Context: The amount of software in solutions provided in various domains is continuously growing. These solutions are a mix of hardware and software solutions, often referred to as software-intensive systems. Companies seek alternatives to improve the software development process to avoid delays or cost overruns related to software development. Component origins such as in-house, outsourcing, COTS or OSS are gaining popularity, therefore, leading to the decision to choose among component origins.

Objective: The overall goal of this thesis is to support decision-making for selecting component origins. Following a decision-making process including all the key decision-making activities is crucial in making decisions. Therefore, the objective of the thesis is to support the decision-makers to create a decision-making process based on their context. In addition, the objective is to improve the decision-making process by incorporating research results and decision-makers' opinion and knowledge in practice.

Method: We identified the factors that influence the choice to select among different component origins through a systematic literature review using an SB strategy and a DB search. We extended the investigation and conducted a case survey of 22 cases. Using design science, we developed solutions including a process-line to support decision-makers, a Bayesian synthesis process to integrate the evidence from literature into practice and a KT framework to facilitate the implementation of research results in practice.

Results: In-house development and alternative component origins (outsourcing, COTS, and OSS) are being used for software development. Several factors such as time, cost and license implications influence the selection of component origins. Solutions have been proposed to support the decision-making. However, these solutions consider only a subset of factors identified in the literature. According to the case survey, the solutions proposed in literature are not aligned with practice. In practice, the decisions are mostly based on opinions. The design objective to support decision-makers with the decision-making process is identified. Therefore, we propose a process-line to address the design objective. In addition, to make the decision-making more informed we propose a KT framework incorporating Bayesian synthesis to help decision-makers make evidence-informed decisions.

Conclusions: The decision to choose among component origins is case dependent. To support the decision-making process, the flexibility and customization of the solution based on the context are important. Therefore, the process-line proposed in the thesis is not prescriptive rather it is customizable to the context. In addition, to facilitate evidence-based decision-making, we provide an application of the KT framework that allows decision-makers to consider research results in addition to their own opinions and knowledge.
Decision-Making Support for Choosing Among Different Component Origins

Deepika Badampudi
Decision-Making Support for Choosing Among Different Component Origins

Deepika Badampudi

Doctoral Dissertation in Software Engineering

Department of Software Engineering
Blekinge Institute of Technology
SWEDEN
ABSTRACT

Context: The amount of software in solutions provided in various domains is continuously growing. These solutions are a mix of hardware and software solutions, often referred to as software-intensive systems. Companies seek alternatives to improve the software development process to avoid delays or cost overruns related to software development. Component origins such as in-house, outsourcing, Components Off-The-Shelf (COTS) or Open Source Software (OSS) are gaining popularity, therefore, leading to the decision to choose among component origins.

Objective: The overall goal of this thesis is to support decision-making for selecting component origins. Following a decision-making process including all the key decision-making activities is crucial in making decisions. Therefore, the objective of the thesis is to support the decision-makers to create a decision-making process based on their context. In addition, the objective is to improve the decision-making process by incorporating research results and decision-makers’ opinion and knowledge in practice.

Method: We identified the factors that influence the choice to select among different component origins through a systematic literature review using an Snowballing (SB) strategy and a Database (DB) search. We extended the investigation and conducted a case survey of 22 cases. Using design science, we developed solutions including a process-line to support decision-makers, a Bayesian synthesis process to integrate the evidence from literature into practice and a Knowledge Translation (KT) framework to facilitate the implementation of research results in practice.

Results: In-house development and alternative component origins (outsourcing, COTS, and OSS) are being used for software development. Several factors such as time, cost and license implications influence the selection of component origins. Solutions have been proposed to support the decision-making. However, these solutions consider only a subset of factors identified in the literature. According to the case survey, the solutions proposed in literature are not aligned with practice. In practice, the decisions are mostly based on opinions. The design objective to support decision-makers with the decision-making process is identified. Therefore, we propose a process-line to address the design objective. In addition, to make the decision-making more informed
we propose a KT framework incorporating Bayesian synthesis to help
decision-makers make evidence-informed decisions.

**Conclusions:** The decision to choose among component origins is
case dependent. To support the decision-making process, the flexibility
and customization of the solution based on the context are important.
Therefore, the process-line proposed in the thesis is not prescriptive
rather it is customizable to the context. In addition, to facilitate
evidence-based decision-making, we provide an application of the KT
framework that allows decision-makers to consider research results in
addition to their own opinions and knowledge.

**Keywords:** Component-based software development, component origin,
decision-making, snowballing, database search, process-line, Bayesian
synthesis and knowledge translation.
OVERVIEW OF PUBLICATIONS

Papers included in the thesis:


Contribution statement

As the lead author of the chapters Deepika Badampudi took the responsibilities in designing, executing and reporting of the studies. As a second author in Chapter 4, Deepika Badampudi contributed in data extraction, analysis, discussion and reporting of the results. The contributions to individuals chapters are as below:

Chapter 2: Deepika Badampudi was the main driver of the systematic literature review. The review consists of two search strategies, Deepika Badampudi designed the review protocol and Claes Wohlin contributed to the design of the snowballing search strategy. Kai Petersen contributed in the database search strategy. Deepika Badampudi extracted the data from 18 (of 24) studies and wrote large parts of the final draft. Claes Wohlin and Kai Petersen reviewed and commented on the final draft of the journal paper.

Chapter 3: Deepika Badampudi took the lead to report experiences from conducting the snowballing search strategy. Deepika Badampudi contributed to the design of the study and reported the findings. Claes Wohlin and Kai Petersen commented and reviewed the final draft of the conference paper.

Chapter 4: Tony Gorschek proposed the idea to conduct a case survey. Kai Peterson, lead the paper and was involved in the entire research process. The remaining authors contributed in data collection. The first two authors (Kai Peterson and Deepika Badampudi) contributed in the data extraction, analysis, discussions and writing. Together with the third author (Syed Muhammad Ali Shah) Deepika Badampudi contributed in reviewing the data extraction scheme and data coding for all research questions. Deepika Badampudi mainly contributed in data extraction, analysis, discussion of results in relation to related work and writing the parts related to decision-making criteria (RQ1.3, RQ2.3 and RQ2.4). Apart from parts of results and discussion, Deepika Badampudi wrote parts of related work (based on the findings from previous conducted SLR), research methodology (analysis method). Deepika Badampudi also reviewed the final paper and presented the paper as a journal first paper at ESEC 2017.

Chapter 5: This journal is an extension of a conference paper. Deepika Badampudi took the lead in extending the conference paper,
and formulated the process-line solution, which was validated in a series of industrial case studies. In addition, Deepika Badampudi designed and conducted the case studies, collected data by conducting the interviews along with the second author (Krzysztof Wnuk), transcribed and analyzed all the interviews and reported the findings. The co-authors commented on the final draft of the journal paper.

**Chapter 6**: Deepika Badampudi mainly contributed with the idea of the study. Claes Wohlin participated and contributed in all the brainstorming sessions. Deepika Badampudi reported the study and Claes Wohlin commented and reviewed the final draft of the conference paper.

**Chapter 7**: Deepika Badampudi mainly contributed with the idea to extend the solution proposed in Chapter 6 into a knowledge translation framework. The idea to develop guidelines and validating solution was jointly discussed by all three authors. The co-authors participated in brainstorming sessions. Deepika Badampudi took the lead in developing the guidelines and initiating industrial collaboration for validating the guidelines. In addition, she wrote the paper which was reviewed by the co-authors.
Papers related but not included in the thesis:


Papers not related to the thesis:


**Paper 2:** Suhrullekha Mutyala, Shravani Nelapudi and Deepika Badampudi. Assessing Barriers and Facilitators to Evidence Use in Decisions: A Tertiary and Interview Study, to be submitted as a journal in 2018.
ACKNOWLEDGMENTS

Firstly, I would like to sincerely thank my advisor Professor Claes Wohlin for his continuous guidance, support, immense knowledge and motivation throughout this thesis. I’m grateful for his timely feedback, constructive criticism and positivity which helped me to grow as a researcher.

I would also like to thank my co-advisor Professor Tony Gorschek for his inputs and guidance. The support and insightful comments from my co-authors are much appreciated. I thank my colleagues for providing a supportive work environment. Last but not the least, I would like to thank all the industrial collaborators.

I thank Farnaz Fotrousi and Indira Nurdiani for the discussions, the sounding board interactions and most importantly their friendship, Siva Dasari, Nina Dzamashvili Fogelstrom, Muhammad Usman and Kennet Henningsson for their company and creating a happy work environment.

Words cannot express my true gratitude and love towards my husband, Gurudutt, who has always supported me in the best possible way.

I thank my parents, Kishore Badampudi and Indira Badampudi for their love and care, my sister, Deepthi Devarakonda for being my best friend and my niece Kanisha Devarakonda for all the happy times.
# CONTENTS

1 INTRODUCTION  

1 Overview ............................................. 1  
2 Background .......................................... 4  
3 Contribution and Research Gaps ......................... 5  
3.1 Chapter 2 .......................................... 5  
3.2 Chapter 3 .......................................... 6  
3.3 Chapter 4 .......................................... 6  
3.4 Chapter 5 .......................................... 7  
3.5 Chapter 6 .......................................... 8  
3.6 Chapter 7 .......................................... 8  
3.7 Research questions .................................. 8  

4 Research Methodology .................................. 10  
4.1 Systematic literature review (SLR) (Chapters 2 and 3) ............................................. 11  
4.2 The case survey (Chapter 4) ......................... 12  
4.3 Case study (Chapter 5) .............................. 13  
4.4 Process-line (Chapter 5): .......................... 14  
4.5 Bayesian synthesis (Chapter 6): ................... 15  
4.6 Knowledge translation (Chapter 7): ............... 15  

5 Overview of the Chapters ............................... 16  
5.1 Chapter 2: Software component decision-making: In-house, OSS, COTS or outsourcing - A systematic literature review ............................................. 16  
5.2 Chapter 3: Experiences from using snowballing and database searches in systematic literature studies ............................................. 17  
5.3 Chapter 4: Choosing component origins for software intensive systems: In-house, OTS, OSS or outsourcing? – A case survey. ............... 18  
5.4 Chapter 5: A Decision-making process-line for selection of software asset origins and components ............................................. 19  
5.5 Chapter 6: Bayesian synthesis for knowledge translation in software engineering: Method and illustration ............................................. 21  
5.6 Chapter 7: Guidelines for knowledge translation in software engineering .............................. 22  

6 Conclusions and Future Work ........................... 23
## SOFTWARE COMPONENT DECISION-MAKING: IN-HOUSE, OSS, COTS OR OUTSOURCING - A SYSTEMATIC LITERATURE REVIEW

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>2</td>
<td>Related Work</td>
</tr>
<tr>
<td>3</td>
<td>Method</td>
</tr>
<tr>
<td>3.1</td>
<td>Need for the review</td>
</tr>
<tr>
<td>3.2</td>
<td>Study identification</td>
</tr>
<tr>
<td>3.3</td>
<td>Data extraction and classification</td>
</tr>
<tr>
<td>3.4</td>
<td>Quality assessment</td>
</tr>
<tr>
<td>3.5</td>
<td>Analysis</td>
</tr>
<tr>
<td>3.6</td>
<td>Validity threats</td>
</tr>
<tr>
<td>4</td>
<td>Results</td>
</tr>
<tr>
<td>4.1</td>
<td>Research types, methods and quality (RQ1)</td>
</tr>
<tr>
<td>4.2</td>
<td>Influencing factors (RQ2)</td>
</tr>
<tr>
<td>4.3</td>
<td>Solutions (RQ3)</td>
</tr>
<tr>
<td>5</td>
<td>Discussion</td>
</tr>
<tr>
<td>5.1</td>
<td>COTS over OSS</td>
</tr>
<tr>
<td>5.2</td>
<td>OSS over COTS</td>
</tr>
<tr>
<td>5.3</td>
<td>In-house over OSS and COTS</td>
</tr>
<tr>
<td>5.4</td>
<td>COTS and OSS over in-house</td>
</tr>
<tr>
<td>5.5</td>
<td>Research gaps</td>
</tr>
<tr>
<td>6</td>
<td>Conclusion</td>
</tr>
</tbody>
</table>

## EXPERIENCES FROM USING SNOWBALLING AND DATABASE SEARCHES IN SYSTEMATIC LITERATURE STUDIES

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>2</td>
<td>Related Work</td>
</tr>
<tr>
<td>3</td>
<td>Research Method</td>
</tr>
<tr>
<td>3.1</td>
<td>Details of snowballing</td>
</tr>
<tr>
<td>3.2</td>
<td>Details of database search</td>
</tr>
<tr>
<td>3.3</td>
<td>Research questions</td>
</tr>
<tr>
<td>4</td>
<td>Results</td>
</tr>
<tr>
<td>4.1</td>
<td>Evolution of the SB process (RQ1)</td>
</tr>
<tr>
<td>4.2</td>
<td>Efficiency of SB (RQ2)</td>
</tr>
<tr>
<td>4.3</td>
<td>Reliability of SB (RQ3)</td>
</tr>
<tr>
<td>5</td>
<td>Discussion</td>
</tr>
<tr>
<td>6</td>
<td>Conclusion</td>
</tr>
</tbody>
</table>

## CHOOSING COMPONENT ORIGINS FOR SOFTWARE INTENSIVE SYSTEMS: IN-HOUSE, COTS, OSS, OUTSOURCING OR SERVICES? - A CASE SURVEY

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Choosing component origins for software intensive systems: in-house, COTS, OSS, outsourcing or services? - a case survey</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>2</td>
<td>Related Work</td>
</tr>
<tr>
<td>2.1</td>
<td>Decision making problem</td>
</tr>
<tr>
<td>2.2</td>
<td>Characterizing decision making</td>
</tr>
<tr>
<td>3</td>
<td>Method</td>
</tr>
<tr>
<td>3.1</td>
<td>Research Questions</td>
</tr>
<tr>
<td>3.2</td>
<td>The case survey method</td>
</tr>
<tr>
<td>4</td>
<td>Results</td>
</tr>
<tr>
<td>4.1</td>
<td>Overview of cases</td>
</tr>
<tr>
<td>4.2</td>
<td>RQ1: How are CSOs chosen?</td>
</tr>
<tr>
<td>4.3</td>
<td>RQ2: What was the result of the decision making process?</td>
</tr>
<tr>
<td>5</td>
<td>Discussion</td>
</tr>
<tr>
<td>5.1</td>
<td>Reflections with respect to the research questions</td>
</tr>
<tr>
<td>5.2</td>
<td>Characterization of decisions</td>
</tr>
<tr>
<td>5.3</td>
<td>Comparison with the general traits of decision making from related work</td>
</tr>
<tr>
<td>5.4</td>
<td>Validity threats</td>
</tr>
<tr>
<td>6</td>
<td>Conclusion</td>
</tr>
</tbody>
</table>

## 5 A DECISION-MAKING PROCESS-LINE FOR SELECTION OF SOFTWARE ASSET ORIGINS AND COMPONENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
<td>153</td>
</tr>
<tr>
<td>2</td>
<td>Background</td>
<td>156</td>
</tr>
<tr>
<td>2.1</td>
<td>Decision-making</td>
<td>156</td>
</tr>
<tr>
<td>2.2</td>
<td>GRADE taxonomy</td>
<td>157</td>
</tr>
<tr>
<td>2.3</td>
<td>Descriptive models</td>
<td>158</td>
</tr>
<tr>
<td>2.4</td>
<td>Evidence-based knowledge repository</td>
<td>163</td>
</tr>
<tr>
<td>3</td>
<td>Related Work</td>
<td>164</td>
</tr>
<tr>
<td>3.1</td>
<td>Deciding on origin</td>
<td>164</td>
</tr>
<tr>
<td>3.2</td>
<td>Decision-making in software business</td>
<td>165</td>
</tr>
<tr>
<td>3.3</td>
<td>Application of process-line in software engineering</td>
<td>166</td>
</tr>
<tr>
<td>4</td>
<td>Research Methodology</td>
<td>166</td>
</tr>
<tr>
<td>4.1</td>
<td>Iteration 1</td>
<td>167</td>
</tr>
<tr>
<td>4.2</td>
<td>Iteration 2</td>
<td>169</td>
</tr>
<tr>
<td>4.3</td>
<td>Iteration 3</td>
<td>170</td>
</tr>
<tr>
<td>4.4</td>
<td>Validity discussion</td>
<td>171</td>
</tr>
<tr>
<td>5</td>
<td>Process-line for Decision-making using a Checklist</td>
<td>172</td>
</tr>
<tr>
<td>5.1</td>
<td>The recommended activities in the decision-making process-line are as follows:</td>
<td>175</td>
</tr>
<tr>
<td>6</td>
<td>Validation and Case studies</td>
<td>179</td>
</tr>
<tr>
<td>6.1</td>
<td>Overview of the companies</td>
<td>179</td>
</tr>
</tbody>
</table>
6.2 Overview of the decision-making process followed by the companies .................................. 182
6.3 Detailed description of the decision-making activities followed in all the cases .................. 183
6.4 Detailed description of the decision-making activities followed in a sub-set of the cases ...... 188
6.5 Discussion and implications from the cases ................................................................. 191

7 General Discussion and Validation Implications ...................................................... 193
7.1 Checklist v1.0 validation and refinement into Checklist v2.0 ...................................... 193
7.2 Checklist v2.0 validation and refinement into process-line (Checklist v3.0) ..................... 194

8 Summary and Further Work ....................................................................................... 195

6 BAYESIAN SYNTHESIS FOR KNOWLEDGE TRANSLATION IN SOFTWARE ENGINEERING: METHOD AND ILLUSTRATION 197
1 Introduction ................................................................. 197
2 Background and Related Work ............................................................. 199
  2.1 Overview of Bayesian synthesis ......................................................... 199
  2.2 Bayesian synthesis in health research .................................................. 200
  2.3 Synthesis and KT in SE ................................................................. 200
  2.4 Bayesian in SE .............................................................................. 201
3 Bayesian Synthesis for KT - Method Description ........................................... 201
  3.1 Step 1: The prior probability .............................................................. 202
  3.2 Step 2: The likelihood ..................................................................... 203
  3.3 Step 3: The posterior probabilities - refining prior probabilities .......... 205
4 Illustration ............................................................................................................ 208
  4.1 Step 1: Prior probability ................................................................... 208
  4.2 Step 2: Likelihood ......................................................................... 209
  4.3 Step 3: Posterior probability - Refining prior probability ................. 211
5 Discussion ............................................................................................................. 214
6 Conclusions .......................................................................................................... 215

7 GUIDELINES FOR KNOWLEDGE TRANSLATION IN SOFTWARE ENGINEERING 217
1 Introduction ........................................................................................................... 218
2 Background and Related Work ........................................................................ 220
  2.1 Sharing tacit knowledge ..................................................................... 220
  2.2 Academia-industry collaborations ....................................................... 220
  2.3 Technology transfer ......................................................................... 221
  2.4 Knowledge transfer and translation .................................................... 222
| 3  | Description of the KT Framework                       | 223 |
| 3.1 | Overview of the KT framework                          | 223 |
| 3.2 | Detailed working of the KT framework                  | 225 |
| 4   | KT Framework Evaluation                               | 232 |
| 5   | Discussion                                            | 238 |
| 6   | Conclusions                                           | 239 |

**REFERENCES**

241
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Mapping of chapters to goals</td>
<td>2</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Overview of the thesis</td>
<td>3</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Mapping of goals, research questions and chapters</td>
<td>9</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Research methodology used in the thesis.</td>
<td>11</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Design framework used in the thesis. <em>Adopted from Wieringa [185]</em>.</td>
<td>14</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Decision levels</td>
<td>29</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Snowball search process</td>
<td>36</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Database search process</td>
<td>37</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Research types and studied combinations of component origins</td>
<td>47</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Research methods and studied combinations of component origins</td>
<td>48</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Positive and negative influences of project metrics factors</td>
<td>53</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Positive and negative influences of external factors.</td>
<td>57</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Positive and negative influences of software development process factors.</td>
<td>60</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Trade-offs for COTS and OSS factors</td>
<td>63</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Trade-offs and dependencies for advantages of OSS over COTS</td>
<td>64</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Trade-offs and dependencies for in-house, COTS and OSS factors</td>
<td>65</td>
</tr>
<tr>
<td>Figure 17</td>
<td>Evolution of identified papers</td>
<td>80</td>
</tr>
<tr>
<td>Figure 18</td>
<td>Citation matrix</td>
<td>83</td>
</tr>
<tr>
<td>Figure 19</td>
<td>Venn diagram for the overlapping papers</td>
<td>90</td>
</tr>
<tr>
<td>Figure 20</td>
<td>Options coverage</td>
<td>91</td>
</tr>
<tr>
<td>Figure 21</td>
<td>Dimension coverage</td>
<td>92</td>
</tr>
<tr>
<td>Figure 22</td>
<td>Decision levels when choosing components</td>
<td>100</td>
</tr>
<tr>
<td>Figure 23</td>
<td>Stakeholder perspectives</td>
<td>107</td>
</tr>
<tr>
<td>Figure 24</td>
<td>Number of options considered</td>
<td>121</td>
</tr>
<tr>
<td>Figure 25</td>
<td>Decision outcomes</td>
<td>123</td>
</tr>
<tr>
<td>Figure 26</td>
<td>Number of roles involved in the decision initiation presented in the cases</td>
<td>126</td>
</tr>
<tr>
<td>Figure 27</td>
<td>Number of roles involved in the decision preparation presented in the cases</td>
<td>126</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Number of roles involved in the decision-making presented in the cases.</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>Criteria trade-offs</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>Number of criteria</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>Effort in decision preparation</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>Effort in decision making</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>Hierarchical clustering of cases</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Mapping of GRADE to concepts in the decision model and the supporting models.</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>An illustration of how the GRADE taxonomy, knowledge repository and descriptive models provided the starting point for the decision-making checklist and process-line.</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>Tasks carried out in the design cycle and case study.</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>A process-line for decision-making supported by a knowledge repository.</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>A possible decision-making process including having a knowledge repository.</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>The process-line activity coverage in related to the decision-making process followed by the companies.</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>Number of stakeholder roles involved in each decision.</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>Number of asset origins involved in each decision.</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>Number of criteria considered in each decision.</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>The process-line for decision-making including variations in activity execution.</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>Three steps of Bayesian synthesis</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>Division of likelihood calculation</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>Four different approaches to refine prior probabilities</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>The use of knowledge transfer (evidence briefings) in knowledge translation.</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>Knowledge translation framework (Adapted from Graham et al. [66]).</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>Knowledge creation cycle (Adapted from Straus et al. [169]).</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>Knowledge creation cycle (Adapted from Straus et al. [169]).</td>
<td></td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 1  Secondary and primary studies related to the different decision levels ............................... 29
Table 2  Systematic reviews related to individual component origins .............................................. 30
Table 3  Inclusion and exclusion decision rules .... 34
Table 4  Number of hits for database search ........... 37
Table 5  Search type and number of relevant papers .. 38
Table 6  Data extraction form ................................. 38
Table 7  Rules defining the themes ......................... 42
Table 8  Total number of primary studies per comparison category ................................................. 45
Table 9  Research types ........................................ 46
Table 10 Research methods ................................ 47
Table 11 Domains of the primary studies ................ 48
Table 12 Rigor and relevance scores of the empirical studies ......................................................... 49
Table 13 Primary studies discussing influencing factors for component origin (RQ2) ...................... 50
Table 14 High-level themes, themes and codes ........ 51
Table 15 Project metrics factors ............................. 52
Table 16 External factors ....................................... 55
Table 17 Software development process factors ........ 58
Table 18 Primary studies proposing solutions for choosing a component origin (RQ3) ..................... 61
Table 19 Advantages of component origins .......... 63
Table 20 Mapping of factors influencing the decision and factors considered in the solutions .......... 67
Table 21 Nine search strings used by SB .................. 75
Table 22 Inclusion and exclusion decision rules ....... 77
Table 23 Search strings used by database search ........ 77
Table 24 Compliance with the SB guidelines ........... 82
ACRONYMS

CBSE  Components-Based Software Engineering  
EBSE  Evidence-Based Software Engineering  
SLR  Systematic Literature Review  
COTS  Components Off-The-Shelf  
OSS  Open Source Software  
SB  Snowballing  
FSB  Forward Snowballing  
BSB  Backward Snowballing  
DB  Database  
KT  Knowledge Translation  
SE  Software Engineering
1 INTRODUCTION

1 OVERVIEW

The use of software is becoming more and more common in solutions provided by different domains such as automotive, automation, health and telecommunication. Systems that consist of large amounts of software that provides value to its users are referred to as software-intensive systems. Companies are constantly seeking to improve the software development/building process to gain profit and competitive advantage. The improvements are usually aimed to provide timely, high quality and cost efficient solutions.

Components-Based Software Engineering (CBSE) is known to increase productivity, save costs, increase quality and reusability [181]. CBSE promotes the development for reuse and building from existing software components [181]. Different alternatives to in-house development such as outsourcing [103], adopting COTS [22] and OSS [71] components are gaining popularity. The different alternatives are referred to as component origins.

A decision to choose among component origins is a complex decision because of the uncertainties involved in software projects. In addition, there is often more than one goal that needs to be considered in such decisions. Such decisions might have major implications that may not be realized immediately. In a long-term strategic perspective, the decision of where to get components from may have important implications, on, for example, effort. For example, when outsourcing development to another company located, for example, in a different country, as we are customers, we can still influence the development processes, and have control over the evolution of the components. When deciding to get COTS components, then we have less influence, but not the challenge of transferring knowledge, learning, etc. that may occur in outsourcing.

It is important to understand the state of existing knowledge (Chapter 2) and practice (Chapter 4) on how the component origins are traded-off, i.e., what criteria are used to select the component origins in which context using what process, techniques or tools. Like any other technology or process, no component origin is universally good or uni-
versally bad, some component origins are more appropriate than others based on the context of the projects and the organizations.

The goals, criteria and decision-outcomes might be different for different contexts. Therefore the solution for such decisions should not be a prescriptive solution. Hence, one of the main contributions of the thesis (Chapter 5) supports the decisions-makers to build their own tailored decision-making process based on their context. Decision-makers usually rely on their own experience and opinions when making decisions according to the findings of Chapter 2. The opinions and knowledge of decision-makers might be error-prone [50]. Therefore, in addition to the decision-makers’ opinion and experience, it is important to consider evidence in terms of the knowledge provided in the research studies. Guidelines for a Bayesian synthesis approach (Chapter 6) and KT in Software Engineering (SE) (Chapter 7) facilitate the alignment of knowledge from research studies and practitioners’ opinions and experience in decisions.

Hence even though the overall goal is to support decision-making for selecting component origins, the need to provide methodological support to integrate knowledge from research studies and decisions-makers opinions has been identified in this thesis. Therefore, we further divide the overall goal of the thesis into two goals:

**Research goal 1:** To support the decision-making process to select among different alternatives.

**Research goal 2:** To provide methodological support to facilitate the use of research results in decisions.

The chapters in this thesis contribute to the goals as shown in Figure 1.

![Figure 1: Mapping of chapters to goals](image)
The overall research goal is to provide a decision-support solution for selecting component origins. Figure 2 depicts the focus of this thesis to support decision-makers. We begin by investigating the existing knowledge on different component origins, the objective is to explore how the decision to choose the component origin is made. To understand the existing knowledge, we conducted a systematic literature review (Chapter 2).

The main findings of the review are a list of criteria used to choose a component origin. The review also allowed us to investigate the existing solutions to address the decision-making process for component origin selection. The solutions were based on optimization models that suggest a component origin that is the best fit for the goal. The systematic literature review was conducted using database (DB) search and snowballing (SB), we reported our experiences on the reliability and efficiency of the SB method in comparison to the DB search (Chapter 3).

After understanding the state of existing knowledge, we investigated the state of practice by evaluating 22 cases through a case survey study reported in Chapter 4. The main findings of the case survey indicate that the criteria, decision-making models, component origins considered and the component origins selected vary from case to case. In addition, expert opinion was the most common decision-making model used in such decisions. In other words, the decision-makers rely on their opinions to select the component origins. However, the solutions
proposed in the research studies were based on optimization models according to Chapter 2. Therefore, we notice a misalignment between the practice and solutions provided in research studies. Furthermore, the perception of the decision-makers in retrospect indicated problems related to the decision-making process. Thereby identifying the need to propose a solution that is aligned with practice and likely to be accepted by decision-makers. Therefore, we propose a solution called process-line to support the decision-making process. The process-line provides a set of all relevant decision-making activities to support the decision-makers in creating their own decision-making process based on their context. The process-line solution is presented in Chapter 5.

In addition, as seen in our case survey study (Chapter 4), most of the decisions are based on expert opinions. Opinions could be biased and error-prone [50]. Therefore, in addition to their opinions, decision-makers should consider external input in terms of knowledge from research studies to make informed and un-biased decisions. We refine and validate the use of a synthesis approach used in healthcare to integrate data collected from research studies with practitioners’ opinions (Chapter 6). To further facilitate the use of research results in practice we refine the KT framework proposed in healthcare by incorporating the synthesis approach described and illustrated in Chapter 6. We provide guidelines to KT in SE and evaluate the framework in Chapter 7.

2 BACKGROUND

This section provides background information on the concepts that are mentioned in the chapters.

Component-based software development (CBSE): CBSE is defined as follows: "The primary role of component-based software engineering is to address the development of systems as an assembly of parts (components), the development of parts as reusable entities, and the maintenance and upgrading of systems by customizing and replacing such parts" [45]. The different parts (components) can be either developed in-house, outsourced or obtained as external components (COTS or OSS).

Component origin: The different options to get components from are called component origins. The description of each development option is as follows:

- **In-house**: This option is the most straightforward. It involves the development of the software component within the company. It
also includes offshoring where the software component is developed within the same company however, in a different location.

- **Outsourcing**: The development of the software component is outsourced to another company often called supplier or vendor. The requirements for the development are provided by the company outsourcing the development.

- **OSS** [79, 114]: It refers to "open source software". The OSS components are available for free and the component is provided along with the source code. These components are pre-built and the requirements for building these components are derived from several factors such as developers’ interest or sometimes even the market trend.

- **COTS** [22, 129]: It refers to "components off-the-shelf". These components like the OSS components are pre-built however, they are not free and often the source code is not provided along with the components. The COTS components are mainly driven by the market trend.

**Decision to choose component origin** [192]: In this thesis the decision-making is on the strategic level, i.e. to choose a component origin. Such decisions are often derived by some goal such as: improving the software development process or return of investment. The decisions are based on criteria such as quality, time and cost. The decision-making process requires different models to estimate the criteria values such as the cost to buy/build the component, or the time it takes to build, integrate and test the component. The decision-making process involves decision-makers from different roles, these roles might have different expertise in terms background and responsibility and different contributions such as decision initiator, supporter and decider.

3 Contribution and research gaps

The contributions of this thesis are as mentioned in the below sections.

3.1 Chapter 2

**Research gap 1**: Primary studies on selection between the component origins have been conducted. However, this evidence has not been aggregated and interpreted so far.

Hence, our objective for Chapter 2 is as follows:
Objective 1: To identify factors that could influence the decision to choose among different component origins and solutions for decision-making in the literature.

The contributions of Chapter 2 are as follows:

- C1.1: The research types and methods used to evaluate the criteria and solutions.
- C1.2: The criteria considered in comparing two or more component origins in a way that facilitates decision-making.
- C1.3: The solutions to decision-making (for example, optimization).
- C1.4: The research gaps and future directions.

3.2 Chapter 3

Research gap 2: In SE there are fewer studies that use SB as compared to a DB search strategy for searching for primary studies. The potential of SB regarding its efficiency and reliability in finding the primary studies is not fully understood. Hence, this leads us to our next objective, which is:

Objective 2: To find efficient and reliable search strategies.

The contribution of Chapter 3 is as follows: Reliability and efficiency of SB in comparison to DB search. In addition, the use of SB as the main search strategy to conduct the systematic literature reviews such as in Chapter 2. It is possible to use SB as a strategy to identify knowledge from research studies to be translated into practice. We have used SB in Chapter 7 to identify knowledge from research studies.

3.3 Chapter 4

Research gap 3: The results from previous systematic literature review (Chapter 2) indicate that only a few case studies were conducted related to the component origins. Hence, the case survey study is conducted to complement the limited state of practice by investigating 22 cases thereby adding to the body of knowledge.

Based on the research gap, the next objective was as follows:

Objective 3: To understand how practitioners choose among the component origins with respect to stakeholders involved, criteria used for making the decision and the decision-making method used. In addition, the objective was to understand the outcome of the decision, the
effort spent, the criteria that had most important on the decision and the perceptions of the practitioner in retrospect.

The contributions of Chapter 4 are as follows:

• C3.1: The component origins and criteria that were considered in the decisions, stakeholders involved and decision-making approach/models used in practice.

• C3.2: The outcome of the decisions in practice, i.e., the component origin chosen, the criteria that were significant in making the decision, effort invested and perceived success degree of the decision.

3.4 Chapter 5

Research gap 5: Based on the results from Chapter 2, we know that few solutions were proposed. The solutions were focused on evaluating a sub-set of criteria and not on the decision-making process and activities that might be important for the decision to choose among component origins. In addition, in the case survey results in Chapter 4, we only have summarized information provided by one stakeholder involved in a decision. In Chapter 5, we continued the investigation by understanding the decision-making process through the perspective of all involved stakeholders.

Therefore, our next objective is:

Objective 5: To support decision-makers in building their own decision-making process by providing a set of decision-making activities.

The contributions of Chapter 5 are as follows:

• C5.1: Complement the findings from Chapters 2 and 4 by investigating the decisions from the perspective of all involved stakeholders.

• C5.2: An approach called process-line which allows decision-makers to create their own tailor-made decision-making process. The process-line allows the decision-makers to consider all the activities at the beginning of the decision-making process and avoid overlooking critical aspects as observed in some cases in Chapter 4.
3.5 Chapter 6

Research gap 6: None of the synthesis methods in SE research is primarily designed to translate evidence from literature into practice by taking the experience of practitioners into account. Therefore, our next objective is:

Objective 6: To improve the methodological support to integrate evidence from literature and practitioners knowledge.

The contribution of Chapter 6 is as follows: An application of Bayesian synthesis in SE to make the evidence available to practitioners in a way that allows practitioners to integrate the evidence from literature into their decisions.

3.6 Chapter 7

Research gap 7: In the previous chapters (Chapters 2 and 4) and some related work \([9], [50]\), the gaps and differences between research and practice have been identified. The solutions proposed in research studies are not always aligned to practitioners’ context and hence they, are not always utilized in practice. There is a need to translate the SE research results to practice. Knowledge translation is done in an ad-hoc manner in SE research and there are no guidelines for KT. Therefore, our next objective is:

Objective 7: To provide the methodological support for utilizing research results in practice.

The contribution of Chapter 7 is as follows: An application of the KT framework in SE research, in particular, to translate research evidence into practice by combing contextualized expert opinions with research evidence.

3.7 Research questions

Research questions help us address the research objectives and gaps. The research questions in this thesis are formulated based on the objectives and the research gaps. Eight main research questions are formulated some of which are further divided into sub-research questions. The research questions addressed in the thesis and their mapping to the chapters are as shown in Figure 3. Note that the diagram shows the key research questions only, the further sub-questions are mentioned in the individual chapters.
RQ 1-3 are jointly answered by Chapters 2 and 4. As seen in Figure 3, Chapter 2 answers some questions which are complemented by Chapter 4. In addition, Chapter 4 provides new answers in RQ 1, 2 and 3 (RQ2.1, RQ2.2, RQ2.3, RQ3.4 and RQ4.5) which were not found in the literature i.e. in Chapter 2. Chapter 4, provides further new insights by answering RQ4.

Chapter 5 directly answers RQ5 as it provides a process-line to support decision-making. The inputs from Chapters 2 and 4 are used in de-
signing the solution (process-line) proposed in Chapter 5. The methodological support provided in Chapters 6 and 7 was partially used to integrate practitioners’ opinions and research results and validate the solution in Chapter 5. Furthermore, Chapters 6 and 7 provide methodological support for using the process-line in practice, which is part of future work.

Chapter 3 directly answers RQ6 by reflecting on the results from Chapter 2. Therefore, Chapter 2 is partially answering RQ6. Chapter 6 provides a detailed description on how practitioners’ opinions and research results can be integrated using Bayesian synthesis, thereby answering RQ7. Chapter 7 answers RQ8 by providing an implementation of the KT framework in SE. The KT framework in Chapter 7 incorporates Bayesian synthesis from Chapter 6 therefore, Chapter 7 partially answers RQ7 and Chapter 6 partially answers RQ8. Knowledge from relevant research studies are used to integrate with practitioners’ opinions in Chapters 6 and 7, and experiences from Chapter 3 can be used to retrieve relevant studies therefore, Chapter 3 partially answers RQ7 and RQ8.

4  RESEARCH METHODOLOGY

Research methods help answering research questions in a systematic and repeatable way. The rigor of research methods not only allows a thorough investigation of a phenomenon but also allows the users/readers to trust and rely on the results. The overall research methodology followed in this thesis is as shown in Figure 4.

We started by investigating the problem using a Systematic Literature Review (SLR), a case survey and case studies. The research problem is investigated in all the chapters. Chapters 2, 3 and 4 are primarily focused on investigating the status of the research area. Meanwhile Chapters 5 and 7 investigated the problem, as well as constructed the solution. The use of Bayesian synthesis was investigated in Chapter 6 based on problem investigated in the previous chapters (Chapters 2, 3 and 4). Once the design objective was known we constructed the solutions in Chapters 5, 6 and 7. The solution in Chapter 5 was validated in three cases and the solutions in Chapter 6 and 7 were implemented and evaluated in two cases which was reported in Chapter 7.

A summary of the research methods used for problem investigation and their application to the chapters is provided in the Sections 4.1, 4.2 and 4.3.
4.1 Systematic literature review (SLR) ( Chapters 2 and 3)

Systematic literature reviews are used to aggregate and interpret the evidence through a scientific and repeatable process. It mainly includes the following steps:

1. Study identification: This step includes searching and selecting the primary studies. Mainly two search strategies are used in SE: DB search and SB. The primary studies are selected from the search results based on defined inclusion/exclusion criteria.

2. Data extraction: In this step, the data from primary studies is extracted in an explicit and consistent way. Data extraction forms are used to extract data. The design of the extraction forms is driven by the research questions.

3. Quality assessment: The quality of the primary studies is assessed in this step. The assessment is based on rigor and relevance of the primary studies.

4. Analysis: The data from the primary studies are analyzed in this step. Analysis methods such as thematic analysis or narrative analysis are commonly used to analyze data from qualitative studies. Whereas meta-analysis is commonly used to analyze data from quantitative studies.

Chapter 2: In Chapter 2, a systematic literature review was conducted to identify factors that could influence the decision to choose
among different component origins and solutions for decision-making in the literature. We used a SB search strategy following the guidelines in [190] to search for primary studies and a DB search following the guidelines in [97] was used to validate and ensure the completeness of the search process. A data extraction form was used to extract data. The quality of the primary studies was evaluated using the rigor and relevance criteria defined in the guidelines proposed in [85]. Thematic analysis following the guideline in [46] was used to analyze the results of the primary studies.

Chapter 3: In Chapter 3, reflections of conducting a systematic literature review using two different search strategies: SB and DB search are reported.

4.2 The case survey (Chapter 4)

We provide an introduction to the case survey method as it has not been widely applied in a software engineering context.

Studies often report only a small number of cases in a single publication. On the other hand, surveys focus on a large number of data points and are mostly quantitative. The case survey method is a combination of the two approaches [107], as Larsson points out “it can overcome the problem of generalizing from a single case study and at the same time provide more in-depth analysis of complex organizational phenomena than questionnaire surveys” (cf. [107], p. 1566). According to Larsson, there are multiple benefits of using the case survey method. As cases are synthesized the case survey adds value to previous individual cases, and the richness of case studies can be incorporated to draw conclusions in the quantitative synthesis. A data extraction form is used for extracting data from the cases, hence it is possible to easily extend the case survey by adding further cases. Overall, Larsson concludes that the case survey is a means to bridge the gap between positivist approaches (such as surveys) and humanistic/interpretivist approaches (such as case studies).

The process of the case survey method comprises of four different steps:

1. Select the cases of interest.
2. Design the data extraction form for elicitation
3. Conduct the coding
4. Use statistical approaches to analyze the coding
Chapter 4: The authors of the study collected cases using convenience sampling. The cases that considered at least two or more component origins were of interest. In addition, we wanted the cases that include additional information such as the stakeholders involved and the outcome of the decisions as we did not find related information in the literature. The data extraction was done using a coding scheme developed based on the input from Chapter 2. The detailed description of the process steps of the case survey study are outlined in Chapter 4 Sections 3.2.1 to 3.2.4.

4.3 Case study (Chapter 5)

Case study is regarded as a suitable research method for SE research as it is hard to study a phenomenon in isolation. A case study allows to investigate the phenomenon in its real-life context. The steps involved in case study research are listed and summarized by Runeson and Höst [154] as follows:

1. Case study design: The objectives of the study are defined and the case study is planned based on the objective and research questions. The plan includes details about the case to be studied, the data collection method to be used, and the selection strategy to be used.

2. Preparation for data collection: The procedures and protocol for data collection are defined in this step. The protocol includes details such as: what questions should be asked?

3. Collecting evidence: The data collection is performed to collect evidence.

4. Analysis of collected data: The collected data is analyzed using different analysis methods that suits the collected data.

5. Reporting: The results of the analyzed data are reported. The report should include an elaborate description of the research work and details of the conclusions. This includes a description of the context in which the results are likely to be valid.

Case study is used in Chapter 5 to do an in-depth investigation of the decision to choose among component origins in two cases. Based on which the solution was designed using a design science approach [185]. The solution was later validated using three more cases. The practitioners involved in the decision-making process, i.e. in the decision initiation, preparation and final decision were interviewed.
Design science is used to construct the design solutions in this thesis. The design framework of the thesis is presented in Figure 5.

![Design Framework Diagram](image)

Figure 5: Design framework used in the thesis. *Adopted from Wieringa [185].*

As mentioned earlier, Chapters 2 and 3 found existing answers to the research questions and Chapter 4 provided additional answers to the research questions which were contributing to the knowledge context. The design objectives were identified in Chapters 2 and 4, which were used as an input in the design phase. In addition, the input from the knowledge context as shown in Figure 5 is used in the designing of the solutions. The design of each main contribution of the thesis is presented in the Sections 4.4, 4.5 and 4.6.

### 4.4 Process-line (Chapter 5):

The design objective from Chapters 2 and 4 was used as an input to construct the solution in Chapter 5. In addition, the design objective was further investigated by conducting a multi-case study (two cases). Existing problem solving knowledge and designs from the knowledge context were used in the design phase.
context were used. The existing designs focused on a specific decision-making aspects such as the context model [28], knowledge repository [38], property model [136], decision-making taxonomy [134] specific to the design objective were used as an input. The design was validated by co-authors and external researchers. After the validation, we re-designed the solution and validated it in three additional cases based on which the solution was refined and called as process-line.

4.5 Bayesian synthesis (Chapter 6):

The overall design objective was identified through the overall goal to support decision-making in selecting component origins. In addition, the design objective was identified in Chapters 2 and 4. The design objective was to support the integration of research results with decision-makers’ opinions. The existing design from healthcare [182], [161], [44] and [150] was adapted and was implemented by simulating the design problem. This chapter is mainly providing a solution which is illustrated using an example related to the decision-making to select among component origins.

4.6 Knowledge translation (Chapter 7):

The solution in Chapter 6 is extended into a KT framework. The existing design from Chapter 6, existing designs from knowledge context in SE [25] and from healthcare [68], [169] and [66] are used to adapt the KT framework in SE. The KT framework to SE was adapted with the intention to support decisions to choose among component origins. However, the main objective of the study in Chapter 6 is to evaluate the KT framework in software practice. Therefore, we approached the companies for an industrial evaluation. The company that was interested to use the KT framework wanted support with GUI test automation decisions at the time we approached them. Since the objective is to evaluate the KT framework for decisions in software practice, we focused on the industrial evaluation of the KT framework. Therefore, we prioritized the industrial evaluation of the KT framework in supporting GUI test automation decisions over choosing the sub-area, i.e. selection among component origins. We investigated the test automation GUI for two software-intensive systems to understand the decision support needed by the decision-makers. The KT framework is implemented and evaluated in the two software-intensive systems investigated.
In this section, an overview of the chapters is provided. The overview includes the objective of the chapter, the methods used to achieve the objective and a description of the main findings.

5.1 Chapter 2: Software component decision-making: In-house, OSS, COTS or outsourcing - A systematic literature review

Four widely used component origins are COTS, OSS, outsourcing and in-house development. Decision-makers make decisions on choosing a component origin for developing or acquiring a component/s. The objective of this chapter is to present results from the literature interpreting the factors that influence such decisions and the existing solutions supporting the decisions. A systematic literature review is conducted to identify primary studies. A total of twenty four primary studies were identified. The details of the review protocol are reported briefly in Section 4.1 and in detail in Chapter 2.

Eleven factors that have an influence on the decision to choose among different component origins were identified. The factors are: Time, cost, effort, market trend, source code availability, technical support, license, integration, requirement, maintenance and quality. Most of the primary studies considered two component origins in the decision. The decision between in-house vs. COTS and COTS vs. OSS were the most researched decisions. The solution models proposed in the literature are based on optimization techniques. The
criteria considered in the solutions models are time, cost and reliability.

**Key findings:** Through our systematic literature review we found that some criteria were considered positive for the adoption of some component origins while negative for some other component origins. For example, the time to test and integrate was positive for the adoption of COTS however, negative for in-house. In addition, same factors were reported to be both positive and negative by different studies. For example, component evolution was in line with the market trend with COTS components. However, some other studies reported that this resulted in frequent upgrades which impacted the stability of the product, therefore being a negative factor. Different trade-offs were identified for example, with frequent updates the maintainability effort increased to make the system stable. On the other hand, if the system is not upgraded to the latest version, then the support from the vendor is considered to be void. Although, using COTS is beneficial when time to market is critical, the time to test and integrate COTS components can be greater than the in-house development effort when the component size is small [73]. This indicates that one option is not always best for all possible scenarios. It is important to know how decisions are made and what criteria are used to select the component origins.

5.2 Chapter 3: Experiences from using snowballing and database searches in systematic literature studies

SLRs are ways to aggregate and interpret findings from primary studies. DB search is commonly used for searching the primary studies in SE. It is recommended to use SB after DB search to find additional papers. There are few studies that use SB as the main or only method. The effort required in terms of the number of papers to be reviewed to find the primary papers (efficiency) and the capability to find all relevant papers (reliability) might raise some concerns. Therefore, the objective of this chapter is to find efficient and reliable search strategies. A brief summary of the results are
The key findings of this chapter are that DB search and SB search are comparable. More papers were reviewed using SB in comparison to the DB search. However, most papers were either duplicates, gray literature or non-English papers which were easy to exclude based on the title. In DB search, such entries can be automatically removed using DB search engines. The total number of abstracts reviewed using DB search and SB were approximately the same. The SB strategy identified 83% of the papers, while DB search identified 46% of the papers and 29% of the papers were found by both DB search and SB. The large percentage of papers found in SB search might be due to the fact that the research area of choosing among component origins is not well established and the terminology used in research studies is not consistent. Therefore, relying on search strings in DB search did not result in identifying as many papers. It can be concluded that SB strategy was efficient and reliable for searching the primary studies to achieve the objective in Chapter 3.

5.3 Chapter 4: Choosing component origins for software intensive systems: In-house, OTS, OSS or outsourcing? – A case survey.

In Chapter 4, the focus was to extend the investigation and find answers that were not found in the literature (Chapter 2). In Chapter 2, we identified the gaps in the literature and designed the case survey to complement the SLR findings and find answers that were missing in the literature. In Chapter 2 we did not have complete information on "a" decision particularly, we did not have information about the outcome of the decisions. For example, we did not have answers to the following questions - Which component origins among those considered were chosen? Which criteria ended-up being significant? And, what was the success rate of these decision as perceived by practitioners?. In the case survey, we notice that the solutions proposed in the literature were not aligned with what the practitioners preferred.
**Key findings:** From our case survey study, we see that new component origins such as COTS and OSS were popular. Even though traditional alternatives such as in-house and outsourcing were considered more often in the decisions, the newer component origins were chosen more often. For example, OSS was chosen almost all the times it was considered in the decision, whereas in-house was chosen only about half of the times. We also noticed that at the beginning of the decision, many criteria were considered to be important. However, at the end of the decision, a key sub-set of the criteria made the difference. In some cases, the criteria that were initially considered important were not taken into account when the final decision was made. When investigating the perception of the decision in retrospect we notice that in some cases, the decision-makers were satisfied with the decision outcome. However, in some cases, the decision-makers were not completely satisfied with the decision outcome. In most cases, the problems were due to the decision-making process itself. For example, huge effort was spent in investigating the vendors but the decision was made to build it in-house. The problems related to decision-making process were identified. Based on this experience the process-line to support decision-makers was proposed in Chapter 5.

5.4 Chapter 5: A Decision-making process-line for selection of software asset origins and components

Based on the investigations in Chapters 2 and 4 a support solution is proposed which supports the decision-making process to select the component origins. The solution proposed in this chapter includes a set of decision-making activities, which are described in the form of a process-line that can be used by decision-makers to build their specific decision-making process. Five case studies were conducted in three companies to validate the coverage of the set of decision-making activities.

We began by validating an existing solution to support decisions to choose component origins [193]. The validation was conducted in two phases. The existing solution was first validated in two cases. In the validation, it was observed that no activity in the existing solution was perceived to be missing, although not all activities were conducted and the activities that were conducted were not executed in a specific order. Therefore, the solution was refined into a process-line approach which increases the flexibility and hence it is better in capturing the differences in the decision-making processes observed in the case studies.
The applicability of the process-line was then validated in three case studies. The process-line consists of thirteen main decision-making activities as follows:

- Identify stakeholders to be involved in the decision.
- Screen and evaluate the suitability of the asset origins.
- Decide criteria from goals.
- Decide on priorities of criteria.
- Decide on how to handle the time aspect.
- Identify and describe the context.
- Look for similar cases in a knowledge repository.
- Decide on property models to use.
- Evaluate the criteria, make estimations using the property models and evaluate the impact of the decision.
- Weigh the estimation results of the selected properties based on the priorities of criteria.
- Make a tentative decision.
- Make a final decision.
- Store the case in the decision knowledge repository.
Key findings: In addition to the process-line, the case studies conducted in this study complement the findings from Chapters 2 and 4. The decision to choose among component origins was investigated through case studies by interviewing all the stakeholders involved in the decision. Whereas in the case survey study in Chapter 4 only one representative of the decision was interviewed. The companies investigated in this study were following most of the activities in the process-line. However, The process-line was perceived to be useful from both a managerial and technical perspective. The practitioners discussed the usefulness of the process-line from the time perspective. For example, if the time to make the decision is short then, such process-line is useful in creating an informed decision-making process. Another perceived use of the process-line was as a tool to improve their decision-making process. The cases investigated in this study have different decision outcomes. For example, in Case 1 and Case 4, the decision to move from COTS to OSS was made. In Case 2 a COTS vendor was chosen and in Case 3 an OSS component was chosen. In Case 5, the decision to not switch to another component was made. The detailed descriptions of the cases and the decisions are provided in Chapter 5. The selected cases had a mix of strategic and tactical decisions. The process-line was aligned with both strategic and tactical decision-making processes followed by the companies.

5.5 Chapter 6: Bayesian synthesis for knowledge translation in software engineering: Method and illustration

Often the aim of SE is to identify or produce best practices, processes, tools and methods. It is important that the best practices are made available to practitioners. Evidence-based software engineering practice encourages integrating practitioner opinions with evidence from literature. This is done to use the evidence from literature and to adapt it to the practitioners’ context using the practitioners’ knowledge and experience. We propose the use of Bayesian synthesis to systematize the integration of evidence from literature and practitioners’ subjective opinion. This contributes towards
decision-making, as most decisions are based on subjective opinions.

**Key findings:** Bayesian synthesis can be summarized in three steps:

- **Prior probability:** Prior probability is formulated by collecting subjective opinions from practitioners and/or researchers.
- **Likelihood:** The likelihood is a summary of evidence from literature.
- **Posterior probability:** Posterior probability is formulated when prior probability and likelihood is combined.

The detailed process to conduct Bayesian synthesis is reported in Chapter 6 along with an example from SE research.

5.6 Chapter 7: Guidelines for knowledge translation in software engineering

The knowledge produced by research studies are not always implemented in practice. The knowledge produced by research studies should be translated beyond the dissemination and towards the application of the knowledge in software engineering practice. The KT framework proposed in this chapter facilitates exchange, synthesis and application of knowledge. The steps (iterative and not sequential) in the KT framework (including Bayesian synthesis steps from Chapter 6) are as follows:

- Entry point for knowledge translation - Need for knowledge implementation.
- Identify, review and select knowledge (Chapter 6).
- Elicit prior (Prior probability) (Chapter 6).
- Assess gaps.
- Align external knowledge to local context.
- Assess barriers and facilitators to knowledge use.
- Select, tailor and implement the KT intervention (Chapter 6).
- Monitor use.
- Evaluate outcomes.
- Sustain use.
Key findings: The KT framework was implemented and evaluated in practice. It was appreciated that the research results were summarized to them and it was considered to be a time efficient process. The KT framework helped in aligning and streamlining the practitioners’ opinions. Most importantly the practitioners identified the usefulness of KT process in supporting decision-making. Particularly when they do not have all the information needed to evaluate all alternatives. The practitioners’ level of confidence increased after the implementation of KT as they were more informed. They were able to have a broader view and were not confined to their own experience and environment.

In summary, the KT framework presented and evaluated in this chapter is:

- Systematic.
- Repeatable.
- Reflective of the needs and mores of the practitioners.
- Iterative.
- Collaborative and consensus-driven.

6 Conclusions and future work

The overall goal of this thesis is to provide decision-support to choose among different component origins. To address this goal, we investigate the existing knowledge and practice. The methods used in this thesis are systematic literature review, case survey, case study and design science. The criteria and existing solutions used to choose between different component origins are identified through a systematic literature review (Chapter 2) and the experiences of conducting the systematic literature review are reported in Chapter 3. The case survey is used to investigate how decisions to choose among component origins are made in practice (Chapter 4). To support decision-makers in creating their own decision-making process, we proposed a process-line in Chapter 5. An application of Bayesian synthesis to integrate the results from literature into practice is illustrated in Chapter 6, which is extended into a KT framework in Chapter 7.

In Chapter 2, we investigated alternative component origins (in-house, COTS, OSS and outsourcing) to build software-intensive systems. The results indicate that the decision is not only between internal (in-
house) and external component origins (COTS, OSS and outsourcing), the decision between the external components (e.g. COTS vs. OSS) is also considered in the primary studies.

The findings of Chapter 4 indicate that the newer component origins such as COTS and OSS are as popular as traditional component origins such as in-house. In the beginning many criteria were considered however, cost remained the most important criterion when the decision was made. The decision-makers’ perceptions on the outcome of the decision indicated challenges in the decision-making process. Furthermore, the existing solutions were not focusing on the decision-making process. Therefore a need to provide a solution to support the decision-making process is identified.

In addition, a deeper investigation of the context or scenarios was not possible as the primary studies did not report sufficient details of the context. In addition, it was not possible to explore the perspectives of all the stakeholders involved in the decision. None of the primary studies and cases in the case survey focused on the perspective of different stakeholders involved in the decision. To address these limitations, we conducted case studies to investigate the decision-making process in its real-life context (software-intensive systems).

Based on the need to support decision-makers in selecting component origins, we propose a process-line which allows the decision-makers to create a tailor-made decision-making process. In addition, to facilitate evidence-informed decisions, we demonstrate the use of Bayesian synthesis and KT in SE research and practice.

As part of future work, we plan to implement the process-line for an on-going decision to select component origin and evaluate it. In addition, we will use the KT framework along with Bayesian synthesis to facilitate evidence-informed decision-making for selecting component origins.
SOFTWARE COMPONENT DECISION-MAKING: IN-HOUSE, OSS, COTS OR OUTSOURCING - A SYSTEMATIC LITERATURE REVIEW

Abstract

Context: Component-based software engineering require decisions on component origins for acquiring components. A component origin is an alternative of where to get a component from.
Objective: To identify factors that could influence the decision to choose among different component origins and solutions for decision-making (e.g. optimization) in the literature.
Method: A systematic review study of peer-reviewed literature has been conducted.
Results: In total we included 24 primary studies. The component origins compared were mainly focused on in-house vs. COTS and COTS vs. OSS. We identified 11 factors affecting or influencing the decision to select a component origin. When component origins were compared, there was little evidence on the relative (either positive or negative) effect of a component origin on the factor. Most of the solutions were proposed for in-house vs. COTS selection and time, cost and reliability were the most considered decision criteria in the solutions. Optimization models were the most commonly proposed technique used in the solutions.
Conclusion: The topic of choosing component origins is a green field for research, and in great need of empirical comparisons between the component origins, as well of how to decide between different combinations of them.

Keywords
Open Source Software (OSS), components off-the-shelf (COTS), in-house, build-or-buy, decision-making, component-based software engineering (CBSE).
Component-based software engineering (CBSE) usually involves assembling and developing reusable components [45]. In this study component is referred to as an asset, which when incorporated into a system adds value (functional and/or non-functional) to the end customer. Hence assembling or developing operating systems components or databases components are out of the scope of this study. Each component can be obtained from a different source, we refer to the sources as component origins in this study.

Four of the widely used component origins are considered in this study as follows:

1. In-house: The component is developed within the company.
2. Components off-the-shelf (COTS): A commercial component is externally bought from another company/vendor.
3. Open Source Software (OSS): An open source component is externally obtained from the OSS community.
4. Outsourcing: The development of a component is outsourced (or subcontracted).

These component origins have been chosen as they are well-defined options related to developing with components (see e.g. [29, 60, 129]). Depending on the goal, the decision-maker might decide to use one of the component origins. For example, if reducing the overall cost of the project is the goal, the decision-maker might select a component origin which is cost effective. However, this decision is not easy due to the uncertainties involved in software projects. Also, there is often more than one goal that needs to be considered in such decisions. In order to make an informed decision, the decision-maker should consider all the factors and their trade-offs that could influence a decision to select a component origin. Factors are project attributes (e.g. time and cost), project activities/processes (e.g. maintenance and integration) and non-technical factors (e.g. technical support and source code availability).

The objective of this chapter is to identify the factors that influence the decision to select a component origin and the solutions on how to make a choice based on the factors (e.g. through an optimization model).

The strategic decision of where to get components from is a complex one, and has major implications that may not be overseen immediately. For example, on the decision level to choose component origin we may begin with a make-or-buy decision, which is to build in-house or to
get the component externally. Then when deciding to get the component externally, additional implications may play into the decision. For example, when outsourcing development to another company located, for example, in a different country, as we are customers, we can still influence the development processes, and have control over the evolution of the component. When deciding to get a COTS component, then we have less influence, but not the challenge of transferring knowledge, learning, etc. that may occur in outsourcing. This indicates that, from a long-term strategic perspective the decision of where to get a component from may have important implications, on for example, effort. Practitioners can benefit from knowing the factors that could influence the decision to choose among different component origins and possible solutions that support decision-making process.

The individual component origins (COTS, OSS, in-house and outsourcing) have already been well explored in a number of existing literature reviews. Hauge et al. [71], for example, conducted a systematic literature review on OSS, and Maras et al. [120] conducted a review on developing software with components such as COTS. The selection of an actual component may take place on multiple decision levels, namely choosing the component origin, choosing the provider, and choosing the actual component. Secondary studies on component provider selection [93] and component selection [128] exist. No systematic literature review addressing decision-making with regard to the component origins was identified, highlighting the need for such a review of the literature. Hence, the inclusion and exclusion criteria focus on the component origin, the decisions on different levels are not independent as is shown in the Figure 6. As there is no synthesis on the component origin decision level, this study focuses on component origin selection. The decisions on how to select and integrate component (after the component origin is chosen) are not considered within the scope of this study.

Systematic review studies are a means to aggregate evidence through a scientific and repeatable process. A systematic literature review was used as the research method to find the factors and solutions in the literature using the guidelines in [190] and [97]. The guidelines in [46, 85, 186] and [138] are used for quality assessment, data analysis, classification of research types and validation of the study respectively. Systematic review studies are a means to aggregate evidence through a scientific and repeatable process. The specific contributions achieved through the systematic review are the identification of:
• C1: the research types and methods used to evaluate the factors and solutions.
• C2: the factors considered in comparing two or more component origins in a way that facilitates decision-making.
• C3: the solutions to decision-making (e.g. optimization).
• C4: the research gaps and future directions.

The remainder of this chapter is structured as follows: Section 2 presents related work on existing literature reviews closely related to our study. Section 3 presents the research method. The results are presented in Section 4 and discussed in Section 5. Section 6 concludes the chapter.

2 RELATED WORK

Three different levels of decisions are recognized as shown in Figure 6. The light grey block represents the decision level. The white block represents the decision space where the different options (dark gray) are traded-off. The arrows represent the decision control flow. As seen in Figure 6 the decision is not taken on the same level, the control shifts across different levels. For example, once the component origin is chosen, the provider is selected followed by the actual component selection. However, while evaluating the components, the decision-maker might want to consider alternative provider or even an alternative component origin. This indicates that once a decision is taken at a decision level, it can be revised by the activities in the next decision level. The decision on each decision level is a research study in itself.

The focus is delimited to decision-making choosing between the four component origins: OSS, COTS, developing the components in-house, and outsourcing the development of the components.

Secondary studies on decision level for provider [93] and component selection [128] have been conducted. No systematic literature review on the topic of component origin selection while using a systematic approach for study identification (snowball sampling [86] and database search [196]) was identified. Primary studies to select component origins exist, although the research problem to select component origin is not among the frequently researched topics [181]. In the study [181] different research topics on component-based software engineering from its inception were reviewed. The results in [181] indicate that none of the most researched topics is based on decision-making to select the component origin. However, industrial research partners have corrob-
orated that it is an area where improvements can be made. Based on the industry interest and the limited number of studies in the area, we conclude that it is an area worthy of further research.

Secondary and primary studies have considered the decision to select the provider and to select components after the component origin is chosen. Examples of secondary and primary studies outside our scope, but still related to decision-making are listed in Tables 1 and 2. The secondary and primary studies related to the provider and component selection decision levels are presented in Table 1. Primary studies in Table 1 are a subset of existing studies related to the provider and component selection. While, Table 2 provides a list of secondary studies related to the different topics with respect to the individual component origins. The primary studies that compare the component origins are included in the result of the review study and are mentioned in the result section.

Table 1: Secondary and primary studies related to the different decision levels

<table>
<thead>
<tr>
<th>Decision Level</th>
<th>Secondary studies</th>
<th>Primary studies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provider selection</td>
<td>Supplier selection [93]</td>
<td>Vendor selection [75], community selection [194]</td>
</tr>
</tbody>
</table>

Using the relevant keywords of each comparison origin in Scopus, in combination with the research method of systematic literature review
Table 2: Systematic reviews related to individual component origins

<table>
<thead>
<tr>
<th>Component origins</th>
<th>Secondary studies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open source</td>
<td>Software evolution [20, 175], software quality (reliability, interoperability) [172], development processes [3], software quality (maintainability) [171], OSS in commercial development [71, 83], challenges on integrating open and inner source [167], empirical methods used in OSS research [168]</td>
</tr>
<tr>
<td>COTS</td>
<td>–</td>
</tr>
<tr>
<td>In-house</td>
<td>Synthesis of research on component based software engineering [120]</td>
</tr>
<tr>
<td>Outsourcing</td>
<td>–</td>
</tr>
</tbody>
</table>

and systematic mapping studies, COTS and outsourcing of components did not yield any results. As the focus of this study is on decision-making in choosing the component origin, no deeper analysis of the studies is provided.

3 Method

3.1 Need for the review

While we could not find any existing review on decision-making to choose a component origin, practitioners would greatly benefit from being aware of possible solutions and relevant factors for choosing between the component origins. This review is conducted to explore the factors that are considered in decisions to choose a component origin and also to understand the existing solutions available to guide the decision-making process. The following research questions were used to drive the review study:

- **RQ1**: What are the research types, methods and quality of the contributions? Through this research question, the identified papers are classified according to the six types of research defined by Wieringa [186]: evaluation research, validation research, solution proposal, philosophical papers, opinion papers, and experience reports. In addition, the research methods, rigor and relevance used by the primary studies to obtain the results are discussed (see contribution C1).

The answer to this question allows to determine the strength of evidence of the existing literature.
• **RQ2:** What are the different factors that influence the decision to choose among different component origins? Four component origins are of particular interest in this study. In order to choose between component origins, comparisons may be carried out. The goal of this research question is to identify the different factors that are compared to evaluate the different component origins. The list of factors considered in the decision to choose a component origin and the evidence supporting the results can be used by decision-makers in their decisions (see contributions C2).

• **RQ3:** What solutions have been proposed to choose the component origin? With the influencing factor as input, different solutions might be proposed to make a decision. For example, prioritization-based approaches making trade-offs, or models that optimize with regard to a specific criterion under given constraints. The available solutions will allow researchers and practitioners to be aware of existing the work. The limitation of the solutions will point the direction future work in proposing new solutions (see contributions C3 and C4).

### 3.2 Study identification

For study identification, two approaches were used. First, the first and second author collaborated on the identification of studies through snowball sampling. Then the third author, independently of the other authors, conducted a database search to complement the snowball sampling. The research on decision-making to choose a component origin is relatively new. In addition, the terminology is not yet established or consistently used in the research papers. Hence, snowball sampling is chosen as the main method instead of database search. That is, the snowball sampling is conducted first and then database search is conducted to validate the snowballing process and to ensure the coverage of the primary studies. As suggested in [190] all papers for a search could be identified only using snowball search, though to add confidence to the study identification process, also a database search was conducted. We evaluated the preliminary search results from snowballing sampling and database search (Chapter 3) and conclude that the implemented search strategy was efficient and reliable. However, it was noted that the primary studies that were identified were not from conferences related to component-based software engineering. To ensure that this is not a threat to validity, a manual search of the proceedings for the component-based software
engineering conference (CBSE) between 2004-2015 was conducted. No new papers were found in the manual search.

3.2.1 Inclusion and exclusion criteria

The same inclusion and exclusion criteria have been used for all searches, i.e. start set in snowballing, backward snowballing, forward snowballing, and the database search. These are explained in the following sections.

In order to include a paper, at least one of the following inclusion criteria should be fulfilled:

- Papers discussing the decision-making process for selecting a component origin, i.e. answer the question: Why select the component origin? Not how to select a component once the component origin is decided.
- Papers comparing two or more component origins, e.g. papers discussing cost, time, scope or quality trade-offs (or proxies of these four aspects) for two or more component origins are of particular interest.
- Papers proposing solutions that support the decision-making process in the selection of the component origin in relation to other component origins, i.e. answer the question: How is it possible to make the decision process for selecting a component origin more effective and efficient?
- All papers should be peer-reviewed.

Papers are excluded if they fulfil any of the following exclusion criteria:

- Papers that are not related to CBSE.
- Papers discussing architectural aspects of CBSE, such as papers answering questions: How to integrate the components in the system once the component origin is decided?
- Papers discussing development solutions (Programming) for CBSE.
- Papers discussing adoption of platforms, technology, IT services or operating systems.
- Papers discussing end-user adoption of software packages.
• Papers that discuss only one component origin without contributing to the decision-making process on why a specific component origin should be used or not.

• Duplicate reports of the same study.

• Grey literature.

• Articles not written in English.

3.2.2 Snowball sampling

The snowball search consisted of the following steps. First, a start set of papers was established through a search in Google Scholar and the application of the inclusion and exclusion criteria. Thereafter, backward snowballing (based on reference lists) and forward snowballing (based on citations) were conducted. The snowball procedure was conducted jointly by the first two authors of the paper.

Establish a start set: Since we are interested in looking at how the different component origins are compared and how the decisions are made, the following search strings were generated:

• In-house vs. outsourcing and software
• In-house vs. COTS
• In-house vs. OSS
• COTS vs. OSS
• COTS vs. outsourcing and software
• Outsourcing vs. OSS and software

Papers were retrieved from Google Scholar as an unbiased start set was wanted, i.e. not papers available in a specific publisher’s database. A cutoff was chosen to get a start set. The first 10 entries of each search string were considered. Among the 60 entries found, there were many entries that were not software related, particularly for the following search strings: In-house vs. outsourcing, Outsource vs. OSS and COTS vs. Outsourcing. The common keyword in the above search strings was “outsource”. As outsourcing is done in various domains, “software” was added to the search string to restrict the papers to the software domain.

In total, the search resulted in 90 entries for which the inclusion and exclusion criteria were applied. The inclusion and exclusion of papers for the start set was carried out in two phases. The objective was to find at least one paper for each search string.
Start set - Phase 1: Both authors reviewed the meta-data (title, abstract and introduction) independently for all 90 entries. The decision point was noted while reviewing the papers, where decision point refers to the point at which the decision of inclusion or exclusion was made, in this case, title, abstract or introduction. The paper was either tentatively accepted for Phase 2 or rejected based on the inclusion/exclusion criteria. Both authors performed the inclusion and exclusion independently and once the authors completed this phase a review meeting was held to compare the review results. The decision rules shown in Table 22 are defined for inclusion and exclusion according to the outcome of the review process conducted by the first two authors. An inclusive approach has been adopted, i.e. when in doubt the paper has been included for further evaluation in Phase 2.

<table>
<thead>
<tr>
<th>Case</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Both authors accept a paper</td>
<td>Include the paper for next step</td>
</tr>
<tr>
<td>Both authors reject a paper</td>
<td>Exclude the paper</td>
</tr>
<tr>
<td>Either one of the authors accepts a paper</td>
<td>Include the paper for next step</td>
</tr>
</tbody>
</table>

Start set - Phase 2: Often the title, abstract, and introduction are insufficient to get a full and comprehensive picture of the articles, and hence a final decision was reached by further reading the papers. The papers were reviewed and a summary of the papers was reported along with the final decision made. The decision was either to include the paper in the start set or exclude it. This process was conducted by the first author and the second author reviewed the report and either agreed or disagreed to the decisions made. At the end of this step, the papers were finally included in the start set when both authors agreed to include the paper. If one of the authors disagreed for a particular paper, the paper was discussed until the authors reached a consensus. After completing Phase 2, a total of five papers were included, which were used to conduct backward as well as forward snowballing (cf. [190]).

Backward snowballing: Backward snowballing is the process of reviewing the references of a relevant paper. Backward snowballing started by randomly selecting a paper from the start set. The process and order of reviewing the references was as follows:

1. Title of the referenced paper.
2. The reference context of the paper. The reference context refers to the text surrounding the reference citation within the primary study.

3. Abstract of the referenced paper.

Phases 1 and 2 were repeated on the references of the start set papers and decision points (title, abstract or full text) were noted. The first two authors performed this step independently. At the end of the reviewing process, a second review meeting was conducted to compare the results. Again the same steps were followed as in Table 22 and the same inclusion and exclusion criteria were applied.

**Forward snowballing:** Forward snowballing is the process of reviewing the citations of a relevant paper. The starting point of forward snowballing was the randomly selected paper from the start set. For each paper citing the paper in the start set, the following was reviewed:

1. Title of the paper citing.
2. Abstract of the paper citing.
3. The reference context to the paper being cited.

Phases 1 and 2 were conducted for the paper citing the papers in the start set and decision points (title, abstract or full text) were also noted. The first two authors performed this step independently. And at the end of this process, a third review meeting was conducted to compare the results. Again the same steps were followed as in Table 22. Backward and forward snowballing were performed for all papers in the start set. The papers accepted in backward and forward snowballing were added to the final set of papers to be included in the review. Backward and forward snowballing were repeated for each newly added paper in the final set of papers until no new papers were found.

Initially, the backward and forward snowballing was done independently by both first and second author. As the reviewing continued the decisions (accept/reject) made by the authors were same, i.e. the authors accepted/rejected same papers. From there onwards only the first author conducted the snowballing and the second author reviewed the snowballing process reported by the first author.

Figure 7 shows the number of papers retrieved at each stage through the snowball sampling process.
3.2.3 Database search

The main purpose of the database search was to increase the confidence in the selection of papers. Thus, the database search is conducted as a complementary search to the snowballing procedure.

The search terms were divided into population, intervention, comparison, and outcome (PICO). The population was restricted by searching for “software (X)” and “component (Y)”.

The intervention relates to making trade-offs, decisions, or selecting a component origin, search keywords are: “trade-off (I₁)”, “decision (I₂)”, and “selection (I₃)”

Outcome for the following component origins:

- A: (COTS OR “components off the shelf” OR “component off the shelf”)
- B: (In-house OR inhouse)
- C: (open-source OR “open source” OR OSS)
- D: (outsource OR out-source OR outsourcing OR “third-party”)

Comparison relates to the different component origin combinations. Given that different combinations of component origins are of interest, the following strings are constructed:

- E: A AND (B OR C OR D)
- F: B AND (C OR D)
- G: C AND D

This resulted in the following search strings:
• **Search 1**: \((X \text{ OR } Y) \text{ AND } (I_1 \text{ OR } I_2 \text{ OR } I_3) \text{ AND } E\)

• **Search 2**: \((X \text{ OR } Y) \text{ AND } (I_1 \text{ OR } I_2 \text{ OR } I_3) \text{ AND } F\)

• **Search 3**: \((X \text{ OR } Y) \text{ AND } (I_1 \text{ OR } I_2 \text{ OR } I_3) \text{ AND } G\)

The databases searched were Scopus and Inspec/Compendex, i.e. three index databases were used that encompass publications from Elsevier, IEEE, ACM, etc. The number of search results per search string and database is shown in Table 4.

<table>
<thead>
<tr>
<th>Database</th>
<th>Search</th>
<th>Hits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scopus</td>
<td>Search 1</td>
<td>60</td>
</tr>
<tr>
<td>Scopus</td>
<td>Search 2</td>
<td>101</td>
</tr>
<tr>
<td>Scopus</td>
<td>Search 3</td>
<td>51</td>
</tr>
<tr>
<td>Inspec/Compendex</td>
<td>Search 1</td>
<td>71</td>
</tr>
<tr>
<td>Inspec/Compendex</td>
<td>Search 2</td>
<td>110</td>
</tr>
<tr>
<td>Inspec/Compendex</td>
<td>Search 3</td>
<td>38</td>
</tr>
</tbody>
</table>

**Figure 8: Database search process**

**Applying the inclusion and exclusion criteria:** The inclusion and exclusion criteria were applied by the third author independently. The results of the database search were reviewed by the first author. In cases of disagreements, the authors discussed the papers until the disagreements were resolved.
3.2.4 Combined results

In total (snowballing + database search), 24 unique papers were identified. The number of included papers unique to snowballing, database search as well as the papers found by both approaches are shown in Table 5.

Table 5: Search type and number of relevant papers

<table>
<thead>
<tr>
<th>Search type</th>
<th>Number of papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snowball sampling</td>
<td>11</td>
</tr>
<tr>
<td>Database search</td>
<td>4</td>
</tr>
<tr>
<td>Both</td>
<td>9</td>
</tr>
</tbody>
</table>

3.3 Data extraction and classification

A data extraction form was designed to extract the data in an explicit and consistent way. Which data to extract was driven by the research questions. Table 6 provides an overview of the data extracted, and how the data fields link to the research questions.

Table 6: Data extraction form

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
<th>RQs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meta-information</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Study focus and methods</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Research facet</td>
<td>Classification of research type based on Wieringa et al. [186]</td>
<td>RQ1</td>
</tr>
<tr>
<td>Factors</td>
<td>Factors considered for decision-making and comparison (emerging)</td>
<td>RQ2</td>
</tr>
<tr>
<td>Solution</td>
<td>Description of the solution to make a decision (emerging)</td>
<td>RQ3</td>
</tr>
</tbody>
</table>

The research type was classified according to Wieringa et al. [186]; Six types (evaluation research, validation research, solution proposal, philosophical papers, opinion papers, and experience reports) were distinguished. Only evaluation and validation research are of empirical nature, where evaluation research is rooted in practice, and validation research is conducted in laboratory settings. Hence, the classification shows the strength of evidence by illustrating the ratio between the research types.
The first author designed the extraction form, and it was reviewed by both the second and third author prior to extraction. In addition, as mentioned in Section 3.2 full text of the papers was read, this gave an overview of the depth and breadth of the available data. Thereafter, the first and third authors divided the papers and extracted the information from the primary studies. The first author extracted data from 18 papers, and the third author extracted data from 6 papers. The extraction was done by identifying the text segments that were related to the research questions. In total 96 text segments were extracted. The text of each text segment was recorded in data extraction form. In most cases, the exact text was recorded with the exception of paraphrasing the text that contained other irrelevant information.

3.4 Quality assessment

The quality of the empirical studies is assessed by evaluating the rigor and relevance. Rigor and relevance are considered as the two main criteria for evaluating the study quality [85]. The guidelines in [85] have been used to score the rigor and relevance of empirical studies. The details of the rubrics used to evaluate are presented in Sections 3.4.1 and 3.4.2. It is to be noted that not all primary studies included in this study are empirical, hence the quality of only a subset of primary studies is evaluated.

3.4.1 Rigor

The rigor is assessed by determining the degree to which the study context (C) validity threats (V), and research design (R) are described in the primary studies. The rigor aspects are rated as strong (1), medium (0.5) or weak (0).

**Strong description (1):** The rigor aspect is rated as strong when the following conditions are fulfilled:

1. **Context (C)** - C equals to 1 if the context describes sufficient details of context such as the size of the company/organization, the profile, size, and domain of the projects investigated.

2. **Validity threats (V)** - C equals to 1 if all the applicable validity treats are considered, and appropriate actions are taken to mitigate the threats.

3. **Research design (RD)** - The research design is rated as 1 when the research methodology is executed following the best practices.
Also, it should report details such as data collection, sampling, analysis, treatments, and outcome variables.

**Medium description (0.5):** The rigor aspect is rated as medium (0.5) if any one of the descriptions mentioned as strong description is missing. **Weak description (0):** The rigor aspect is rated as low (0) if no descriptions that are mentioned as strong descriptions are provided.

### 3.4.2 Relevance

Relevance evaluation consists of evaluating the realism of the environment in which the results are obtained. Evaluating the subjects/users (S), context (C) and scale (S) determines the relevance. Also, the suitability of the research method (R) used to obtain the results is evaluated. The relevance aspects are rated as either high (1) or low (0).

1. **Subjects/users (SU) – Contribute to relevance (1):** S equates to high (1) when the subjects/users used in the evaluation are representative of the intended users of the process, i.e. industrial practitioners.
   **Do not contribute to relevance (0):** S equates to low (0) when the subjects/users used in the evaluation are not representative of the intended users of the process, i.e. students, researchers or subjects are not mentioned.

2. **Context (C) – Contribute to relevance (1):** C equates to high (1) when the context in which the results are obtained is representative of the intended usage context, i.e. industrial context/setting.
   **Do not contribute to relevance (0):** C equates to low (0) when the context in which the results are obtained is not representative of the intended usage context, i.e. laboratory setting or other simulations.

3. **Scale (S) – Contribute to relevance (1):** S equates to high (1) when the scale of application used in evaluation is of realistic size, i.e. projects investigated are of an industrial scale or the number of interviews in the case study, or number respondents of the survey are of realistic size.
   **Do not contribute to relevance (0):** S equates to low (0) when the evaluation is performed using applications of unrealistic size, i.e. down-scaled or toy example.

4. **Research Method (RM) – Contribute to relevance (1):** RM equates to high (1) when the research method used allows to investigate
real situations, and that is relevant for practitioners, i.e. industrial survey or case study.

**Do not contribute to relevance (0):** RM equates to low (0) when the research method used does not lead itself to investigate real situations or if no research method is used, i.e. laboratory experiment (software or human subjects).

The results of the quality assessment of the empirical papers are provided in Section 4.1.3.

### 3.5 Analysis

Thematic analysis is used for analyzing the data extracted from the primary studies. The thematic analysis identifies, analyses and reports patterns (themes) within data [46]. After the text segments are extracted, they are coded into themes by the first and the third author independently using the coding rules defined in Table 7. Some text segments can be assigned to more than one theme, for example, if the text compares the integration effort for COTS and OSS and also discusses integration effort as an influencing, the text segment is coded into the theme called "effort" and "integration". The themes that include only one text segment are not considered since the analysis aims to identify recurring themes.

<table>
<thead>
<tr>
<th>Themes</th>
<th>Rules</th>
<th>Example of Text Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Description of the time in terms of the duration</td>
<td>“Time to market was the common project profiles which was most emphasized” [114]</td>
</tr>
<tr>
<td>Cost</td>
<td>Any costs in terms of the money involved in a project</td>
<td>“Relying on OSS helped the project to overcome budget related issues” [131]</td>
</tr>
<tr>
<td>Effort</td>
<td>Effort in terms of man-hours to perform an activity</td>
<td>“Unsatisfactory integration effort estimation is a common risk of OTS components” [114]</td>
</tr>
<tr>
<td>Quality</td>
<td>Discussing quality attributes such as reliability and security</td>
<td>“Consumers make product quality inferences based on cues such as brand and store name through a process called affect-referral.” [165]</td>
</tr>
<tr>
<td>Market Trend</td>
<td>The ability of the component to evolve based on the market trend</td>
<td>“Market rules force COTS vendors to release a new version of their products even if they know they are not fully working.” [51]</td>
</tr>
<tr>
<td>Source Code Availability</td>
<td>Text describing the availability of source code</td>
<td>“Open access to source code allows greater opportunities for customizing OSS software to the specific needs” [125]</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Technical Support</td>
<td>Discussions on the support provided or available for fixing bugs and required changes in functionality</td>
<td>“As soon as the attack the OSS community fixes or countermeasures are devised quickly” [77]</td>
</tr>
<tr>
<td>License</td>
<td>Text describing the license agreement, obligations or restrictions</td>
<td>“License requires that modified versions of the software also be open (often referred to as a “copy-left” provision)” [165]</td>
</tr>
<tr>
<td>Integration</td>
<td>Text describing the integration activity</td>
<td>“Required integration work is used in evaluation criteria” [73]</td>
</tr>
<tr>
<td>Requirement</td>
<td>Discussion on requirements as a phase and nature of requirements (complex or unique)</td>
<td>“Recommended changes to a customer’s requirement when we felt that by doing so we could satisfy it with an OSS component.” [131]</td>
</tr>
<tr>
<td>Maintenance</td>
<td>Related to upgrading the system and maintaining system stability</td>
<td>“The OSS user is not obliged to upgrade.” [51]</td>
</tr>
</tbody>
</table>

Table 7: Rules defining the themes

The coding rules for assigning text segments to the themes were commonly defined by the first and the third author. Such rules reduce the threat of bias and mistakes during the data extraction. The text segments from the primary studies are summarized and coded into existing themes, new themes are added and defined whenever necessary in an inductive manner. The text segments coded into the themes is compared within and across the themes to compare and contrast the properties of different themes. In order to keep the traceability between the data and the primary studies, the text relevant for the extraction (text segments) within primary studies is highlighted, and tagged by a comment linking it to the themes using Adobe reader. This helps in checking the reliability and revising the extraction if needed.

In total 11 themes were identified which were the factors that could influence the decision to choose a component origin.
3.6 Validity threats

Petersen and Gencel [138] proposed to discuss five types of validity threats, namely descriptive validity, theoretical validity, generalizability, interpretive validity, and reliability in software engineering research in general.

3.6.1 Descriptive validity

Descriptive validity concerns threats to the ability to accurately capture and describe observations made.

This threat is mainly relevant in the data extraction activity. To reduce the threat, a data extraction form was designed, which was then reviewed by two experienced researchers having conducted a number of systematic literature reviews (authors two and three). Furthermore, it helped to keep the traceability between the original text of the papers and the data extracted by highlighting and tagging the text within the document. That way it was always possible to return to the source when in doubt. Hence, this threat is well controlled.

3.6.2 Theoretical validity

Theoretical validity threats may lead to a lack of ability to capture what is intended to be captured, e.g. due to confounding factors that the researcher may not be aware of.

Study identification and sampling: In this study, different search strategies were utilized, i.e. snowball sampling and database search to increase the reliability of the study identification. Furthermore, multiple researchers have been involved in the selection of studies. As the complementary database search only yielded a small set of additional papers, the confidence in the sample is increased. The database search added papers in the component origins comparison category for which there was no paper found through the snowballing process.

Researcher bias: In the data extraction as well as study selection researcher bias is a threat. To reduce bias, more than one researcher was involved in each step. The snowball sampling procedure was conducted by two authors. The result of the database search was checked by the first author. In addition, during the data extraction when in doubt both researchers were in the same room during the extraction to immediately discuss the doubt. Hence, the threat due to the extraction not being reviewed is minimized.
3.6.3 Generalizability

The generalizability is concerned with generalizing within groups (i.e. the same community or organization) and between groups (i.e. between different communities and groups).

**Internal generalizability:** The ability to generalize within the group (decision-making for choosing a component origin) is highly dependent on the studies published (e.g. the different contexts studied). Hence, internal validity is limited to the research contexts of the studies. In this case, limited information is available on the context reported, hence it is impossible to make a judgment regarding this. Furthermore, the total number of primary studies is low. Per category (comparison of component origins), only a small number of papers was identified. In particular, in-house vs. OSS and in-house vs. outsourcing are only represented by two papers each.

**External generalizability:** To determine whether the same approaches are also applicable when having a different component origin to choose from is a subject for further investigation and cannot be determined here.

3.6.4 Interpretive validity

Interpretive validity is concerned with threats related to the ability to draw objective and valid conclusions based on the data collected, i.e. this threat is relevant when interpreting the findings. Coding rules were defined and regular meetings were conducted during the data analysis to ensure that the data is interpreted in the same way by all authors. Hence, the threat was mitigated by involving multiple researchers in the interpretation and discussion of the findings.

3.6.5 Reliability

In order to achieve reliability, the research steps must be repeatable. Repeatability is the ability of other researchers replicating the results. The detailed steps taken in the searches are reported. For example, the search strings used, databases used and inclusion/exclusion criteria are documented, hence reliability is increased. Repeatability of the data extraction is important. The relevant text segments were highlighted and tagged with appropriate codes within the document. This ensures traceability and repeatability of the data extraction process.
The answers to the research questions are provided in this section. The contribution of the primary studies with respect to the comparison categories is shown in Table 8.

<table>
<thead>
<tr>
<th>Component origin</th>
<th>Count</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>OI-OC</td>
<td>8</td>
<td>[22, 39, 41, 88, 90, 109, 143, 162]</td>
</tr>
<tr>
<td>OC-OS</td>
<td>6</td>
<td>[51, 77, 79, 112, 114, 125]</td>
</tr>
<tr>
<td>OS</td>
<td>5</td>
<td>[36, 100, 152, 165, 178]</td>
</tr>
<tr>
<td>OI-OS</td>
<td>1</td>
<td>[173]</td>
</tr>
<tr>
<td>OI-OO</td>
<td>2</td>
<td>[101, 103]</td>
</tr>
<tr>
<td>OI-OC-OS</td>
<td>2</td>
<td>[73, 131]</td>
</tr>
</tbody>
</table>

OI-OC: In-house vs. COTS, OC-OS: COTS vs. OSS, OS: Only OSS, OI-OS: In-house vs. OSS, OI-OO: In-house vs. outsourcing.

4.1 Research types, methods and quality (RQ1)

In this section we discuss the research types, the research methods and the quality of the primary studies. Section 4.1.1 describes the research types, Section 4.1.2 describes the research methods and Section 4.1.3 describes the results of quality assessment of the primary studies.
4.1.1 Research types

Identifying the research types gives a good overview on the strength of evidence of the primary studies. Table 9 provides an overview of the research types of the primary studies based on Wieringa et al.’s classification [186]. Evaluation research is an investigation of a problem in practice or implementation of a technique in practice. Solution papers are papers proposing solutions and arguing its relevance. Philosophical papers are papers that sketch a new way of looking at things. Validation research investigates the properties of a solution proposal that has not been implemented in practice and opinion papers are papers that contain opinions of the authors. The most common type is “Evaluation research”, which indicates that the data collected is originating from the software engineering practice and is empirical. Validation research also represents empirical studies, indicating that 9 out of 24 studies are empirical. Also, 8 out of 24 papers are neither empirical studies nor solutions papers.

<table>
<thead>
<tr>
<th>Research types</th>
<th>Count</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation research</td>
<td>8</td>
<td>[22, 36, 73, 109, 112, 114, 165, 173]</td>
</tr>
<tr>
<td>Solution proposal</td>
<td>7</td>
<td>[39, 41, 88, 90, 103, 143, 162]</td>
</tr>
<tr>
<td>Philosophical papers</td>
<td>4</td>
<td>[77, 100, 125, 178]</td>
</tr>
<tr>
<td>Validation research</td>
<td>1</td>
<td>[101]</td>
</tr>
<tr>
<td>Opinion papers</td>
<td>2</td>
<td>[51, 79]</td>
</tr>
<tr>
<td>Experience papers</td>
<td>2</td>
<td>[131, 152]</td>
</tr>
</tbody>
</table>

Figure 9 shows how the primary studies are distributed in relation to the comparison of component origins. Even though the highest number of studies is classified as evaluation research, the evaluation research studies are distributed among “In-house vs. COTS” to “In-house vs. COTS vs. OSS”, which means that in each category only a few studies are rooted in practice. Of the solutions proposed for “In-house vs. COTS”, only two are evaluated in practice [88, 90]. It is also interesting that, even though contributing with fewer studies, all research types proposed by Wieringa et al. [186] could be identified in the set of primary studies.
Figure 9: Research types and studied combinations of component origins

4.1.2 Research methods

Research methods for the evaluation and validation research studies (see Table 9) have been studied. Table 10 provides an overview of the research methods of the primary studies. The most common method

<table>
<thead>
<tr>
<th>Research type</th>
<th>OI-OC</th>
<th>OC-OS</th>
<th>OS</th>
<th>OI-OS</th>
<th>OI-OO</th>
<th>OI-OC-OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation research</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Solution proposal</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Philosophical papers</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Validation research</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Opinion papers</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Experience papers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

OI-OC: In-house vs. COTS, OC-OS: In-house vs. OSS, OC-OS: COTS vs. OSS
OI-OO: In-house vs. Outsourcing, OS: Only OSS, OI-OC-OS: In-house vs. COTS vs OSS

is “Survey”, which indicates that the results are generalized across different domains. Figure 10, shows the distribution of research methods in relation to the comparison categories studied. The surveys address “COTS vs. OSS” and “OSS” adoption decisions. The comparison category “In-house vs. COTS” has a mix of case study and interview study methods. However, for the remaining comparison categories primary studies with same research methods are addressing them. The number of primary studies using the same research method per comparison category is also low (maximum two per category).

Among the four surveys conducted as shown in Table 10, only two primary studies have reported domain details [112, 114]. The domains covered by the two surveys is provided in Table 11. Among the interview studies [109] and case studies [22], only one study each provides domains details which is also provided in Table 11.

The primary studies [114] and [112] belong to “COTS vs. OSS” comparison category and the primary studies [22] and [109] belong to “In-house vs. COTS” comparison category. From the results in Table 11 we
can see that the results for “In-house vs. COTS” and “COTS vs. OSS” comparison categories are from different domains. Hence, the results are generalized with respect to the domains.

4.1.3 Quality assessment results

Table 12 shows the results of quality assessment. The empirical studies are evaluated with respect to rigor and relevance based on the rubrics discussed in Section 3.4. Note, the strength of evidence of empirical studies is only assessed, as the results of non-empirical papers such as opinion and philosophical papers are not based on any evidence. The total rigor and relevance is a sum of the scores assigned to each rigor and relevance aspect. If the total of rigor score is 1.5 or greater, the rigor is said to be high. And if the sum is less that 1.5, rigor is low. Similarly when the sum of the relevance score is 2 or above, the relevance is high, and when the sum is below 2 the relevance is low.

Overall most of the empirical studies included in this study have high relevance. Except for the empirical study [101], which is performed in an academic context. Four out of nine studies are evaluated as having high rigor. Five out of nine studies are low in rigor. Most of the studies do not report any validity threats that might be taken into account. The context descriptions are not completely described. For example, in studies [36, 101] only the company size is discussed, the size or domains of the projects are not mentioned. Also, the research de-
Table 12: Rigor and relevance scores of the empirical studies

<table>
<thead>
<tr>
<th>Ref.</th>
<th>C</th>
<th>V</th>
<th>RD</th>
<th>Total Rigor</th>
<th>SU</th>
<th>C</th>
<th>S</th>
<th>RM</th>
<th>Total Relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>[114]</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
<td>High (2.5)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>High (4)</td>
</tr>
<tr>
<td>[112]</td>
<td>0.5</td>
<td>1</td>
<td>0.5</td>
<td>High (2)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>High (4)</td>
</tr>
<tr>
<td>[73]</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td>Low (1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>High (4)</td>
</tr>
<tr>
<td>[22]</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td>Low (1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>High (4)</td>
</tr>
<tr>
<td>[165]</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Low (1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>High (4)</td>
</tr>
<tr>
<td>[109]</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>High (3)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>High (4)</td>
</tr>
<tr>
<td>[36]</td>
<td>0.5</td>
<td>0.5</td>
<td>1</td>
<td>High (2.5)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>High (4)</td>
</tr>
<tr>
<td>[101]</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>Low (0.5)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Low (0)</td>
</tr>
<tr>
<td>[173]</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Low (1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>High (4)</td>
</tr>
</tbody>
</table>

C - Context, V - Validity threats, RD - Research design, SU - Subjects/users, S - Scale, RM - Research Method

Sign descriptions are not completely reported in primary studies. For example, the study presented in [73] mentions that semi-structured interviews are conducted. However, the domain and the sampling of the interviewees are not mentioned and in the study presented in [22] the number of interviews and the roles interviewed are missing.

4.2 Influencing factors (RQ2)

The primary studies that discuss factors are mapped to the component origin comparisons as shown in Table 13. The “count” column indicates the number of studies specifically answering RQ2 out of the total number of primary studies found in this review study.

The majority of primary studies considered OSS, followed by COTS, and In-house. Although OSS is discussed in most studies, only one study considers the comparison between in-house and OSS. The most researched combination of component origins is “COTS vs. OSS”. There were no papers identified for the comparisons “In-house vs. Outsourcing”. It is important to note that, the primary studies reported in Table 13 only discuss factors that are used in evaluation to choose a component origin (RQ2). They do not however discuss how the decision should be made or what decision-making process to follow (RQ3), this
Table 13: Primary studies discussing influencing factors for component origin (RQ2)

<table>
<thead>
<tr>
<th>Component origin</th>
<th>Count</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>OI-OC</td>
<td>2/8</td>
<td>[22, 109]</td>
</tr>
<tr>
<td>OC-OS</td>
<td>6/6</td>
<td>[51, 77, 79, 112, 114, 125]</td>
</tr>
<tr>
<td>OS</td>
<td>5/5</td>
<td>[36, 100, 152, 165, 178]</td>
</tr>
<tr>
<td>OI-OS</td>
<td>1/1</td>
<td>[173]</td>
</tr>
<tr>
<td>OI-OO</td>
<td>0/2</td>
<td>•</td>
</tr>
<tr>
<td>OI-OC-OS</td>
<td>2/2</td>
<td>[73, 131]</td>
</tr>
</tbody>
</table>

OI-OC: In-house vs. COTS, OC-OS: COTS vs. OSS, OS: Only OSS, OI-OS: In-house vs. OSS, OI-OO: In-house vs. outsource, OI-OC-OS: In-house vs. COTS vs OSS.

is discussed in Section 4.3. Therefore even though there are no papers discussing the comparison of in-house and outsourcing, the solutions to choose either in-house or outsource exists which are discussed in 4.3. Hence outsourcing is not discussed in Sections 4.2.1, 4.2.2 and 4.2.3.

The relevant text segments from the primary studies describing the influencing factors were coded into themes as shown in Table 14. The themes are then grouped into higher-level themes which are also shown in Table 14.

The project metrics factors are factors that can be quantitatively measured and estimated. These factors can be estimated using estimation models. The external factors are those that are depended on the provider (COTS vendor or OSS community). These factors usually define the access and control agreement. The software development activities that affect the decision are grouped into software development activity factors.

Sections 4.2.1, 4.2.2 and 4.2.3 provide the description of the factors (bold) and code (italics). The tables (15, 16 and 17) indicate the primary studies addressing the factors for each comparison category. The figures (11, 12 and 13) represent the summary of the description is terms of the effect the factors have on the component origin. It also indicates the level of evidence supporting the effect.

4.2.1 Project metrics factors

The project metrics factors extracted from primary studies with respect to the comparison categories are presented in Table 15, along with the count (C) of the primary studies reporting the factor and the number of primary studies that are empirical (E). The project metrics factors are
Table 14: High-level themes, themes and codes

<table>
<thead>
<tr>
<th>High-level themes</th>
<th>Themes/factors</th>
<th>Text segments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project metrics factors</strong></td>
<td>Time</td>
<td>Time to test and integrate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Time to market</td>
</tr>
<tr>
<td></td>
<td>Cost</td>
<td>Cost of components</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total cost of ownership</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cost of replacing components</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maintenance cost</td>
</tr>
<tr>
<td></td>
<td>Effort</td>
<td>Selection and integration effort</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Development effort</td>
</tr>
<tr>
<td></td>
<td>Quality</td>
<td>Quality in general</td>
</tr>
<tr>
<td><strong>External factors</strong></td>
<td>Market trend</td>
<td>Component evolution</td>
</tr>
<tr>
<td></td>
<td>Source code availability</td>
<td>Access and use of source code</td>
</tr>
<tr>
<td></td>
<td>Technical support</td>
<td>Source code documentation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Code customization support</td>
</tr>
<tr>
<td></td>
<td>License</td>
<td>Changes in requirements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>License fee</td>
</tr>
<tr>
<td></td>
<td></td>
<td>License obligations</td>
</tr>
<tr>
<td><strong>Software development activity factors</strong></td>
<td>Integration</td>
<td>Ease of integration</td>
</tr>
<tr>
<td></td>
<td>Requirements</td>
<td>Task complexity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Task uniqueness</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Requirement stability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Requirements negotiations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Requirements suitability</td>
</tr>
<tr>
<td></td>
<td>Maintenance</td>
<td>Ease of maintenance</td>
</tr>
</tbody>
</table>
Table 15: Project metrics factors

<table>
<thead>
<tr>
<th>Factors</th>
<th>OI-OC</th>
<th>OC-OS</th>
<th>OS</th>
<th>OI-OS</th>
<th>OI-OC-OS</th>
<th>C</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>-</td>
<td>[114]</td>
<td>-</td>
<td>-</td>
<td>[73]</td>
<td>2</td>
<td>2/2</td>
</tr>
<tr>
<td>Cost</td>
<td>[22, 109]</td>
<td>[51, 114]</td>
<td>[36, 165]</td>
<td>-</td>
<td>[73, 131]</td>
<td>8</td>
<td>6/8</td>
</tr>
<tr>
<td>Effort</td>
<td>-</td>
<td>[114]</td>
<td>-</td>
<td>-</td>
<td>[73]</td>
<td>2</td>
<td>2/2</td>
</tr>
<tr>
<td>Quality</td>
<td>-</td>
<td>[114, 125]</td>
<td>[165, 178]</td>
<td>[173]</td>
<td>[131]</td>
<td>6</td>
<td>3/6</td>
</tr>
</tbody>
</table>

OI-OC: In-house vs. COTS, OC-OS: COTS vs. OSS, OS: Only OSS, OI-OS: In-house vs. OSS, OI-OC-OS: In-house vs. COTS vs OSS.

C: no.of primary studies, E: no.of empirical studies

discussed for COTS, OSS and In-house decisions. Among the primary studies in Table 15, most of the primary studies are discussing “COTS vs. OSS” (6) and “In-house vs. COTS vs. OSS” (5) comparison category.

Cost is the most discussed factor and has the most number of empirical studying reporting it as factor. This indicates that cost is an important project metric factor that influences the decision. The description of how the primary studies are discussing the factors are as below (text segments - italics).

**Time:** The inclusion of COTS components causes delays as testing and integrating components (COTS) takes longer than in-house developed components [73]. However, when time to market is critical, COTS and OSS are preferred [114].

**Cost:** Cost of component is a motivating factor for adopting OSS as it is available for free [36, 114, 165]. However, it was not necessarily a disadvantage for COTS component users as some projects preferred COTS because it is paid, therefore, assume it to have better quality [114]. In spite of the differences in the price of a component, the total cost of ownership, which includes the cost to integrate, test and maintain the components can end up to be the same for OSS and COTS components [51]. When the cost of replacing components is considered, there are no budget related issues as there was no investment involved in buying the component [131]. However, replacing COTS components based on customers’ requirements was not cost effective [114]. Maintenance cost is used as a decision criterion in in-house vs. COTS decisions [73, 109]. The maintenance cost of externally acquired components like COTS and OSS can be really high in comparison to in-house components [22, 73].

**Effort:** Selection and integration effort: The reduced development effort is the common motivation for selecting either COTS or OSS [114]. However, estimating the selection and integration effort for COTS and OSS components has been identified as a challenge [114]. The component
selection is on a different decision level than the decision to select component origin (see Figure 6). However, the effort to select component is considered in the decision to choose component origin. The actual process of selecting the right components by considering all the risks is part of component selection decision level. Integrating external components requires considerable effort and if the component is small and requires more effort to integrate the component than to get it developed in-house, it is not the best option to acquire components externally [73].

**Quality:** OSS components might have better quality than COTS components however, it depends on various factors such as compensation, software demand, number of programmers, reward systems, programmer’s cost of effort, and coordination of programmers [125]. In addition, it highly depends on the field testing and post-delivery fault reporting [173]. Quality has been regarded as an motivation factor for adopting OSS [114, 165, 178]. The quality of OSS components has been compared with COTS and in-house components [125, 131]. In some cases the quality of OSS was better than COTS and in-house components, as a hastily developed in-house component might not match the quality of an OSS component which has been widely used by many OSS users [131].

Figure 11 provides a summary of project metric factors with respect to the quality of the primary studies. Though the results for time and ef-

<table>
<thead>
<tr>
<th>Project Metrics Factors</th>
<th>COTS</th>
<th>OSS</th>
<th>In-house</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>H.H</td>
<td>L.H</td>
<td>N.E</td>
</tr>
<tr>
<td>Time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time to test and integrate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time to market</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Cost</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cost of components</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Total cost of ownership</td>
<td>=</td>
<td>=</td>
<td>=</td>
</tr>
<tr>
<td>Cost of replacing components</td>
<td>=</td>
<td>=</td>
<td>+</td>
</tr>
<tr>
<td>Maintenance cost</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Effort</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Selection and integration effort</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Development effort</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Quality</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quality in general</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

Figure 11: Positive and negative influences of project metrics factors.
supporting these factors is from empirical studies (see Table 15). The results for cost and quality are provided by empirical studies (H.H and L.H) and non-empirical studies (N.E). The empirical and non-empirical studies have different contribution towards the cost factors. For example cost factors such as cost of buying component, cost of replacing components and maintenance cost for COTS come from empirical studies. Where as, the cost factors from non-empirical studies are discussing total cost of ownership. Therefore, providing more information related to cost factors. In addition, the evidence and non-evidence studies allow comparisons. For example, according to empirical studies COTS has negative impact due to the cost of replacing component and the results from non-empirical studies indicate that OSS has positive impact. This allows us to compare how COTS and OSS are affected by cost when the results from both empirical and non-empirical studies are considered. Therefore, the results from empirical and non-empirical studies are complementary and there are no conflicts in results. OSS has more number of positive results except for maintenance cost and selection and integration effort factors. Which indicates that, when time, cost, effort and quality are important then OSS can be a suitable option.

4.2.2 External factors

The external factors extracted from primary studies with respect to the comparison categories are presented in Table 16 along with the count (C) of the primary studies reporting the factor and the number of primary studies that are empirical (E). These factors depend on external provider (COTS vendor and OSS community) and market.

The external factors are discussed for COTS, OSS and In-house decisions. Among the primary studies in Table 16, most of the primary studies are discussing ‘COTS vs. OSS’ (17) and ‘OSS’ (5) comparison category. Source code availability and technical support are the most discussed factors in the primary studies. However less than half of these studies are empirical as shown in column E. The description of how the primary studies are discussing the factors are as below (text segments - *italics*).

**Market trend:** (Component evolution) Features that are novel today may become commodity tomorrow, and hence the market trend in terms of features affects which components that are suitable to, for example, develop in-house or use open source. Market trends are the needs of marketplace [22]. OSS development might evolve due to the market trends however, the evolution might also be due to political or social reasons within the OSS community [114] or OSS developer ideas.
Table 16: External factors

<table>
<thead>
<tr>
<th>Factors</th>
<th>OI-OC</th>
<th>OC-OS</th>
<th>OS</th>
<th>OI-OS</th>
<th>OI-OC-OS</th>
<th>C</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market trend</td>
<td>[22, 109]</td>
<td>[51, 79, 114]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>5</td>
<td>3/5</td>
</tr>
<tr>
<td>Technical support</td>
<td>[22]</td>
<td>[51, 77, 79, 112, 114, 125]</td>
<td>[152, 178]</td>
<td>-</td>
<td>[131]</td>
<td>10</td>
<td>4/10</td>
</tr>
<tr>
<td>License</td>
<td>[22, 109]</td>
<td>[51, 114]</td>
<td>[152, 165, 178]</td>
<td>-</td>
<td>[131]</td>
<td>8</td>
<td>4/8</td>
</tr>
</tbody>
</table>

OI-OC: In-house vs. COTS, OC-OS: COTS vs. OSS, OS: Only OSS, OI-OS: In-house vs. OSS, OI-OC-OS: In-house vs. COTS vs OSS.
C: no.of primary studies, E: no.of empirical studies

[79] which, might not be necessarily in the direction of market trend. COTS is preferred over in-house when following market trends and the need to include the newest technology is a priority [22, 109]. The market pressure can however negatively affect the development of COTS components [51]. The market pressure can force the COTS vendor to release the products before they are fully developed or tested. Hence, following market trend can have positive as well as negative effect on COTS adoption as shown in Figure 12.

**Source code availability:** Access to the source code has been stated as one of the advantages of using OSS components [114] and unavailability of the source code is stated as one of the biggest disadvantages of using COTS components [51]. The existing literature explores the usefulness of source code. Code visibility allows the integrators to make necessary changes in the component [79, 125]. However, a survey indicates that source code has been read to some extent but seldom changed [112]. Hence, if no such need arises to change the code, the availability of the source code does not seem important.

During the testing and maintenance phase, source code makes it easier to locate and fix errors as an integrator has access to the source code [73]. The integrators can learn and understand the OSS components and thereby save consulting costs [36]. Although source code is initially not available for COTS components, a survey [112] indicates that 33 percent of the COTS vendors opened their source code. The availability of the OSS source code can also be a disadvantage with respect to trust related
issues [77]. It becomes easier to correlate all the information and plan an attack. Hence, the availability of source code can make the system vulnerable to such attacks [77]. Reviewing the source code written by someone else can be a laborious task [152]. Hence, the availability of source code can be an advantage, however the vulnerability to attacks can make it a disadvantage as shown in Figure 12.

It can be difficult to understand the OSS source code documentation [79, 165] particularly, by inexperienced developers [178]. However, in one particular instance the documentation of the OSS code was better than the source code of a COTS component that was acquired at an additional cost [131]. The differences in results might be due to the dependencies on developer skills [178] and project profile [131]. The discussions on dependencies in continued in Section 5 and depicted in Figure 14.

**Technical support** Responsiveness of the vendor/community: Response rate of COTS vendor support has been reported to be good [114]. However, delayed response from COTS vendor are also reported [131]. Similarly the responsiveness of OSS community is reported to be good [77, 131]. However, not having enough support from OSS community is also reported [114]. Support availability: The COTS vendors might refuse to provide support, if the users do not have the latest version and if there are not enough users that are affected. The support provided by OSS community does not pose such limitations [51, 178]. OSS users widely practice customization of the code and COTS vendors who open their source code do not support such changes and proclaim that any modification will result in loss of support [125]. However, conflicting results indicating that OSS community might not extend their support if the code is customized is reported [114].

It is also possible to get the required changes from the COTS vendors [112]. However, smaller organizations adopting COTS and OSS components have little influence on the component evolution [79]. Developing and maintaining an effective relationship with vendors is a new activity, which is not needed in in-house development [22]. The results related to technical support are conflicting and it is uncertain how the COTS vendors or OSS communities react to changes in the source code. However, it highly depends on the profile of the vendor/community [125], size of the organization adopting COTS and/or OSS [79] and number of users [51, 178]. The discussions on dependencies in continued in Section 5 and depicted in Figure 14.

**License:** License includes the terms, conditions, and costs for a given product for use by an organization over a particular period of time
It also covers the vendor relationship including integration support [22]. Low **cost of licenses** has been a motivation for adopting OSS components [36, 114]. However, the **legal obligations** such as the general public license are considered too restrictive as, any modifications made to the source code should also be open which is referred to as copyleft [51, 165]. In addition, all applications linked to the code must also be made available [131, 165]. In order to avoid this and due to intellectual property (IP) concerns, the components with less restrictive licenses are preferred [36]. In addition, obtaining such a non-restrictive license has been regarded as a daunting process for a mission critical applications [131]. Hence, the license obligations must be thoroughly evaluated before adopting OSS components [109, 152].

Figure 12 provides a summary of external factors with respect to the quality assessment of the primary studies.

As Figure 12 is summarizing external factors, these factors do not have an impact on in-house development. Hence, in-house option is not mentioned in the table. The results are from empirical studies (H.H, L.H) as well as non-empirical studies (N.E). The results from empirical studies are contradicting results from non-empirical studies. The evidence from empirical study indicates that **COTS** is preferred when following market trend is important. However the results of non-empirical studies state that the market pressure can have negative impact as discussed earlier in this section. The conflicts in source code availability and technical support factors are due to the dependencies.
mentioned earlier, these dependencies are also discussed in Section 5 (see Figure 14). From the Figure 12, we can see that overall OSS is preferred when external factors are considered.

4.2.3 Software development process factors

The software development process factors extracted from primary studies with respect to the comparison categories are presented in Table 17 along with the count (C) of the primary studies reporting the factor and the number of primary studies that are empirical (E).

<table>
<thead>
<tr>
<th>Factors</th>
<th>OI-OC</th>
<th>OC-OS</th>
<th>OS</th>
<th>OI-OS</th>
<th>OI-OC-OS</th>
<th>C</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integration</td>
<td>-</td>
<td>[51, 79, 114]</td>
<td>-</td>
<td>-</td>
<td>[73]</td>
<td>4</td>
<td>2/4</td>
</tr>
<tr>
<td>Requirements</td>
<td>[22, 109]</td>
<td>[114]</td>
<td>[100, 178]</td>
<td>-</td>
<td>[73, 131]</td>
<td>7</td>
<td>4/7</td>
</tr>
<tr>
<td>Maintenance</td>
<td>[22]</td>
<td>[51, 79, 114]</td>
<td>-</td>
<td>-</td>
<td>[73]</td>
<td>5</td>
<td>3/5</td>
</tr>
</tbody>
</table>

OI-OC: In-house vs. COTS, OC-OS: COTS vs. OSS, OS: Only OSS, OI-OS: In-house vs. OSS, OI-OC-OS: In-house vs. COTS vs OSS.
C: no.of primary studies, E: no.of empirical studies

The software development process factors are discussed for COTS, OSS and In-house decisions. Among the primary studies in Table 15, most of the primary studies are discussing ‘COTS vs. OSS’ (6) and “In-house vs. COTS vs. OSS” (5) comparison category. Requirements factor is the most discussed factor and has the most number of empirical studying reporting it as factor. This indicates that the decision to choose the component should be taken as early as the requirements phase. The description of how the primary studies are discussing the factors are as below (text segments - italics).

Integration: Ease of integration: OSS integration and COTS integration is similar if the source code is not exploited [51]. It is difficult to identify if the defects are inside or outside the COTS or OSS components, even if the source code is available for OSS most users usually do not look at the code [114]. However, without the source code it is difficult to understand the working of the component (COTS) in particular when there are issues such as environmental differences, deviations from supported protocols, etc. Consequently making it difficult to integrate COTS components. [79]. The effort required to integrate COTS components is considered in the in-house vs. COTS decisions [73].

Requirements: Task complexity and uniqueness: The decision to choose between in-house, COTS and OSS is mostly taken in the requirements
The developers prefer to use pre-built components such as OSS libraries when the task complexity is high to improve productivity [178]. However when unique development efforts (developing innovative functionality) are required to fulfil a requirement, OSS is unlikely to be used [178]. In addition, when there is requirement uncertainty, the tendency to adopt existing OSS components is high [100].

In-house development starts with a specific set of system requirements and builds a system that meets those requirements. However, the COTS based system development starts with a generic set of requirements and then the market place is explored to see how closely they match the needs of the system [22]. Whether or not it applies to all COTS based systems is unknown.

The requirements should be carefully analyzed before the decision to acquire external components such as COTS is made [73]. The requirement negotiations and evaluations of external components are too long. During the negotiations and evaluations the technical requirements might change and the component might no longer be needed [73]. The available OSS or COTS component might not perfectly match the requirements and therefore, might not be suitable to use. The customers might agree to the requirements changes that are necessary to utilize OSS component [131]. However, not adapting sufficiently to the customers’ requirements and not having the possibility to negotiate customer requirements has been perceived as a common risk for COTS and OSS components [114]. This could be the reason why OSS or COTS components may not be preferred when the requirements require unique development effort and are not very flexible to changes.

**Maintenance:** One of the challenges in COTS deployment and maintenance is to strike a balance between system stability and being up to date with the market place [22, 114]. The COTS vendor usually decides the rate at which the component changes and the organization that acquires the COTS component should decide if they want to upgrade or lose the support for the component they are currently using [79]. Maintenance policies can force the component to be upgraded to a newer version so that the technical support provided from the vendors can be retained. Therefore, maintenance policies are used as an evaluation criterion in the in-house vs. COTS decisions [73]. The OSS users are not obliged to change whenever there is any upgrade [51].

Figure 13 provides a summary of software development process factors with respect to the quality of the primary studies. The results are
supported by empirical (H.H, L.H) and non-empirical (N.E) studies. The results from empirical studies are supporting non-empirical studies and there are no conflicts in results. OSS and in-house have more number of positive results compared to COTS. Which indicates that, when integration and maintenance are important then either OSS or in-house can be a suitable option.

4.3 Solutions (RQ3)

The primary studies that discuss solutions to choose a component origin (RQ3) for the software components are listed in Table 18. The “no.of studies” column indicates the number of studies specifically answering RQ3 out of the total number of studies found in this review study.

4.3.1 O1: In-house vs. COTS

For the “O1: In-house vs. COTS” decisions six studies provided solutions. The solutions [39, 41, 88, 90, 143, 162] define optimization models.

Papers [39, 41] help to select the component that should be developed in-house and the components that should be bought. An optimization model is proposed which minimizes cost under reliability and delivery time constraints by selecting the right origin (in-house or COTS) of components. It also considers the amount of testing required to develop the component in-house into the decision-making. The selection of component origin is done after the software design is available. For
Table 18: Primary studies proposing solutions for choosing a component origin (RQ3)

<table>
<thead>
<tr>
<th>Component origin</th>
<th>Count</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>OI-OC</td>
<td>6/8</td>
<td>[39, 41, 88, 90, 143, 162]</td>
</tr>
<tr>
<td>OC-OS</td>
<td>0/6</td>
<td>-</td>
</tr>
<tr>
<td>OS</td>
<td>0/5</td>
<td>-</td>
</tr>
<tr>
<td>OI-OS</td>
<td>0/1</td>
<td>-</td>
</tr>
<tr>
<td>OI-OO</td>
<td>2/2</td>
<td>[101, 103]</td>
</tr>
<tr>
<td>OI-OC-OS</td>
<td>0/2</td>
<td>-</td>
</tr>
</tbody>
</table>

OI-OC: In-house vs. COTS, OC-OS: COTS vs. OSS, OS: Only OSS, OI-OS: In-house vs. OSS, OI-OO: In-house vs. outsource, OI-OC-OS: In-house vs. COTS vs OSS.

COTS components, the time is measured in terms of the time taken by the vendor to provide the component and the time taken to adapt the component. For the in-house component, the time taken to develop the component and to achieve the desired level of reliability is measured. Cost is measured in terms of the cost required to buy and adapt the COTS components, similarly the cost for developing in-house components is also estimated.

The paper [143] is an extension of the solution proposed in [39], suggesting an optimization model by taking into account the reliability, delivery time and cost. The addition to the model is the ability to take the decision as soon as the requirements are available.

A multi-objective approach for an optimal in-house vs. COTS decision for a fault tolerant system was formulated by [88]. The two objectives in decision-making are maximizing system reliability and minimizing the system cost under the constraint of delivery time. In addition, it also considers compatibility between available alternative components.

The solution in the paper [162] propose an architecture optimization approach based on a meta-heuristic swarm intelligence algorithm to decide the origin (In-house or COTS) of the components. The algorithm considers delivery time, cost and reliability constraints. It also considers the number of test cases for in-house components.

The paper [90] presents a solution to maximize the intra-module coupling density under a number of constraints such as delivery time, cost, reliability, compatibility, cohesion, coupling, requirements fit and test success.
4.3.2 In-house vs. Outsourcing

The solution in paper [101] proposes a methodology and tool support to decide the components that need to be outsourced and which ones should be developed internally. The tool classifies the projects’ software components by means of a graph based model of the components’ requirements and their corresponding clustering with respect to requirements dependencies and priorities.

The paper [103] proposes a solution that determines the outsourcing potential (if it is high the component should be outsourced), input is knowledge specificity (based on business, functional and technical), and interdependencies (priority, between software components and communication intensity among developers). A decision is made with the help of decision tables.

5 Discussion

From the high-level themes discussed in Sections 4.2.1, 4.2.2 and 4.2.3, we can see that OSS has more positive influence on the decision when project metrics and external factors are considered. In-house and OSS have more positive influence when software development process factors are considered. However, when the individual factors/themes are considered each component origin has some advantage over some other component origins. The results of this study indicate that some trade-offs and dependencies need to be considered which is discussed in this section along with the advantage and disadvantages of the component origins.

The advantages are discussed in Table 19. The advantages of outsourcing are not discussed as none of the primary studies have mentioned them. The information presented in Table 19 can be used as an initial set of factors to be considered by practitioners in the decision-making.

In addition, there are some trade-offs that need to be considered along with the advantages. Figures 14, 15 and 16 illustrate the trade-offs between the different factors. The trade-offs that should be considered along with advantages of COTS over OSS are depicted in Figure 14 and discussed in Section 5.1. The trade-offs that should be considered along with advantages of OSS over COTS are depicted in Figure 15 and discussed in Section 5.2. Whereas, Figure 16 discusses trade-offs that are related to the advantages of in-house development over COTS and OSS, which is discussed in Section 5.3. Section 5.4 discusses the advantages of COTS and OSS over in-house development, no trade-
Table 19: Advantages of component origins

<table>
<thead>
<tr>
<th>Component origin</th>
<th>Advantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>COTS over OSS</td>
<td>Market Trend [114]</td>
</tr>
<tr>
<td>OSS over COTS</td>
<td>Low cost of replacing component [114, 131], Source code available [36, 73, 79, 114, 125], Low licensing cost [36, 114], Ease of debugging integration defects [79], Ease of maintaining system stability, Better quality [114, 125, 165, 178]</td>
</tr>
<tr>
<td>In-house over COTS &amp; OSS</td>
<td>Ability to add unique functionality [178], Reduced maintenance cost [22, 73], Easy integration [79, 114], Reduced testing time [73]</td>
</tr>
<tr>
<td>COTS &amp; OSS over In-house</td>
<td>Reduced time to market [114], Reduced development effort [114], Ability to add complex functionality [178]</td>
</tr>
</tbody>
</table>

offs were mentioned for the advantages discussed in Section 5.4 in the primary studies. The trade-offs and dependencies depicted in Figures 14, 15 and 16 are aggregated from different primary studies. The “↔” arrow represents the trade-off between different factors and “→” arrow represents the dependencies on the context that affects the factors. Depending on the context, the factors have positive or negative impact.

5.1 COTS over OSS

If following the market trend is the system requirement, practitioners might choose COTS components as COTS components have the advantage of following the market trend. One should also consider the trade-off between the following factors: market trend, maintenance and technical support as shown in Figure 14. Keeping up with the market trend

![Figure 14: Trade-offs for COTS and OSS factors](image-url)
can mean that the component must be frequently upgraded [22, 51, 114]. Frequent upgrades might hinder the system stability. In addition, if the component is not updated to the latest version, there is a risk of losing technical support offered by the COTS vendor [79]. Hence, the trade-off between market trend, maintainability and technical support should be considered. Technical support offered depends on the license agreements and context information such as size of the organization (large-scale organization have better chances with license negotiations) and profile of the vendor or community (depends on how flexible the vendor/community providing the component).

5.2 **OSS over COTS**

OSS is preferred over COTS when there is a possibility for lot of requirement changes, as there is no impact on the budget [114], [131]. If OSS components are chosen because the source code is available, the trade-off between source code availability, technical support and license must be considered as shown in Figure 15.

![Figure 15: Trade-offs and dependencies for advantages of OSS over COTS](image)

The source code can help to identify defects during integration however, it depends on the developers’ skill or experience in finding the defects in source code [178]. In high-risk projects, the source code might need to be reviewed [131]. However, in low-risk projects, such reviews might not be necessary. Hence, if the project has a high-risk profile and the developers or integrators are skilled and experienced, the availability of source code is beneficial. On the other hand, if the project does not have a high-risk profile and the developers or integrators do not
have the required skill or experience, the availability of source code is not of any benefit to the practitioners.

Another benefit of source code being available is the possibility to change the code. However, the trade-off between technical support and license needs to be considered. Some OSS communities might refuse to support if the component is customized, although it highly depends on the profile of the vendor of community [125], size of the organization using the component [79] and number of users using the component [51], [178]. In addition, OSS communities might impose a restrictive license, any changed code must be given back to the OSS community [114].

5.3 In-house over OSS and COTS

One of the advantages of COTS and OSS is reduced development effort [114]. However, integrating COTS and OSS is time consuming [114]. Hence, the trade-off between development effort and integration effort must be considered as shown in Figure 16.

![Figure 16: Trade-offs and dependencies for in-house, COTS and OSS factors](image)

If integrating COTS or OSS components might take longer than developing components, then they should be developed in-house rather than acquired externally. The development time might vary based on the size of the component. In addition, some studies [73] and [22] report that maintenance cost of in-house developed components is lower than COTS and OSS components.

5.4 COTS and OSS over in-house

The pressure to develop faster due to market competitiveness is identified as a challenge in in-house development [10]. COTS and OSS are known to be viable options when time to market is a criterion [114]. The constant requirements changing resulted in a lot of wasted development effort [10], which can be avoided when external components are used especially, OSS as there are no additional costs involved in buying the component [114]. Challenges related to understanding complex requirements were identified [10]. The developers prefer to use
pre-built components such as OSS libraries when the task complexity is high to improve productivity [178].

5.5 Research gaps

The purpose of this systematic review is also to identify research gaps and directions for future studies. Overall, only a few studies have been found in the area researched. Thus, very specific research gaps could be identified looking at the existing literature.

As shown in Table 18 the majority of solutions has been proposed for “In-house vs. COTS”. The solutions that have been proposed were optimization models using different factors such as time, cost and quality (reliability in specific). Other component origins have no solutions, or very few (at most two studies). Given that solutions are often dependent on the context in which they are applied, more evaluations of them are needed, and a wider array of new solutions may be needed to address the complex decision problem of choosing a component origin.

The factors that influence the decision to choose a component origin mentioned in Sections 4.2.1, 4.2.2 and 4.2.3 and the factors that are used in solutions proposals (Section 4.3) are mapped in Table 20.

As seen in Table 20, the solutions only consider project metric factors (cost, time and quality) excluding effort. The external factors and software development activity factors are not considered by any of the solutions. In addition, the cost considered in solutions only considers the cost of buying and adapting COTS components, however in the primary studies that discuss the influencing cost factors have considered cost of buying [36, 114, 165], replacing [165] and maintaining component [22, 73, 109]. Also even though time to obtain and adapt the component is considered, time to market is not considered by any of the solutions. Reliability is considered in almost all proposed solutions however, it did not emerge as a most important factor in the studies that compared the different component origin. Unlike the solutions, non-technical aspects such as vendors’ profile are used to evaluate quality [165]. The integration effort has not been considered by any of the solutions, even though it is mentioned as an influencing factor in the in-house vs. COTS and OSS decisions [51, 73, 79, 114]. In addition, an important trade-off between integration effort and development effort has been identified as shown in Figure 16. One possible reason for not including integration effort may be because it is difficult to estimate the effort required to integrate external components such as COTS and OSS.
Table 20: Mapping of factors influencing the decision and factors considered in the solutions

<table>
<thead>
<tr>
<th>Factors influencing the decision to choose a component origin</th>
<th>Factors considered in solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project metric factors</strong></td>
<td></td>
</tr>
<tr>
<td>Time to market and time to test and integrate</td>
<td>COTS - Time taken by the vendor to provide the component and adaptation time.</td>
</tr>
<tr>
<td></td>
<td>In-house - Time taken to develop the component.</td>
</tr>
<tr>
<td>Cost of buying, replacing and maintaining component</td>
<td>Cost of buying and adapting COTS components</td>
</tr>
<tr>
<td>Selection, development and integration effort</td>
<td>Not considered</td>
</tr>
<tr>
<td>Quality</td>
<td>Reliability</td>
</tr>
<tr>
<td><strong>External factors</strong></td>
<td></td>
</tr>
<tr>
<td>Market trend</td>
<td>Not considered</td>
</tr>
<tr>
<td>Source code availability</td>
<td>Not considered</td>
</tr>
<tr>
<td>Technical support</td>
<td>Not considered</td>
</tr>
<tr>
<td>License</td>
<td>Not considered</td>
</tr>
<tr>
<td><strong>Software development activity factors</strong></td>
<td></td>
</tr>
<tr>
<td>Integration</td>
<td>Not considered</td>
</tr>
<tr>
<td>Requirements</td>
<td>Not considered</td>
</tr>
<tr>
<td>Maintenance</td>
<td>Not considered</td>
</tr>
</tbody>
</table>

Among the solutions provided for in-house vs. COTS and in-house vs. outsourcing, in-house is the common component origin however, the solutions provided do not have any common factors that are considered in the solutions.

Considering the above remarks, it can be said that the proposed solutions have not necessarily considered the most important factors that are needed in the decision to choose a component origin.

Few studies (maximum 6, see Table 13) per combination of component origins were identified. Thus, it is hard to judge whether the component origins discussed are the most relevant ones for the decision to be made. One study [152] highlighted, according to what the authors believed, the most important factors that need to be considered while selecting OSS components. The relative importance of factors for the different combinations of component origins is unknown and hence based on the existing studies it is impossible to suggest the ones that are most important. Also, the order in which the factors should be considered...
is not known. Though, the findings can serve as an initial inventory of factors to be considered. If a factor is mentioned for one of the component origins, but not for another, also does not necessarily mean that it is irrelevant for that component origin.

Conflicting statements with regard to quality and technical support factors were found, but not for all other combinations. This may be due to the low number of studies and indicates that to make a reliable statement with regard to a positive or negative effect in relation to different factors, it is not possible to provide reliable guidance to practitioners in their decision-making. In particular, to have good decision support it is important to understand the strengths and weaknesses of the alternatives with regard to the factors and outcomes empirically. With this understanding, in combination with knowledge about the importance of the factors, more informed decisions can be made. With the current solutions, no data is available to do so.

6 Conclusion

A systematic review of the factors affecting the decision to choose a component origin has been conducted. A component origin is a source to get the components from (here in-house development, open-source, COTS and outsourcing). Three research questions were asked:

RQ1: What are the research types, methods and quality of the contributions? All the research types classified by Wieringa [186] were identified. Hence, the primary studies consisted of a good mix of empirical and non-empirical results. This distribution of the research types allowed to identify conflicting results based on context dependencies. The most common research type is evaluation research indicating that the research is empirical and rooted in industrial practice, which is a positive result. The most common research method is survey indicating that the results are not specific to a particular context. However the context is not described sufficiently in the primary studies. The results of quality assessment indicate the lack of rigor in the primary studies in terms of reporting the context, validity threats and research design.

RQ2: What are the different factors that influence the decision to choose among different component origins? In total eleven factors grouped into three high-level themes that influence the decision to choose a component origin were identified. The project metrics factors are: time, cost, effort and quality. External factors such as market trend, source code availability, technical support and license were identified. In addition, the software development activity factors such as integration, require-
ments, and maintenance activities that have an impact on the decision were identified. The results from empirical are supporting and complementing results from non-empirical studies. However in some cases, conflicts are identified (see for example, Figure 12). The conflicts are due to the context dependencies.

The project metrics and external factors positively affect the adoption of OSS components. However, the trade-off between development effort and integration effort needs to be considered when choosing between in-house and OSS components (see Figure 16). Whereas, when external factors are considered, the context dependencies on the source code availability and technical support factors need to be considered (see Figure 14). The software development activity factors positively affect the decision to adopt OSS components and in-house. Overall COTS has a negative impact on adoption however, it is preferred over OSS and in-house when following the market trend is important.

**RQ3: What solutions have been proposed to choose the component origin?**

The solutions were mainly focused on choosing between in-house and COTS and were mainly based on optimization models.

The overall implication is that there are too few studies, and hence too few contexts investigated to provide recommendations for practitioners on how to make decisions between the component origins. That being said, the initial set of studies identified may serve as a basis for future studies.

Three decision levels are shown in Figure 6. The factors influencing provider selection are different from component origin selection [93]. The factors influencing component selection are similar to the factors influencing component origin selection [128]. For example, the component integration effort is considered in component selection. However, the comparison with development effort is not considered. The decision to select the component origin must be taken first before selecting the component. While there are secondary studies on provider selection [93] and component selection [128] level, this systematic literature review is the first secondary study on selecting component origin. The research field of choosing between component origins is in great need of future studies to provide comprehensive approaches to support decision-making. This is true in terms of the number of solutions proposed, and the extent to which empirical evaluations have been conducted.

As a consequence, future work has to focus on (a) determining the order of importance and magnitude of the factors; (b) providing empirical evidence on comparisons of component origins with regard to
the factors; and (c) proposal of novel solutions taking (a) and (b) into consideration.

Acknowledgments

The work is partially supported by a research grant for the ORION project (reference number 20140218) from The Knowledge Foundation in Sweden.
EXPERIENCES FROM USING SNOWBALLING AND DATABASE SEARCHES IN SYSTEMATIC LITERATURE STUDIES

Abstract

**Background:** Systematic literature studies are commonly used in software engineering. There are two main ways of conducting the searches for these type of studies; they are snowballing (SB) and database (DB) searches. In snowballing, the reference list (backward snowballing - BSB) and citations (forward snowballing - FSB) of relevant papers are reviewed to identify new papers whereas in a database search, different databases are searched using predefined search strings to identify new papers.

**Objective:** Snowballing has not been in use as extensively as database search. Hence it is important to evaluate its efficiency and reliability when being used as a search strategy in literature studies. Moreover, it is important to compare it to database searches.

**Method:** In this chapter, we applied snowballing in a literature study, and reflected on the outcome. We also compared database search with backward and forward snowballing. Database search and snowballing were conducted independently by different researchers. The searches of our literature study were compared with respect to the efficiency and reliability of the findings.

**Results:** Out of the total number of papers found, snowballing identified 83% of the papers in comparison to 46% of the papers for the database search. Snowballing failed to identify a few relevant papers, which potentially could have been addressed by identifying a more comprehensive start set.

**Conclusion:** The efficiency of snowballing is comparable to database search. It can potentially be more reliable than a database search however, the reliability is highly dependent on the creation of a suitable start set.

**Keywords**

Snowballing (SB), backward snowballing (BSB), forward snowballing (FSB), database (DB) search, efficiency, reliability.
Evidence-based software engineering (EBSE) was introduced in 2004 by Kitchenham et al. [99]. The objective of EBSE is to synthesize the evidence from multiple primary studies. A means to synthesize the evidence are systematic literature reviews. Guidelines have been proposed for conducting literature studies. Kitchenham and Charters [98] recommended a search strategy using well-defined search strings in databases (DBs) and also the review of reference lists from relevant primary studies and review articles. In the guidelines [98] it is recommended to use BSB, i.e. the use of the reference lists of studies to identify additional relevant papers, as a complement to DB searches. Webster and Watson [184] have recommended to use snowballing (SB) as the main method for conducting the search in literature studies.

As different search strategies exist, studies [86], [190] have been conducted to evaluate if the search strategy impacts the actual outcomes of the systematic literature studies. Jalali [86] compared DB search and BSB and found that the results are not dependent on the search strategy. More recently, Wohlin [190] has proposed guidelines for conducting SB in systematic literature reviews and performed a replication study using those guidelines. He concludes that using SB as a search strategy might be a good alternative to DB searches. However there are few studies comparing the search strategies and more studies are needed to investigate under which circumstances SB is preferred over DB search.

This study complements the studies [86] and [190] by evaluating the efficiency and reliability of SB and DB search. For the research presented in this chapter a systematic mapping study using different search strategies was performed by different authors independently during the same time period. Snowballing (FSB and BSB) and DB search were the two different search strategies used in the mapping study. The same research questions and inclusion/exclusion criteria is used in both search strategies. The following research questions are answered in this evaluation study:

1. **RQ1 How are the relevant papers identified through the evolution of the SB process?**
   The main contribution of this research question is to recognize the identification pattern of the accepted papers.

2. **RQ2 How efficient is the SB?**
   Efficiency is the noise vs. relevance ratio. In order to answer this research question, the efficiency of SB is evaluated and also compared with DB.
• RQ3 How reliable is SB in capturing all relevant papers?

In this context reliability is regarded as the ability to identify all relevant papers. We contribute to this research question by evaluating the common and unique results identified by SB and DB search.

The remainder of the chapter is outlined as follows. Related work is presented in Section 2. Section 3 describes the research method. The result of this study are presented in Section 4 and discussed in Section 5. Section 6 concludes the chapter.

2 RELATED WORK

Only a few studies have focused on the reliability and efficiency of systematic literature studies. MacDonell et al. [118] evaluated the reliability of systematic reviews. They compared the results of two systematic reviews, which had common research questions performed by two independent groups of researchers. The conclusions of this study was that the reviews were robust to difference in the review process and people involved.

An audit was conducted to find how the primary sources were originally identified by Greenhalgh and Peacock [67]. The audit result was that only 30% of the studies were found by the DB search, whereas 51% of papers were found through SB. The conclusion of the audit was that a systematic review of complex evidence cannot completely rely solely on DB searches (predefined, protocol driven), instead browsing library shelves, asking colleagues, pursuing interesting references (SB) may have better efficiency.

Skoglund and Runeson [159] evaluated the reference based search with the objective to reduce the large number of papers to be reviewed. The results for the technically focused reviews were satisfactory with only a few missing relevant papers. However, for reviews where the search area is wide or included general terms, the results were not satisfactory, i.e. large number of papers were missing. Hence they conclude that the performance in terms of precision is context dependent.

In another study conducted by Jalali and Wohlin [86], they investigated the reliability by comparing the results of two systematic reviews on the same topic. Both reviews were conducted by the same authors, however, these reviews were not conducted in the same time period. The first study was conducted using DB search, and the second study was conducted after a couple of months using BSB as a search strategy. In Jalali and Wohlin’s study they found that the actual results are not
highly dependent on the search strategy. Another conclusion was that the efficiency of SB might be higher when the keywords for searching include general terms.

More recently, in 2014, Wohlin conducted a replication study [190] to compare SB results including BSB and FSB and DB search. The outcome of the replication study using SB was similar to the original study conducted using DB search, also SB is considered to be a good alternative to DB search.

The reliability of mapping studies was discussed in a study [191], it includes potential areas of improvement to increase the reliability. One of the improvement suggested is to use SB as search strategy as SB based on researchers expertise and knowledge in an area is more efficient than finding optimal search strings. However the study indicates the need of more studies comparing the search strategies to understand which search strategy is better in which circumstances. Hence this study is conducted as a complement to studies [86] and [190]. In particular, More studies are needed to be able to determine when SB is better than DB search and in what circumstances.

3 RESEARCH METHOD

The main objective of this study was to reflect on the lessons learned from using SB as a search strategy in literature studies. Snowballing was used as a main method to find the existing literature related to the strategic decision-making process to select among different development options. The four main development options considered in the study were: in-house, outsource, COTS and open source. Hence the inclusion criteria for the study was to include papers if one of the following criteria was fulfilled: [1] Papers discussing decision-making process for selecting an option. [2] Papers comparing two or more of the options. [3] Papers proposing solutions that support the decision-making process in the selection of an option in relation to other options. Papers were excluded if they were not related to component-based software or if papers discussed architectural and development aspects of components based system. Papers discussing adoption of software packages, IT services or operating system were excluded. Grey literature and non-English papers were also excluded from the study.

A DB search was also performed as a complementary step to validate the SB results. The two searches, i.e DB and SB were conducted by independent authors in the same time period. Deepika Badampudi and Claes Wohlin conducted the SB, while the DB search was conducted
by Kai Petersen. The same inclusion and exclusion criteria were used by the authors to avoid the threats related to the judgment of inclusion and exclusion criteria, and hence making the results of the two searchers comparable. The treats related to the reliability of DB results as single researcher conducted it is under control as the researcher has vast experience in conducting DB search. In this study we focus on lessons learned from SB and compare the SB and DB search results. The details of the searches are described in the following sections.

3.1 Details of snowballing

The SB search was conducted by first creating a start set, and thereafter conducting BSB and FSB of the start set in an iterative fashion.

**Creation of the start set:** The search strings to create the start set are shown in Table 21. They were applied on Google scholar. The keyword “Software” was added to three of the search strings as most of the papers retrieved were not related to software engineering.

Google scholar is not restricted to specific publishers, and hence it was selected as the index DB to create the start set. From the results returned by the Google scholar search, the first 10 results of each of the nine search strings (see Table 21), were reviewed using the inclusion and exclusion criteria. That is, a total of 90 search results were reviewed. The papers that fit the inclusion criteria were added to the start set in two phases.

<table>
<thead>
<tr>
<th>Table 21: Nine search strings used by SB</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-house vs. outsourcing</td>
</tr>
<tr>
<td>In-house vs. COTS</td>
</tr>
<tr>
<td>In-house vs. OSS</td>
</tr>
<tr>
<td>COTS vs. OSS</td>
</tr>
<tr>
<td>COTS vs. outsourcing</td>
</tr>
<tr>
<td>Outsourcing vs. OSS</td>
</tr>
<tr>
<td><strong>Additional search strings</strong></td>
</tr>
<tr>
<td>In-house vs. outsourcing and software</td>
</tr>
<tr>
<td>Outsource vs. OSS and software</td>
</tr>
<tr>
<td>COTS vs. Outsourcing and software</td>
</tr>
</tbody>
</table>
In *Phase 1*, the papers were tentatively included based on title, abstract and introduction. In some cases more sections were reviewed. However an extensive full text review was not conducted in this Phase.

In *Phase 2* the inclusion or exclusion was done based on full text reading. The reviewing in Phase 1 and 2 was done by first two authors independently. At the end of each phase a review meeting was held to analyze the review process. The decision rules (cf. [137] for an overview) shown in Table 22 were applied for the final inclusion or exclusion.

Five papers were selected that were used as input for the SB activity (including BSB and FSB).

**Snowballing activities:** BSB and FSB were carried out in iterations. The reference list of the papers in the start set were reviewed during BSB and citations were reviewed during FSB. The citations were retrieved from Google scholar. The papers that were included in the iterations were added to the start set and SB of the newly added papers was done in the next iterations. This process was followed until no new papers were found. When all the papers were added to the start set it was considered as the final set, which included all primary studies. The same inclusion and exclusion criteria were used and the decision to include or exclude was made in two phases (see above) as in the start set. However the review process for Phase 1 was different from the start set, in BSB the following was reviewed in Phase 1:

1. Title of the referenced paper.
2. The point of reference (reference context) of the paper.
3. Abstract of the referenced paper.

Whereas in FSB the following order is followed:

1. Title of the paper citing.
2. Abstract of the paper citing.
3. The point of reference (reference context) to the paper being cited.

The reference context refers to the text surrounding the reference citation within the primary study. This was done as the text surrounding the reference citation allows to understand the context of the citation, for example the reason for citing the paper. Note that steps 2. and 3. was reversed in FSB as it was easier to read the title and abstract first when looking at the papers citing a relevant primary study.
Table 22: Inclusion and exclusion decision rules

<table>
<thead>
<tr>
<th>Case</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Both authors accept a paper</td>
<td>Include the paper for next step</td>
</tr>
<tr>
<td>Both authors reject a paper</td>
<td>Exclude the paper</td>
</tr>
<tr>
<td>Either one of the authors accepts a paper</td>
<td>Include the paper for next step</td>
</tr>
</tbody>
</table>

3.2 Details of database search

The DB search was performed to increase the confidence in the selection of papers. The search terms were divided into population, interventions, comparison, and outcome (PICO). The search strings as shown in Table 23 were applied in Scopus and Inspec/Compendex. The papers were either included or excluded based on the same inclusion and exclusion criteria used for SB. Thereafter, the first author reviewed the included papers to check their relevance. In cases of disagreements the authors discussed the papers until they were resolved.

Table 23: Search strings used by database search

<table>
<thead>
<tr>
<th>Database search</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search 1: (X OR Y) AND (I1 OR I2 OR I3) AND E</td>
</tr>
<tr>
<td>Search 2: (X OR Y) AND (I1 OR I2 OR I3) AND F</td>
</tr>
<tr>
<td>Search 3: (X OR Y) AND (I1 OR I2 OR I3) AND G</td>
</tr>
<tr>
<td>E: A AND (B OR C OR D)</td>
</tr>
<tr>
<td>F: B AND (C OR D)</td>
</tr>
<tr>
<td>G: C AND D</td>
</tr>
<tr>
<td>A: (COTS OR “components off the shelf” OR “component off the shelf”)</td>
</tr>
<tr>
<td>B: (In-house OR inhouse)</td>
</tr>
<tr>
<td>C: (open-source OR “open source” OR OSS)</td>
</tr>
<tr>
<td>D: (outsource OR out-source OR outsourcing OR “third-party”)</td>
</tr>
<tr>
<td>“software (X)” and “component (Y)”</td>
</tr>
<tr>
<td>“trade-off (I1)”, “decision (I2)”, and “selection (I3)”</td>
</tr>
</tbody>
</table>
3.3 Research questions

In order to reflect on SB and compare it with DB search, the following research questions are formulated -

• **RQ1 How were the relevant papers identified through the evolution of SB process?**
  As stated earlier, in SB the first step is to create the start set and keep adding papers through BSB and FSB. In this research question we analyze the pattern in which the papers were found. With additional studies reporting patterns, if we recognize repeating patterns we can improve the guidelines for the SB procedure. For example, this can lead to more concrete guidelines on how to construct the start set, and which papers to apply SB on. Overall, with a higher number of data points, in the future we could propose steps based on reliable evidence that lead to a more effective and efficient SB procedure.

• **RQ2 How efficient is SB?**
  Efficiency is the number of papers included in relation to the total number of papers reviewed. One of the risks in mapping studies is to review a large number of non-relevant papers, i.e noise [190]. Hence to answer RQ2 the noise vs. relevance ratio is analyzed. As SB was carried out in iterations, the efficiency of all the iterations as well as BSB and FSB was analyzed independently in this research question. The efficiency was also compared to the efficiency of the DB search. Besides the title and abstract, the reference context was also used as a decision base in SB. Reviewing the reference context is an additional step which is not considered in the DB search. Hence, we evaluated the usefulness of the reviewing reference context in decisions to include or exclude papers. In this research question we also considered the total number of decisions made based on the reference context of the primary studies and how many of these decisions changed when the full text of the paper was reviewed. Hence the following sub-questions were answered in this research question
    - RQ2.1 What was the efficiency of start set, iterations, BSB and FSB?
    - RQ2.2 How useful was the reference context as a decision base?
    - RQ2.3 How efficient was SB in comparison to DB search?
• **RQ3 How reliable was SB in capturing all relevant papers?**

In this context the reliability of SB is the ability to capture all relevant papers. A complimentary DB search was conducted to evaluate and compare the reliability of SB. If the papers found in DB search are mostly the same as papers found in SB, with only a few unique papers found in DB search, then we can claim that in this case the SB process was reliable. The DB search used additional keywords related to the dimension (selection, trade off and decision). Hence, we evaluated if the SB process is as reliable as the DB search in identifying papers addressing different dimensions, and also development options. Hence the research question was divided into the following sub questions:

- **RQ3.1** What were the common and unique papers identified in SB and DB searches?
- **RQ3.2** What were the common and unique development options identified in SB and DB searches?
- **RQ3.3** What were the common and unique dimensions identified in SB and DB searches?
- **RQ3.4** To what extent did we have the same conclusion using two different searches? Note that the same question has been asked in [86].

### 4 Results

#### 4.1 Evolution of the SB process (RQ1)

The first step in SB was to identify a start set from which the SB can start. The goal was to find papers comparing two or more development options. Based on the four options considered in the study we have six possible pairs to compare the development options. Each comparison pair is considered as a category in the start set. Hence we have six comparison categories as shown in Figure 17.

Thus, one search strings was formulated for each combination of development options as shown in Table 21.

The first ten results from Google Scholar for each search were reviewed. The evolution of identified papers is depicted in Figure 17.

**Start set:** In total, five papers were included into the start set. Three papers from the COTS vs. OSS search string, one from In-house vs. COTS and one from In-house vs. OSS search string. However, the paper found in the In-house vs. OSS search string was also comparing COTS hence,
the paper was added to a new category (In-house vs. COTS vs. OSS) as shown in Figure 17. No papers were found for the In-house vs. Outsource, COTS vs. Outsource and Outsource vs. OSS comparisons. No papers were identified for any combination with outsource as one of the development option.

We found a high number of papers that were not software related for the outsource combination search strings hence, we included three additional search strings by adding software as a keyword. The inclusion of software in the search strings did not provide any additional papers. Although we were hopeful that the categories that had papers would contribute to the empty categories as they have some common options.

Most papers (3/5) were added to the category COTS vs. OSS. In total 5 papers were added to start set represented by circles in Figure 17.

**Iterations:** Four iterations were needed to reach saturation.

1st Iteration: In the 1st iteration the reference list and citations of 5 papers were reviewed. Two papers in start set did not generate any papers. Snowballing papers in the COTS vs. OSS category added 3 papers in new category (only OSS), it also added papers in two existing categories (In-house vs. COTS and In-house vs. COTS vs. OSS). In total 10 papers were added in this iteration represented by triangles in Figure 17.

Figure 17: Evolution of identified papers
2nd Iteration: In this iteration a paper is added in a new category i.e. Make vs. buy vs. share. One paper each is added to COTS vs. OSS and In-house vs. COTS categories. In total 3 papers were added in this iteration represented by squares in Figure 17.

3rd Iteration: 2 papers were added in this iteration represented by diamonds in Figure 17.

4th Iteration: No papers were added in this iteration as SB of the two papers identified in 3rd did not find any new papers. Hence this is the final iteration in the SB process.

In the start set the COTS vs. OSS category had most papers, but at the end of the first iteration in-house vs. COTS category had the highest number of papers. Also 5/8 papers in the in-house vs. COTS category were from the COTS vs. OSS category. The 2 papers in the start set did not generate any papers. Interestingly, even though some categories were not covered in the start set, the SB procedure allowed us to find papers in these categories (here Only OSS and Make vs. buy vs. share).

Characteristics of start set: The creation of the start set was an important step as the papers identified in the SB process depend on the papers in the start set. The characteristics of a good start set are discussed in the SB guidelines [190]. The characteristics are stated in Table 24, along with the compliance to the guidelines in the SB procedure conducted for this paper.

The papers in the start set were organized in clusters. Each cluster contained papers that had no citation relations to each other, i.e. they did not refer to each other and/or did not have common authors. One paper was referring to another paper in the start set, this also was the same paper with common authors. Hence, they belonged to the same cluster. In this way the start set represented to the best of the authors knowledge at this stage different clusters. It may be fewer cluster, since other papers found later may bridge between two clusters identified at this stage in the process. Even though not all papers in the start set were adhering to the guidelines, each cluster had the characteristics of a good start set as the papers were referring to each other and did not have same authors. The papers found in start set were not published recently. Hence, based on this there is a good chance of finding papers in both FSB and BSB.

Citation Matrix: The referencing and citing between the papers is shown in Figure 18. The referencing is denoted as (x), for example it can be seen that P5 was referencing papers P6 and P11. Similarly P5 was cited by papers P12 and P15. It was not possible to cite the papers that had not been published yet, this is denoted as (−). There was one
exception, i.e. P12 (2007) is referencing P17 (2008) even though P17 was published later. This is because the reference was to an unpublished version of the same paper. The blank cells indicate that the papers were not referencing or citing other papers. The papers shown in Figure 18 are grouped according to the iterations:

- P1 to P5: start set
- P6 to P15: 1st iteration
- P16 to P18: 2nd iteration
- P19 to P20: 3rd iteration.

Furthermore, the papers in each iteration are arranged according to the publishing year. P4 was referencing six papers, which was the highest number of references to other included papers in this study. P11 had the highest number of citations, which was five papers. Both P4 and P11 were published in 2006. As the year 2006 was halfway in the publishing timeline (2000 to 2014), it gave the papers published in 2006 a good chance to be both cited and referring to papers included. P5 was also published

<table>
<thead>
<tr>
<th>Guideline</th>
<th>Compliance</th>
</tr>
</thead>
<tbody>
<tr>
<td>The papers in start set should not refer each other.</td>
<td>4 out of 5 papers are not referring to each other.</td>
</tr>
<tr>
<td>The number of papers must be reasonable. Focused (specific) research areas requires fewer papers than broader research area.</td>
<td>The number of papers depend on the research topic. Considering that the research conducted on this topic is not perceived to be extensive, we believe that the papers in start set were of a good size for SB, i.e. five papers.</td>
</tr>
<tr>
<td>The start set should cover several different publishers, years and authors.</td>
<td>The start set covers papers from three different years, two papers have common authors.</td>
</tr>
<tr>
<td>The start set ought to be formulated from keywords in the research questions.</td>
<td>The search strings were formulated using keywords in the research questions.</td>
</tr>
</tbody>
</table>
in 2006 although it had two citations and two references to included papers. This may be due to the fact that P4 and P11 belonged to a cluster which had a higher number of papers identified compared to the cluster that contained P5.

Here, it is interesting to observe that, even though papers belong to different categories, as a common topic was studied (development options), cross-referencing between categories could be found (see also Figure 17). Though, relying on this assumption is a risk, one should aim to cover categories as well as possible with at least one paper (see Section 5 for further elaboration).

4.2 Efficiency of SB (RQ2)

4.2.1 Efficiency of identifying start set (RQ2.1)

In total 90 papers were reviewed, out of which five papers were included in the start set, which resulted in the efficiency of finding a start set as 5.6 % (5/90). However, a closer examination is required to correctly determine the efficiency. A paper is excluded based on title if:

| Year | Ref. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|------|------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|
| 2002 | 1    | x |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2005 | 2    | x |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2006 | 3    | x |   |   | x |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2006 | 4    | x |   | x | x | x |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2006 | 5    | x |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2000 | 6    | x |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2001 | 7    |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2002 | 8    |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2004 | 9    |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2005 | 10   | x |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2006 | 11   |   |   |   | x |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2007 | 12   | x |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2008 | 13   |   | x | x | x |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2012 | 14   | x |   | x |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2014 | 15   | x |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2002 | 16   |   |   |   | x |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2008 | 17   | x |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2011 | 18   |   |   |   |   |   |   |   |   | x |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2012 | 19   |   |   |   |   |   |   |   |   |   |    | x |    |    |    |    |    |    |    |    |    |    |    |
| 2014 | 20   |   |   |   |   |   |   |   |   |   |    |   | x |    |    |    |    |    |    |    |    |    |    |

Figure 18: Citation matrix
a: The title is not related to the research topic/questions,
b: The paper is in the grey literature or it is not in English,
c: The paper has already been reviewed.

Exclusion based on title requires substantially less effort compared to abstract, introduction or full text. Hence efficiency is recalculated using the total number of abstracts and introductions reviewed in Phase 1 and full text reviewed in Phase 2 which is 6.4 % (5/78).

Overall, the total number of papers to be reviewed was relatively low for the start set when being compared to traditional search-based literature reviews (e.g. [35, 149] both selected from an initial set of over 600 papers). Though, this changes when taking the iterations also into consideration (see the following section).

4.2.2 Efficiency of iterations (RQ2.1)

The efficiency of each iteration is calculated and shown in Table 25.

As SB was continued until no new papers were found, the efficiency of the last iteration will always be zero. The efficiency of the second and third iteration was very low. In total 1044 papers were reviewed in all four iterations out of which 785 decision were based on titles. Thus, similar to the start set, many decision were based on titles. Moreover, the number of papers that were already reviewed increased in the iterations. Therefore, there was a need to revise the efficiency calculation by considering only the number of abstracts and reference contexts reviewed. The revised efficiency based on the number of abstract and reference context reviewed in the iterations is shown in Table 26.

As seen in Table 25 and Table 26, the efficiency drastically improved when papers, where decisions were made based on the title, were removed from the calculations.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>First iteration</td>
<td>5.5 % (10/181)</td>
</tr>
<tr>
<td>Second iteration</td>
<td>0.5 % (3/627)</td>
</tr>
<tr>
<td>Third iteration</td>
<td>1.0 % (2/188)</td>
</tr>
<tr>
<td>Fourth iteration</td>
<td>0.0 % (0/48)</td>
</tr>
</tbody>
</table>
4.2.3 Comparison of BSB and FSB (RQ2.1)

In this section BSB and FSB were analyzed to find the number of papers identified in each process, and hence identify the most efficient process. In total 15 papers were identified in the iterations out of which 7 papers were identified through BSB and 8 were identified through FSB, which is quite similar. Also the difference in the total number of papers reviewed in BSB (470) and FSB (574) SB was not large. Hence, in this particular case BSB 1.5% (7/470) and FSB 1.4% (8/574) SB were equally efficient.

It is also of interest to study the relevance of papers, where relevance refers to papers that should be considered for inclusion based on the relevance to the topic. In BSB, references not related to the research topic were quite often identified, such as references on research methods or tools used in the research. Also there was a chance of finding a lot of grey literature in the FSB as citations could be any document, such as master theses or project reports. Hence, we looked at the number of duplicates, grey literature and non English papers in BSB and FSB, respectively. The results are shown in Table 27.

The duplication percentage and grey literature count were nearly same. Surprisingly, less grey literature was found in FSB than expected. Hence, in total the noise in BSB was 463 (470-7) and in FBS 566 (574-8). The noise includes grey literature, non-English papers, duplicates and non-relevant papers. Among the considered noise, it is fairly easy to exclude grey literature, non-English papers and duplicates, in comparison to non-relevant papers.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>First iteration</td>
<td>14.5 % (10/69)</td>
</tr>
<tr>
<td>Second iteration</td>
<td>2.1 % (3/140)</td>
</tr>
<tr>
<td>Third iteration</td>
<td>4.4 % (2/46)</td>
</tr>
<tr>
<td>Fourth iteration</td>
<td>0.0 % (0/4)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase</th>
<th>Duplicates</th>
<th>Grey literature</th>
<th>Not English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backward</td>
<td>22.1% (104/470)</td>
<td>19.8% (93/470)</td>
<td>0.2% (1/470)</td>
</tr>
<tr>
<td>Forward</td>
<td>20.7% (119/574)</td>
<td>16.0% (92/574)</td>
<td>8.0% (46/574)</td>
</tr>
</tbody>
</table>
Therefore, we compare how many non-relevant papers (excluding other types of noise as described above) were identified in both BSB and FSB. We already know the number of grey literature, non-English papers and duplicates in our study. The total number of non-relevant papers in BSB was 265 (463-(104+93+1)) and in FSB 309 (566-(119-92-46)). Hence, the percentages of non-relevant papers in relation to the total number of papers reviewed were 56.3% (265/470) for BSB and 53.8% (309/574) for FSB. In this case FSB has slightly lower number of non-relevant papers in comparison to BSB.

Hence, neither BSB and FSB is considered to be more or less efficient in this case, both are equally important in finding relevant papers. Given the strategy to find the start set, there was a risk in missing a substantial number of papers if only choosing one of the two SB processes (BSB or FSB). This situation may vary depending on how the start set is chosen, for example if having a highly cited seminal paper the situation may be better than in the case described here.

4.2.4 Reference context as decision base (RQ2.2)

Inclusion and exclusion based on title and abstract was straightforward. In BSB the reference context was reviewed before reviewing the abstract of the paper being evaluated for the simple reason that it is better to use as much information as possible from the current paper before going to the new paper. However, reviewing the reference context alone did not lead to decisions in that many cases. For example, only 16 decisions were based on the reference context. Out of these papers, only one paper was decided to be included, and one was tentatively included, although the latter got excluded based on reading the full text. The remaining 14 papers were excluded based on reference context. The following examples of reference contexts have been formulated in a way that they led to a decision.

- It is easier to exclude than to include using the reference context - For example, in the following reference context “Detailed discussions on sample selection in this study are reported in...” it is clear that the paper is a method paper. Hence, it was easy to exclude the paper based on the reference context.

- Only one paper was finally included based on the reference context, which was as follows “Some studies compared the differences between COTS and OSS products per se and concluded that there is still no empirical evidence that OSS fosters faster system growth and that OSS is more modular than closed source software.”. It was clear from
the reference context that OSS and COTS were compared, which was one of the inclusion criteria of the study.

• However, sometimes the reference context can be deceiving. For example, consider the following reference context: “Pizka reported experience on building the same system with three different strategies, such as wrapping an OSS component, adapting/changing the source code of the OSS component, or building the same component from scratch”. This looks very promising as it compares OSS and in-house (inclusion criterion). However, the paper is related to Operating System development, which is not within the scope of the study. Thus, the paper was excluded based on reading the full text.

Though, in most cases the reference context did not prove as useful as anticipated, i.e. no decision could be made. This means that the decision could not be made based on the reference context because of the following reasons:

• The reference context might not reflect the goal of the paper - Sometimes the reference context might be something that is of no interest, although the paper actually might have other parts that are interesting. For example, in the following reference context “Other argued that it is the quality, not the number, of the eyes looking at code that count”. The reference context is about quality, but the papers also compares OSS with COTS, which cannot be seen in the reference context.

• The reference context is vague or unclear - “Previous studies have looked at using COTS and OSS components in software development.”. In this reference context, the reference is vague. It is not clear what “looked at” means. Sometimes the reference context includes just a keyword which is not very useful to understand. Whenever the reference context is not understandable, the enclosing paragraph was read to understand the context.

• Difficulty in finding reference context - Often different reference styles are used. Some use numbering while some other use author names as a reference index. In both cases it becomes difficult to track the reference context if no ordering is used. For example, the first reference context could be 15 instead of starting with 1 which makes navigation difficult.

To make the reference context more useful for systematic literature studies, it is required that authors describe the references more clearly.
4.2.5 Total efficiency of SB (start set and iteration)

Based on the total number of papers reviewed in start set and iterations, the efficiency becomes 1.8 % \((5 + 10 + 3 + 2 + 0)/(90 + 181 + 627 + 188 + 48)\)

As efficiency is the ratio of noise vs. relevance, we calculate the total amount of noise comprised of grey literature, duplicates, non-English papers and non-relevant papers. In our study the amount of noise in SB was 1114 (1134-20) papers, including the total of grey literature, duplicates and non-English papers, which was 483 (195+239+49). Hence, the number of non-relevant papers is calculated as noise - (grey literature + duplicates + non-English papers), which were 631 (1114-483) papers. Hence, the percentage of total noise was 98.23 % \((1114/1134)\) out of which the percentage of non-relevant papers was 55.64% \((631/1134)\) and 42.59% \((483/1134)\) of the total papers consisted of grey literature, duplicates and non-English papers.

It was easy to exclude papers based on grey literature, duplicates and non-English papers in comparison to the exclusion of papers based on non-relevance of the paper. Hence, we recalculate the efficiency based on the number of relevant papers in comparison to the total number of non-relevant papers, which was: 3.17% \((20/631)\)

As progress was made, more decision were based on titles, the percentages of decisions made based on titles for each iteration was as follows: Iteration 1 = 61.9 %, Iteration 2 = 77.7 %, Iteration 3 = 75.5 %, and Iteration 4 = 91.7 %. Thus in this case, the percentages for exclusion based on titles increased for each iteration.

There were some factors that affected the efficiency of SB such as inclusion/exclusion criteria and data extraction. Clear inclusion and exclusion criteria were a must in SB, otherwise there was a risk to include unwanted papers. The latter will result in conducting SB on unwanted papers that later get excluded. Thus, all papers potentially identified from a paper that later was excluded must also be excluded. Including papers that later were excluded will result in a waste of time and effort. It was beneficial to extract the data of the included papers before conducting the SB on the new papers identified. Not only did it assure a valid inclusion as detailed reading was done, but also it gave a good idea of the reference context. The inclusion or exclusion based on reference context can be done during extraction.
4.2.6 Efficiency of SB vs. DB search (RQ2.3)

Finally we compared the efficiency of the two searches. The efficiency indicates the amount of noise and relevance in DB search and SB. To calculate, first the total number of papers were considered in the calculation, which was the total efficiency. Then the revised efficiency was calculated based on the total number of abstracts reviewed.

As seen in Table 28 the total efficiency of DB search was more than SB, i.e. in SB more papers were reviewed. The efficiency was also calculated by considering the total number of abstracts reviewed. The percentages shown in Table 28 indicate that, based on the total abstract reviewed SB was more efficient than DB search. Hence, in this study the efficiency of SB is comparable to the efficiency of DB search.

4.3 Reliability of SB (RQ3)

In this section the two searches are compared on the following aspects

- Papers identified by both studies
- Development options identified by both studies
- Dimension of research identified by both studies
- Conclusion derived from both studies

In each comparison the number of papers uniquely identified by the SB and DB searches and papers commonly identified by both are presented. This presentation allows us to compare SB and DB search in three ways. 1: all papers identified by each search; 2: unique papers identified by each search; 3: papers commonly identified by both searches.

4.3.1 Common and unique papers (RQ3.1)

As seen in Figure 19, in total 24 papers were identified, out of which 7 were commonly identified by both SB and DB search. 13 papers were uniquely identified by SB and 4 were uniquely identified by DB. In total

<table>
<thead>
<tr>
<th>Phase</th>
<th>Total efficiency</th>
<th>Only abstracts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snowball</td>
<td>1.76% (20/1134)</td>
<td>6.23% (20/321)</td>
</tr>
<tr>
<td>Database</td>
<td>3.21% (13/404)</td>
<td>5.70% (13/228)</td>
</tr>
</tbody>
</table>
SB found 20 papers and DB search found 11 papers. The overlap between two searches was more than the papers identified only through DB search (7>4). Snowballing found more papers than DB search, this indicates that SB is more reliable. However we need to investigate how the identified papers support the different development options, which is discussed in the next section.

4.3.2 Common and unique dev. options (RQ3.2)

The main objective of the literature mapping was to identify studies that compare different development options. For this reason it was important to compare the searches based on development options identified. No papers were identified for COTS vs. Outsource and Outsource vs. OSS by both searches, hence they are not mentioned in the comparison.

As shown in Figure 20, considering the unique and common papers identified by both searches, we can see that in total both SB and DB were comparing 5 options each.

Among the 5 options, 3 were commonly identified by both searches, which were In-house vs. CCOTS, COTS vs. OSS and Make vs. buy vs. share (OSS). This means that two options were uniquely identified by each search. The options In-house vs. OSS and In-house vs. Outsourcing were only identified by DB. It should be noted that for the com-

![Venn diagram for the overlapping papers](image-url)

Figure 19: Venn diagram for the overlapping papers
parison In-house vs. OSS, SB identified one paper. Though, it was also comparing COTS in addition, hence it was mentioned as a separate comparison i.e In-house vs. COTS vs. OSS. Hence it can be said that In-house vs. Outsourcing was the only option that was not identified by SB. Half (2/4) of the papers found only through the DB search contributed to the In-house vs. Outsource category, for which SB process did not find any papers.

As shown in table 21 three additional search strings with the inclusion of software keyword were used to identify papers comparing outsource option. However, the additional searches did not result in finding new papers. Since DB search was successful in finding the papers comparing the outsource option, it implies that a more extensive search should have been used in finding the papers in start set. All the unique paper found through the DB search were found in Google scholar using the search strings in Table 21, although not in the first 10 search results. Hence, it is clear that more search results should have been reviewed, specially for categories where there were no papers identified.

Options Only OSS and In-house vs. COTS vs. OSS were uniquely identified by SB. It is interesting to note that the search strings used to identify the papers were comparing only two options. However, in SB a paper comparing three options was also identified. Also papers discussing only one option were only found through SB.

Most of the common papers identified (5/7) belong to the In-house vs. COTS category, and 5/6 papers in COTS vs. OSS category were found through SB.

![Figure 20: Options coverage](image-url)
4.3.3 Combined and unique dimensions (RQ3.3)

As seen in Figure 21, both searches have been successful in finding papers in all dimension. Database search used additional keywords such “decision”, “trade-off” and “selection” in the search strings. Snowballing was successful in finding a good number of papers in each dimension, even though additional keywords were not included.

4.3.4 Conclusions consistency (RQ3.4)

In the mapping study for which our SB and DB search procedures were used, conclusions were formed from collective results of the DB search and SB. We investigate whether the same conclusions could be drawn when only SB and only DB search would have been used.

Table 29 presents the results of the investigation, a tick (✔) mark indicates that the results still hold true, whereas a cross mark (×) indicates same conclusion cannot be drawn.
<table>
<thead>
<tr>
<th>No.</th>
<th>Conclusions</th>
<th>Options</th>
<th>Only Only</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SB</td>
<td>DB</td>
</tr>
<tr>
<td>1</td>
<td>Majority of primary studies considered in-house development, followed by OSS and COTS.</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>2</td>
<td>Some studies only consider one option in decision making.</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>3</td>
<td>Only two studies consider outsource.</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td><strong>Dimensions</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Solutions mainly concentrate on In-house vs. COTS.</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>5</td>
<td>Comparative studies are mainly focused on COTS vs. OSS.</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>6</td>
<td>In-house vs. COTS vs. OSS is the only option that does not provide any decision criteria.</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td><strong>Decision Criteria Factors</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Software quality:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 quality criteria were identified.</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>8</td>
<td>Project metrics:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>In-house vs. COTS considered all project metrics with most studies considering cost.</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>9</td>
<td>Conflicting results were reported with regard to cost.</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td><strong>Context factors</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Source code reliability has highest number of studies in COTS vs OSS category.</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>11</td>
<td>Studies disagree on market evolution, ease of use and vendor support is being an issue.</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td>12</td>
<td>Studies disagree on whether OSS affects maintainability negatively or positively.</td>
<td>✔</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td><strong>Solutions</strong> (5/8 solution papers were commonly found)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Most studies define optimization models.</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td><strong>Research Types</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Most studies are empirical.</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>15</td>
<td>All research types proposed by Wieringa[186] could be identified in the set of primary studies.</td>
<td>✔</td>
<td>✗</td>
</tr>
</tbody>
</table>

Table 29: Conclusion analysis
Interestingly papers identified through the SB process had some conflicting results, which were not reported through papers identified through DB. The entries 9, 11 and 12 in Table 29 are such conflicting results.

Only a subset of research types proposed by Wieringa [186] were identified by DB, whereas papers identified through SB had covered all research types.

Overall only one conclusion would not hold true if only SB was used and nine conclusions would not hold true if only DB search was used. If more care was taken in the creation of start, then SB seems to be highly reliable in this case.

5 Discussion

The results of this study help to understand some potential factors affecting the efficiency and reliability of SB. Such improvement areas or factors are discussed in this section. Organizing papers in start set into different categories based on the different concepts of the study can help in finding the relevant papers. It is important that each category has at least one paper in the start set. Even though papers in the categories contribute to finding papers in other categories through SB, relying on it to find papers in empty categories did not work. Hence proceeding with empty categories in the start set is considered as a risk of missing some relevant papers. Also clear inclusion/exclusion criteria and extracting data from included papers before the papers are used in SB can improve efficiency.

The efficiencies of SB and DB search techniques do not differ a lot when the total number of abstracts reviewed are considered. However, when the total number of papers reviewed are considered, SB seems less efficient. As the search results might include duplicates, grey literature or non-English papers, reviewing titles does not take much time and effort. Hence, comparing efficiencies based on abstracts seems more reasonable. The BSB and FSB are both equally efficient in this study. Hence, the results of this study support the recommendation suggested in [86] that both BSB and FSB process should be implemented. The efficiency might also be affected by the effort required to review the reference context. In this study only 16 decisions were based on reference context. Therefore, in this case the reference context did not proof to be useful in including or excluding papers. However, the usefulness of the reference context highly depends on how well the reference context is described in the primary studies.
In this study 45.9% of the studies were identified through DB search and 83% of the studies were identified through SB, out of which 29.1% were overlapping papers. In total SB identified more papers in relation to DB search which is similar to the findings of [67]. However, the SB procedure failed to identify a few important papers. The impact of additional and missing papers identified is stated below.

**Impact of additional papers:** The papers found in SB provided a deeper investigation on the factors influencing the adoption decision. For example, some studies suggest that OSS affects maintenance positively, while some other studies suggest that OSS affects maintenance negatively. Such conflicting results were not found in papers identified only through the DB search. The DB search strings were more detailed and specific in comparison to SB search strings, yet SB found more relevant papers than the DB search. However, more general search strings (as those used in the SB procedure, see Table 21) resulted in reviewing a lot of noise, which was 98.23% and the noise in the DB search was 95.17%. This contradicts with the findings of [86] where the noise in SB was relatively less in spite of the more general search strings.

**Impact of missing papers:** Snowballing identified more relevant papers and contributed in better analysis. However it failed to identify papers comparing in-house vs. outsource which were identified through DB papers.

6 Conclusion

In this study, we evaluated the efficiency and reliability of SB by comparing it with DB search used in a mapping study on choosing among different development options. The results of this evaluation study are summarized in the following research questions.

- **RQ1 How were the relevant papers identified through the evolution of SB process?**
  The start set consisted of 5 papers, out of which 2 papers did not generate any papers on SB them. Half (10/20) of the total papers were identified in the first iteration. The start set was organized in different categories representing different concepts. We found that SB of one paper in one category added papers in other categories. 5/8 papers in one category were added by a paper in other category. Also new categories emerged as the SB progressed.

- **RQ2 How efficient is SB?**
  Although the total number of papers reviewed by SB were more
compared to DB, 42.59% of the total papers were either grey literature, duplicates or non-English papers, which did not take much time and effort to exclude. The total efficiency of SB is 3.17% based on the papers included in relation to the total non-relevant papers (excluding grey literature, duplicates or non-English papers). The difference of efficiency based on the total number of abstracts reviewed by DB search and SB is not too drastic hence, we conclude that in our case the efficiency of SB and DB search is comparable. Also we found that, within SB, BSB and FSB are equally efficient and first iteration was most efficient.

- **RQ3 How reliable is SB in capturing all relevant papers?**  
The DB search identified 45.9% of the papers, SB found 83% of the papers. Snowballing provided a richer analysis, included papers representing all dimension of decision making and all types of research types classified by Wieringa [186] were identified. However, SB failed to find papers in in-house vs. outsource comparison category, whereas DB search was successful. This is a result of leaving the categories in the start set empty. The missing papers were found in Google scholar using the same search strings. However, since only first 10 results from each search string were used, they were not identified. Hence, if more care is taken to find at least one paper in each category then SB would be highly reliable and be considered as a good alternative to DB search.
CHOOSING COMPONENT ORIGINS FOR SOFTWARE INTENSIVE SYSTEMS: IN-HOUSE, COTS, OSS, OUTSOURCING OR SERVICES? – A CASE SURVEY

Abstract: The choice of which software component to use influences the success of a software system. Only a few empirical studies investigate how the choice of components is conducted in industrial practice. This is important to understand to tailor research solutions to the needs of the industry. Existing studies focus on the choice for off-the-shelf (OTS) components. It is, however, also important to understand the implications of the choice of alternative component sourcing options (CSOs), such as outsourcing versus the use of OTS. Previous research has shown that the choice has major implications on the development process as well as on the ability to evolve the system. The objective of this study is to explore how decision making took place in industry to choose among CSOs. Overall, 22 industrial cases have been studied through a case survey. The results show that the solutions specifically for CSO decisions are deterministic and based on optimization approaches. The non-deterministic solutions proposed for architectural group decision making appear to suit the CSO decision making in industry better. Interestingly, the final decision was perceived negatively in nine cases and positively in seven cases, while in the remaining cases it was perceived as neither positive nor negative.

1 INTRODUCTION

Architectural decision making distinguishes different types of decisions, Kruchten [105] divided decisions into structural and behavioral decisions. Structural decisions are concerned with the elements and their interfaces of architectural components. The choice of which components to use in a software-intensive system is thus a structural architecture decision. The choice of the right components is an important factor for the success of a system developed with Off-the-Shelf (OTS) components [9], which includes Components off the shelf (COTS) and Open Source Systems (OSS). Currently, there is a lack of empirical evidence and understanding how practice selects OTS components, as
pointed out by Ayala et al. [9] “to improve OTS component selection practices; the research community must understand what the actual industrial OTS selection practices are in order to envisage more realistic and effective solutions”. Ayala et al. [9] and other researchers [63, 111] have provided insights of how practice selects components. Their focus was OTS development. However, when choosing a component another consideration is the source of the component, leading to the following question: Should the component be developed in-house, should an OTS component be squired, or should the development of the component be outsourced?

This question is of high relevance as different component sourcing options (CSOs) have distinct characteristics that have to be taken into consideration. COTS based development implies a lack of control over evolution and the quality of the component (cf. Torchiano and Morisio [177]). Torchiano and Morisio also point out that the development with COTS also has an implication on the development process, which needs to focus on the combination and testing of the components, as well as dealing with the evolution of the components that is not in the control of the integrator. When choosing outsourcing the issue of the lack of control does not arise. Though, specific issues to outsourcing may materialize. In particular distances (cultural, temporal, and geographical) have an effect on the development process [81], which has to be adjusted to cope with the distances [180]. In addition, software architects are challenged in mentoring and facilitating learning at the outsourced organization, and have to guard the integrity of the architecture during the learning period [21]. Given the significance of the potential effects on the organization making a choice for a CSO, it is important how organizations choose between them. The evidence of how decisions are made for CSOs is very limited with only two industrial case studies in the area [Chapter 2].

The need to better understand decision making in industry for OTS selection [9] and CSO selection [Chapter 2] has been highlighted. Furthermore, the need for evaluating the outcome of the decision is important [70]. To address the above mentioned needs we provide an analysis of 22 cases about how decision making took place when choosing among CSOs for adding components in industrial software-intensive systems. We identified the CSOs considered, the stakeholders involved, the criteria considered, and the actual decision taken. Furthermore, reflections on the decision reached were obtained, such as whether the “right” decision was reached.
We considered four CSOs: In-house developed components, components off-the-shelf (COTS), open source components (OSS), and outsourced development of components, which are defined as follows:

- **In-house**: The component is developed within the same company. Badampudi et al. [Chapter 2] highlight that it is still considered in-house development when the development is distributed, as long as it takes place within the company.
- **COTS**: This option stands for “components off-the-shelf” or “commercial-off-the-shelf”, which are already developed (pre-built) and for which the source code is usually not available to the buyer.
- **OSS**: Open source components are also pre-built, but the source code is available. OSS components are commonly built by a community.
- **Outsource**: Another company is developing the component and is given the contract by the company wanting to obtain the component.

The research method used was case survey [107]. The cases were gathered in the context of the ORION project. The results add to the limited empirical knowledge (cf. [9]) of how component decisions have been made in industrial practice, in particular the choice between CSOs. The cases were collected based on the researchers’ experiences in industrial settings and based on interviews with experts from companies. Overall, 22 cases are included in this case survey.

The remainder of the chapter is structured as follows. Section 2 presents the related work. Section 3 describes the research method used. Section 4 explains the results, followed by their discussion (Section 5). Section 6 concludes the chapter.

## 2 RELATED WORK

We first present the decision making problem targeted in the chapter by characterizing the alternatives for the decision (In-house, OSS, COTS, and Outsourcing) based on the knowledge presented in the literature. Furthermore, the different options are compared based on their strengths and weaknesses. Thereafter, we characterize the decision-making for components based on the literature. We always indicate which findings were theoretical or empirical.

1 http://orion-research.se
2.1 Decision making problem

Component decision making takes place on different levels. Figure 22 depicts the different levels of decision making, distinguishing between the CSO choice, vendor choice, and the selection of the actual component. On the top (strategic) level, the CSO is selected. On the provider selection level, depending on the option, either vendors, suppliers, or communities may be chosen. Finally, on the lowest level a concrete component is chosen. Note that we do not imply a particular order in which the decisions are made.

To facilitate and support the choice of components, different researchers provided insights about the different CSOs. The selection of insights are summarized in Table 30 to characterize the options companies can choose from. The table shows the distinguishing characteristics between the CSO options as they were presented in the literature. In the case of the characterization of the CSOs all findings are empirical. Given that the CSOs have different characteristics they have different implications on the criteria governing the choice between them.

Badampudi et al. [Chapter 2] synthesized the findings from the literature about the strengths and weaknesses of the CSOs in comparison to each other, and distinguished between theoretical and practical influences. A summary of their synthesis is shown in Table 31. The table shows the synthesized evidence (cf. Badampudi et al. [Chapter 2]) of the performance of CSOs in comparison to each other, and states whether they are empirical (E) or theoretical (T). If a CSO affects a cri-
Table 30: CSO Characteristics

<table>
<thead>
<tr>
<th>CSO</th>
<th>Characteristics cited from papers</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-house</td>
<td>Control over features and evolution of the product</td>
<td>[65]</td>
</tr>
<tr>
<td></td>
<td>Control over the development process and projects initiated</td>
<td>[65]</td>
</tr>
<tr>
<td></td>
<td>Awareness and knowledge of the system lies with the developing organization</td>
<td>[21]</td>
</tr>
<tr>
<td>COTS</td>
<td>Lack of control (evolution, quality, functionality) as components are black-boxes</td>
<td>[177]</td>
</tr>
<tr>
<td></td>
<td>COTS development influences the development process</td>
<td>[177]</td>
</tr>
<tr>
<td></td>
<td>COTS are mostly used in real-time, embedded and distributed computing</td>
<td>[181]</td>
</tr>
<tr>
<td>OSS</td>
<td>Motivation for choosing OSS: higher quality, shorter time to market, and cost reduction</td>
<td>[63]</td>
</tr>
<tr>
<td></td>
<td>Most important criterion for choosing between different OSS is the vitality of the community, functionality, standard compliance, and ease of integration</td>
<td>[63]</td>
</tr>
<tr>
<td></td>
<td>OSS components are used without modification</td>
<td>[111]</td>
</tr>
<tr>
<td></td>
<td>Cost of locating and debugging defects in OTS-based systems is substantial</td>
<td>[111]</td>
</tr>
<tr>
<td>OTS</td>
<td>Issues and challenges in estimating integration effort and debugging</td>
<td>[116]</td>
</tr>
<tr>
<td></td>
<td>Cost estimation factors are: time to understand the OTS, OTS inflexibility, and dealing with OTS evolution and corresponding updates needed by integrators.</td>
<td>[111]</td>
</tr>
<tr>
<td></td>
<td>OTS rarely affect quality negatively (reliability, performance, security were problematic if problems occurred)</td>
<td>[111]</td>
</tr>
</tbody>
</table>

Terminology: A criterion positively in comparison to others, this is indicated by a “+”; if it is negatively affected this is indicated by a “−”; if no difference is observed this is indicated by a “=”; If both, positive and negative effect is observed this is indicated by a "+,-". Both empirical data (E) as well as findings based on theoretical reasoning (T). The synthesized evidence is based on a limited set of studies with varying scientific rigor and
relevance (cf. [Chapter 2]. Only nine empirical studies were identified by Badampudi et al. that compared CSOs.

In the case of the study by [63] one of the interviewees pointed out that “we do not want our developers start writing programs from scratch. At least in the Java world we find very often that the highest quality libraries are the ones that are open source and not the commercial”. In this case, the CSO has already been chosen (OSS). However, in this case survey study we explore decisions where the CSO choice had to be made, which is of interest to practice and research given the distinguishing characteristics of the CSOs (see Tables 30 and 31).

Table 31: Synthesis of CSO comparisons

<table>
<thead>
<tr>
<th>Criterion group</th>
<th>COTS</th>
<th>OSS</th>
<th>In-house</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time to test and integrate [74]</td>
<td>-</td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>Time to market [115]</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cost of components [37, 115, 166]</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Total cost of ownership [52]</td>
<td>=</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>Cost of replacing components [115, 130]</td>
<td>-</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Maintenance cost [23, 76, 110]</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td><strong>Effort</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Selection and integration effort [76, 115]</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Development effort [115]</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td><strong>Quality</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quality in general [115, 126, 130, 166, 174, 179]</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td><strong>Market trend</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component evolution [23, 52, 80, 110, 115]</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>Source code</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access and use of source code [37, 52, 76, 78, 113, 115, 153]</td>
<td>-</td>
<td>+</td>
<td>,,-</td>
</tr>
<tr>
<td>Source code documentation [80, 166]</td>
<td>-</td>
<td>+</td>
<td>,,-</td>
</tr>
<tr>
<td><strong>Technical support</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vendor response time [78, 115, 130]</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Support availability [52, 179]</td>
<td>-</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Code customization [126]</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Changes in requirements [23, 80, 113, 179]</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>License</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>License fee [37, 115]</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>License obligations [52, 130, 153, 166]</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
2.2 Characterizing decision making

As mentioned earlier the choice of a component is an architectural decision, specifically a structural decision according to Kruchten [105]. Thus, we first review the literature on architectural decision making to elicit important characteristics of the decisions made. Thereafter, we separately present the literature on the different decision levels presented in Figure 22.

2.2.1 Architectural decision making

Theoretical: Kruchten [105] defined an ontology for architecture decision making. Kruchten divides architectural decisions into different categories:

- **Structural decisions**: These decisions are concerned which elements (such as components) to include in an architecture, as well as the design of their interfaces. Also, ban-decisions of what should not be included can be specified here. In addition, the behavior of the components is specified, which describes the interaction between different architectural elements.

- **Property decisions**: These decisions are concerned with how to design the architecture (e.g. architectural style) in order to achieve certain properties (e.g. performance). As relevant properties Kruchten mentions usability, security, politics, cost and risk.

- **Executive decisions**: These decisions are concerned with contextual factors, such as the technology or the processes used.

To support the documentation of architectural decisions van Heesch [72] proposed a framework. Elements to be documented were the state of the decision, the decision making group, the problem specifying why the decision was made, the decision taken, the alternatives considered, system concerns, and the history of the decision. An interesting aspect mentioned is that decisions are often not taken in isolation, but rather in groups, Hence, van Heesch proposes to also capture links to all related decisions. According to Heesch typical stakeholders in the decision making are architects, reviewers, managers, customers, requirements engineers, new project members, and domain experts.

Given that architectural decisions are commonly prepared and made by a group of people, group decision making (GDM) plays an important role. The literature suggests a variety of approaches to support GDM. Malavolta et al. [119] created a meta-model allowing to capture
the different viewpoints of stakeholders. The meta-model captures rationales, issues, concerns and criteria of the decision and is linked to a group decision model. The group decision model captures and tracks the decisions that are related to a group of stakeholders. Features for conflict resolution strategies and traces to other relevant artifacts are also captured. Similarly, Nowal and Pautasso [132] provided an approach for the systematic recording of argumentation viewpoints. Argumentation viewpoints should raise situational awareness of teams and hence provide a means to build a consensus. To record an argumentation viewpoint design issues, alternatives and the stakeholder positions in relation to the alternatives are captured. Nowal and Pautasso developed a tool supporting brainstorming and the evaluation of design elements. A Software architecture warehouse facilitates the tracking and sharing of decisions by a team. Zimmermann et al. [197] highlight that a common problem of architecture decision making is the lack of a documented rationale. Zimmerman et al. emphasize the identification of reusable decisions, and hence highlight the importance of capturing the information about the decision. They also point out that decisions are not made in isolation from other decisions. They propose a conceptual framework for proactive decision identification, collaboration and enforcement. The framework provides a wide range of features, such as proposals for templates and the identification of reusable decisions, process support for identifying, making and enforcement of the decisions, pushing to-do-lists to the architectural teams, and providing support techniques to reflect on the decisions. The feature of decision enforcement allows to directly inject decisions into the code. The proposal has been proven to be practical in a service-oriented architecture (SOA)-based application.

**Empirical:** Capilla et al. [26] describe how to capture architectural design decisions and provide tool (Architecture Design Decision Support System) support to facilitate knowledge transfer and provide deeper insights into the rationales behind architectural decisions. They conducted a case study with students using the tool and captured the effort of the architecting activities and the effort spent on the decision. The usability and ease of use was positively assessed. Reasoning activities for capturing the design decisions required 47% of the overall effort spent. To support reasoning activities Razavian et al. [146] introduced design reflections in student groups by asking reflective questions to the students. This led to backtracking and rethinking design decisions. Overall, external triggered reflections improved the quality of discourse and had a positive effect on reflections taking place within the groups.
Tofan et al. [176] evaluated a decision making process (GADGED) based on the repertory grid approach [34] to determine whether it increases the consensus in GDM for architectures. The repertory grid approach systematically elicits the decision alternatives, constructs by which they are compared and the ratings for each alternative in relation to the constructs. A statistical analysis is thereafter conducted [34]. The findings showed that GADGED was useful for group decision making, in particular for inexperienced architects.

To determine whether existing GDM approaches provide sufficient supports to groups wanting to make architectural decisions Rekha and Muccini [147] propose an evaluation framework. The framework checks the presence of features to facilitate learning, problem analysis, the ability to rate alternatives, as well as conflict resolution, to name a few. Rekha and Muccini applied the framework to existing GDM approaches and found that they do not fully support GDM in their current form. Examples of features missing are the ability for stakeholders to explicitly indicate their preferences, as well as conflict resolution mechanisms and rules determining how the preferences of stakeholders should be taken into account. Groher and Weinreich [70] asked student groups with significant practical experience and GDM knowledge to individually develop GDM tools. Later, the tools were compared against the framework proposed by Rekha and Muccini [147]. They found that the tools mostly fulfilled the features. New interesting ideas emerged, in particular the possibility to provide features for the review of decisions after they have been made and communication means between stakeholders inside the tool.

As architectures evolve continuously decision makers have to conduct assessments on a continuous basis. For each decision a trade-off has to be made between different properties of the architecture, such as cost, performance and reliability. Cortellessa et al. [43] proposed a model-based framework utilizing an optimization model with the aim of minimizing cost while thresholds are set with regard to reliability and performance. Initial evaluations showed that the framework performed better in decision making compared to humans, which provided indications of the usefulness of optimization models in decision making.

2.2.2 Choosing CSOs

*Theoretical:* The solutions identified in the literature by Badampudi et al. (cf. [Chapter 2]) only aiming at deciding between CSOs were solution proposals without a rigorous empirical component. To decide between
in-house and COTS development optimization models have been pro-
posed [40, 42, 110, 130, 144]. As an example, an optimization model may
be used to make a choice of components to maximize reliability, while
minimizing delivery time. Different constraints can be defined, such as
costs. For making a trade-off between in-house development and out-
sourcing Kramer and Eschweiler [102] propose to utilize clustering to
group components and utilize requirements dependencies and priori-
ties to make the choice. Kramer et al. [104] propose utilizes outsourcing
potential, knowledge specificity, and interdependencies to make a deci-
sion utilizing decision tables.

2.2.3 Choosing Vendors

Empirical: The selection of vendors and suppliers has been considered
in several studies. A secondary study has been presented by [94], which
identifies the success factors in the selection of offshore outsourcing
vendors. In total 22 factors were identified from the primary studies.
Cost-saving, skilled human resource, appropriate infrastructure and
quality of products were among the factors that were identified in more
than 50% of the primary studies. Studies [76, 194] present findings re-
lated to vendor selection and community selection, respectively. The re-
relationship between component integrator and external vendor (COTS
vendor and OSS community) is considered important to minimize the
time and effort on technical, legal and business negotiations [76]. In ad-
dition, the collaboration between the OSS developers and actives users
is considered important to maintain good communication and relation-
ship with the OSS community.

2.2.4 Choosing OTS components

Empirical: Vale et al. [181] conducted an extensive systematic mapping
study on component-based software engineering (CBSE) to find open
issues and research trends. With regard to component selection they
found that COTS are commonly selected for embedded, real-time and
distributed systems. Looking at the field of CBSE limited evaluations
and a lack of empirical knowledge have been identified. Similarly Ayala
et al. [9] found limited evidence in relation to selection practices in
practice. To address this gap multiple researchers conducted studies
on how OTS components are chosen.

Ayala et al. [9] found a gap in the processes proposed in the litera-
ture versus what has been used in practice. For example, component
repositories are proposed, but not often used in practice.
Ayala et al. [9] and Gerea [63] found that common steps in the component selection process are identification, evaluation, learning and knowledge management, use of the component, and choosing. Gerea also found that the process of selection process is impacted by the component size. Larger components were selected earlier in the development life-cycle. The process used for selection is rarely formal and rather ad-hoc in nature, which has been reported by multiple authors (cf. [9, 111, 177]). For COTS selection Li et al. [111] found that companies use prototyping to learn about COTS.

In the case of OSS the stakeholders involved in the process are the owners of the OSS, developers (distinguishing between core and general developer), system testers, user support as well as problem reporters and users [17]. The initiation of the decision for OSS as well as the preparation and investigation is mostly done by software developers. The leaders (such as the CEO) then take the decision and have the final word [63]. End-users are also involved in the decision making [63]. To systematize the stakeholder roles three perspectives can be identified, the decision function (initiation, decision preparation, and decision making) [63], the relationship to the component (vendor of the component, integrators, and customers/ end-users) [177], as well as the roles in the development life-cycle (such as developers and testers) [17]. The perspectives and corresponding roles are summarized in Figure 23. The figure shows the different stakeholder perspectives derived from the literature based on which a role can be described. For example, a software tester initiating the decision on integrator side.

![Figure 23: Stakeholder perspectives](image)

Figure 23: Stakeholder perspectives
Gerea et al. [63] found that the most important criteria for decision making for OSS are the compliance to standards and the matching of the functionality provided by the component to the needs. Another important criterion for OSS component selection is the vitality of the OSS community. Architectural considerations are also of great importance [177]. In addition to the criteria, risk factors need to be considered during component selection, integration and maintenance, which were identified by Morandini et al. [128] and Li et al. [116]. The risk factors were related to ill-estimation of selection, integration and maintenance effort, wrong component selection, component integration and maintenance failure. Furthermore, legal risk with respect to intellectual property and license were identified. Risk reduction activities were to invest in learning the relevant components and integrating components that are unfamiliar first [116]. In addition extensive testing of the components is important [116].

The best practices for COTS selection in literature and in industry were identified by Rikard et al. [106] and three COTS selection methods (Comparative evaluation process, COTS-based requirements engineering and framework of COTS selection) were compared by Wanyama and Far [183].

3 Method

We first present the research questions, and thereafter provide details on the case survey method and how it has been used in this study. We utilized the guidelines by Larsson [107] during the design of the research.

3.1 Research Questions

The first research question was concerned with how CSOs were selected in the context of software-intensive system development. In order to determine how the decisions were made, several sub-questions needed to be answered:

- **RQ1: How are CSOs chosen?**
  - **RQ1.1:** Which CSOs (a) were considered and (b) which CSOs among those considered were chosen? The first sub-question (a) provides an insight of the decision making problem formulated by the companies. Knowing the combinations frequently considered by companies helps to guide future re-
search, in particular it shows which CSOs should be compared empirically so that companies may use this information as input to their decision making process. In the second sub-question (b) we investigate whether trends are visible of one CSO being preferred over another, which provides early indications of preferences between CSOs with respect to their advantages and disadvantages.

- **RQ1.2**: Which stakeholders were involved in the decision process? Whether a decision is taken individually or in a group is an important aspect when designing decision support for CSOs. Also, understanding the different roles involved in the stages of decision making provides practitioners with the possibility to reflect on whether the identified roles are also relevant in their decision scenarios.

- **RQ1.3**: Which criteria (a) were considered for making the decision and (b) which criteria initially considered ended up as significant for the final decision? Similar to RQ1.1, the criteria show which variables need to be studied when comparing CSOs empirically. Criteria considered in the preparation for the decision making, though not considered as relevant in the final decision, point to a potential for improvements in decision making processes. Thus, it is of interest which criteria are actually those that were essential in the final decision.

- **RQ1.4**: Which decision making approach/model was used? In the literature several methods are proposed (such as optimization models). Understanding the methods used in practice allows to determine whether the solutions proposed in research found their way into practice, and which methods used in practice need to be integrated into decision support systems.

The second research question aims at understanding the outcome of the decision making process:

- **RQ2**: What were the decision outcomes of the CSO selection process?
  - **RQ2.1**: What was the effort invested in the decision making process? When looking at solutions in software engineering (in this case how CSOs are chosen) the cost factor has to be considered. Thus, we investigated the estimated effort spent on decision making.

  - **RQ2.2**: Were the CSOs chosen considered the “right” choice retrospectively? In this question we reflect on the selected options
for making a decision from the retrospective point of view: finding that decisions are mostly positive indicates a success of CSO decision practices. Negative results on the other hand point to the need for decision support and adjustments in the practices.

3.2 The case survey method

The case survey steps as executed in this study are outlined in Sections 3.2.1 to 3.2.4.

3.2.1 Step 1: Select the cases of interest

The focus of this chapter is on choosing CSOs, such as choosing between in-house development versus open source for software-intensive systems. The components chosen should be utilized as parts of the system-intensive system developed. For example, if an automotive component is developed, and the choice is where to obtain the integrated development environment (IDE), this case would not be included in the study. On the other hand, if the focus of the software development is the IDE itself, then this case would be included. As pointed out by van Heesch et al. [72] architectural decisions are often not independent and thus are bundled into a single decision problem. This was also the case in this case survey.

The inclusion criteria thus can be summarised as follows:

- The case provides information of how the decision making between at least two CSOs has been taking place where the component should become part of a software-intensive system. For example, a database component becomes part of the system, while the development environment does not.
- The system for which the CSO decision is made is industrial (can involve academics if they are supporting the industry).
- Cases should at least be explicit about the CSOs considered, the persons involved in the decision making process, the CSO chosen, the methods used in decision making, and the criteria used when preparing and making the decision.
- Cases were elicited from two sources, namely researchers with industry experience of the ORION project reporting cases from industrial systems where they have been involved in the decision, and interviews with industry practitioners outside of the project.
The target population are cases of making decisions for CSOs for software-intensive systems. All cases are based on decisions for industrial systems. The sampling strategy was convenience sampling, i.e. we reported cases that we could access through the ORION project and industrial contacts. In order to obtain the data, two approaches were used, namely members of the ORION project, and interviews with industry practitioners outside of the project (see Table 32). All cases focus on decisions for industrial systems. Own experiences refer to researchers who in the role of industry practitioners have been involved in CSO decision making (11 cases). In addition interviews were conducted with industry practitioners (11 cases).

<table>
<thead>
<tr>
<th>Source</th>
<th>Number of cases</th>
<th>Case IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cases reported by ORION researchers</td>
<td>11</td>
<td>1, 2, 3, 4, 5, 6, 7, 8, 9, 16, 19</td>
</tr>
<tr>
<td>Interviews with industry stakeholders (outside ORION)</td>
<td>11</td>
<td>10, 11, 12, 13, 14, 15, 17, 18, 20, 21, 22</td>
</tr>
</tbody>
</table>

Members of the ORION project reported the cases using the data extraction scheme (Table 34). As the researchers had access to different networks a diverse sample could be obtained focusing on different domains and including experience from industrial and academic environments. The experiences of the research participants originate from knowledge obtained about relevant cases during former industrial projects where the subjects provided support as researchers (3 cases), the work done in an own company or work (3 cases), as well as consultancy work (5 cases). The interviewees were identified through the researchers’ networks. All decision cases presented were done for real-world industrial systems. The introduced data extraction scheme (Section 3.2.2) served as the interview guide. The interviews lasted for 45 minutes to 1 hour.

Table 33 provides the years of industry experience of the subjects in this study. The table shows the experience of the subjects that were the sources of the cases. We state the average, median, minimum, and maximum experience in years. Overall, the subjects have substantial experience in the software industry.
Step 2: Design data extraction scheme for data elicitation

There is a risk that the participants in the study eliciting the cases misinterpret the items in Table 34. Thus, the data extraction scheme was reviewed by all ORION project participants. The quality of the form and its understandability were essential in the extraction process, and was important for the validity of the results. An initial form was designed by the first author of the study. The co-authors reviewed the form and submitted change requests to the first author, each reviewer could see the comments already submitted. Each review was considered and a rejoinder was written to keep track of changes, including a response motivating the change, and an action specifying what was changed. After no further comments had been received, the form was considered of sufficient quality to start the data extraction.

The initial version of the data extraction form was defined based on the literature on decision making for CSOs (see related work), and ongoing work to create a taxonomy to describe the decision making for choosing among CSOs, the so-called GRADE taxonomy [135]. The taxonomy defines criteria, roles, decision making methods, environments, and decision making goals. We also explicitly asked the practitioners to expand on context and decision criteria relevant to the decision if they were not captured yet. Thus, all factors presented here were the ones that the practitioners were aware of, or raised as essential.
3.2.3 Step 3: Conduct the coding

Table 34 provides the extraction scheme that shows the category of extraction items, the name of the item, its description, value domain (data type) and links to research questions. Furthermore, we indicate where coding of the information was provided. The extraction scheme was used as the guide for interview. Not all information in the form needed to be coded, and we only coded information directly linked to the research questions. Many items were either integer values, enumerations, or boolean (present or not present in the case). That is, only items 13, 14, 15, 27, 28, 29, 33, and 34 in Table 34 were coded. These concern, among others, decision outcomes, lessons learned, or decision criteria not captured in the initial version of the extraction form. The initial coding was conducted by the first author. An open coding strategy was followed. For example, items 12 - 14 in the data extraction scheme (Table 34) referring to the stakeholders can be found in Tables 38 and 39. Also, through the coding a terminology control was performed so that we consistently refer to roles and quality attributes. For example, the subjects provided the criteria “API fit, compatibility with minimal adaptation” and “Very important that the solution is compatible with the other component in the system”, which were grouped under “Product - compatibility”. The coding was reviewed by Deepika Badampudi and Syed Muhammad Ali Shah. The coding was done based on notes taken during the interviews as well as the information filled in by the ORION members.
<table>
<thead>
<tr>
<th>Item ID</th>
<th>Category</th>
<th>Item</th>
<th>Description</th>
<th>Research question</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Meta-information</td>
<td>Code</td>
<td>Unique identifier for case</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>Meta-information</td>
<td>Author</td>
<td>ORION research participant providing the case</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>Meta-information</td>
<td>Company</td>
<td>Case company</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>Meta-information</td>
<td>Source</td>
<td>Origin of the case</td>
<td>X</td>
</tr>
<tr>
<td>5</td>
<td>Meta-information</td>
<td>Decision scenario</td>
<td>A short summary of the decision case</td>
<td>X</td>
</tr>
<tr>
<td>6</td>
<td>Context</td>
<td>Domain</td>
<td>Domain in which the decision was taken (e.g. automotive, avionics)</td>
<td>X</td>
</tr>
<tr>
<td>7</td>
<td>Context</td>
<td>Application type</td>
<td>Type of the application developed (e.g. embedded, information system)</td>
<td>X</td>
</tr>
<tr>
<td>8</td>
<td>Context</td>
<td>Company size</td>
<td>Size of the whole company</td>
<td>X</td>
</tr>
<tr>
<td>9</td>
<td>Context</td>
<td>Development unit size</td>
<td>Size of the development unit where the component was used</td>
<td>X</td>
</tr>
<tr>
<td>10</td>
<td>Context</td>
<td>Development methodology</td>
<td>Software Development Methodology used</td>
<td>X</td>
</tr>
<tr>
<td>11</td>
<td>Context</td>
<td>Other</td>
<td>Other relevant context factors to consider</td>
<td>X</td>
</tr>
<tr>
<td>12</td>
<td>CSOs</td>
<td>CSOs considered in the decision</td>
<td>CSOs = In-house, COTS, OSS, Outsource</td>
<td>RQ1.1</td>
</tr>
<tr>
<td>13</td>
<td>Stakeholders</td>
<td>Decision initiator</td>
<td>Stakeholders involved in the initiation of the decision (identification of the need to make a decision and formulation of the decision problem)</td>
<td>RQ1.2</td>
</tr>
<tr>
<td>14</td>
<td>Stakeholders</td>
<td>Stakeholders in decision prepartion</td>
<td>Stakeholders preparing the information and reflections needed to make a decision</td>
<td>RQ1.2</td>
</tr>
<tr>
<td>15</td>
<td>Stakeholders</td>
<td>Decision makers</td>
<td>Stakeholders taking the decision</td>
<td>RQ1.2</td>
</tr>
<tr>
<td>16</td>
<td>Decision criteria</td>
<td>Performance</td>
<td>Response time, timing behaviour of the system</td>
<td>RQ1.3</td>
</tr>
<tr>
<td>17</td>
<td>Decision criteria</td>
<td>Maintainability</td>
<td>Ease of updating the system (corrective, enhancements)</td>
<td>RQ1.3</td>
</tr>
<tr>
<td>18</td>
<td>Decision criteria</td>
<td>Reliability</td>
<td>Reliability of the system</td>
<td>RQ1.3</td>
</tr>
<tr>
<td>19</td>
<td>Decision criteria</td>
<td>Security</td>
<td>Security of the system</td>
<td>RQ1.3</td>
</tr>
<tr>
<td>20</td>
<td>Decision criteria</td>
<td>Time</td>
<td>Time (duration) to develop the system</td>
<td>RQ1.3</td>
</tr>
<tr>
<td>21</td>
<td>Decision criteria</td>
<td>Cost</td>
<td>Cost to develop the system</td>
<td>RQ1.3</td>
</tr>
</tbody>
</table>

Table 34: Extraction scheme
<table>
<thead>
<tr>
<th></th>
<th>Decision criteria</th>
<th>Information about the market (mass market or bespoke development)</th>
<th>RQ1.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>Market and contract</td>
<td>Information about the market (mass market or bespoke development)</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Access and control</td>
<td>Ability to access the code and control the evolution of the component</td>
<td>RQ1.3</td>
</tr>
<tr>
<td>24</td>
<td>Component usage in the system</td>
<td>Ease of use when using the component in the system</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Component history</td>
<td>Evolution of the component in terms of maturity and change history</td>
<td>RQ1.3</td>
</tr>
<tr>
<td>26</td>
<td>Certification</td>
<td>Certification of the component, certification of the company offering a component</td>
<td>RQ1.3</td>
</tr>
<tr>
<td>27</td>
<td>Other</td>
<td>Other criteria not mentioned</td>
<td>RQ1.3</td>
</tr>
<tr>
<td>28</td>
<td>Decision model</td>
<td>Method used to make the decision</td>
<td>RQ1.4</td>
</tr>
<tr>
<td>29</td>
<td>Property model</td>
<td>Method used to estimate the impact of the decision</td>
<td>RQ1.4</td>
</tr>
<tr>
<td>30</td>
<td>Decision outcome</td>
<td>CSOs chosen</td>
<td>RQ1.1</td>
</tr>
<tr>
<td>31</td>
<td>Decision preparation effort</td>
<td>Effort in preparing the decision (estimated)</td>
<td>RQ2.2</td>
</tr>
<tr>
<td>32</td>
<td>Decision making effort</td>
<td>Effort in making the decision (estimated)</td>
<td>RQ2.2</td>
</tr>
<tr>
<td>33</td>
<td>Evaluation of the decision and the decision impact</td>
<td>Important criteria of the decision and reflections on the success/failure</td>
<td>RQ2.3/ RQ2.4</td>
</tr>
<tr>
<td>34</td>
<td>Noteworthy comments</td>
<td>Remarks considered important by the person extracting the case</td>
<td>X</td>
</tr>
</tbody>
</table>
Step 4: Analysis

Yin and Heald [195] report the results of the case survey in terms of vote counting. A similar approach is utilized in this case survey (e.g. the number of cases considering a CSO, the number of cases considering different criteria, etc.).

Odds ratio is used as a statistical analysis method to quantify how strongly the presence or absence of property A is associated with the presence or absence of property B in a given population. In this case, the association between the presence or absence of a decision criterion and the presence or absence of a decision option is measured through odds ratio (see RQs 1.3 and 2.3). To compute odds ratio we determine the following variables:

- \(a\) = Number of cases where the criterion is considered and the decision option is chosen
- \(b\) = Number of cases where the criterion is considered and the decision option is not chosen
- \(c\) = Number of cases where the criterion is not considered and the decision option is chosen
- \(d\) = Number of cases where the criterion is not considered and the decision option is not chosen

The OR is thereafter calculated as:

\[
OR = \frac{a/c}{b/d} = \frac{a \times d}{b \times c}
\]  

The confidence interval (cf. [82]) for OR values is calculated as:

\[
\text{Upper}95\%\text{CI} = e^{\ln(\text{OR}) + 1.96\sqrt{1/a + 1/b + 1/c + 1/d}}
\]  

\[
\text{Lower}95\%\text{CI} = e^{\ln(\text{OR}) - 1.96\sqrt{1/a + 1/b + 1/c + 1/d}}
\]

The OR values are interpreted as follows:

- OR = 1 The criterion does not affect odds of decision option being chosen.
- OR > 1 indicates that the criterion is associated with higher odds of the decision option being chosen.
- OR < 1 indicates that the criterion is associated with lower odds of the decision option being chosen.
The values of confidence interval are considered to determine if the results are statistically significant. If the range of confidence intervals include the value 1 then the result is determined as not statistically significant, which is the case for the included cases (see Table 42). Power analysis is a statistical test to determine the sample size needed to detect an effect with a given degree of confidence. It is based on statistical assumptions and data characteristics. In particular, the characteristic that we would have to know is the relative precision implying skewness of the distribution of the odds ratio value in order to do a power analysis. However, we do not have historical data to specify the desired value with confidence given that our study is of exploratory nature. Ayala et al. [9] also highlighted this, in particular sourcing/ component decision practices in industry are not widely investigated. Hence, at the exploratory stage we do not know these characteristics to the degree to make a reliable power analysis. Miller [124] points towards qualitative direction as an alternative to statistical significance testing. Thus, in our results we utilize the statistical analysis from odds ratio to triangulate the findings with the responses from the subjects of this study (see Section 4.2.3).

4 RESULTS

4.1 Overview of cases

Table 35 provides an overview of the 22 cases included in the case survey. The table shows contextual information about the cases. The size refers to the overall (including different development organizations), while the size of the development unit refers to the organization where the selection of the component took place. The development units were located in Sweden. Furthermore, we characterized the domain, the application type, and the development methodology used in the development unit. The cases were characterized by the size of the company, the development unit where the component should be used. Furthermore, the domain, application type, and development methodology were specified. Half of the cases were in the context of the automotive domain (11 of 22), while other contexts have been considered as well. Given the proportionally high number of automotive cases, the most frequently reported application type was embedded systems. A variety of software development methodologies has been used, including agile and plan-driven processes. Furthermore, multiple cases reported hybrid processes combining agile and plan-driven concepts.
Table 35: Overview of the cases

<table>
<thead>
<tr>
<th>Case ID</th>
<th>Company Size</th>
<th>Domain</th>
<th>Application type</th>
<th>Development methodology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>100,000</td>
<td>Automotive</td>
<td>Embedded systems</td>
<td>Iterative development, Lean manufacturing</td>
</tr>
<tr>
<td>Case 2</td>
<td>1.200</td>
<td>Utilities</td>
<td>Embedded + Software + Apps</td>
<td>Agile SCRUM variant</td>
</tr>
<tr>
<td>Case 3</td>
<td>40</td>
<td>Human resource management</td>
<td>Software</td>
<td>Hybrid Plan driven, Agile</td>
</tr>
<tr>
<td>Case 4</td>
<td>21</td>
<td>Trade, Security, Consumer</td>
<td>Software, Hardware, apps</td>
<td>Hybrid Plan driven, Agile</td>
</tr>
<tr>
<td>Case 5</td>
<td>500</td>
<td>Financial</td>
<td>Software + Hardware (servers)</td>
<td>Hybrid Plan driven, Agile</td>
</tr>
<tr>
<td>Case 6</td>
<td>17,000</td>
<td>Surveillance/Security</td>
<td>Software + Hardware</td>
<td>Plan driven, iterative</td>
</tr>
<tr>
<td>Case 7</td>
<td>NA</td>
<td>Telecommunication</td>
<td>Embedded system</td>
<td>NA</td>
</tr>
<tr>
<td>Case 8</td>
<td>NA</td>
<td>Automotive</td>
<td>Embedded control systems</td>
<td>Iterative, informal process</td>
</tr>
<tr>
<td>Case 9</td>
<td>100,000</td>
<td>Automotive</td>
<td>Interactive tool for calibration</td>
<td>N/A</td>
</tr>
<tr>
<td>Case 10</td>
<td>100,000</td>
<td>Automotive</td>
<td>Embedded software architecture</td>
<td>Iterative development, time-boxed deliveries, incremental</td>
</tr>
<tr>
<td>Case 11</td>
<td>100.00</td>
<td>Automotive</td>
<td>Embedded system architecture</td>
<td>Waterfall</td>
</tr>
<tr>
<td>Case 12</td>
<td>100.00</td>
<td>Automotive</td>
<td>Embedded system architecture</td>
<td>Waterfall</td>
</tr>
<tr>
<td>Case 13</td>
<td>20</td>
<td>Defense</td>
<td>Search engine</td>
<td>Agile</td>
</tr>
<tr>
<td>Case</td>
<td>Budget (in K)</td>
<td>Number of Employees</td>
<td>Industry/Segment</td>
<td>Development Methodology/Style</td>
</tr>
<tr>
<td>-------</td>
<td>---------------</td>
<td>---------------------</td>
<td>-----------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Case 14</td>
<td>20</td>
<td>3</td>
<td>Marketing &amp; Advertising Analytics tool</td>
<td>Waterfall</td>
</tr>
<tr>
<td>Case 15</td>
<td>14,000</td>
<td>180</td>
<td>Defense</td>
<td>Mission critical</td>
</tr>
<tr>
<td>Case 16</td>
<td>140,000</td>
<td>300</td>
<td>Process automation and robotics for several industries</td>
<td>Embedded control system</td>
</tr>
<tr>
<td>Case 17</td>
<td>100,000</td>
<td>30</td>
<td>Automotive</td>
<td>Embedded control system</td>
</tr>
<tr>
<td>Case 18</td>
<td>10,000</td>
<td>10</td>
<td>Automotive</td>
<td>Control-dominant software</td>
</tr>
<tr>
<td>Case 19</td>
<td>N.A. (large scale)</td>
<td>1,000</td>
<td>Telecom</td>
<td>Information system</td>
</tr>
<tr>
<td>Case 20</td>
<td>100,000</td>
<td>100</td>
<td>Telecom</td>
<td>Charging system</td>
</tr>
<tr>
<td>Case 21</td>
<td>100,000</td>
<td>5</td>
<td>Telecom</td>
<td>Embedded system</td>
</tr>
<tr>
<td>Case 22</td>
<td>3</td>
<td>3</td>
<td>Telecom</td>
<td>NA</td>
</tr>
</tbody>
</table>

119
4.2  *RQ1: How are CSOs chosen?*

In the context of *RQ1* we investigated the CSOs considered, the involved stakeholders, the decision criteria, and the decision model used.

4.2.1  *RQ1.1: Which CSOs (a) were considered and (b) which CSOs among those considered were chosen?*

The decision making problem constitutes the choice of CSOs for a software intensive system, namely in-house, COTS, OSS and Outsource. In addition the practitioners considered services as a separate option in their decision making. Looking at the definition of services they were considered in the category of COTS by Ayala et al. [9]. More specifically, they are defined as a way of delivering functionality: “*Software-intensive services, often delivered as cloud or internet services, can also be products from all industries like financial, insurance, gaming, social software, or personal services based on software*” (cf. [84]). In this chapter our aim was to present the decision making problem as it was formulated by the practitioners, hence services has been presented as a separate option. It is interesting to observe that what is conceptually incorrect (i.e. including services as a CSO) was not a factor for the practitioners to exclude them as an option in their decision making.

Table 36 shows the most frequently considered CSOs. As is evident from the table the most frequently considered CSOs were In-house, COTS and Outsource. This information is important to, for example, decide which options have to be well supported by evidence with regard to benefits and limitations of one alternative over another (e.g. COTS in comparison to In-house). Only in five cases the practitioners considered services as an option.

<table>
<thead>
<tr>
<th>CSO</th>
<th>Frequency of consideration</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-house</td>
<td>17</td>
<td>32.08</td>
</tr>
<tr>
<td>COTS</td>
<td>14</td>
<td>26.42</td>
</tr>
<tr>
<td>Outsource</td>
<td>11</td>
<td>20.75</td>
</tr>
<tr>
<td>OSS</td>
<td>6</td>
<td>11.32</td>
</tr>
<tr>
<td>Services</td>
<td>5</td>
<td>9.43</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>53</strong></td>
<td><strong>100.00</strong></td>
</tr>
</tbody>
</table>
To understand which CSOs are traded off against each other we analyzed the frequency of combinations for comparisons between the alternatives, which is shown in Table 37. The most frequent trade-offs were between In-house vs. COTS, In-house vs. Outsource, and COTS vs. OSS. It is no surprise that the most frequent comparisons comprise of the CSOs being most frequently considered (see Table 36).

The number of options considered has an implication on decision support methods, i.e. one needs to determine whether solutions can support a trade-off between two or more alternatives. The number of options considered in the cases are thus illustrated in Figure 24. The figure shows a histogram of the number of options considered for the set of cases. It shows that it is common to consider two options, while there are still a number of cases that involve three options and more.
Figure 25 shows the options considered for each case as well as the options chosen. The elements in the figure are to be interpreted as follows:

- CSOs that were considered, but not chosen, are represented as black circles. For example, in case 12, In-house has been considered, but it was not chosen.
- Decisions that deviated from the recommended choice based on the investigation during the decision preparation are represented as black diamonds. For example, in Case 1 the recommendation was to go with COTS based on the decision preparation, but then the choice made was Outsource, which was also a CSO considered from the beginning in the decision making process.
- CSOs that were considered, chosen, and did not deviate from the recommended choice, are illustrated as circles with a white diamond inside.

In 7 of 22 cases in-house development has been chosen among the alternatives. Also in 7 cases components off-the-shelf (COTS) was the CSO chosen. In-house development has been considered in 17 out of 22 cases. Thus, the reflection of developing internally, or obtaining a component externally from different sources seems to be a key decision in practice. In a few cases only, the recommended alternative has not been chosen, which was true for Cases 1 and 14.

For case 1, the decision was prepared on a technical level considering the wide range of factors (performance, reliability, etc.). The technical people in preparation then passed their recommendation to the decision maker on business level, who decided to ignore the findings and make a political decision. In conclusion, for this case politics and the management of relationships overruled technical considerations.

For case 14, a social media analytics platform required a component for sentiment analysis from Facebook and Twitter. In-house development was recommended and a similar library was already developed that could have been used, but a web-service has been obtained also. In the end, quality issues were observed for the web service (textAnalytics) while also an alternative service (Mashape) was considered. Overall, this led the company to initiate a new decision, namely replace the textAnalytics service with Mashape, or to utilize the in-house developed component.

As mentioned earlier decisions are often combined [72]. In the case survey, the cases where more than one CSO (see Figure 25) has been chosen represent combined decisions. For example, in Case 5 a system
for access control, back-end trade, and reporting for a large financial system was developed with high demands on accuracy and performance. This required three components, which were all handled as
Table 38: Decision making roles involved in the decision

<table>
<thead>
<tr>
<th>Roles</th>
<th>Initiation Abs</th>
<th>Initiation %</th>
<th>Preparation Abs</th>
<th>Preparation %</th>
<th>Deciding Abs</th>
<th>Deciding %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software management</td>
<td>15</td>
<td>62.5</td>
<td>32</td>
<td>43.24</td>
<td>21</td>
<td>80.77</td>
</tr>
<tr>
<td>Software construction/dev.</td>
<td>7</td>
<td>29.17</td>
<td>9</td>
<td>12.16</td>
<td>4</td>
<td>15.38</td>
</tr>
<tr>
<td>External support</td>
<td>1</td>
<td>4.167</td>
<td>8</td>
<td>10.81</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>Software test/quality control</td>
<td>1</td>
<td>4.167</td>
<td>2</td>
<td>2.70</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>Customers</td>
<td>0</td>
<td>0.00</td>
<td>3</td>
<td>4.05</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>Expert group (group of roles, unspecified)</td>
<td>0</td>
<td>0.00</td>
<td>5</td>
<td>6.75</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>Legal</td>
<td>0</td>
<td>0.00</td>
<td>2</td>
<td>2.70</td>
<td>1</td>
<td>3.84</td>
</tr>
<tr>
<td>Sales (business/customer relations)</td>
<td>0</td>
<td>0.00</td>
<td>4</td>
<td>5.40</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>Software architecture/design</td>
<td>0</td>
<td>0.00</td>
<td>6</td>
<td>8.10</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>Sub-contractors (providers of components)</td>
<td>0</td>
<td>0.00</td>
<td>3</td>
<td>4.0554</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>Total</td>
<td>24</td>
<td>100.00</td>
<td>74</td>
<td>100.00</td>
<td>26</td>
<td>100.00</td>
</tr>
</tbody>
</table>

Table 39: Management roles involved in the decision

<table>
<thead>
<tr>
<th>Roles</th>
<th>Initiation Abs</th>
<th>Initiation %</th>
<th>Preparation Abs</th>
<th>Preparation %</th>
<th>Deciding Abs</th>
<th>Deciding %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executive management (CEO/CTO)</td>
<td>6</td>
<td>30.00</td>
<td>11</td>
<td>31.43</td>
<td>8</td>
<td>32.00</td>
</tr>
<tr>
<td>Management (type unspecified)</td>
<td>7</td>
<td>35.00</td>
<td>9</td>
<td>25.71</td>
<td>9</td>
<td>36.00</td>
</tr>
<tr>
<td>Product management</td>
<td>1</td>
<td>5.00</td>
<td>7</td>
<td>20.00</td>
<td>2</td>
<td>8.00</td>
</tr>
<tr>
<td>Project management</td>
<td>4</td>
<td>20.00</td>
<td>7</td>
<td>20.00</td>
<td>4</td>
<td>16.00</td>
</tr>
<tr>
<td>Line management</td>
<td>2</td>
<td>10.00</td>
<td>1</td>
<td>2.86</td>
<td>2</td>
<td>8.00</td>
</tr>
<tr>
<td>Total</td>
<td>20</td>
<td>100.00</td>
<td>35</td>
<td>100.00</td>
<td>25</td>
<td>100.00</td>
</tr>
</tbody>
</table>

one decision problem to solve account, trading, and reporting. When we consider this as a decision problem, the company considered build or service for the first component, build or outsource for the second component, and build or get a COTS for the third component.
4.2.2 RQ1.2: Which stakeholders were involved in the decision process?

We distinguish three groups of stakeholders in the decision making. According to Strydom [170] the key stakeholders in decision making are:

1. **Decision initiation:** The decision initiator who is raising the need to make a decision to solve a specific problem (in this case the selection of CSOs).

2. **Decision preparation:** People in the decision preparation are concerned with the study of documentation, the presentation and production of internal reports to share the findings of investigations of CSOs, meetings in the form of workshops and seminars, as well as informal discussions. Furthermore, they document a rationale. Only a few cases (4 cases) the rationale for the decision was not documented, but rather stated in discussions. In the remaining cases the documentation took place in the form of reports.

3. **Decision makers:** The decision makers (taking the decision and thus “making the final choice between alternatives” (cf. [170])). The following ways of making a decision were found: A leader takes a decision and hence no consensus was required; The stakeholders agreed and hence no negotiation needed to take place and consensus was reached; A demonstrator/simulator was used to illustrate the effects of the solution to facilitate the decision.

Table 38 provides an overview of the roles involved in the decision making, grouped by initiation, preparation, and decision making. Table 39 shows the distribution for the management roles involved.

**Decision initiation:** With regard to decision initiators software management is the most frequent initiator for the decision making cases. Non-managerial roles have initiated the decision making process in three cases (software architecture and design/construction).

In three cases (16, 17 and 19) multiple roles initiated the decision. For software management the roles could be further refined. There it is noteworthy that in six cases executive management (CTO/CEO) have initiated the decision making. Other roles initiating the decision making were project leaders/manager, line manager, and product manager. When multiple roles were involved, it was a combination of management and technical roles.

**Decision preparation:** Compared with the decision initiation, more distinct roles were involved as stakeholders in the process of support-
ing the decision with discussion and expert input. Table 38 shows that the most common roles in decision preparation were software management and software architecture and design/construction. It was evident that expert decision support was obtained in twelve cases from either researchers or consultants. Additionally, customer relations/sales and software architects and architects/designers were involved frequently. Only in a few cases software test (Cases 6 and 7), the actual customers (Cases 2, 4 and 9) and sub-contractors (Cases 5, 9 and 16) were involved.

Figure 26: Number of roles involved in the decision initiation presented in the cases

Figure 27: Number of roles involved in the decision preparation presented in the cases
Figure 28: Number of roles involved in the decision-making presented in the cases

**Decision making:** The decision makers were primarily managers. In comparison, only a few decision makers were in the category of software construction. In two cases (4 and 8) a consensus decision between management and software design/construction was made.

**Number of roles per decision making process:** Figures 26, 27 and 28 shows the number of roles involved in the decision making process related to initiation, preparation, and decision making. Understanding the number of roles involved has important implications on how to support the decision making process. For example, as soon as multiple roles are involved there is a need to support consensus building and incorporating multiple points of view during the process. Only a few roles (primarily management) are involved in the initiation (one individual role in 19 cases) and making the decision, while a larger set of roles is involved in the preparation (in average three roles, see Figures 26, 27 and 28).

4.2.3 RQ1.3: Which criteria (a) were considered for making the decision and (b) which criteria initially considered ended up as significant for the final decision?

(a) **Criteria considered:** The criteria based on which the decision options are evaluated are shown in Table 40.
<table>
<thead>
<tr>
<th>Group</th>
<th>Criterion</th>
<th>Cases</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
<td>Product Performance</td>
<td>17</td>
<td>77.27</td>
</tr>
<tr>
<td></td>
<td>Reliability</td>
<td>13</td>
<td>59.09</td>
</tr>
<tr>
<td></td>
<td>Maintainability</td>
<td>13</td>
<td>59.09</td>
</tr>
<tr>
<td></td>
<td>Certification</td>
<td>12</td>
<td>54.55</td>
</tr>
<tr>
<td></td>
<td>Level of openness and access/controld</td>
<td>11</td>
<td>50.00</td>
</tr>
<tr>
<td></td>
<td>Security</td>
<td>9</td>
<td>4.91</td>
</tr>
<tr>
<td></td>
<td>Compatibility</td>
<td>9</td>
<td>4.91</td>
</tr>
<tr>
<td></td>
<td>Functionality</td>
<td>7</td>
<td>31.82</td>
</tr>
<tr>
<td></td>
<td>Architectural dependencies</td>
<td>4</td>
<td>18.18</td>
</tr>
<tr>
<td></td>
<td>Compliance to standards and regulations</td>
<td>3</td>
<td>13.64</td>
</tr>
<tr>
<td></td>
<td>Licensing rules</td>
<td>3</td>
<td>13.64</td>
</tr>
<tr>
<td></td>
<td>Portability</td>
<td>2</td>
<td>9.09</td>
</tr>
<tr>
<td></td>
<td>Quality (general)</td>
<td>3</td>
<td>13.64</td>
</tr>
<tr>
<td></td>
<td>Safety</td>
<td>1</td>
<td>4.55</td>
</tr>
<tr>
<td></td>
<td>Stability</td>
<td>2</td>
<td>9.09</td>
</tr>
<tr>
<td></td>
<td>Ease of integration</td>
<td>1</td>
<td>4.55</td>
</tr>
<tr>
<td></td>
<td>Extendability</td>
<td>2</td>
<td>9.09</td>
</tr>
<tr>
<td></td>
<td>Longevity of asset</td>
<td>2</td>
<td>9.09</td>
</tr>
<tr>
<td></td>
<td>Scalability</td>
<td>2</td>
<td>9.09</td>
</tr>
<tr>
<td></td>
<td>User experience</td>
<td>1</td>
<td>4.55</td>
</tr>
<tr>
<td></td>
<td>Availability</td>
<td>1</td>
<td>4.55</td>
</tr>
<tr>
<td></td>
<td>Fitness for purpose</td>
<td>1</td>
<td>4.55</td>
</tr>
<tr>
<td></td>
<td>Number of users</td>
<td>1</td>
<td>4.55</td>
</tr>
<tr>
<td></td>
<td>Robustness</td>
<td>1</td>
<td>4.55</td>
</tr>
<tr>
<td>Financial</td>
<td>Cost - general (time, effort, resources)</td>
<td>17</td>
<td>86.36</td>
</tr>
<tr>
<td></td>
<td>Cost - buy/rent</td>
<td>3</td>
<td>1.62</td>
</tr>
<tr>
<td></td>
<td>Cost - acquisition</td>
<td>1</td>
<td>4.55</td>
</tr>
<tr>
<td></td>
<td>Cost - Adaptation</td>
<td>1</td>
<td>4.55</td>
</tr>
</tbody>
</table>
Cost - Licensing 1 4.55
Cost - risks incurred 1 4.55
Cost - total cost of ownership 1 4.55
Cost - maintenance 1 4.55

<table>
<thead>
<tr>
<th>Project</th>
<th>Level of support</th>
<th>5 22.73</th>
</tr>
</thead>
<tbody>
<tr>
<td>Familiarity with technology</td>
<td>1 4.55</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Business</th>
<th>Ecosystems</th>
<th>1 4.55</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Market trend</td>
<td>1 4.55</td>
</tr>
<tr>
<td></td>
<td>time to market</td>
<td>1 4.55</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>158</td>
</tr>
</tbody>
</table>

The table indicates the number of cases that have considered the criteria and also the percentage of the total number of cases that consider the criteria. Frequently considered criteria related to the product are quality criteria, such as performance, reliability, maintainability, compatibility and security. Further product-related characteristics were considered frequently — the most frequently mentioned ones being certification, level of openness and access/control. Furthermore, cost was an important criterion. A sub-set of cases further specifies the type of cost (e.g., to buy/rent, licensing, etc.).

As seen in Table 40, some of the criteria are considered more frequently. We investigated the criteria that are considered together among the most frequently considered criteria (frequencies greater than 10) in relation to the total number of times a criterion is considered as shown in Figure 29. The percentages are calculated by dividing the number of times the criterion is considered together with another criterion by the total number of times a criterion is considered. For example, performance is considered 17 times in total and out of the 17 times, it is considered 13 times together with reliability. Therefore the percentage of performance and reliability considered together and

<table>
<thead>
<tr>
<th>Performance</th>
<th>Reliability</th>
<th>Maintainability</th>
<th>Cost</th>
<th>Certification</th>
<th>Level of openness</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>76.47</td>
<td>58.82</td>
<td>70.59</td>
<td>58.82</td>
<td>58.82</td>
</tr>
<tr>
<td>100</td>
<td>X</td>
<td>61.54</td>
<td>92.31</td>
<td>69.23</td>
<td>46.15</td>
</tr>
<tr>
<td>76.92</td>
<td>61.54</td>
<td>X</td>
<td>93.31</td>
<td>53.85</td>
<td>46.15</td>
</tr>
<tr>
<td>73.68</td>
<td>63.16</td>
<td>63.16</td>
<td>X</td>
<td>57.89</td>
<td>47.57</td>
</tr>
<tr>
<td>91.67</td>
<td>83.33</td>
<td>58.33</td>
<td>91.67</td>
<td>X</td>
<td>50</td>
</tr>
<tr>
<td>90.91</td>
<td>81.82</td>
<td>54.55</td>
<td>63.64</td>
<td>54.55</td>
<td>X</td>
</tr>
</tbody>
</table>

Figure 29: Criteria trade-offs
in this order is \((13/17)\times100 = 76.47\%\). For reliability and performance the value is 100\% as shown in Figure 29. This means that every time the reliability is considered, it is always considered together with performance. Higher percentages indicate that possible trade-offs between the criteria might need to be considered in the decision. For example, improving reliability might be done under performance constraints.

The number of criteria considered are shown in Figure 30. The number of criteria taken into consideration impacts the requirements on the solution, e.g. with respect to optimization this means to choose an approach for multi-objective optimization. When conducting an analysis of trade-offs between criteria (i.e. how they positively or negatively influence each other) the complexity of the analysis increases.

(b) Criteria considered significant in the final decision: To assess which criterion has a significant impact on the final decisions two approaches are used. Firstly, we captured which criteria the subjects of the study mentioned as considered important in the final decision (Table 41). The criteria that were important in the final decision are divided into two groups: (1) Criteria that were considered in the preparation and are a sub-set chosen as important by the decision maker, (2) New criteria not considered in decision preparation, but in decision making. Secondly, we statistically explored the association between a criterion and a CSO in order to determine if the decision options are more favorable when
a particular criterion is considered. To evaluate the association we con-
sidered the odds ratio (OR) between the criterion and decision option
(Table 42). Using both statistical and qualitative approaches provides a
means for triangulation and qualitative reflection.

We first present reflections for the data as explained by the subjects
shown in Table 41. Several interesting observations can be made from
the table 41. In several cases only a small sub-set of criteria was consid-
ered in decision making compared to the preparation (see e.g. Cases
4, 5, 8, 17, and 19). That is, if the relevant decision criteria could have
been identified earlier, this has a potential to save investigation effort
in the preparation phase (such as pre-studies). An undesirable case is
where the input from the preparation is not considered in the final de-
cision, and new criteria become important. That is, the effort spent in
preparation is spent on investigating criteria that were not important
(see, for example, Cases 1, 6, 11, 18, and 19). For example, in Case 1 a
large car manufacturer performed a pre-study to investigate pros and
cons as a basis for opting between In-house, COTS, and OSS consider-
ing 15 criteria, namely: performance, reliability, cost (general), acquisition
cost, adaptation cost, maintenance cost, functionality, quality (general), fa-
miliarity with technology, ecosystem, architectural dependencies, longevity of
the component, extensibility, level of openness and access/control, compatibil-
ity. However, the best option with respect to the criteria considered in
preparation has not been chosen. Rather a political decision was made
to maintain a good relationship with a supplier company.
<table>
<thead>
<tr>
<th>Case</th>
<th>Criteria considered in preparation</th>
<th>Criteria considered important in decision making (subset of preparation)</th>
<th>Criteria considered important in decision making (new)</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-house</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 4</td>
<td>Performance, reliability, security, cost, user experience, compatibility, level of support, level of openness and access/control, stability, and certification</td>
<td>Security</td>
<td></td>
</tr>
<tr>
<td>Case 6</td>
<td>Performance, reliability, security, cost, user experience, compatibility, level of support, level of openness and access/control, stability, and certification</td>
<td>Traditions of developing in-house</td>
<td></td>
</tr>
<tr>
<td>Case 18</td>
<td>Performance, maintainability, reliability, cost, and certification</td>
<td>Competence</td>
<td></td>
</tr>
<tr>
<td>Case 21</td>
<td>Performance, reliability, and certification</td>
<td>Performance, reliability</td>
<td></td>
</tr>
<tr>
<td>COTS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 11</td>
<td>Performance, maintainability, reliability, security, cost, safety, robustness, compatibility, and certification</td>
<td>Cost</td>
<td>Ease to provide solutions to customers</td>
</tr>
<tr>
<td>Case 13</td>
<td>Performance, cost, level of support, market trend, compatibility, architectural dependencies, level of openness/access and control, component history, certification</td>
<td>Component history, market trend, performance, (most important), compatibility, cost, level of support, architectural dependencies, level of openness and access/control</td>
<td></td>
</tr>
<tr>
<td>Case 16</td>
<td>Maintainability, cost, fitness for purpose</td>
<td>Cost, fitness for purpose</td>
<td></td>
</tr>
<tr>
<td>Case 22</td>
<td>Cost, functionality</td>
<td>Cost, functionality</td>
<td></td>
</tr>
<tr>
<td>OSS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 3</td>
<td>Performance, reliability, security, cost, scalability, level of openness/access and control</td>
<td>Scalability, functionality</td>
<td></td>
</tr>
<tr>
<td>Case 7</td>
<td>Maintainability, cost</td>
<td>Maintainability, cost</td>
<td></td>
</tr>
<tr>
<td>Case 15</td>
<td>Performance, maintainability, reliability, security, cost, architectural dependencies, level of openness and access/control</td>
<td>Architectural dependencies, reliability</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Outsource</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 1</td>
<td>Performance, reliability, cost (general), acquisition cost, adaptation cost, maintenance cost, functionality, quality (general), familiarity with technology, ecosystem, architectural dependencies, longevity of the component, extendibility, level of openness and access/control, compatibility</td>
<td>Relationship with supplier company (maintain)</td>
<td></td>
</tr>
<tr>
<td>Case 12</td>
<td>Performance, maintainability, reliability, cost, functionality, compatibility, certification</td>
<td>Performance, cost, maintainability, functionality, compatibility, reliability</td>
<td></td>
</tr>
<tr>
<td>Case 19</td>
<td>Performance, maintenance cost, reliability, security, cost (general), quality (general), time to market, extendibility</td>
<td>Maintenance cost</td>
<td>Adaptability, level of openness and access/control</td>
</tr>
<tr>
<td>Case 20</td>
<td>Cost, certification</td>
<td>Cost</td>
<td></td>
</tr>
<tr>
<td><strong>Service</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 14</td>
<td>Performance, cost, functionality, availability, level of support, level of openness and access/control</td>
<td>Cost, functionality, performance, availability, level of support</td>
<td></td>
</tr>
<tr>
<td><strong>Combinations</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case 2</td>
<td>Performance, maintainability, reliability, security, cost, risks incurred, licensing rules, level of support, portability, compatibility, certification</td>
<td>N.A.</td>
<td></td>
</tr>
<tr>
<td>Case 5</td>
<td>Performance, reliability, security, cost, compliance to standards and regulations, cost for buying/renting, scalability, component history, certification</td>
<td>Performance, reliability</td>
<td></td>
</tr>
<tr>
<td>Case 8</td>
<td>Performance, portability, cost, licensing, functionality, portability, licensing rules, level of openness and access/control, ease of integration, component history</td>
<td>Level of openness and access/control, component history</td>
<td></td>
</tr>
<tr>
<td>Case 10</td>
<td>Performance, maintainability, reliability, security, cost, certification, level of openness and access/control, compatibility, stability, number of users</td>
<td>Cost, maintainability, reliability, certification, level of openness and access/control, performance, security</td>
<td></td>
</tr>
<tr>
<td>Case 17</td>
<td>Performance, maintainability, cost, quality (general), compliance to standards and regulations, functionality, level of openness and access/control, compatibility</td>
<td>Cost, functionality, compliance to standards and regulations</td>
<td></td>
</tr>
</tbody>
</table>
We now triangulate the findings of the statistical analysis (Table 42) with the information provided by the subjects (see Table 41).

Overall, in-house has higher odds ratio values. In-house seems to be a favorable decision option when all the frequently considered criteria (excluding general cost and maintainability) are considered. In particular, certification has the highest odds ratio (OR = 6.86) when in-house is chosen as the outcome. We can also see in Table 41, certification is considered in all the cases (Case 4, 6, 18 and 21) where in-house is chosen. Though, certification was considered in all cases, it was not the criterion that turned out to be the most significant. As some of the decisions were taken based on new criteria (tradition and competence) as in Table 41. However, since all the cases that had chosen in-house considered certification as a criterion and the higher values of odds ratio in Table 42 indicate that certification is important criterion for choosing in-house.

As seen from Table 42, performance (OR = 2.18) and reliability (OR = 2.19) also have the highest odds ratio values when in-house is chosen. This supports the qualitative data in Table 41 as performance and reliability is considered in all the cases where in-house is chosen and also turned out to be important criteria in one of the case (Case 21). Level of openness has higher odds ratio however, it was only considered in half of the cases when in-house was chosen.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>In-house</th>
<th>COTS</th>
<th>OSS</th>
<th>Services</th>
<th>Outsourcing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OR</td>
<td>Conf. Int.</td>
<td>OR</td>
<td>Conf. Int.</td>
<td>OR</td>
</tr>
<tr>
<td>Performance</td>
<td>2.18</td>
<td>0.70</td>
<td>0.82</td>
<td>0.17</td>
<td>1.17</td>
</tr>
<tr>
<td></td>
<td></td>
<td>24,21</td>
<td>6,34</td>
<td>14,52</td>
<td></td>
</tr>
<tr>
<td>Reliability</td>
<td>2.19</td>
<td>0.32</td>
<td>0.56</td>
<td>0.10</td>
<td>0.64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15,04</td>
<td>3,25</td>
<td>5,61</td>
<td></td>
</tr>
<tr>
<td>Maintainability</td>
<td>2.19</td>
<td>0.32</td>
<td>1,25</td>
<td>0.21</td>
<td>2,40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15,04</td>
<td>7,41</td>
<td>27,72</td>
<td></td>
</tr>
<tr>
<td>Cost</td>
<td>0.395</td>
<td>0.04</td>
<td>0.16</td>
<td>0.00</td>
<td>0.39</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3,52</td>
<td>0,00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Certification</td>
<td>6.86</td>
<td>0.66</td>
<td>1,25</td>
<td>0.21</td>
<td>0.17</td>
</tr>
<tr>
<td></td>
<td></td>
<td>71,72</td>
<td>7,41</td>
<td>1.96</td>
<td></td>
</tr>
<tr>
<td>Level of openness</td>
<td>3,75</td>
<td>0.54</td>
<td>0.45</td>
<td>0.08</td>
<td>3,75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>26,04</td>
<td>2,67</td>
<td>43,31</td>
<td></td>
</tr>
</tbody>
</table>
Maintainability (OR = 2.40) and level of openness (OR = 3.75) have highest odds ratio when OSS is chosen according to Table 42. This is also supported by qualitative data in Table 41 as both maintainability and level of openness is considered in two out of three cases where OSS was chosen. In addition, maintainability ended up being important criterion in one of the case (Case 7).

According to Table 42, cost is associated with lower odds of any CSOs being chosen as all values of odds ratio are below one. However, according to the qualitative data in Table 41, cost has been considered important in choosing COTS, outsourcing and services.

4.2.4 RQ1.4: Which decision making approach/model was used?

Table 43 shows the approaches used in decision making. Expert opinion/judgment has been utilized in all cases. Expert judgment was supported by a number of approaches. Prioritizing and ranking the alternatives with respect to weighted criteria (4 cases), listing the Pros and Cons (5 cases), and Pugh analysis (4 cases) were identified. More formal approaches for estimating (e.g. COCOMO) or the use of models for decision making (e.g. optimization) was not observed. It should be noted that we explicitly considered more structured decision making techniques such as the Analytical Hierarchical Process (AHP), elicitation of weights under decision model [56] (Item 28 in Table 34). However, they were not found in the cases as the decision making was discussion based and ad-hoc.

4.3 RQ2: What was the result of the decision making process?

The second research question focuses on the description of the decision outcomes while following the decision making processes characterized in Section 4.2.

Table 43: Approaches used for decision making (frequency)

<table>
<thead>
<tr>
<th>Decision making approaches used</th>
<th>Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert opinion / expert judgment</td>
<td>22</td>
</tr>
<tr>
<td>Pros and Cons</td>
<td>5</td>
</tr>
<tr>
<td>Prioritization, ranking, weighted criteria</td>
<td>4</td>
</tr>
<tr>
<td>Pugh analysis (Decision-matrix method)</td>
<td>4</td>
</tr>
<tr>
<td>Total</td>
<td>22</td>
</tr>
</tbody>
</table>
4.3.1 RQ2.1: *What was the effort invested in the decision making process?*

The effort spent on preparing the decision for the decision maker is shown in Figure 31, the x-axis shows effort spent in person hours. The intervals are illustrated in intervals of 500 person hours. The effort data was available in 16 of 22 cases. The average time spent on preparation was around 780 person hours.

In comparison Figure 32 shows the effort spent on decision making, but it is available for 16 cases, only. This indicates only a small fraction of the effort spent in preparation (on average 30 person hours) is spent on decision making.

4.3.2 RQ2.2: *Were the chosen CSOs considered the “right” choice retrospectively?*

Looking at how the decisions were evaluated it is visible that a high number of decisions were perceived as sub-optimal, which applied to a total of 9 cases (see Table 44). The table shows the outcome for those decision cases where a decision has been reached at the point of time of data collection. The case number is shown. The year of decision shows the year in which the decision was taken. The evaluation of the outcome was based on perception. We show the assessment of the subjects in terms of whether the decision was positive, they were indifferent, or negative. Furthermore, the rationales for the assessment provided by

![Figure 31: Effort in decision preparation](image-url)
the subjects have been stated. In only 7 cases the decision was perceived as clearly positive. This highlights the need for support in the decision making processes to improve the decision making outcomes.

Considering the decision year it is visible that in the cases the majority of cases at least two years are in-between the decision year and the year the data was collected (2016). Hence, this allowed for a reflection by the participants as there was sufficient time allowing for an assessment.

Table 44 also provides an insight on why one option was preferred over another. For example, it is visible that in-house was preferred over COTS and services for security reasons (Case 4), in-house over COTS and outsource for performance and stability reasons (Case 21), etc. We briefly present the rationale stated by the subjects for either evaluating the decision positively (see evaluations with the √-symbol in Table 44) or negatively (see evaluations with the †-symbol).

Positive (√): When choosing in-house the practitioners perceived the best decision was taken in comparison to outsourcing. Furthermore, another reason to assess the choice of in-house development over outsourcing was the improvement in the reliability of the product. When COTS has been chosen, the absence of issues were raised as the reason for the positive assessment when being compared to in-house development. OSS has been found to save costs, in Case 15 a factor of cost savings by the factor of 10 has been mentioned. When choosing a combination of options Cases 10 and 17 reported positive results.

![Figure 32: Effort in decision making](image-url)
Table 44: Decision outcomes and their evaluation

<table>
<thead>
<tr>
<th>Case</th>
<th>Decision year</th>
<th>Outcome of the decision</th>
<th>Evaluation of the final decision $\sqrt{\vdash}$ = positive, o = indifferent, † = negative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 4</td>
<td>2012</td>
<td>In-house over COTS and Services</td>
<td>†: <strong>Product - Security:</strong> Company needed to build themselves, as security requirements could not be met by other options, i.e. other options would have been preferred if possible</td>
</tr>
<tr>
<td>Case 6</td>
<td>2011</td>
<td>In-house over COTS and outsourcing</td>
<td>†: <strong>Decision making process:</strong> Huge effort invested in the investigation of vendors, and in the end it was done in-house anyways</td>
</tr>
<tr>
<td>Case 18</td>
<td>N.A.</td>
<td>In-house over outsourcing</td>
<td>√: <strong>General:</strong> Perception that the best decision was made (no rationale given)</td>
</tr>
<tr>
<td>Case 21</td>
<td>N.A.</td>
<td>In-house over outsourcing</td>
<td>√: <strong>Product - Reliability:</strong> Reliability was improved</td>
</tr>
<tr>
<td>Case 11</td>
<td>2006</td>
<td>COTS over in-house</td>
<td>√: <strong>Absence of issues:</strong> No issues later on, decision was considered a success</td>
</tr>
<tr>
<td>Case 13</td>
<td>2013</td>
<td>COTS over OSS</td>
<td>†: <strong>Product - Performance:</strong> Issues arose in terms of computing performance that was not seen in advance, a larger pre-study could have helped</td>
</tr>
<tr>
<td>Case 16</td>
<td>2009</td>
<td>COTS over in-house</td>
<td>o</td>
</tr>
<tr>
<td>Case 22</td>
<td>N.A.</td>
<td>COTS over in-house</td>
<td>o</td>
</tr>
<tr>
<td>Case 3</td>
<td>2004</td>
<td>OSS over COTS</td>
<td>√: <strong>Financial - Cost:</strong> Cost of the chosen solution was low</td>
</tr>
<tr>
<td>Case 7</td>
<td>2006</td>
<td>OSS over COTS</td>
<td>o</td>
</tr>
<tr>
<td>Case 15</td>
<td>2013</td>
<td>OSS over COTS and in-house</td>
<td>√: <strong>Financial - Cost:</strong> Very successful with regard to cost reduction by a factor of 10</td>
</tr>
<tr>
<td>Case 1</td>
<td>2008</td>
<td>Outsourcing over COTS and in-house</td>
<td>†: <strong>Product, financial, as well as project criteria:</strong> Sub-optimal solution with respect to criteria considered in preparation</td>
</tr>
<tr>
<td>Case 12</td>
<td>2005</td>
<td>Outsourcing over in-house</td>
<td>†: <strong>Product - Performance:</strong> Underestimated computing resources needed (performance) of the chosen solution</td>
</tr>
<tr>
<td>Case 19</td>
<td>2015</td>
<td>Outsourcing over in-house and services</td>
<td>o</td>
</tr>
<tr>
<td>Case 20</td>
<td>N.A.</td>
<td>Outsourcing over in-house</td>
<td>†: <strong>Product - Quality (general):</strong> Quality issues arose</td>
</tr>
<tr>
<td>Case 14</td>
<td>2012</td>
<td>Services over in-house</td>
<td>†: <strong>Product - Quality (general):</strong> Solution not as successful as hoped for, quality issues cost time to market, service needs to be replaced.</td>
</tr>
<tr>
<td>Case 2</td>
<td>2012</td>
<td>COTS and Services over OSS</td>
<td>o: <strong>Project - Level of Support, Financial - Cost:</strong> Trade-off between level of support (safer solution), but with a higher cost</td>
</tr>
<tr>
<td>Case 5</td>
<td>2014</td>
<td>COTS, Services and Outsourcing over In-house</td>
<td>†: <strong>Decision making process:</strong> Challenging to estimate and assess the impact and decision required long lead-time</td>
</tr>
<tr>
<td>Case 8</td>
<td>2014</td>
<td>COTS over OSS (first iteration) and OSS over COTS (second iteration)</td>
<td>†: <strong>Decision making process:</strong> Problems in details could not be foreseen (old version of programming language became an issue for chosen mature component)</td>
</tr>
<tr>
<td>Case 10</td>
<td>2007</td>
<td>In-house and COTS over an individual option</td>
<td>√: <strong>General:</strong> Perceived as the right decision</td>
</tr>
<tr>
<td>Case 17</td>
<td>2013</td>
<td>In-house and Outsourcing over an individual option</td>
<td>√: <strong>Project - Familiarity with technology:</strong> Perceived as positive decision with regard to a combination of in-house and outsourcing in terms of available competence and responsibility</td>
</tr>
</tbody>
</table>

**Negative (†):** For the cases choosing in-house development two reasons for negative assessments have been given, namely the decrease in product quality with respect to security and issues with the decision making process itself. When choosing COTS, in one case product qual-
antity issues arose with regard to performance. When outsourcing was chosen only negative assessments were found in the cases, all of them being related to product quality issues. For the choice of combinations of CSOs issues arose with regard to the lack of ability to foresee problems and to conduct estimations.

5 DISCUSSION

The discussion is structured along the research questions. In particular, the results of the systematic review [Chapter 2] and the case survey are compared as both studies had a similar focus.

5.1 Reflections with respect to the research questions

The first research question explored how CSOs are chosen in practice, four research questions were formulated and are discussed in the following. Three of the four questions are shared with the literature review by Badampudi et al. [Chapter 2], namely options considered (RQ1.1), criteria considered (RQ1.3), and decision making method (RQ1.4). Stakeholder roles were not explicitly discussed in the primary studies included by Badampudi et al., and hence are new results provided by the case survey in the context of choosing CSOs.

Options considered (RQ1.1): The contributions of existing literature and case survey are mapped as shown in Table 45. The table shows the CSOs considered, the number of studies, as well as the number of cases. Comparisons refer to studies investigating the properties of the CSOs in comparison to each other. Solutions refer to proposed approaches to decide between CSOs. The number of cases show how many times the CSOs were considered in comparison to each other in our case survey. The primary studies include empirical comparisons between CSOs and

Table 45: Mapping of existing literature and case study contributions with respect to CSOs

<table>
<thead>
<tr>
<th>CSOs considered</th>
<th>Number of primary studies</th>
<th>Number of cases</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Comparisons</td>
<td>Solutions</td>
</tr>
<tr>
<td>COTS vs. OSS</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>In-house vs. COTS</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>In-house vs. Outsource</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>In-house vs. OSS</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>COTS vs. Outsource</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OSS vs. Outsource</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
solution proposals of how to choose among them. Even though there are comparisons, no decision making solutions assist the COTS vs. OSS decision. The case survey reported six cases that consider COTS and OSS in the decision.

While there are no primary studies comparing in-house and outsourcing, two papers propose decision making solutions for the selection of CSOs. Overall the number of primary studies (two papers) discussing in-house and outsource is low despite being the most frequent decision as revealed by our case survey (11 of 22 cases).

In-house and COTS have also been compared in primary studies identified in our systematic literature review [Chapter 2], and decision making solutions are proposed to make in-house vs. COTS decisions. The frequencies of the primary study and case survey are consistent for the in-house vs. COTS options.

COTS vs. outsource and OSS vs. outsource are neither compared nor any decision making solutions are proposed. Therefore, based on the existing literature, it is unknown whether such decisions are even considered by decision makers. However, the case survey identify cases where COTS vs. outsource and OSS vs. outsource decisions are considered. There is thus a need for better support in practice.

The number of times options were chosen in comparison to the times they were considered indicates that all options are viable choices for the software-intensive systems considered. In-house has been selected in 41% of all cases it was considered (7/17), COTS in 50% (7/14), Outsource in 55% (6/11), OSS in 83% (5/6) and Services in 40% (2/5).

**Stakeholders (RQ1.2):** In most cases management roles initiated the decision, while the decision making preparation usually involved several roles from management, software construction and development, but also software design. Also, consultants were used to support the preparation of the decision, which indicates that additional competencies compared to those in the organization are needed to make the decision.

**Criteria (RQ1.3):** We first look at trade-offs between criteria, and compare them to the trade-offs discussed in our systematic literature review [Chapter 2].

**Trade-off 1:** Trade-offs between market-trend, technical support and maintainability are observed in the literature (cf. [Chapter 2]). Following market trend indicates frequent updates which involves additional maintenance effort. At the same time, the need for a high pace in releasing new features is required, which may result in technical debt as shortcuts may be taken. Also, if the component is not upgraded to the
latest available version, then the support offered from supplier/vendors might not be extended. Hence, a trade-off needs to be made between following market trends, maintaining the system stability and retaining the technical support. In the case survey 14 cases consider maintainability, 5 cases consider technical support, and 1 case considers market-trend. This means that maintainability, technical support, and market-trend are not considered together in the decisions.

Trade-off 2: Another trade-off identified in the literature is between source code availability, technical support, and license [Chapter 2]. The availability of source code might be a criteria for selecting a decision option so that the code can be changed. However, some licenses require changes in the code to be open. Also, technical support might not be extended for the modified code. Although source code availability (12) has high frequencies of cases, the frequencies of license (3) and technical support (5) are comparatively lower, which implies that the trade-off is not considered.

Trade-off 3: Development effort and integration effort trade-off is identified in the literature [Chapter 2]. Development effort can be saved if the development is not done in-house. However, if the saved development effort is less than the additional integration effort, then the decision is not optimal. No such trade-off is considered in any of the cases.

Overall the trade-offs observed in the cases and literature are not consistent, i.e. the existing studies do not support the processes observed in industry. This indicates a potential gap between industry practice and the focus of research with regard to sourcing decisions when looking at the trade-offs made. Overall, this merits the investigation of trade-off practices in the industry to support researchers in the selection of future research topics and questions with regard to trade-offs.

The need for prioritizing the criteria and identifying the important ones also became evident when analyzing the final decisions taken, and the criteria ending up as important among those considered. The decision problem may be simplified if criteria are identified and removed early. Much can be learned from the requirements community in that regard, in particular requirements prioritization techniques may be of value [15]. Barney et al.’s [11] study gives an overview of approaches for trade-offs between quality attributes, which may also be useful to not only trade-off between quality attributes, but by considering other factors as well.

Decision Making Methods (RQ1.4): The methods for decision making devoted to in-house vs. COTS and in-house vs. outsource exist in
the literature [Chapter 2], both trade-offs being frequently considered in the case survey. All the in-house vs. COTS decision making solutions proposed in the literature consider technical factors, notably time, cost, and reliability. These factors are easy to calculate; probably this is the reason why the methods have considered them. Most of the cases in this case survey considered cost and reliability in their decisions. Time has not been considered that often. However, the case survey identified many other criteria that are considered in the decisions (which also include non-technical criteria), which are not included in the methods proposed in literature.

The methods for choosing between in-house vs. outsource proposed in the literature consider requirement dependencies [Chapter 2]. However, this is not identified as a criterion in any of the cases in the case survey.

All the decision making methods proposed in literature specifically for CSOs are automated and mathematical, i.e. they are highly formalized. However, the cases indicate that the most popular techniques used in making decisions are expert based, which involves subjective opinions. This indicates that the methods proposed in the CSO literature are not consistent with the practice in industry. This also suggests that practitioners are looking for decision making methods that aid in decision-making and not the solutions that give the decision/outcome. However, the solutions proposed for GDM in the context of architectures seem more appropriate (see Section 2.2.1).

Further, according to the case survey, the management takes most decisions. The decision making methods proposed in the literature are quite complex. Due to the complexity and learning curve involved, the managers might not accept or use the solutions proposed in the literature.

**Effort invested (RQ2.1):** The effort invested in preparing the decision is quite significant, with the mean effort being 780 person hours, while in several cases the effort was over 1000 hours. In order to understand the factors we investigated the correlation between the effort invested and the complexity of the decision problem (number of criteria and number of options considered). In addition we calculated the correlation between the number of people involved in the preparation and the effort. Table 46 shows the results of the correlation (using the non-parametric method proposed by Spearman). The table shows the correlation with effort in relation to the people involved as well as the complexity of the decision problem. The complexity of the decision problem is indicated by two variables, the number of CSOs and the
number of criteria considered. Rubin [151] defines boundaries for the strength of correlations. A strong positive correlation is observed for the number of CSOs, and a moderate positive correlation for the number of decision criteria. A lower correlation was observed with regard to the number of people involved. Given that the number of CSOs as well as the number of criteria seem to be related to effort in preparation we suggest a staged process for choosing among options to not invest preparation effort on more obvious exclusions. Similar reflections were presented in the field of requirements engineering, where it was found that more complex decisions take more time [188]. As a consequence requirements triage has been introduced that removes the most obvious options first to avoid investigative effort [49]. Thus, similar ideas may be relevant for choosing among CSOs. Even though the correlation between effort and the number of people was lower in comparison to the other measures (see Table 46) communication overhead during the preparation should still be considered as factor besides the number of CSOs and decision criteria as the overhead increases with the number of people involved.

**Retrospective reflection on CSO choice (RQ2.2):** In seven cases only, the result of the decision was perceived as positive. In particular, if high investments have been made in preparing the decision, and the final decision is not perceived as successful, then the preparation effort can be considered wasted. The methods used for decision making were mostly experience based, and no decision support systems or methods have been used (such as estimations, existing evidence from research); thus the results indicate that there is an industrial challenge relevant for the research to address. In particular, decision support systems aiding the experts may be of interest to design for this particular decision problem. Early attempts have been made and preliminary results are available in that regard (cf. Wohlin et al. [193]). Furthermore, the related work on architectural decision making provides solutions to record rationales

---

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Spearman Correlation to Prep. Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of CSOs</td>
<td>0.559</td>
</tr>
<tr>
<td>Number of Decision Criteria</td>
<td>0.350</td>
</tr>
<tr>
<td>Number of People in Preparation</td>
<td>0.075</td>
</tr>
</tbody>
</table>

Table 46: Correlation with effort
and drive reflections in architectural decision making in Section 2.2.1 (e.g. [119, 132, 197]).

5.2 Characterization of decisions

According to Larsson [107] case surveys focus on identifying patterns (similarities and differences) between cases. With regard to the research questions each individual question investigated a specific aspect of the decision making case. We used hierarchical cluster analysis, in particular clustering of binary data (presence and absence of variables or attributes in cases) for this purpose. The method to calculate the clusters was Squared Euclidean Distance. The clustering is used as a reflective tool. It takes the values obtained across research questions into account to determine the similarity and differences between the cases. We only included variables related to the decision case (CSOs, stakeholders, and decision criteria) to find commonalities of how decision making has been made. Figure 33 shows the Dendrogram. The Dendrogram shows the similarity and dissimilarity between cases to explore whether interesting or shared patterns emerged in the data. The more similar two cases, the closer the vertical lines are to the left-hand side of the figure. For example, cases 20 and 22 are more similar than cases 8 and 13. It is evident that four cases are closely related, namely 18, 20, 21 and 22. Furthermore, cases 10 and 11 are very closely related. Overall, the Dendrogram shows that no clear patterns could be obtained as the distances between clusters were high. Using two-step clustering as an alternative approach showed that the shapes of clusters are not clearly identifiable, which is a sign that, besides the groups above, no clear patterns across cases are observable.
Figure 33: Hierarchical clustering of cases
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Case 18</th>
<th>Case 20</th>
<th>Case 21</th>
<th>Case 22</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Context</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Domain</td>
<td>Automotive</td>
<td>Telecom</td>
<td>Telecom</td>
<td>Telecom</td>
</tr>
<tr>
<td>Company size</td>
<td>10,000</td>
<td>100,000</td>
<td>100,000</td>
<td>3</td>
</tr>
<tr>
<td>Development unit size</td>
<td>10</td>
<td>100</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Development method</td>
<td>Agile</td>
<td>Agile (Scrum)</td>
<td>Agile (Scrum)</td>
<td>N.A.</td>
</tr>
<tr>
<td><strong>Decision case characteristics</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CSOs considered</td>
<td>In-house, Outsource</td>
<td>In-house, Outsource</td>
<td>In-house, Outsource</td>
<td>In-house, COTS</td>
</tr>
<tr>
<td>Stakeholders (initiation)</td>
<td>Software construction</td>
<td>Management</td>
<td>Management</td>
<td>Management</td>
</tr>
<tr>
<td>Stakeholders (preparation)</td>
<td>Management, Expert Group</td>
<td>Management</td>
<td>Management, Sales</td>
<td>Management</td>
</tr>
<tr>
<td>Stakeholders (decision making)</td>
<td>Management</td>
<td>Management</td>
<td>Management</td>
<td>Management</td>
</tr>
<tr>
<td>Criteria considered</td>
<td>Performance, maintainability, reliability, cost, and certification</td>
<td>Cost, certification</td>
<td>Performance, reliability, and certification</td>
<td>Functionality, Cost</td>
</tr>
<tr>
<td>Method</td>
<td>Expert judgment</td>
<td>Expert judgment</td>
<td>Expert judgment</td>
<td>Expert judgment</td>
</tr>
<tr>
<td><strong>Outcome</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decision impact</td>
<td>Positive</td>
<td>Negative</td>
<td>Positive</td>
<td>Indifferent</td>
</tr>
<tr>
<td>Effort</td>
<td>2400</td>
<td>640</td>
<td>480</td>
<td>N.A.</td>
</tr>
</tbody>
</table>
It has already been established that the cases share decision making characteristics, and a number of cases (more than two) is concerned, hence it is of interest to take a closer look at them. Hence, their contexts and outcomes are of interest to compare, which may explain why they are in the same cluster. The cluster of cases 18, 20, 21, and 22 is interesting because three of the four cases were perceived as positive. Despite of being in different domains (cases 18 and 22 are in the automotive domain and cases 20 and 21 are in the telecom domain) both of them are similar. The cases characteristics, the context, and the outcome for the four cases in the cluster are briefly summarized in Table 47.

5.3 Comparison with the general traits of decision making from related work

In the earlier sections we compared the findings of the case survey with the literature specific to each research question. The more general characteristics of decision making for architecture, CSO, and OTS decision making are presented in this section. We summarized eight characteristics of decision making in Table 48. The table shows the main findings for the general characteristics of decision making for architecture, CSO decision making and OTS decision making. It also indicates whether the finding is supported by the case survey, distinguishing between fully, partially, and not supported. A reflection for the degree of support is also stated. The table shows that two findings are not well supported, in particular L3, L6, L10 and L11. These show potential ways of improving industrial decision making (e.g. more explicitly considering risks and the experience and familiarity with the technology), while also may influence research solutions (e.g. providing solutions for CSO decision making that facilitate non-deterministic decisions). Solutions are present for architecture decision making and may be tailored to suit CSO decision making (see Section 2.2.1).
<table>
<thead>
<tr>
<th>ID</th>
<th>Related work findings</th>
<th>Supported by case survey?</th>
<th>Reflections</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Architecture decision making</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L1</td>
<td>Architectural decisions types are structural, property and executive decisions.</td>
<td>Yes</td>
<td>In the case survey structural decisions were taken</td>
</tr>
<tr>
<td>L2</td>
<td>Architecture decisions are made in a group</td>
<td>Yes</td>
<td>Prepared in the group, reflections and rationales identified, but the actual decisions (final word) is made by individuals (mostly management)</td>
</tr>
<tr>
<td>L3</td>
<td>Tool support for group decision making, and the possibility to review decisions after they have been made</td>
<td>No</td>
<td>Not present and thus a source of improvement. Could have facilitated to collect more information in the case survey in comparison of what could be obtained</td>
</tr>
<tr>
<td>L4</td>
<td>Architecture decisions are not taken in isolation</td>
<td>Yes</td>
<td>Bundled related decisions using the same criteria and treated them as a single decision problem</td>
</tr>
<tr>
<td>L5</td>
<td>Decision making is non-deterministic and argumentative</td>
<td>Yes</td>
<td>No prescribed / systematic decision making method used, rather discussion</td>
</tr>
<tr>
<td></td>
<td><strong>CSO decision making</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L6</td>
<td>Optimization solutions for CSO selection have been proposed assuming deterministic decision making</td>
<td>No</td>
<td>Decisions were based on discussions and without structured approaches (i.e. ad-hoc)</td>
</tr>
<tr>
<td></td>
<td><strong>OTS decision making</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L7</td>
<td>The solutions in the literature do not match the practice in companies</td>
<td>Yes</td>
<td>Non-deterministic approaches for CSO selection were used in the cases, while CSO literature (L6) assumes deterministic decision making.</td>
</tr>
<tr>
<td>L8</td>
<td>Decision making approaches are not formal, but rather ad-hoc</td>
<td>Yes</td>
<td>No well structured decision making method used (the most structured was Pugh-analysis, while otherwise the listing of pros and cons and less structured discussions took place)</td>
</tr>
<tr>
<td>------</td>
<td>-------------------------------------------------------------</td>
<td>-----</td>
<td>----------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>L9</td>
<td>In OSS decision imitative is mostly taken by developers, while leaders (such as CEOs) have the final word</td>
<td>Partially</td>
<td>The initiation of the decision was mostly driven by management, the final word was mostly given by management (CEOs/CTOs)</td>
</tr>
<tr>
<td>L10</td>
<td>Risks are considered during the decision</td>
<td>No</td>
<td>Only the risk of cost increase was considered in one case</td>
</tr>
<tr>
<td>L11</td>
<td>Experience with the component is an important factor</td>
<td>No</td>
<td>Only in one case familiarity with the technology has been raised as an important criterion.</td>
</tr>
<tr>
<td>L12</td>
<td>Developers move towards OSS development</td>
<td>Partially</td>
<td>In 5 of 6 cases where OSS was considered it was chosen. Though it was not the default choice by companies, and they invested substantially to assess alternative options.</td>
</tr>
</tbody>
</table>
5.4 Validity threats

Larsson [107] highlights a number of limitations related to the case survey method. Furthermore, threats described in Petersen and Gencel [62] are highlighted where applicable.

**Generalizability:** First, Larsson points out that a limited number of cases could be studied, given that a case survey extracts more detailed information than a survey. Furthermore, the available cases are limited and not easily accessible. In this study, a total of 22 cases have been obtained. As it can be seen from Table 35 different domains and application types were studied. The automotive and telecommunication domains are most frequently represented, which introduces a bias in the dataset. Ayala et al. [9] observed that there is an increasing adoption of OSS components over proprietary (in-house) solutions. In Ayala et al.’s study most of the experiences were reported from software consultancies. In our case survey study all sourcing options were considered, and also frequently chosen for all CSOs (as shown in Figure 25). The difference in findings between the two studies indicates the need for further investigations.

**Descriptive validity - factual accuracy:** There is a validity threat in that the coding is misunderstood and that data are not available/missing. Given that all researchers reviewed and iterated the instrument (see extraction scheme in Table 34) the risk of wrong interpretations of the scheme was reduced. Interviews were conducted over the phone, which allowed to clarify and ask follow-up questions to receive accurate answers. Another threat to factual accuracy is missing data. Effort data has been elicited for 16 out of 22 cases. Not all respondents were confident in being able to provide an accurate number, thus 6 cases did not include the effort. We aimed at including only reliable data where the subjects felt confident in the information they were providing.

**Theoretical validity - confounding factors and controllability:** It would be desirable to make an inference from the characteristics of the decision making process to the success of the decision. Reflections related to the relationship between the two (process and success) are prone to confounding factors. Of particular interest are the context characteristics of the cases (see Table 35) where we captured size of the development unit developing the system, the domain, the application type, and the development methodology used in projects. In the case survey we did not identify additional context factors that may be of relevance. Though, looking at the literature a variety of factors may play a role (cf. Carlsson et al. [28])), such as organizational and team complexity, organizational
models, developer experience with respect to the project, and individual development practices used (such as pair programming). Given that the practitioners only considered a subset of the contextual factors and criteria presented by Carlsson et al. the need for providing a more systematic approach of integrating context information into decision making is highlighted.

**Interpretive validity - objectivity of the researcher:** Based on the findings from the survey conclusions and recommendations to practitioners are provided. The recommendations follow the data, and there is a risk that an individual researcher draws biased conclusions. The risk is reduced due to the number of authors involved in the study who provided their input to the reflections and key findings.

**Interpretation and coding of the data:** Mistakes and potential bias are probable when interpreting and coding a large dataset. The coding activity is similar to what would be conducted in a systematic literature review when coding extracted data from papers. Kitchenham et al. [95] recommends to peer-review the coding. Consequently, the second and third authors of the paper reviewed the coding done by the first author.

**Repeatability:** A data extraction form and the GRADE taxonomy [135] support the data extraction increases its objectivity, though a threat to the repeatability of the results remains due to the characteristics of the research.

## 6 Conclusion

In this chapter we investigated how CSOs are chosen by conducting a case survey supported by Larsson’s guidelines (cf. [107]). We identified and described 22 decision cases for choosing CSOs in the contexts of software-intensive systems. The CSOs were based on the experience of experts participating in a research project for choosing CSOs (ORION), and interviews with industry practitioners. 11 cases were based on experiences of the research project members, and 11 based on interviews.

All options considered were viable and have been selected by the practitioners, which includes In-house, COTS, OSS, Outsourcing or combinations of them. We found a mismatch between industry and practice. In particular, CSO related solutions for decision making were mostly deterministic, while decisions in practice were non-deterministic. Thus, solutions proposed for supporting decisions in the context of architecture in general are of interest for practitioners, as those take non-deterministic decision making into account. One example is the repository grid technique. Furthermore, solutions to record
past decisions in a systematic way are of interest to gain deeper insights of what decisions can be reused in similar contexts. Such learning is important as in many cases the decision made was not perceived as successful. We found that recommendations in the literature were not followed, such as considering risks and experiences and the familiarity with the technology as explicit criteria. These are potential avenues for improvements. Also, decision making approaches were mostly ad-hoc and not well structured.

A common reason for the decision to be perceived as negative were issues with the quality, such as performance and security. Means for an early assessment and estimation of these properties are of use. A small set of cases used simulators.

Future work: In summary, future work needs to focus on the following avenues:

- Conduct more in-depth empirical studies of how CSOs are chosen in industrial practice. In particular complementary case studies and large-scale surveys are of interest.

- Provide support for group decision making and consensus building as in the final decision aspects of the investigation (such as criteria and recommendations) were not followed, and the final word for the decision lied with individuals (mostly management). In particular, the tailoring of existing solutions for software architecture GDM is interesting.

- Provide tools to systematically capture and thus identify reusable decisions and create an evidence base for CSO decision making.

Acknowledgment

The work is partially supported by a research grant for the ORION project (reference number 20140218) from The Knowledge Foundation in Sweden.
A DECISION-MAKING PROCESS-LINE FOR SELECTION OF SOFTWARE ASSET ORIGINS AND COMPONENTS

Abstract: Selecting sourcing options for software assets and components is an important process that helps companies to gain and keep their competitive advantage. The sourcing options include: in-house, COTS, open source and outsourcing. The objective of this chapter is to further refine, extend and validate a solution presented in our previous work. The refinement includes a set of decision-making activities, which are described in the form of a process-line that can be used by decision-makers to build their specific decision-making process. We conducted five case studies in three companies to validate the coverage of the set of decision-making activities. The solution in our previous work was validated in two cases in the first two companies. In the validation, it was observed that no activity in the proposed set was perceived to be missing, although not all activities were conducted and the activities that were conducted were not executed in a specific order. Therefore, the refinement of the solution into a process-line approach increases the flexibility and hence it is better in capturing the differences in the decision-making processes observed in the case studies. The applicability of the process-line was then validated in three case studies in a third company.

Keywords: Component-based software engineering; decision-making; case study.

1 INTRODUCTION

In the early days of software development, it was not uncommon to internally develop software product features, operating system or even programming languages and compilers (e.g. AXE10 developed by Ericsson used an operating system and programming language developed in-house). As the software business matured, two significant trends emerged: specialization and commoditization [48]. Specialization may be viewed as a result of commoditization, as many companies embraced specialization as a means to stay competitive. The commodity parts of their products were most often taken off the shelf. The increas-
The growing popularity of Open Source Software (OSS) helps accelerating the commoditization process and encouraged many software companies to look for alternative or multiple revenue streams and new sources of novelty and value. As a result, the primary focus is now on developing software that provides a competitive advantage, e.g. killer apps.

Nowadays, companies need to decide what to develop themselves and what to get from elsewhere. On the strategic (executive) level, the strategy of mergers and acquisitions is a relevant option for obtaining software and the organizations that develop it [156]. However, acquisitions may not always be feasible or possible, including for example OSS assets. Decision-making efficiency in relation to software assets becomes important as they can be realized using internal development resources (in-house), buying Components off-the-shelf (COTS), subcontracting (outsourcing) or utilizing OSS software. The four asset sourcing alternatives provide different benefits and consequences (e.g. competitiveness), and hence affects or shapes the business models. For example, using OSS software is in many cases related to joining and participating in a software ecosystem [87]. Furthermore, the selection of one of the four alternatives directs the company towards one of the four business model archetypes: creator, distributor, lessor and broker [142]. However, for many software companies, the time for being only creators and solve technical challenges is history.

We use the term “software asset” to denote any type of software, including components that can be used for achieving the business objective for a specific system or product being developed. Software assets may be divided into four main types based on the source or origin of the asset (henceforth denoted asset origin): in-house, COTS, OSS and outsourcing. Within each of these asset origins, different assets may fulfil the identified needs, e.g., several different COTS may provide the same functionality to the user. In-house refers to assets developed or reused internally within an organization. Thus, in-house includes software having been developed within the same organization, independent of location (e.g. sites in another country), subsidiaries or organizational structure (e.g. different business area). The other three types of asset origins are external, COTS and OSS components are provided from an external source and outsourcing is here used as a sourcing option outside the organization that needs a software asset [160]. Component-based software engineering has been an important area of research for almost three decades [163] and [181].

The asset sourcing strategy that is the most optimal for an asset is an important decision for companies. Should it be developed in-house
or should it be sourced/looked for elsewhere? To date, research has focused on comparing just a few of these asset origins, e.g. in-house versus COTS, and in-house versus outsourcing. To the best of our knowledge, no paper has addressed all four asset origins [Chapter 2]. To address this gap, we proposed a set of different decision-making activities and packaged them in a decision-making process for selecting software asset origins [193]. Three types of descriptive models: decision model, property model and context model, as well as a knowledge repository were used as inputs for formulating the set of decision-making activities and the process. The input helped in identifying the different activities that support in answering several key questions required to make a decision. To build a decision-making process, the scope of the actual decision needs to be determined, i.e. what to decide. The decision-making process as such illustrates how a decision may be made. Furthermore, who makes the decision is determined by the identification of the stakeholders. The main reasons for the decision, i.e. why a decision is made, are captured through the criteria in the decision model.

The solution presented in our previous work is presented as a process [193]. However, based on the validations conducted in the case studies presented here, we concluded that the concept of a common process might not be the best fit to reality. In addition, our view of the previous solution was closer to a checklist rather than a prescribed process. Therefore, from now on we refer to the previous solution as a checklist, which here more appropriately is described as a process-line, and not a process. The initial goal of the set of activities captured in our previous solution [193] was to support the selection between the four different types of asset origins (in-house, COTS, OSS and outsourcing), although the solution proposed in [193] was expected to be adaptable also to select between different components of the same type of asset origin. Therefore, in this chapter the selection between different components of the same type of asset origin is considered as one of the cases for validating the checklist. We do not focus here on mergers or acquisitions as a sourcing strategy for software assets [156]. The evaluation is aimed at validating the coverage of the activities represented by the checklist. We also noticed during the validation that the decision-makers neither executed the decision-making activities orderly nor did they follow the proposed order of execution in our previous work [193].

The main contribution of this chapter is to provide a process-line from which decision-makers can include or exclude activities depending on their case. Therefore, the process-line presented in this chapter is a set of activities (without any prescribed order) that enables the
decision-makers to build a tailored decision-making process to select between the four asset origins and between different components. The goal is to provide a set of decision-making activities and not decision-making criteria for the selection of asset origins and components.

The extension of our previous work [193] is focused on validating the coverage of the activities represented by the checklist, formulating a process-line and validating the applicability of the process-line in supporting the decision-makers to build their decision-making process, in particular the following additional aspects are covered: 1) an in-depth description of research methodology related to the design of the checklist and the process-line and the execution of the five case studies; 2) description of the case studies that validate the proposed checklist; 3) an extended discussion of the nature of the decision-making process-line; and 4) validating the applicability of the process-line from a coverage point of view and the implications that can be drawn from conducting the industrial case studies.

The remainder of the chapter is outlined as follows. Section 2 presents background information from general decision-making theory, and a specific taxonomy intended to help formulating three descriptive models for the decisions discussed in this chapter. It also introduces the frame of reference in terms of the three descriptive models and a knowledge repository that were used as an input to identify the set of decision-making activities that form the checklist. The related work on decision-making related to different asset origins and decisions in software business is discussed in Section 3. In Section 4, we present the research methodology utilized to formulate the checklist, the process-line and its validation using five case studies. The process-line is presented in Section 5. Section 6 presents the two case studies where the checklist was validated at two companies and three cases where the process-line was validated. A discussion of implications for research and practice is provided in Section 7. Finally, Section 8 provides a summary and pointers to further work.

2 BACKGROUND

2.1 Decision-making

Decision theory largely deals with actors making decisions (e.g. bring an umbrella or not) in the face of uncertain events (e.g. rainfall or not), leading to different outcomes (e.g. wet or dry) and pay-offs (e.g. it rained and even though burdened by the umbrella, you are dry). There
are many textbook introductions to the subject, e.g. [148], as well as extensive literature reviews on theories of decision-making under risk [164].

In the area of software engineering research, decision theory has been applied to diverse problems such as evaluating software components [108], testing [158], architecture [27] and requirements engineering [69]. Decision theory is also one of the cornerstones in the theory of value-based software engineering [18]. Empirical research includes studies on how people make decisions about service level agreements [59].

The objective here is not to make a theoretical contribution to decision theory in software engineering and software business, but rather to apply it to a particular problem class: how to select an appropriate asset origin for a particular piece of software component or to choose the software component itself. In so doing, we use decision theory terminology and concepts to reason about the problem and present a decision-making process-line that will make it possible to reuse previous experience and published results alike to make the best possible decision, given the knowledge available.

2.2 GRADE taxonomy

The work presented in this chapter is grounded in the GRADE taxonomy [134]. The taxonomy summarizes the relevant concepts and definitions for decision-making for selecting between asset origins. On the highest level, it combines five fundamental concepts of decision-making for software intensive systems: Goals, Roles, Assets, Decision and Environment (GRADE) as illustrated in Figure 34. These five fundamental concepts can be used as building blocks for creating models supporting decision-making.

![Figure 34: Mapping of GRADE to concepts in the decision model and the supporting models.](image-url)
Goals represent the starting point for a decision. They represent the internal business goals and customer goals, and have a broad impact on the entire product or even organization. The goals form an important input to the decision-making.

Roles represent individuals involved in the decision-making. The roles are classified into types, functions, levels and perspectives.

The assets describe the decision assets (often encapsulated in a software component) characterized by: origin, attributes, type, usage and realization options.

The decision contains the decision methods that can be used for estimating outcomes for a specific option among those evaluated in the decision-making process.

The environment describes the environment before the decision was analysed or made. It includes the characteristics of organizations, products, stakeholders, markets and business prior to making a decision.

Our previous work [193] is based on three descriptive models that capture the concepts for decision-making as proposed by the GRADE taxonomy. We present these related concepts in the following section. It should be noted that the descriptive models and the evidence-based knowledge repository described below were used as inputs to formulate the checklist. The checklist provides an overview of the set of activities that could be considered in the decision. The working of each activity is case dependent, for example, the checklist recommends that the decision-makers should consider the selection of the appropriate property model/s as an activity to estimate or evaluate the criteria. However, the selection of the specific property model depends on the specific decision. A brief description of the descriptive models such as the property model is provided below. The detailed working of the different models is described in [28], [38] and [136], and it is considered out of the scope of this chapter as it is only used as an input to identify the set of decision-making activities.

2.3 Descriptive models

Three descriptive models are built from the GRADE taxonomy to ensure that no decision-making aspect is missed. The three descriptive models correspond to the five fundamental concepts in GRADE, as described and mapped in Figure 34. In particular, the five concepts comprise: 1) the three decision model cornerstones: stakeholders (roles), origins (assets) and criteria (goals); and 2) two supporting models – property models (decision) and context models (environment).
In addition to experience of the involved stakeholders, it is beneficial to support the decision-making with related historical evidence and experiences. This can be captured in an evidence-based knowledge repository, which is elaborated in more detail in Section 2.4.

2.3.1 Decision model

The decision model consists of three main cornerstones:

Stakeholders – which stakeholders (and hence different perspectives) need to be involved? The stakeholders should be identified from the roles in GRADE that should be involved in the decision-making. The stakeholders involved into the decisions can be categorized into: initiators, influencers/contributors (preparation) and decision-makers [Chapter 4].

The stakeholders have different perspectives (as described through the Roles concept in GRADE) that should be taken into account in the decision-making process. The perspectives include product requirements aspects that are more short-term (business, functionality and quality aspects) as well as life-cycle aspects that are usually more long-term (architecture, support and maintenance) [187].

Origins – which type of asset origins should be considered (in-house, OSS, COTS and/or outsourcing)? In this case, the asset concept in GRADE is defined as potentially coming from four different asset origins. Thus, it is assumed that the main decision to be taken relates to where a software component needed in a product or system is developed, obtained or acquired. The actual choice maybe between all relevant asset origins or a subset of the asset origins. This may also mean that the suitability of only one asset origin is evaluated to select between competing alternative assets of the same origin.

Criteria – which criteria should be evaluated to ensure an informed decision? The criteria are based on the Goal concept in GRADE. Since the goals may be quite general, some goals may not be relevant for a specific decision. It is important to acknowledge here that criteria can have at least three perspectives: customer perspective, internal-business perspective, and community (or ecosystem) perspective. The goals and criteria should be identified and tagged by the relevant perspective and potential conflicts between perspectives should be identified and mitigated. The involved stakeholder roles should review the goals, mitigate potential conflicts and translate them into defined decision criteria to be used in the decision-making. Criteria should be more detailed than the goals and need to be measurable, i.e. contain a threshold for a certain property attribute (e.g. 99.99% service availability or gaining 1
users of a software service within two months after the service component is launched). Thus, criteria should be possible to evaluate, e.g., they could state that a certain property should be above a certain threshold, and each criterion should be evaluated for each viable asset origin. The chosen criteria should be evaluated, where business risk is most likely one of the criteria. Risk is a criterion by itself in relation to a specific asset origin, e.g. the risk of a COTS supplier going bankrupt. However, risk is also related to the uncertainty in specific decisions, their criteria, and the data they are based on, e.g. uncertainty in historical cost or reliability figures.

The stakeholders contribute to the decision model as experts in their own area, e.g., business, architecture or requirements. They are involved in evaluating possible asset origins viable for the specific case and formulating the criteria for the decision based on the goals. Furthermore, the experts provide input to the property models (see Section 2.3.2), they should describe the context of the decision (see Section 2.3.3) and they should help in identifying similar historical evidence and experiences using the evidence-based knowledge repository (see Section 2). The latter includes prioritizing among important factors to compare with historical evidence.

2.3.2 Property model

The decision concept in GRADE includes both models to estimate specific properties and methods to, e.g., weigh different criteria. The property models come into play in estimating outcomes of the non-functional\(^1\) criteria [136] for different asset origins, i.e. there is a need to make the estimations with respect to different criteria for the relevant origins. Non-functional properties of the component candidates correspond to all properties beyond functionality and describe “how” a component performs or delivers its functionality. Properties are closely related to quality aspects and external aspects of a component, e.g. number of active users, source code quality or dependability. The origin of the component determines the scope of property attributes and often constrains estimation methods for these properties.

The property model ontology [136] introduces the two main elements of a property model: 1) non-functional properties that have name, data format and documentation and 2) evaluation method(s) that have names, output, unit, applicability, parameters, driver, formula, description and implementation. A valid property model has to include at least

---

\(^1\) Non-functional criteria/properties are sometimes also referred to as extra-functional or quality criteria/properties.
one property and at least one evaluation method [136]. Recent research on property model ontology identified the following non-functional properties as being important in the automotive domain: cost (development effort), performance, configurability (variability), flexibility, maintainability, testability, power consumption, reliability, safety, evolvability and security.

A property model may contain other property models. Examples of properties include coordination costs, IT service costs and maintenance costs for selecting cloud-computing services [121]. The evaluation method may be quite simplistic, e.g., expert opinion or based on a sophisticated formal mathematical decision model [7]. Recent work identified that COCOMO or estimation by analogy or expert estimation are used in the automotive domain for estimating the development effort and worst-case execution time is used to estimate complexity [136]. The decision could either have positive (goals achieved) or negative (goals not achieved) outcome. Documenting the decisions irrespective of the method used to log the success and failure stories is beneficial. This is further discussed in Section 2.4.

Property models can also be more advanced, e.g. for the reliability criterion using software reliability growth models (SRGM) based on historical data from similar situations. Furthermore, some evaluation methods use generic statistical methods such as regression analysis, while others are based on general methods but still are tailored for a specific purpose such as SRGMs. Properties can and should also be estimated for aspects relevant for communities, ecosystems and markets and not only for a company’s internal or a project’s internal aspects. A good example here could be the degree of influence on ecosystem members or the state of a company’s reputation in a given ecosystem [87].

Property models provide estimates of values for the different criteria, and in most cases the property models only handle one or a few properties at the time. Thus, there is a need to decide the priorities of the different criteria and hence the weighing between them, e.g. is cost more or less important than security. The methods for managing the priorities between criteria, or for combining outcomes in different ways are referred to as decision methods. For this purpose, it would be possible to use, e.g., methods such as AHP [155] and HCV [16]. Some initial work on these prioritization and trade-off problems can be found in [58].

As part of the decision-making, it should be decided, e.g., whether the stakeholders should try to take different time perspectives into ac-
count “manually” or if the property models should instead be used more than once, e.g., to make estimations both for a short-term and a long-term perspective respectively.

2.3.3 Context model

The context model is a representation of the environment in which the decision is made. There are two main objectives of the context model. First, it helps in identifying relevant criteria, property models and solutions previously used by others. Second, it structures the decision at hand for future use in an evidence-based knowledge repository (see Section 2.4). An example of a context model representation is presented in [139]. It comprises six dimensions of the environment, four that capture the organizational characteristics (including practices and tools) and two that are external to the organization (business environment characteristics). The context model also extends the environment concept in GRADE as it helps the decision-maker to understand the context in the future and is integrated with an evidence-based knowledge repository described in Section 2.4.

In [28], the context model is structured into the following five dimensions: 1) organization characteristics including organizational structure (management model, business strategy, maturity, capacity, velocity, etc.), 2) product characteristics and all correct contextual information associated with the product before the decision (maturity, technical debt, complexity, openness, certification, etc.), 3) stakeholder characteristics (level of involvement in the decision, experience, competence etc.), 4) development method and technology (development process and methods, practices, environment and tools used etc.) and 5) market and business (type and structure of the market, market trends, ecosystem effects and agreements etc.). Carlson et al. [28] suggested an open hierarchical model for context representation that can be dynamically adjusted and provide various granularity levels. Moreover, understandability and readability are important quality attributes of the context model. We believe that for a comprehensive context description that includes business characteristics and can be effectively used for guiding business decisions, a possible future area of research is to expand the six dimensions described in [139] to better cover aspects such as the market, ecosystems and also business models.
2.4 Evidence-based knowledge repository

Historical information should be structured so that it is possible to find relevant or similar cases, e.g., similar context, similar prioritized criteria or an interest in the same asset origins. The stored information may facilitate decision-making, and also provide what is generically known as traceability of a decision: what a decision was about, who made the decision, and why the decision was made. This is often referred to as the rationale for a decision. In this respect, any repository should record all relevant aspects of a decision-making scenario. Furthermore, a repository ought to contain other available information such as research articles on the topic, and in particular systematic literature reviews, as well as publicly available data or data shared between trusted partners that can help support different activities of decision-making.

Information from previous decisions can represent an important support in the decision-making process, at least to avoid errors made in the past. However, if the repository was considered as a mere post-decision storage support, it is difficult to justify and motivate the effort of documenting decisions in detail. Furthermore, the repository would miss a lot of its potentials: 1) as mentioned before, recurring decisions might contain important lessons learned; and 2) multiple decisions could entail an agreement about a more general development vision (e.g., different properties derivable from the same goal by different stakeholders), thus requiring consistency. Hence, continuous and reliable data collection, as well as use of the data, should be performed to unlock the full potential that an evidence-based knowledge repository offers.

In our context, the technical implementation of the knowledge repository is realized by taking into account the following observations. Storing decisions about selecting various software assets revolves around the Decision entity of the GRADE taxonomy [38]. The knowledge repository ontology decomposes the decision into the seven key entities on the first level: Environment, Aspect, Value Perspective, Asset Usage, Role Level, Decision Level, and Method Family. The repository should be able to smoothly manage large amounts of data and should offer meaningful mechanisms to retrieve decisions as filtered by their prominent characteristics (i.e., the cornerstones of the decision model), and pointers to relevant studies on the topic. Compatibility and interoperability are important quality attributes of a good decision knowledge repository and therefore we recommend using open data standards supported by reliable quality management measures, e.g. ISO/IEC
Concretely, the decision-making activities and the knowledge repository have been embedded in a prototype application\(^2\). The application allows to enter (a subset of) the decision items for documenting a certain decision case, and after they are stored in the repository. The repository is realized as a graph database through Neo4j technology\(^3\). The choice of graph databases is motivated by their efficiency and scalability for large amounts of data, characterized by flexible structure and arbitrary relationships. Decision items are represented as nodes while relationships are used to trace links between items pertaining to the same decision case. Moreover, the storage and retrieval of information to/from the database is controlled by the application and is completely transparent to the user. In this way, it is possible to keep the required degree of consistency across different decisions and hence to perform queries seeking for similar scenarios.

3 RELATED WORK

3.1 Deciding on origin

The research related to selecting between different software asset origins is quite limited. In a recent systematic literature review [Chapter 2], which is summarized here, no papers addressing all four types of asset origins were identified. However, some papers addressing two or in a few cases three origins were found.

The decision models for in-house vs. COTS are mainly based on optimization models. The optimization models proposed in \([40], [42], [89], [91], [144]\) and \([163]\) help to decide which components should be developed in-house and which should be bought. Cost, delivery time, and reliability are the common objectives and constraints considered in all the proposed optimization models. The optimization models consider single objective or multiple objectives in the decision model.

The objective in the optimization models proposed in \([40], [42], [144]\) and \([163]\) is to minimize cost under reliability and delivery time constraints. The CODER framework proposed in \([40]\) consists of a decision model based on optimization and accepts UML notations as an input. In \([158]\) and \([163]\), the authors propose an architecture optimization approach based on a swarm intelligence algorithm. The CODER frame-

\(^2\) https://github.com/orion-research/coach
\(^3\) https://neo4j.com
work [121] is extended in [142] and [144], allowing decision-making as early as requirements are available. Similarly, a general non-linear optimization model is proposed in [42] for the same objective and constraints, i.e. minimizing cost under reliability and time constraints.

Multi-objective optimization models have been proposed in [89] and [91]. A decision model for fault-tolerant systems is proposed in [1] and [89] with two objectives – to maximize reliability and minimize cost under a time constraint. In addition, coupling and cohesion have been considered in the decision model proposed in [91]. The objectives in [91] are to maximize intra-modular coupling density and functionality under time, cost and reliability constraints.

Two papers focus on deciding between in-house and outsourcing – [102] and [104] – were identified. The model in [102] provides tool support for requirements clustering to find a cohesive group of requirements using a graph-based model. In [104], Kramer et al. propose a decision model using decision tables. The input is the knowledge specificity (business, functional and technical), and interdependencies (priority between software components and communication intensity among developers).

3.2 Decision-making in software business

Running a software business requires making several decisions on multiple levels [7], ranging from strategic decisions about mergers, acquisition and takeovers [156], via tactical decisions on which ecosystem to join and support [87] to highly technical decisions on how to realize customer requirements in software. An increasing number of software companies are evolving from the pure creator business archetype, that implies code ownership but also development risk, high maintenance cost and full responsibility for delivering the required quality, towards mixed or hybrid business models that imply taking on several business archetype roles [142]. At the same time, small and large companies take on outsourcing initiatives to reduce development costs and obtain valuable knowledge and inspiration. This shifts the centre of gravity towards integration work and coordination of outsourced (often also offshored) sites into software products that deliver the value that customers expect. Finally, joining or creating an ecosystem entails a series of decisions regarding growing a healthy influence or disrupting markets by commoditization of ecosystem software. Each of the mentioned four asset origins thus has different implications both in the short term and in the long term. They come with different costs and prices and can
bring different benefits. Decision-makers responsible for running their software ecosystems \cite{87} and businesses are faced with increased decision complexity and frequency that they need to cope with to succeed with their business endeavours. An example here is decision-making in cloud computing environments for selecting appropriate services from different providers \cite{121}.

3.3 Application of process-line in software engineering

Software process-lines has gained more interest recently. However, it has been identified as an immature area and there are few papers reporting the use of process-lines according to a systematic literature review conducted on software process-lines in 2014 \cite{31}. The main use of process-lines is in areas such as software design and architecture with concepts similar to product lines such as variability management \cite{6}, product management and project management \cite{32} and \cite{4}. In our study, we use process-lines for the decision-making process, which is a rather new application. Since, process-lines have not been applied in relation to decision-making we do not discuss the above studies in detail.

4 RESEARCH METHODOLOGY

In this section, we discuss the overall research approach used to construct and validate the decision-making checklist and process-line. The relations between the decision-making process-line, checklist and the associated studies are presented in Figure 35.

The decision-making process-line is formulated based on our previous work \cite{193}, which was built upon the GRADE taxonomy \cite{134} and the descriptive models (context model discussed in \cite{28} and property model discussed in \cite{136}) as illustrated in Figure 35. A knowledge repository \cite{38} then also supports the checklist. These studies and a recent literature review [Chapter 2] on software asset origin selection have influenced and inspired the construction of the checklist. The research team (authors of this chapter) reviewed findings from the checklist validated by the industrial experts and external researchers and prioritized goals and challenges to create the focus for the decision-making process-line.

The overall research approach followed in this study is a combination of design science \cite{185} and case study \cite{57}. The tasks in the design science approach and case study are presented in Figure 36 and elabo-
rated in Sections 4.1, 4.2 and 4.3. The design science approach consists of three iterations, which begin with problem investigation, followed by design and finally validation. In addition, case studies in Companies A, B and C were conducted. Note that the checklist and the process-line were not implemented in a real context. In design science, the solution needs to be validated before it is implemented in a real context. Therefore, in this study we validate the process-line, which is an important step before implementing and evaluating the process-line in use. The limitation of the process-line evaluation is discussed in Section 4.4. The details of each of the iterations and tasks followed in each iteration are described in the following subsections.

4.1 Iteration 1

Problem investigation: In this step, we identified and formulated the design problem. Based on the template designed by Wieringa [185], we formulate our design problem for iteration 1 as follows: Identify different decision-making activities to be considered, and hence to support
decision-makers in selecting suitable software asset origins and components.

**Checklist v1.0 design – identification of the decision-making activities:** The researchers involved in this study selected the relevant objectives from the associated studies. This helped in identifying the first set of the decision-making activities. The decision-making activities were iteratively discussed via several brainstorming sessions where the researchers discussed opinions, compared possible solutions and scrutinized them. This version of the checklist (v1.0) was published in [193].

**Explorative case studies at Companies A and B:** We conducted two exploratory case studies in Companies A and B to explore the decision-making process followed in the companies. The case studies were based on semi-structured interviews with stakeholders involved in two decisions about two software components, one decision for each company. The interview questions were rather broad and did not impose any particular decision-making process.

The questionnaire consisted of three parts: introductory, decision-making process details and concluding remarks. The introduction con-
sisted of questions related to general questions about the interviewee, organization, project and product. In the next part, the interviewees were asked to describe their decision-process for each case. If the interviewees did not mention any particular activity in Checklist 1.0, they were specifically asked about it. In order to not influence the research outcome, such responses were differentiated as activities followed but not explicitly mentioned (See Section 6.3 and Figure 39). In the last part, questions related to outcome (positive or negative) of the decision were asked.

Three interviews in Company A and four interviews in Company B were conducted. Two researchers were involved in conducting the interviews that were recorded and transcribed. No information of the checklist was discussed in the interview nor any specific questions regarding the checklist activities were asked. Coding rules were established so that the coding was done consistently, the text related to the decision-making activities in [193] should be highlighted and tagged in the transcript with the corresponding activity name. The details of the exploratory case studies in Company A and B are provided in Section 6.

**Checklist v1.0 validation:** As the solution is not prescriptive, the validation here is not in terms of “effectiveness” but to validate the Checklist v1.0 coverage. The coverage is validated by ensuring that none of the decision-making activities followed by Companies A and B are missing in the checklist. The decision-making processes followed by the companies were mapped to Checklist v1.0 and were sent to the industrial experts, i.e. to the interviewees in Companies A and B after the interview for validation. It is to be noted that the researchers did not intervene and the validation was done independently by the practitioners. The outcome of the validation indicated that the industrial experts perceived that none of the activities they performed were missing from Checklist v1.0 however, the authors of this study identified the need to refine the Checklist v1.0 activity descriptions so that they are aligned with the activities carried out in Companies A and B. The refinement was carried out in iteration 2.

4.2 **Iteration 2**

**Problem investigation:** The validation conducted in the previous iteration indicated a need to add details to the activity descriptions based on the execution of the decision-making activities by Companies A and B. Particularly, the activities needed to be adapted to the process of replac-
ing the component in use and selecting between different components within the same asset origin, which were the two cases in Companies A and B.

**Checklist v2.0 design – reformulating Checklist v1.0 into Checklist v2.0:** The refinement was done by adding descriptions to include details on how the decision-making activities could be executed. In order to refine the Checklist v1.0, the decision-making processes followed by Companies A and B were considered as an input. The refinement outcome is discussed in Section 7.1.

**Checklist v2.0 validation:** Checklist v2.0 and the descriptions of the decision-making process followed in the companies were validated by external researchers. The validation was done to ensure that the Checklist v2.0 is capturing the decision-making processes followed in the companies. The aim of the validation was to correct any potential inconsistencies. The validation pointed out some inconsistencies in the representation of the Checklist v2.0 in particular, regarding the order of execution of the activities. Therefore, indicating that the representation of the solution needs to be refined. The refinement of Checklist v2.0 is carried out in iteration 3.

### 4.3 Iteration 3

**Problem investigation:** The validation conducted in the previous iteration indicated that the representation of the Checklist v2.0 suggests a prescriptive order of execution, which is misleading, as the goal of this study is not to propose a prescriptive solution. Therefore, the Checklist v2.0 needs to be packaged differently than into a process. The proposal is to formulate the Checklist v2.0 as a process-line which serves as a checklist that supports the decision-makers in building their own decision-making process without any prescribed order of execution.

**Process-line design – reformulating Checklist v2.0 into a process-line:** The refinement was done to change the sequential representation of the Checklist v2.0 to a process-line which acts as a checklist consisting of a list of possible activities divided into preparation, investigation and decision-making phases. The descriptions of the decision-making activities followed by Companies A and B and the outcomes of the validation done by external researchers were considered in the formulation of the process-line. The details of the process-line are described in Section 5.

**Case studies at Company C:** We conducted three exploratory case studies at Company C to explore the decision-making process followed
in three different decisions for different products. Semi-structured inter-
views were used to explore the decision-making processes followed by
Company C. Three interviews were conducted, two researchers were
involved in conducting the interviews that were recorded and tran-
scribed. No information of the process-line was discussed in the in-
terview nor any specific questions regarding the process-line activities
were asked. The same coding rules as mentioned in iteration 1 were
followed. The details are provided in Section 6.

**Process-line validation:** After the interview, the process-line de-
signed in iteration 3 was presented to the industrial experts in Com-
pany C for validation. The objective was to validate the applicability of
the process-line in building the decision-making process as perceived
by the industrial experts.

4.4 **Validity discussion**

**Researcher bias:** There is a threat of introducing researcher bias in the
design, and validation tasks of the design cycle and in the exploratory
case studies.

- **Decision-making process-line and checklist design:** Data trian-
gulation helped to minimize the researcher bias in the design
phase. The checklist was constructed by considering inputs from
various associated studies [Chapter 2], [28], [38], [134] and [136].
In addition, the inputs from the decision-making activities fol-
lowed in the industrial context and external researchers were con-
sidered in the formulation of the process-line. The validation us-
ing expert opinions of external researchers was done to ensure
that all aspects of the associated studies and the facts from the in-
terviews were considered in the process-line. Thereby, individual
researcher bias was minimized during the design process.

- **Validation:** The validations were conducted without any interven-
tion, i.e. Checklists v1.0 and v2.0 as well as the process-line were
not discussed in the interviews. We believe that this approach
minimizes the risk that our interviewees consider some decision-
making activities as performed because they are confronted with
the checklists and process-line. Since the aim of the process-line
is not to suggest one way of making decisions the limitation of
not evaluating the effectiveness is not applicable in our study.

- **Exploratory case studies:** The open-ended interview questions
minimized the research bias in the data collection. The interviews
were recorded, transcribed and coded to capture all the information. Coding rules were established and the coding process was reviewed to avoid inconsistencies and avoid misinterpretation in the coding process. The most relevant stakeholders responsible for the decision in Companies A, B and C were interviewed. Therefore, the threat of not identifying all activities followed in the decision-making process by Companies A, B and C is minimized.

**Generalizability:** The inability to generalize from the five cases can be perceived as a threat. Flyvberg [57] addresses five case study misunderstandings, one of which is generalizability. He suggests that if knowledge cannot be formally generalized, it does not mean that it cannot be used to accumulate knowledge in a given field or in a society. This indicates that the knowledge can be used to contribute towards generalizing the findings. To contribute towards generalizability, the cases have been described in as much detail as possible without compromising the confidentiality. In addition, the selection of case studies also provides support towards generalizability [57]. The selected cases are diversified in terms of the size of the company, decision goals and the decisions. This is further elaborated in Sections 6.1 and 6.2. We selected decisions (cases) after recommendations from the practitioners to ensure their relevance and representativeness for the study.

5 PROCESS-LINE FOR DECISION-MAKING USING A CHECKLIST

The main objective of the checklist designed in iterations 1 and 2 as discussed in Sections 4.1 and 4.2 was to provide a set of decision-making activities to select between different software asset origins and components.

The solution proposed in our previous work consists of a set of decision-making activities that are not prescriptive and should be regarded as a checklist rather than a process. This may be captured in a process-line which serves as a checklist, i.e. activities that may be selected (or not) to create different decision-making processes. Process-line has been defined as “a set of software processes with a managed set of characteristics that satisfy the specific needs of a particular organization and that are developed from a common set of core processes (referred as activities in this study) in a prescribed way” [6]. Note that the set of core processes/activities are not prescriptive rather, the development of the process from a common set of core processes is prescriptive.
Each user of the process-line may decide to include or exclude specific activities and hence make informed decisions of which activities to include and exclude. The process-line was designed in iteration 3 as discussed in Section 4.3. Figure 37 depicts the process-line consisting of three phases (preparation, investigation and decision-making) and a repository. Note that the activity numbers mentioned in Figure 37 are identifiers and do not represent the order of execution. The arrows between the three phases represent the dependencies between the phases, and the arrows between the phases and the repository represent the information flow (one- or two-way communication). The input to the repository from the preparation phase is the context information for the specific decision. Based on the context information, similar cases from the repository are retrieved that could be used to execute the other activities in the preparation phase (elaborated further in Section 5.1) as well as in the investigation phase. In the decision-making phase, similar cases and decisions could be retrieved from the repository and after the decision is made, the final decision could be documented in the repository. The activities within each block are selective activities that could be used. The description of each activity is provided in Section 5.1.

Using the process-line, decision-makers can build their own decision-making process. One possible decision-making process with one possible order, including all activities, is depicted in Figure 38 using the numbering of the recommended activities below. The arrows between the activities indicate the information flow (input and output). However, it should be stressed that the activities do not necessarily have to be followed in the order depicted in Figure 38. Some activities may be perceived as more important than others. However, it has been chosen to present all activities as recommended activities, since the actual usefulness and effectiveness of the different activities and preferred order of the activities may vary from case to case. Thus, the order of the activities should be seen as one possible suitable order. Furthermore, not all activities may be perceived as needed for all decisions. However, it is better to make conscious decisions to not conduct all activities. Thus, the process-line should not be seen as prescriptive; it is intended to make decisions more transparent and to provide support to decision-makers so that important aspects are not overlooked. Furthermore, an evidence-based knowledge repository may not be available in all cases, and hence those activities may not be applicable in all cases. It should also be noted that iterations are expected. They may appear between
any activities depending on the specific decision, or the specific circumstances in relation to a decision.

Figure 37: A process-line for decision-making supported by a knowledge repository.

Figure 38: A possible decision-making process including having a knowledge repository.
5.1 The recommended activities in the decision-making process-line are as follows:

1. **Identify stakeholders to be involved in the decision** – It is important to ensure coverage of roles and persons to make sure that the decision made is possible to implement efficiently. Each stakeholder that is relevant for the decision and its consequences for the business should be identified here.

2. **Screen and evaluate the suitability of the asset origins** – The possible origins for a software asset should be identified. The selection of an asset origin could be for a new component or to replace a component in use with another component by considering two or more asset origins or even from the same origin. For example, there might be a need to replace a component in use developed in-house with an OSS or COTS component. This includes investigating the technical and business compatibilities and the short- and long-term costs of selecting each asset option. In certain cases, not all asset origins are allowed or suitable. In some cases, the main decision is whether to do development in-house or going externally. Sometimes, OSS solutions are not an option. Thus, the possible asset origins need to be identified carefully.

3. **Decide criteria from goals** – Both business and technical criteria are decided based on the goals and targets. The targets should be set so that different asset origins can be evaluated and compared with each other. The goal could be either to choose an asset origin to replace a component in use or choose an asset origin for a new component. In case of replacing a component, the process most likely begins with generic criteria to overcome the existing challenges or shortcomings. For example, a company might want to reduce the time for correcting defects of a component in use, then “reduced time for correcting defects” becomes the generic criterion. Apart from generic criteria there might be additional specific criteria such as “better performance” based on which the selection of an alternative asset origin is made. In most cases, risk needs to be considered as one criterion, since it may differ substantially for different asset origins (in-house, COTS, OSS and outsourcing).

4. **Decide on priorities of criteria** – It is also important to decide how the criteria should be prioritized, e.g. using AHP [155] or HCV [16]. It may also be the case that certain stakeholders have
more power in a decision, i.e. different stakeholder roles may need to be weighed differently in the prioritization process.

5. **Decide on how to handle the time aspect** – Certain solutions may be perceived better or worse in the short-term and long-term respectively. For example, a certain solution may be very good to get a product on the market, but not very good for the long-term architecture of the product. The time aspect is highly relevant when long term maintenance cost is substantial (e.g. when developing in-house) or can be minimized (e.g. by adapting OSS). Decision-makers either have to take time aspects into account when prioritizing between different asset origins or evaluations have to be done separately for different time aspects, e.g. short- and long-term, and the trade-off between them has to be agreed upon.

6. **Identify and describe the context** – Context information such as product context, organizational context and product context may have an impact of the decision-making process. For example, organizational context such as governance structure of the organization might indicate the types of stakeholders that should be considered in the decision (Activity 1). In addition, depending on the regulations/policies some asset origins might not be possible to use. For example, the organizational policy might not allow any code to be open. Thus, OSS might not be a suitable asset origin (Activity 2). The selection criteria and priorities (Activities 3 and 4) depend on the context of the product. For example, if the product has many users, then one of the criteria would be to choose a component that support many users. Therefore, identifying the context is important in a decision-making process. The context description is important to enable comparison with previous cases internally and externally as well as with the research literature, for this purpose the case has to be described. This should be done using the context model, where salient aspects have to be captured. This may include business model(s) used, application domain, system size and development method as well as a range of other aspects [139]. Independently, it is crucial to capture these aspects to enable identification of similar cases and hence relevant evidence and experiences.

7. **Look for similar cases in a knowledge repository** – The identification of similar cases is done using the context information as well as the asset origins considered as suitable and the criteria.
Thus, a similar case is defined as having some key aspects of the context in common (from Activity 6) as well as a focus on similar criteria (from Activities 3 and 4) and similar suitable asset origins (Activity 2). Similar cases are identified and studied to identify relevant evidence and experience and to uncover potential alternative decision scenarios [28]. The knowledge repository could be solely based on internal cases or a more elaborate database containing both internal and external cases. The information in the knowledge repository may indicate that, other asset origins, criteria, property models or decisions have been considered in other similar cases. Thus, it is important to be able to challenge the choices made in the other activities as illustrated in Figure 37 and Figure 38.

8. Decide on property models to use – Once the criteria are decided, there is a need to decide how the criteria should be evaluated. If using a knowledge repository, this can be done by retrieving valuable information in terms of what others have used in similar cases (Activity 7). If there is no knowledge repository, the property models for each criterion have to be decided without additional support, whether they are expert opinions or more advanced estimation models.

9. Evaluate the criteria, make estimations using the property models (including expert judgment) and evaluate the impact of the decision – The criteria can be either measured or estimated. For example, it is possible to measure the performance of an existing COTS or OSS component. However, the cost to develop a new component can only be estimated. Given the chosen property models, estimations need to be done for each criterion for the asset origins under consideration and potentially for different time aspects based on the approach decided in Activity 5. The impact of the decision and the changes that the new decision will bring must also be evaluated. Such evaluations provide better insight into the decision and to perform cost-benefit analysis. The benefits of the decision should outweigh the costs introduced due to the changes needed to implement the decision. Such evaluations are particularly important when an component in use is replaced with a new component from another source (asset origin). The benefits should not only outweigh the costs but also should be better than the component in use. Trade-offs should
also be considered. For example, reducing license cost might require additional effort.

10. **Weigh the estimation results of the selected properties based on the priorities of criteria** – The estimation results from the different property models should be weighed together. This is non-trivial given that the values as such cannot be combined easily in many cases. It is rather the estimation of each criterion and its distance from the targets that need to be weighed together.

11. **Make a tentative decision** – Once the outcomes from the property models have been weighed together, it should be possible for the decision-makers to make a tentative decision. If a knowledge repository is available, it is recommended to browse previous decisions and review relevant tentative scenarios and compare the tentative decision with decisions from similar cases as described in Activity 7. Relevant business context factors should be evaluated here based on similar cases. This should be done to make a final evaluation of the decision, and ensure that the reasoning done is as correct as possible and that no relevant available information is ignored. The tentative decision may be implemented in the form of prototypes.

12. **Make a final decision** – This has been the objective of the decision-making process and hence it is a very important activity for the development. It is important that the stakeholders are able to communicate both the actual decision and the rationale for the decision.

13. **Store the case in the decision knowledge repository** – The case information, including the context information, the criteria used, the stakeholders involved and the asset origins considered should be carefully documented. This activity is important as it allows for transparency if the case is properly documented (including the decision rationale) and helps to organically grow the evidence-base knowledge repository. It is important to add new cases given the speed of change and hence ensure that recent cases are available for decisions to come.

The decision-makers can build their own decision-making processes by including or excluding the set of activities. At the end of the decision-making process, the objective is that the stakeholders should
Table 49: Overview of the companies

<table>
<thead>
<tr>
<th>Company</th>
<th>Size</th>
<th>Profile</th>
<th>Customer Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company A</td>
<td>~100k</td>
<td>Global networking and telecommunications equipment and services</td>
<td>B2B and B2C</td>
</tr>
<tr>
<td>Company B</td>
<td>~500</td>
<td>Global telecommunications</td>
<td>B2B and B2C</td>
</tr>
<tr>
<td>Company C</td>
<td>~600</td>
<td>Global provider of cloud-based Business Support Solutions (BSS)</td>
<td>B2B</td>
</tr>
</tbody>
</table>

have come to either a consensus or at least that the involved stakeholders know why the decision was made, and are able to communicate it in the organization.

6 Validation and Case Studies

6.1 Overview of the companies

The validation was done in two phases, first the coverage of the checklist designed in iteration 1 (Section 4.1) was validated in Companies A and B. The validation outcome indicated that the checklist was not missing any decision-making activities however, some reformulation of the activities was needed. The reformulated checklist was then reviewed by external researchers based on which the representation of the checklist was reformulated into a process-line. The coverage of the process-line was validated in Company C. The case studies were conducted at three companies. Due to the strategic nature of this type of decision, the names of the companies are confidential. However, it does not affect the outcomes from the case studies, since the sole purpose is to validate the coverage of the checklist and process-line and not in detail report data from the company. The case studies were designed as described in Section 4. The overview of the companies including the size of the company in terms of number of employees, profile and customer category is presented in Table 49.

From Table 49 we can see that the companies are diversified in terms of the size of the companies. The size of the company might affect the decision-making process adopted by the companies. Therefore, identi-
fying the decision-making activities followed by diversified companies will help in validating the coverage of the process-line. In addition, the companies have diversified customer category therefore, the process-line can be validated in different contexts.

**Case 1 description at Company A** – In this case the company wanted to replace a component already in use. Due to the existing problems such as lack of timely support from suppliers and substantial license costs, a need to replace the component was identified. In-house and outsourcing were not suitable options due to the immense development effort required. Based on the positive result of a proof-of-concept, and success stories of using the same OSS component by other companies, the decision to use the OSS component was made. The stakeholders involved in the decision, i.e. the architect, line manager and system manager were interviewed. The architect was the decision initiator, contributor and was also involved in the final decision along with the line manager and system manager.

**Case 2 description at Company B** – The hardware for the products is partly internally developed and partly purchased from suppliers. The software that runs on this hardware is based on an OSS platform with extensive adaptations and unique functionality developed in-house. The stakeholders involved in the decision were interviewed, i.e. the product experience planner who was the decider, project scope manager who contributed in the decision with the required information and knowledge and personnel from the legal and sourcing department who was the influencer in the decision. The case at Company B was a decision between two COTS suppliers. The decision-makers had already selected COTS as the asset origin. However, other options were considered, but they were ruled out very early in the process, and hence the actual decision-making became a decision between two different COTS components. Considering OSS components, it became apparent that OSS could not provide the necessary functionality and support. The in-house and outsourcing options were the least favourable already from the beginning due to the immense effort required to provide the solution comparable with the currently available COTS components. Thus, the focus was set on deciding between two COTS components. The decision process took several months and was heavily influenced by technical investigations led by so-called product experience planners. Top management used their technical recommendations when making the decision. The sourcing department at the company negotiated the contract details with the selected COTS supplier. Legal aspects (have control over the patent portfolio and enough resources to fight in the court
Table 50: Overview of the cases with respect to the goal of the decision, the final decision made and the decision outcome.

<table>
<thead>
<tr>
<th>Company</th>
<th>Case</th>
<th>Goal</th>
<th>Final decision</th>
<th>Decision outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company A</td>
<td>Case 1</td>
<td>Replace component in use</td>
<td>COTS→OSS</td>
<td>Positive</td>
</tr>
<tr>
<td>Company B</td>
<td>Case 2</td>
<td>Choose a COTS vendor for building a new component</td>
<td>COTS vendor chosen</td>
<td>Positive</td>
</tr>
<tr>
<td>Company C</td>
<td>Case 3</td>
<td>Add new functionality for a component in use</td>
<td>OSS component chosen</td>
<td>Not positive</td>
</tr>
<tr>
<td>Case 4</td>
<td>Change the deployment framework component</td>
<td>COTS→OSS</td>
<td>Positive</td>
<td></td>
</tr>
<tr>
<td>Case 5</td>
<td>Replace component in use</td>
<td>No replacement</td>
<td>Positive</td>
<td></td>
</tr>
</tbody>
</table>

if some other company initiates or pursues legal procedures) were significant in this case and considered next to technical functionality and quality aspects. Finally, the interaction process between the suppliers and Company B, as well as how they supported the components took a substantial role in the decision.

**Case 3, 4 and 5 descriptions at Company C** – The architects were mainly involved in the decision. The strategic decision as a company was to use as much OSS as possible. Therefore, there was no approval needed for using an OSS component. The decision was mainly focused on fit to functionality criterion and the chief technical officer agreed to decisions proposed by the architects. Hence, we interviewed only the architects for exploring the cases in Company C. In all three cases, OSS was the preferred choice of asset origin. In Case 3, a decision to use a new OSS plugin was made to improve the functionality of an OSS component already in use. In Case 4, the decision was to replace the COTS deployment framework in use with an OSS deployment framework. In Case 5, the decision was between changing to a new OSS component, or continue to use the current OSS component. Therefore, all three cases were distinct since, Case 3 was a new decision, Case 4 was to change from one asset origin to another, and Case 5 was a decision between two components from the same asset origin. The overview of the cases in terms of the goals, final decision and outcome is presented in Table 50.
The goal of three decisions (Cases 1, 4 and 5 in Table 50) were to replace the current COTS component with a new OSS component. In Case 1, the decision was to switch to the new component however, in Case 5 after the investigation the decision was to continue using the component in use. In Case 5, the component in use was perceived to be a sub-optimal solution in terms of functionality fitness. Therefore, the goal was to replace the component in use with a component having better functionality. However, the team was not familiar with the new component and it required additional training or hiring trained personnel. The additional cost and time to train or hire new personnel prevented the decision-makers to switch to the new component. In retrospect, the decision-makers perceived the decision to be positive as switching to the new component would increase the cost (increased head count/training) and delivery time. Therefore, maintaining the cost and timeliness received more importance than functionality fitness (this is further elaborated in Section 6.5-Activity 4). In conclusion, it was considered better to keep the component although it was not the original goal. It was perceived as positive to have done the evaluation, although it did not result in any change. In Case 4, the decision was to change the COTS framework in use to a new OSS framework.

In Case 2, the goal was to choose a new COTS component. Even though the other asset origins were usually considered in the decision-making process by Company B. For this component, the other asset origins did not have any suitable solution. Therefore, the decision was mainly to choose a COTS vendor.

6.2 Overview of the decision-making process followed by the companies

The activities followed or not followed in the companies are illustrated in Figure 39 and elaborated in Sections 6.4 and 6.5. In addition, we depict how the activities were mentioned (explicitly or not) in the interviews. The total number of activities followed by the companies is indicated on the x axis and the frequency of the activities followed in the cases is indicated on the y axis. The circle symbol in Figure 39 represents the coverage of the activities followed by the companies. The circle with diamond inside indicates that the activities did not come up in the interviews when asked about the decision-making process. However, when asked about the specific activities they mentioned that they followed the activities. The diamond symbol indicates that the activities were followed however, either partially or not for its intended purpose.
Figure 39 represents the coverage of the activities followed by the companies using the process-line (and not checklist), as the process-line is the final solution of our study. The cross on the dotted arrow lines indicates that there was no information flow between the phases and the repository. The number of activities followed by the cases is similar. However, Case 3 followed the least number of activities and it was also the decision that was perceived to be negative as shown in Table 50. Activity 5: Timeframe that was not followed was perceived to be an important activity that resulted in having a negative outcome.

Seven activities of the process-line were followed in all five decisions as shown in Figure 39. The activities related to the repository (i.e. 7: Similar cases and 13: Document case) were the least followed activities. In addition, the documentation was done however, not for reusing in future decisions. Success stories from similar cases were considered in the decisions however, the success stories were based on the decision-makers’ knowledge and were not retrieved from a repository as they were not documented.

In the preparation phase, 4: Criteria prioritization and 5: Timeframe activities were not followed in all the cases. Since criteria are not always prioritized, 10: Weighing activity is also not followed in all the cases. Tentative decisions were made in the three out of four cases where the decision was made to take an action.

Overall, the practitioners did not perceive any decision-making activities to be missing in the process-line. The details of how the activities are followed by the companies are discussed in Sections 6.4 and 6.5.

6.3 Detailed description of the decision-making activities followed in all the cases

Activities 1, 2, 3, 6, 8, 9 and 12 were followed in all the cases. The descriptions on how these activities were followed is provided below.

**Activity 1: Identify stakeholders to be involved in the decision** -

Table 51 provides the stakeholders involved in the decision committee and their roles for each case. In most cases the architects played an important role in the decisions. Except for Case 2, the architects were the initiators, contributors and deciders as mentioned in Table 51. In Case 1, the budget approval was required from the design centre head and product manager. In Case 2, the legal and sourcing department was responsible for negotiating and signing the contract with the component suppliers. The number of roles involved in the decision varies from case to case.
Figure 39: The process-line activity coverage in related to the decision-making process followed by the companies.

Figure 40 provides the number of roles involved in the decision. Case 2 has the highest number of stakeholders with different roles involved in the decision. As the decision was based on selecting between different COTS vendors, involving COTS vendors in the decision and negotiating the legal aspects were part of the decision. In Case 1, there was no stakeholder responsible for discussing legal aspects however, it was realized as an important role in the decision-making and was considered for future decisions. Since, Cases 3, 4 and 5 were conducted in a comparatively small company, the decision committee consisted of only two members. The decision was made by the architect and chief
Table 51: Stakeholders involved in the decision committee and their roles in the decision.

<table>
<thead>
<tr>
<th>Case no.</th>
<th>Decision committee</th>
<th>Decision-making role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>Design centre head and product manager</td>
<td>Budget approver</td>
</tr>
<tr>
<td></td>
<td>Line and system manager</td>
<td>Deciders</td>
</tr>
<tr>
<td></td>
<td>Architect</td>
<td>Decision initiator and contributor and decider</td>
</tr>
<tr>
<td>Case 2</td>
<td>Product experience planner and project manager</td>
<td>Deciders</td>
</tr>
<tr>
<td></td>
<td>Project scope manager, developers and testers</td>
<td>Decision contributors</td>
</tr>
<tr>
<td></td>
<td>Legal and Sourcing department</td>
<td>Decision influencers</td>
</tr>
<tr>
<td></td>
<td>Component providers</td>
<td>Decision influencers</td>
</tr>
<tr>
<td>Cases 3, 4 and 5</td>
<td>Chief technical officer</td>
<td>Decider</td>
</tr>
<tr>
<td></td>
<td>Architect</td>
<td>Decision initiator and contributor and decider</td>
</tr>
</tbody>
</table>

The technical officer agreed to the decision in all three cases in Company C.

![Number of roles](image.png)

Figure 40: Number of stakeholder roles involved in each decision.

The number of stakeholders involved, their roles and the personality of the stakeholders affect the inclusion or exclusion of the decision-making activities and the time spent on each activity as perceived as one of the architect interviewed. For example, some stakeholders might need more information to approve a decision.

**Activity 2: Screen and evaluate suitability of the asset origins** – In all cases OSS was the preferred origin and only two origins were considered as shown in Figure 41. In case 2, various OSS components
were available and evaluated regularly but due to lack of full support for text input in over 60 languages they could not be considered. COTS remained as the only option with two potential COTS suppliers. In-house and outsourcing were not considered as suitable alternatives due to the extensive effort (100+ developers working for several months) in Cases 1 and 2. Company C has a strategy to use OSS as the first choice. Developing the component was considered as the last option.

![Figure 41: Number of asset origins involved in each decision.](image)

**Activity 3: Decide criteria from goals** – The number of criteria considered is case dependent as shown in Figure 42. In Case 1, the goal was to replace the component in use with a new component that overcomes the problems identified with the component in use. The general criteria selected based on this goal were: maintenance effort, technical support, costs, distribution, fault tolerance and fitness in terms of functionality. Apart from these criteria, specific criteria related to the actual decision to select the component were: scalability (architecture for scaling), security, performance for big data, replication, source code availability for defect fixing, maturity of the OSS community and the number of users using the component.

In Case 2, the criteria used in the decision were: support for multiple languages, support for defect fixes, ability to survive “patent fights”, reduced costs, better user experience, possibility to add customized user experience elements (like buttons), and technical features (next word prediction and text recognition from sloppy writing).

In Cases 3, 4 and 5, functionality fit was the criterion to select the component. In case 5, the skills needed to maintain the new OSS component was also considered as a criterion in addition to functionality fit.
Figure 42: Number of criteria considered in each decision.

**Activity 6: Identify and describe the context** – In Case 1, organizational, product and supplier contexts were considered in the decision. The governance structure of Company A was such that all relevant stakeholders should be involved in the decision. The product context such as complexity, criticality, the number of customers, and subscribers or users using the product was considered. The context of the supplier community such as maturity and number of recent releases was considered.

In Case 2, the context is implicitly discussed but not modelled. The company does not have enough knowledge within text recognition to internally develop the component and all competitors use external suppliers. The text recognition component is important for the product offering but not considered as the main driver for purchasing the product.

In Cases 3, 4 and 5, the context was considered important. For each decision, the following question is answered – “Are we the organization that can support the decision (i.e. do we have the required skills or support required to implement the decision?) or can we become that organization?”

**Activity 8: Decide on property models to use** – In all cases, expert opinion was used to evaluate the criteria. Deciding on how to evaluate/estimate the criteria did not come up in the interview but when asked about the process the interviewees mentioned that they followed the activity. The evaluation was based on stakeholders’ opinions, experience and knowledge.

**Activity 9: Evaluate the criteria, make estimations using the property models and evaluate the impact of the decision** – In Cases 1, 4 and 5, where the goal was to change or replace the component in use, the experts were aware of the attributes such as time, cost and effort for
the component already in use. The evaluation was done using expert opinion method to find alternatives that improve the existing values and not make it worse. In Case 3, the functionality of the plugin was evaluated by the decision-makers by testing the functionality of the plugin.

Company B (Case 2) had the following properties: level of support (measured by checking if the supplier has regular meetings with the procurer and how quickly they react for issue requests), and code quality was important but only externally measured by checking how many crashes the component had. Internal code quality could not be measured. Support for multiple languages was measured by checking how good the functionality and the next word prediction is in several languages. User testing and prototype-based evaluations were heavily used in this case.

In Case 1 the scalability and redundancy of the OSS component were evaluated. The impact of the decision in terms of risks was also evaluated. For example, if the OSS component did not evolve, then the need to modify the component is identified. In order to modify the component, the source code needs to be fully understood. This involves a learning curve and results in increased head count. The patents and proprietary rights were also evaluated.

The number of additional resources required to use the new component was estimated in Cases 1 and 5. The trade-off between the risk of having to contribute to the OSS community and monetary value gained by not having to pay for external support was evaluated in Case 1. Similarly, in Case 5, the trade-off between the additional time and cost to train employees or hire new employees and its fit to functionality was evaluated.

**Activity 12: Make a final decision** - A final decision was made in all cases. A COTS component was replaced with an OSS component in Cases 1 and 4. In Case 2, a decision to acquire a COTS component from a COTS vendor and a new OSS plugin component in Case 3 was made. In Case 5, it was decided to continue using the component in use and not change to a new component.

6.4 *Detailed description of the decision-making activities followed in a subset of the cases*

Activities 4, 5, 7, 10, 11 and 13 were followed only in some of the cases. The descriptions on how these activities were followed is provided below.
Activity 4: Decide on priorities of criteria – In case 1, prioritization of the criteria was not mentioned explicitly in the interviews. However, the criteria driven from current issues i.e. reduced maintenance time and lower license cost was perceived more important than finding the best possible component replacement based on criteria such as scalability. Therefore, timely support and cost of license were prioritized over other criteria.

In Case 2, the selection criteria were prioritized. Support for multiple languages and support for maintenance and defect fixes received the highest priorities. Technical support and cost were prioritized over accessibility of the source code. Code quality and intended user experience were also important.

In Cases 3 and 4, there was no prioritization as only one criterion was considered. In Case 5, the skills required to maintain the OSS component were considered more important than functionality fit. Mismatch in the required and available expertise of the current team was a bigger risk than the benefit of functionality fitness. If the people need to be trained to use the new functionality of the OSS component or new people need to be hired, then the OSS component was not considered as a good alternative.

Activity 5: Decide on how to handle the time aspect - The short- and long-term impact of the solution was not explicitly mentioned in the interviews. In Case 2, the interviewees talked about long-term implications of the decision (support and patent fights and support for maintenance over 2-5 years) versus short-term (providing bug fixes on time and quick updated for the upcoming product releases).

In Cases 4 and 5, the long-term implications of the decision were considered. In addition, there was no need for an immediate solution as the current solution acted as a backup.

However, in Case 1 the patent issues were realized much later after the decision was taken, which indicates that patent issues have a long-term impact. In addition, in Case 3, the compatibility issues were perceived to have a long-term impact as the stakeholders realized that the updates between the component and the plugin were not always synced.

Activity 7: Look for similar cases in a knowledge repository – In Case 1, the decisions were not documented and similar cases were not retrieved from the repository. However, the same OSS component was previously used by the company successfully, also other large companies used the same OSS component. Although the criteria in the previous decisions were not reused the successful implementation of the
component in previous cases increased the confidence in making the decision.

In Case 2, it was not perceived that there were similar cases. There was no component that offered a similar level of functionality, quality aspects and long-term support from the supplier. However, “similar” could be interpreted a little broader, i.e. to learn from similar cases in the company and not only in the current product-line. Thus, implicitly the previous decision to select between COTS providers could be considered as similar. In a broader sense, any COTS component purchase creates a question about later maintenance and bug fixes and also support for additional functionality.

In Cases 3, 4 and 5, there were no similar decisions made as perceived by the interviewees.

Activity 10: Weigh the estimation results of the selected properties based on the priorities of criteria – Weighing activity was conducted in Cases 1, 2 and 5. It was explicitly mentioned in Case 5 and in Case 1 and 2 the interviewees mentioned it when asked about the weighing activity.

As discussed in Activity 4 followed in Case 1, reduced maintenance time and lower license cost were prioritized criteria. These criteria were traded-off with the additional people (head count) needed to build a local support team in order to replace the technical support provided by COTS vendor. The additional head count cost was traded-off with the COTS vendor license cost. At the end of the weighing, it was perceived that adding additional people to support the maintenance locally was better (faster and more transparent) than relying on the COTS vendor. In addition, the cost of adding people was perceived to be lower than the COTS vendor license cost.

In Case 5, the benefit of having better functionality was trade-off with the addition time and cost in training or hiring the people. At the end of the weighing, it was perceived that using the current component was better (workable solution) than adding additional people to support the maintenance of the component.

In Case 2, the company implicitly compared the service level agreements between the COTS providers. Cost was weighed against the other decision criteria in the final phase and at the end cost was not a factor since both offerings had the same price. Initially, the prices were different, but at the end the price was the same. In other words, one supplier lowered their price to ensure that the price as such was not a decisive factor. Thus, the level of support and the “relationship with the supplier” were the key criteria for the decision.
Activity 11: Make a tentative decision - A proof-of-concept was created and the tentative selection to use the component on a smaller product in Cases 1 and 4 was made. In addition, in Case 2, a tentative decision was made to narrow it down to two options. These two options were evaluated after which one of the components came out as better with respect to some aspects. The key aspects evaluated here were the performance levels in different languages and possible differences in that and how they will impact the expected user experience. As a part of the proof-of-concept, the company also requested several adaptations from suppliers and experienced their response to extension requests.

Activity 13: Store the case in the knowledge repository - The decision was not documented in the classical sense in Cases 1 and 2. It turned into the knowledge about what the decision-makers selected and why. The minutes of the meetings with the decision committee were stored in Cases 2, 4 and 5. However, they were not stored for the purpose of reusing the knowledge for future decisions.

Finally, it should be noted that no new activities were identified through the interviews and discussions. Different asset origins were considered as viable options by Company B. However, only one asset origin was a suitable option and the decision-making was mainly to select a component from the asset origin. It is noteworthy that the decision-making process-line fits well with the decision to select between different asset origins and to select between different suppliers (two components from the same asset origin, i.e. COTS in this case).

6.5 Discussion and implications from the cases

It is noteworthy that the decision-making process-line presented in Section 5 was found useful for selecting between components of the same asset origin. This is validated through the component selection at Company B. Selecting between asset origins is primarily a strategic decision, while the selection between components is mostly performed on a tactical level [7]. The process-line presented in this chapter is likely to support both strategic as well as tactical level decisions.

The decision-making process followed by the companies is case dependent as we see in Section 6 and Figure 39. Thus, the process-line requires substantial flexibility not only in how the activities are conducted but what activities are conducted. The order of the activities (which activity comes first and which second) is also flexible. However,
there is sequential order of execution between the preparatory, investigation and decision/making phases.

Although the activities executed were similar there were some variations in the way the processes were executed. This led us to identify the possible variation in the execution of all the activities in the process-line based on the cases. In addition, the execution of some activities could be dependent on the execution of other activities. The variations and dependencies are presented in Figure 43.

Each activity execution could vary, for example, as shown in the first box in Figure 43 a decision-maker could identify all the relevant stakeholders at the beginning of the process or only a sub-set of the stakeholders and add more stakeholders at later stages whenever required. There were no variations identified based on the interviews for the activity to identify and evaluate the context, criteria and the weighing activity. Similarly, no variations in the tentative decision and the final decision were identified in the interviews. The “weighing” activity depends on the execution of the “criteria prioritization” and “evaluate” activities. If the criteria are prioritized and evaluated (represented by AND gate in Figure 43), then the weighing activity can be executed by weighing the priority of the criteria together with the estimated results obtained from the evaluation.

The execution of the process-line could be iterative for example, a sub-set of stakeholders could initiate the decision and in each iteration more stakeholders could be added. After each iteration, a decision if more investigation is needed to reach to a decision could be made. If further investigation is needed, then activities could be executed again. This finding is from Case 5, where the decision was made to not continue the investigation and use the component in use without any replacement.

The process-line in Figure 37 and the process-line with variations in execution in Figure 43 were shown to the interviewees in Company C. The interviewees’ perceived the process-line in Figure 37 to be applicable from a managerial perspective and the process-line with variations in Figure 43 was perceived to be applicable from a technical perspective. In addition, having a process-line with a set of activities was perceived to be useful for the decision when the time is short as the process of selecting the decision-making activities to include in their decision-making process could be faster. On the other hand, having more time would allow the decision-makers to use the process-line and conduct the activities more thoroughly according to another interviewee. Another use stated by the interviewees was that the process-line is appli-
Figure 43: The process-line for decision-making including variations in activity execution.

cable when there is a problem in the current decision/making process and they are looking for ways to improve it. However, a wider survey is needed to capture the applicability of the process-line.

7 GENERAL DISCUSSION AND VALIDATION IMPLICATIONS

7.1 Checklist v1.0 validation and refinement into Checklist v2.0

The Checklist v1.0 validation conducted in Companies A and B, resulted in the formulation of Checklist v2.0. The following changes were made to the activity descriptions based on the validation.

- Evaluating the suitability of only one origin is added to the description of the “Screen and evaluate the suitability of the asset origins” activity.

- In “Decide criteria from goals” activity, we made a distinction between criteria used to reject an asset origin or replace an existing component, and the actual criteria used to make the decision.
• The context description activity is refined to identify the context information that could impact the decision in addition to describing the context information.

• As some criteria can be measured and evaluated, we reformulated the activity to estimate the criteria to include evaluations and not only estimations as described in the original proposal [193]. For example, scalability and functionality of an existing component can be evaluated. However, time to build a new component may potentially be estimated using some form of cost estimation model. In addition, the impact of the decision, cost-benefit analysis and changes that need to be made in order to incorporate the new decision are added to this activity.

• We also noticed that the tentative decision implemented in the form of prototypes and proof-of-concepts are more reliable and the decision-makers are more confident about their decision when they implement prototypes and proof-of-concepts. Thus, we have added these types of evaluations to the tentative decisions.

In both cases the decision was not documented. Documenting the decision and storing it in a repository allows information to be reused from previous similar decisions. As we can see a shift in the way software is development, reusing existing components is becoming increasingly common. Hence, as more such decisions to select an asset origin will be made, documenting such decisions will create a rich repository that can be used for further decisions.

7.2 Checklist v2.0 validation and refinement into process-line (Checklist v3.0)

We mainly changed the representation of the solution to eliminate a specific order between each activity. The following changes have been made –

• The activities are divided into three phases: preparation, investigation and decision-making.

• The information flow between the phases and repository is made explicit.

• The variation in execution and dependencies between activities and phases is also represented as shown in Figure 43.
The development of today’s software products, systems and services is far from trivial and often result in rather complex decision support models and decision-making processes. The decisions of choosing software components from different origins, such as in-house development vs. COTS, OSS and outsourcing, are most often strategic and have significant consequences on competitiveness. The decision-making process-line presented in this chapter provides a starting point for supporting such decisions and addresses the research gap identified in a recent systematic literature review [Chapter 2].

The presented decision-making process-line for selection of software asset origins can be applied for both B2B and B2C contexts as long as relevant stakeholders are identified and involved in decision-making. For B2C contexts, end users and other external stakeholders need to be involved and accurately represented.

In future work, we plan to further evaluate the generalizability of the decision-making process-line to support different levels of decisions, i.e. strategic decisions (to select between different asset origins), tactical decisions (to select between different suppliers or different components) and selection between different services. In addition, we plan to survey several business scenarios that involve diverse business models, asset origins, company characteristics and ecosystem participation models. We aim at clearly outlining short- and long-term consequences of each variation of activity execution. These should form guidelines that software business practitioners may use when considering various sourcing options.

We plan to implement the process-line in practice and evaluate the implementation as part of future work. The applicability of the process-line is important to evaluate through a large survey. A survey to collect the feedback on the applicability of the process-line is planned as future work. Moreover, we plan to expand our research on the evidence-based knowledge repository in the following ways: to create the first implementation of a repository that can support decision-makers and to create tool support for executing the process and storing the data in the knowledge-based repository. Finally, we plan to conduct an empirical study to evaluate the use of presented decision-making process-line to formulate specific decision-making processes and identify future work directions.
Acknowledgments

The work is supported by a research grant for the ORION project (reference number 20140218) from The Knowledge Foundation in Sweden. We would also like to thank our colleagues in the ORION project for fruitful discussions. Finally, we would like to thank the reviewers for valuable input that have helped improving the chapter.
Abstract

Systematic literature reviews in software engineering are necessary to synthesize evidence from multiple studies to provide knowledge and decision support. However, synthesis methods are underutilized in software engineering research. Moreover, translation of synthesized data (outcomes of a systematic review) to provide recommendations for practitioners is seldom practiced. The objective of this Chapter is to introduce the use of Bayesian synthesis in software engineering research, in particular to translate research evidence into practice by providing the possibility to combine contextualized expert opinions with research evidence. We adopted the Bayesian synthesis method from health research and customized it to be used in software engineering research. The proposed method is described and illustrated using an example from the literature. Bayesian synthesis provides a systematic approach to incorporate subjective opinions in the synthesis process thereby making the synthesis results more suitable to the context in which they will be applied. Thereby, facilitating the interpretation and translation of knowledge to action/application. None of the synthesis methods used in software engineering allows for the integration of subjective opinions, hence using Bayesian synthesis can add a new dimension to the synthesis process in software engineering research.

Keywords

Bayesian, evidence-based decision-making, synthesis, knowledge translation.

1 Introduction

The outcome (knowledge) of SLR should be useful for practitioners [25, 96]. It should be translated into recommendations that can enable and support evidence-informed decision-making in SE practice [25].
The fourth step of the five-step process for adapting the practices of Evidence-Based Software Engineering (EBSE) is referred as KT [25, 96].

Greenhalgh and Wieringa [68] argue that “objective, impersonal research findings” are unhelpful. Therefore KT should not be viewed as just supplying the outcomes of an SLR to professionals. Instead, it should be considered as a research activity involving researchers, subjective opinions of practitioners and policy-makers/decision-makers to make evidence-informed decisions [96].

KT in SE is defined as “the exchange, synthesis and ethically sound application of knowledge - within a complex system of interactions between researchers and users - to accelerate the capture of the benefits of research through better quality software and software development processes. [96].

Budgen et al. [25] state that KT in SE is done in an ad-hoc manner and lacks adequate documentation. In addition, highlights that “KT should itself be systematic and repeatable as possible, and it should also reflect the needs and mores of practitioners as well as of the different forms of organizational context within which they work” [25, 96].

The need to develop guidelines for undertaking KT in SE has been identified [25]. The aim of this Chapter is to adopt/adapt Bayesian approaches to synthesis used in health research to provide guidelines to KT. Though Bayesian approaches are referred as synthesis methods [53], they are essentially synthesis methods extended to support KT. They synthesize data and provide interpretation of the outcome in the application context by incorporating knowledge and experience of intended users (therefore supporting KT). As mentioned earlier, synthesis is not a separate activity as per the definition of KT in SE. The focus of this chapter is on synthesis process that facilitates KT rather that the KT activity itself.

In health research, Bayesian synthesis has been used to provide decision support by incorporating both subjective opinions of decision-makers and evidence/knowledge [141]. Bayesian synthesis is particularly useful when there is not enough evidence to confirm its suitability and the decisions need to be taken nevertheless in a reasonable and informed way [141]. Another advantage of Bayesian synthesis is that it takes the potential user of the analysis i.e. the practitioners’ or decision-makers’ and policy-makers’ perspective into consideration [141]. It is more flexible and efficient in using evidence from all available sources [161] and can synthesize findings from methodologically diverse studies [44]. Hence, the Bayesian approach is an attractive method to synthesize evidence and support KT as it provides interpretations of what
evidence mean in a particular context by incorporating subjective opinions of the potential users of synthesized data.

Bayesian synthesis starts with a subjective opinion, and these opinions are updated based on the evidence available. Although conceptually it seems straightforward, it is not easy to implement, and some of the methodological issues remain unresolved [53]. Bayesian synthesis methods have been used in health research to synthesize primary studies [150]. However, it has not been implemented in SE research.

In this chapter, we show how Bayesian approaches can be used to synthesize evidence and make use of the knowledge in practice. We also demonstrate the working of Bayesian approaches to synthesize and KT through examples in SE research.

The remainder of the chapter is structured as follows. Section 2 describes related work and Bayesian approaches. We then provide the method description of Bayesian to synthesize and KT in the context of SE in Section 3. The Bayesian synthesis is illustrated in Section 4. Finally in Section 5 we compare our method with the alternatives and conclude in Section 6.

2 BACKGROUND AND RELATED WORK

We begin by providing a brief overview of the working of Bayesian principles in Section 2.1 (note that only the description of Bayesian principles based on which the Bayesian synthesis method is developed is provided here. The description of the Bayesian synthesis method for knowledge translation is provided in Section 3). In Section 2.2, we describe how Bayesian synthesis has been implemented in health research so far. In Section 2.3, we discuss how synthesis is done in SE research. Although Bayesian methods are not used for synthesis in SE, it has been used in other ways which is discussed in Section 2.4.

2.1 Overview of Bayesian synthesis

It is important to get an overview of how Bayesian theory applies to synthesis to understand the working of Bayesian synthesis. The basic idea of Bayesian synthesis is described in health research [141, 161]. A brief summary of the method is provided below:

1. **Prior probability** - State a subjective opinion based on personal experience, excluding evidence.
2. **Likelihood** - Evaluate the evidence obtained from primary studies.

3. **Posterior probability** - Combine the prior probability and likelihood to produce a final opinion.

### 2.2 Bayesian synthesis in health research

Bayesian meta-analysis is a Bayesian approach used in health research that suits the EBSE requirement of incorporating prior knowledge and experience into the synthesis [53]. Roberts et al. use Bayesian meta-analysis to synthesize evidence from eleven qualitative and thirty two quantitative primary studies [150]. The study also incorporated subjective opinions of five experts. The subjective opinions and data collected from qualitative studies were used to form the prior probability. The range of prior probabilities is known through qualitative data. Hence, it is called as an informative prior probability where the range of probabilities (uncertainties) is narrow. However, we believe that the prior probability should be purely subjective based on personal experience and elicited before acquiring additional information. This allows tracking of how subjective opinions differ from collected data and how the subjective opinions are altered/refined based on the collected data. The prior probability is then updated through data collected from quantitative studies.

Two approaches called “quantitizing” [182] and “qualitizing” [44] have been proposed in health research. However, these approaches do not address the EBSE recommendation of incorporating subjective opinion into the synthesis hence, not discussed further.

In summary, the Bayesian meta-analysis method allows to incorporate subjective opinions. Although, it seems suitable to be used in SE it must be adapted so that the subjective opinion is unbiased.

### 2.3 Synthesis and KT in SE

A study was conducted to evaluate how evidence is synthesized in SE research [47]. The findings of the study show that limited attention is paid to the synthesis of evidence in SLRs. While 41% of studies did not report following any synthesis methods, among the studies that reported using synthesis methods thematic analysis (22.6%) and narrative synthesis (16.1%) were most used [47].
KT is not widely practised in SE. According to a tertiary study [24], among the 143 SLRs reviewed, only few provided recommendations [25]. However, the recommendations were produced by experts and not by incorporating the subjective opinions of potential users [25]. In addition, the recommendations were generated in an ad-hoc manner without following a systematic method or guidelines. Though, knowledge translation activity is discussed and defined in software engineering, there are no methods or guidelines for undertaking knowledge translation in software engineering [25, 96].

2.4 Bayesian in SE

Even though Bayesian theory is not yet used for synthesizing evidence in SE research, Bayesian networks have been applied to address various SE research problems. Approximately 72% of Bayesian networks applications are in the software quality (46.15%) and SE management (26.5%) areas [127]. The use of Bayesian networks for evidence-based decision-making in SE has been discussed in [127]. Three Bayesian network models are proposed to predict software reliability [127]. The study also claims that the use of Bayesian networks is not well recognized in SE research as compared to other disciplines such as health research [127]. Even though Bayesian networks are used for decision-making in SE, it has been used to analyze data from single studies. For example, Bayesian networks are used to represent the software life cycle phases by incorporating expert judgement (collected from qualitative surveys) into quantitative data collected from software repositories [127]. The use of Bayesian synthesis to synthesize evidence from multiple studies has not been implemented yet.

3 Bayesian Synthesis for KT - Method Description

In this section, we describe how Bayesian synthesis can be used for synthesizing evidence and KT. The Bayesian synthesis proposed in this chapter is adopted from health research. It consists of three main steps as depicted in Figure 44.

The Bayesian synthesis starts with prior probability which consists of subjective opinions based on personal experiences on a particular topic. Thereafter, the data is extracted from the studies to understand what is known or what is already been studied about the topic. Likelihood is the probability that the extracted data is true. Now that the prior probability and likelihood are known, the posterior probability is formu-
lated by refining the prior probability given the likelihood. The steps involved in Bayesian synthesis are described in following sections.

3.1 Step 1: The prior probability

This step includes the following two sub steps:

1. Selecting individuals: Sampling of individuals

2. Eliciting opinions: Capturing subjective opinions.

The advantage of Bayesian synthesis is that it allows incorporating subjective opinions and beliefs into the synthesis. In this way, the synthesis is not limited to the observed or collected data. Before collecting the data, the subjective opinions are captured.

Selecting individuals: In SE, the subjective beliefs of practitioners, decision-makers/policy-makers are relevant. The selection of individuals who will be the users’ of the knowledge is important. For example, if a decision needs to be made in a software project then, all the practitioner roles that should be involved in making the decision should be selected to elicit their subjective opinions.

Elicit opinions: Opinions are elicited to collect prior probability. Prior probability can be captured in terms of percentage ranging from 0 to 100 % or in terms of absence/presence (0/1) of a parameter value. One of the advantages of Bayesian synthesis is that it is flexible. Hence, the prior probability can be formulated in a way that suits the research objective. Spiegelhalter et al. [161] state that there is no “correct” prior and that Bayesian synthesis should be seen as a means of transforming prior into posterior opinions, rather than producing the posterior probabilities. The subjective opinions form the prior probabilities is represented as P(parameters).
3.2  *Step 2: The likelihood*

Likelihood is the representation of what is known. In other words, it is the summary of all the research studies within a specific research objective. For example, likelihood is the outcome of a SLR.

This step includes the following two sub steps:

1. **Data extraction**: Extract relevant data from the primary studies

2. **Likelihood calculation**: Calculate the likelihood of the data for the given parameters

**Data extraction**: The extracted data can be summarized in a table format where, each extracted parameter is the column name and each row represents the results (parameter value) from a single primary study. The parameter values entered in the table depends on the research objective and the available data. For example, if the research objective is to find factors that affect the decision. Then, the factors from the primary studies should be extracted. The studies that report the factor can be represented as 1 and the studies that do not report the factor can be represented as 0. However, if the research objective is to identify the factors that affect the decision positively and the factors that affect negatively or have no effect. Then the positive effect can be represented as 1, negative as 0 and no effect as 0.5. The extraction also depends on the data that is available. If the primary studies include statistical analysis such as odds ratio of the factors then odds ratios are extracted that provide the magnitude of the factors and not just the presence or absence of the effect. Similarly for quantitative (experimental) studies, the effect sizes must be extracted.

The extracted data can be organized based on the evidence provided to support the results or based on the data type. The division of extracted data depends on what is extracted from the primary studies. If the extracted data is of different data types, then it is recommended to divide the likelihood based on the data type i.e. qualitative and quantitative. It will allow to aggregate similar data together. However, if the extracted data from all studies is of same data type then it is recommended to divide the likelihood based on the evidence supporting the results. This will allow to interpret the likelihood based on the evidence provided in the primary studies.

The difference between empirical and non-empirical studies might be more relevant to analyse in SE. Empirical studies might have more importance as they are based on empirical evidence. Thus, we suggest dividing studies into non-empirical and empirical studies in SE instead
of qualitative and quantitative studies. However, as Bayesian synthesis is flexible, it allows to separate the likelihood based on qualitative and quantitative studies or empirical and non-empirical studies as illustrated in Figure 45.

![Likelihood Calculation Diagram](image)

**Figure 45: Division of likelihood calculation**

**Likelihood calculation:** The likelihood is written as the function of the observed data for the given parameters represented as \( P(\text{data} | \text{parameters}) \). In other words, the proportion of studies reporting the parameter to be true. For example, if 5 out of 10 studies report the parameter then, it is 50% \([5/10]*100\) likely that the observed data is true to the population. Higher values of likelihood indicates that the observed data is more likely to occur.

The likelihood calculation can be divided for better interpretation and analysis of the observed data as shown in Figure 45. Based on how the extracted data is divided, the likelihood can be calculated for each division. There is further diversity in the results with respect to the context and the quality in the way the study has been conducted. The importance of reporting context details in SE research has been identified \([55, 139]\). Hence, it is advisable to use the context information and separate the likelihood based on the context and quality assessment. Such representation of likelihood provides a detailed analysis of the collected data and provides a richer input to the next step which is discussed in Section 3.3.
3.3 Step 3: The posterior probabilities - refining prior probabilities

Posterior probability is the refinement of the prior probability given the likelihood. The equation of posterior probability is stated in [44] as shown in Equation 4.

\[
P(\text{parameters}|\text{data}) = \frac{P(\text{data}|\text{parameters})P(\text{parameters})}{P(\text{data})}
\]

(4)

Where,

\(P(\text{data}|\text{parameters})\) is the likelihood and \(P(\text{parameters})\) is the prior probabilities.

However in our approach we do not follow a mathematical approach in the calculation of posterior probability as it may ignore some of the interpretations. For example, practitioners might want to refine the probabilities according to the context and the quality assessment of the primary studies and their individual opinions. Therefore, the practitioners who state the prior probability will refine their probabilities based on their interpretation of the observed data and likelihood calculation.

The posterior probability is the combination of prior probability and likelihood. The posterior probabilities are formulated step-wise by refining the prior given the likelihood. The step-wise formulation is based on how the likelihood calculation is separated. In the first step, the prior probability is refined when the likelihood of first division of papers is available and in the next step, when the likelihood of second division of papers is available. This final probability is regarded as the posterior probability. The prior probability and likelihood can be combined independently by each individual or collectively by discussing the differences in prior probability and a common interpretation of likelihood.

If the posterior probability is to be formulated with a common prior probability and common understanding of likelihood then, internal and external conflicts should be resolved. Internal conflicts refer to the conflicts between the individual opinions and external conflicts refer to the conflicts in interpretation of likelihood.

**Discussing internal conflicts:** Once the prior probability of each individual is known, the individuals discuss the differences in the probabilities and try to resolve conflicts. The differences in the probabilities could be due to the differences in the individual roles. In some cases it is important to resolve the differences rather than proceeding with the different probabilities. For example, a developer might not see the same
factors to be important in the decision as the architect. The developer might not be aware or might not have encountered similar experience as the architect. The differences in the probabilities are discussed until they reach consensus. If the differences are due to lack of knowledge then the individual probabilities should be refined.

Spiegelhalter et al. have summarized the different strategies to refine probabilities from different individuals [161]:

1. Elicit a consensus: The diverse probabilities of all the individuals are brought into consensus using either informal or formal Delphi methods. During the process of eliciting consensus all individuals should be given equal importance. This way the dominant individuals’ influence will not impact opinions of others. Once the prior probabilities are refined and are similar, an average of the probabilities is taken.

2. Calculate a pooled prior: A simple average of all individual priors is taken. In this case the individual prior probabilities are not similar before taking the average. If the prior probabilities represent the absence or presence of a factor then the proportions of the practitioners mentioning the presence or absence is calculated. For example if 3/6 practitioners mention that the presence of factor X affects adherence then the combined probability is 50%.

_Discussing external conflicts_: The individuals who assigned the prior probabilities discuss how to interpret the extracted data and likelihood calculation. Particularly, how much importance should they give to the extracted data and likelihood calculation and how they should refine their probabilities. The context and quality are also taken into consideration. For example, the individuals might decide to only consider the high quality studies from a particular domain. In this way the external conflicts based on the interpretation of likelihood are resolved.

We propose four approaches to discuss internal and external conflicts as follows:

1. Discuss internal and external conflicts
2. Discuss only internal conflicts
3. Discuss only external conflicts
4. Discuss only at the end

The working of the four approaches is depicted in Figure 46.
Steps
Step 1: Prior probability
  Select individuals
  Elicit opinions
Step 2.1: Elicit opinions or calculate pooled prior probability
Step 2.2: Retain individual prior probabilities
Step 3: Likelihood
  Extract data
  Calculate likelihood
Step 4: Discuss extracted data and likelihood calculation
Step 5: Formulate posterior probability
Step 6: Elicit opinions or calculate pooled posterior probability

Approach 1
1
2.1
3
4
5
2.2
3
4
5
6

Approach 2
1
2.1
3
4
5
2.2
3
4
5
6

Approach 3
1
2.1
3
4
5
2.2
3
4
5
6

Approach 4
1
2.1
3
4
5
2.2
3
4
5
6

Figure 46: Four different approaches to refine prior probabilities

**Approach 1: Discuss internal and external conflicts:** In this approach, both the prior probabilities and likelihood are discussed. The individual prior probabilities are combined by either eliciting consensus or calculating pooled prior probability. The common prior probability will be updated based on common understanding of the likelihood.

**Approach 2: Discuss only internal conflicts:** In this approach the prior probabilities are combined by either eliciting consensus or by calculating pooled priors. However, the likelihood is not discussed and the individuals do not have a common interpretation of the likelihood. The common prior probability will be updated based on individual understanding of the likelihood. Therefore, each individual formulates posterior probability by refining the common prior probability based on their individual understanding of the likelihood. This approach can be adopted when the differences in interpreting the evidence are important.

**Approach 3: Discuss only external conflicts:** In this approach the prior probabilities are neither discussed nor combined. In other words,
the diversity of opinions is retained. Only the likelihood is discussed i.e. all the individuals would have a common understanding on how to interpret the collected data and update their individual probabilities. Therefore, each individual formulates posterior probability by refining their individual prior probability based on the common understanding of the likelihood. One possible reason to not discuss or combine probabilities could be that all the practitioners have the same role and there are no confounding factors affecting the opinions. In this case the differences become relevant to capture.

**Approach 4: Discuss only at the end:** In this approach neither the prior probabilities nor the collected data and likelihood calculation are discussed. The individuals independently refine their probability without discussing their individual opinions and the likelihood resulting in individual posterior probabilities.

Based on the approach used to resolve conflicts, the posterior probability is formulated by refining either the pooled or individual prior probability based on the common or individual understanding of likelihood. In approach 1 a common posterior probability is formulated. However, in approaches 2, 3, and 4 individual posterior probabilities are formulated. The individual posterior probabilities are then combined by discussing the differences by eliciting consensus or by taking an average.

## 4 Illustration

In this section we present an example to show the working of Bayesian synthesis for KT. In this example, the posterior probabilities are computed using approach 1 (described in Section 3.3). The research problem is the decision to choose between in-house development and acquiring OSS components for building software systems. The objective is to identify the factors that impact the decision.

### 4.1 Step 1: Prior probability

**Selecting individuals:** As the research problem is related to decision-making in software engineering practice, the decision-makers’ experiences and opinions in practice are relevant. However, in this example the prior probability values are hypothesized values provided by the authors of this chapter for illustration purpose. Thus, the authors are acting as practitioners/decision-makers. In a real context, actual decision-makers’ experiences and opinions should be considered.
Eliciting opinions: The decision-makers provide the factors that they think impact the decision to choose between in-house development and OSS. Table 52 represents the probabilities assigned by the practitioners (in this case, the authors). The probabilities assigned by the decision-makers are independent and are solely based on their personal experience and opinion without any information from the scientific research. If the decision-makers state that the factor is important in the decision, then, the value 1 is assigned. The value 0 indicates that the decision-maker has not mentioned the factor being important. Since the decision-makers only indicate if they think a factor is important of not, the values are in binary form i.e. either 1 or 0. Depending on the role (practitioner’s perspective) some of the factors may or may not be considered as important factors.

Table 52: Prior probabilities - decision-makers’ experience/opinion

<table>
<thead>
<tr>
<th>Decision-maker Role</th>
<th>Time</th>
<th>Cost</th>
<th>Effort</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manager</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Developer</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Architect</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Integrator</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Tester</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

4.2 Step 2: Likelihood

Likelihood is the representation of what is known i.e the representation of the evidence from the literature. The primary studies mentioned is this example are a sub-set of the primary studies considered in the SLR reported in Chapter 2. Note that the outcome of the SLR reported in Chapter 2 is not translated in recommendations. However, it is regarded as future work to translate knowledge in recommendations for practitioners using Bayesian synthesis for KT.

Organise extracted data: The extracted data regarding the factors that impact the adoption decision are organised as shown in Table 53. The primary studies consist of empirical and non-empirical studies. The non-empirical studies are mostly opinion, experience or philosophical papers (based on classification proposed be Wieringa et al. [186]). Personal opinions, views or experience are more focused on individ-
ual researchers or is project specific. Whereas empirical studies such as case studies and surveys are more generalized and are based on stronger evidence. The value “1” in Table 53 represents that the factor has been mentioned in a primary study, value “0” represents absence of the factor. Since the data extracted from all the studies is of same data type, we organized the extracted data based on empirical and non-empirical research types. In the empirical studies we found that some of the factors were mentioned but the conclusion was that the factor did not have any significant effect. We still record this evidence as the factor has been mentioned as they are validating a possible myth. Hence this is important to be considered in the synthesis. Such factors that are mentioned as not having any effect are assigned the value 0.5. Three new factors are identified by the papers which were not identified by the practitioners in step 1.

Table 53: Organisation of extracted data from non-empirical and empirical studies

<table>
<thead>
<tr>
<th>Ref.</th>
<th>T</th>
<th>C</th>
<th>E</th>
<th>Q</th>
<th>TS</th>
<th>L</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Non-empirical Papers

<table>
<thead>
<tr>
<th>Ref.</th>
<th>T</th>
<th>C</th>
<th>E</th>
<th>Q</th>
<th>TS</th>
<th>L</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>[131]</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>[178]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>[152]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Empirical Papers

<table>
<thead>
<tr>
<th>Ref.</th>
<th>T</th>
<th>C</th>
<th>E</th>
<th>Q</th>
<th>TS</th>
<th>L</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>[73]</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>[165]</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>[36]</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

T: Time, C: Cost, E: Effort, Q: Quality
TS: Technical support, L: License, R: Requirements

Likelihood calculation based on empirical and non-empirical studies: In this illustration the likelihood calculation is divided into empirical and non-empirical studies. The likelihood is the proportions of the primary studies reporting the factor being important for the decision. The context and quality of the primary studies are as mentioned in Table 54. The likelihood for each factor is calculated by considering the percentage of primary studies that have mentioned the factor as an important factor in making adoption decisions. For example, four primary studies have indicated that the technical support (TS) factor is an
important factor. Hence, the likelihood for technical factor (TS) equates to 67% (\(\frac{4}{6} \times 100\)). As seen in Table 55, the first row represents the total likelihood and the likelihood is separately calculated for empirical and non-empirical studies which is represented in the following two rows. The total likelihood of cost (C), technical support (TS) and license (L) is the same. However, when the likelihood is separated based on evidence, we see that all the empirical studies report cost being important and all the non-empirical studies report technical support and license being important. Based on this separate calculation the total likelihood of 67% may be interpreted differently based on the evidence supporting the factors.

It is recommended that the likelihood is further divided based on the context and quality, this can be done by the rigor and relevance assessment method proposed by Ivarsson and Gorschek [85] during data extraction while conducting SLRs. However as seen in Table 54, at most one paper is supporting the same domain hence, the likelihood is not divided further based on context. In addition, since the quality of all empirical studies is high and non-empirical studies is low, the likelihood calculation would be as listed in Table 55. However, in a real case contextual factors may be viewed as very important when rating the outcome from different studies and hence, having a big impact on the likelihood.

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Context</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>[131]</td>
<td>Mission-critical domain</td>
<td>Low</td>
</tr>
<tr>
<td>[178]</td>
<td>Not mentioned</td>
<td>Low</td>
</tr>
<tr>
<td>[152]</td>
<td>Not mentioned</td>
<td>Low</td>
</tr>
<tr>
<td>[73]</td>
<td>Telecommunication domain</td>
<td>High</td>
</tr>
<tr>
<td>[165]</td>
<td>Not mentioned</td>
<td>High</td>
</tr>
<tr>
<td>[36]</td>
<td>Multi domain</td>
<td>High</td>
</tr>
</tbody>
</table>

4.3 Step 3: Posterior probability - Refining prior probability

We have the probabilities of the decision-makers as shown in Table 52. Following approach 1, the differences in the probabilities of the decision-makers should be resolved before continuing. Hence, a pooled prior is calculated by considering the percentage of decision-makers
Table 55: Likelihood based on empirical and non-empirical research papers

<table>
<thead>
<tr>
<th>Likelihood</th>
<th>T</th>
<th>C</th>
<th>E</th>
<th>Q</th>
<th>TS</th>
<th>L</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>8%</td>
<td>67%</td>
<td>17%</td>
<td>33%</td>
<td>67%</td>
<td>67%</td>
<td>50%</td>
</tr>
<tr>
<td>Non-empirical</td>
<td>0%</td>
<td>33%</td>
<td>0%</td>
<td>33%</td>
<td>100%</td>
<td>100%</td>
<td>66%</td>
</tr>
<tr>
<td>Empirical</td>
<td>16%</td>
<td>100%</td>
<td>33%</td>
<td>33%</td>
<td>33%</td>
<td>33%</td>
<td>33%</td>
</tr>
</tbody>
</table>

T: Time, C: Cost, E: Effort, Q: Quality
TS: Technical support, L: License, R: Requirements

that have mentioned the factor as an important factor in making adoption decisions. For example, as seen in Table 52, three decision-makers have indicated that time is an important factor. Hence, the prior probability for time (T) equates to 60% (\(\frac{3}{5}\) *100). The first row (Prior probability) in Table 56 represents the pooled prior probabilities computed from values in Table 52. The second and forth rows (Non-empirical likelihood and Empirical likelihood) are the same as the second and third rows of Table 55. The third and fifth rows (Refined probability and Posterior probability) are new rows related to the formulation of posterior probabilities which are discussed below.

We have the probabilities of the decision-makers and likelihood from the non-empirical papers and empirical studies. The likelihood of empirical and non-empirical studies (Table 55) is provided to the decision-makers along with the context and quality of the papers (Table 54). Once the decision-makers receive this information, they either decide to update (either increase or reduce) their prior probability or stay with their initial prior probability. The prior probabilities are updated step-wise, once after the data from non-empirical studies is available (third row in Table 56 - Refined probability) and later when data from the empirical studies is available (fifth row in Table 56 - Posterior probability). It depends on how the decision-makers react based on the additional information provided. Not all decision-makers will have the same interpretation of the likelihood (data from empirical and non-empirical studies). As we use approach 1 to formulate posterior probability, any conflicting interpretations should be resolved until consensus is achieved.

For example, one of the decision-maker might decide to lower the probability of cost (third column in Table 56 - C) based on the non-empirical likelihood (second row in Table 56 - Non-empirical likelihood) since only \(\frac{1}{3}\) (33%) non-empirical studies (paper [131] as seen in Table 53) has reported this factor. However, other decision-makers
might not change the probability as they decide to give more importance to paper [131] as it is in the same domain (Table 54) as the decision-makers. Hence, the decision-makers discuss if they should consider the overall likelihood of the effort factor (C) or only focus on the studies that are within the same domain. The decision-makers collectively decide not to lower the initial prior probability. Hence, the refined probability (third row in Table 56 - Refined probability) is same as the first row in Table 56 - (Prior probability). In the next step when the decision-makers look at the likelihood from empirical studies (forth row in Table 56 - Empirical likelihood), all the empirical studies have reported cost as an important factor. Hence the decision-makers collectively decide to increase the probability as shown in the fifth row in Table 56 - Posterior probability. The knowledge and experience of decision-makers should facilitate in the interpretation of the data. If consensus is not achieved, the average of the probability should be considered. The refined probabilities are as shown in Table 56. Note that in this illustration the authors refined the probabilities however, in a real context the decision-makers should refine the prior probability. The arrows indicate the change from prior probabilities, i.e. either the prior probability is increased or decreased.

Table 56: Revised probabilities

<table>
<thead>
<tr>
<th></th>
<th>T</th>
<th>C</th>
<th>E</th>
<th>Q</th>
<th>TS</th>
<th>L</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior probability</td>
<td>60%</td>
<td>80%</td>
<td>40%</td>
<td>60%</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Non-empirical likelihood</td>
<td>0%</td>
<td>33%</td>
<td>0%</td>
<td>33%</td>
<td>100%</td>
<td>100%</td>
<td>66%</td>
</tr>
<tr>
<td>Refined probability</td>
<td>50%</td>
<td>80%</td>
<td>40%</td>
<td>50%</td>
<td>100%</td>
<td>100%</td>
<td>66%</td>
</tr>
<tr>
<td>Empirical likelihood</td>
<td>16%</td>
<td>100%</td>
<td>33%</td>
<td>33%</td>
<td>33%</td>
<td>33%</td>
<td>33%</td>
</tr>
<tr>
<td>Posterior probability</td>
<td>20%</td>
<td>90%</td>
<td>40%</td>
<td>30%</td>
<td>75%</td>
<td>75%</td>
<td>60%</td>
</tr>
</tbody>
</table>

T: Time, C: Cost, E: Effort, Q: Quality
TS: Technical support, L: License, R: Requirements

Table 56 highlights the importance of including non-empirical studies in the synthesis. If non-empirical studies were not included then the factors: technical support (TS), license (L) and requirements (R) would not have been considered as important in the decision-making process. If the decision-makers took the decision without considering the evi-
evidence then they would have only considered four out of seven factors. And only one among the four factors: i.e. cost (C) has high probability. The three new factors have high probability which means that if they were ignored it would have a significant impact on the decision. The non-empirical studies have the most impact on the high probabilities, hence, it indicates the importance of non-empirical studies. Non-empirical studies are criticized as they lack rigor and relevance. However, since decision-makers themselves are involved in the synthesis they can validate the information from the non-empirical studies. Involving the decisions-makers in the synthesis allows to interpret the synthesized data (SLR outcomes) in the application context and facilitates in providing recommendations for adapting the outcomes in practice. The examples of recommendations that can be provided to the decision-makers are as follows:

- Time is not as important as it is perceived by decision-makers. As likelihood suggests [73] that the time saved in developing in-house might not result in overall reduction in time as the selection and integration of OSS consumes the time saved by not developing in-house. The decision-makers accept and agree to the likelihood hence, we can say that time is not an important criteria for the decision.

- External factors such as technical support provided and license obligations are important decision criteria. Initially the decision-makers do not mention external factors as they might be overseen. However, once they receive the likelihood, they agree that external factors could potentially influence the decision.

Depending on the quality of the primary studies supporting the evidence and the experience of the involved practitioners, the recommendation can be regarded as strong or weak.

5 DISCUSSION

Bayesian synthesis takes into account a wide range of evidence, including subjective opinions into the synthesis. None of the synthesis methods used in SE research allows to incorporate subjective opinions. Bayesian synthesis is flexible and works well with other synthesis methods such as thematic analysis. The novelty of Bayesian synthesis is in the extension of traditional synthesis methods to support KT.

Bayesian synthesis gives the probabilities of the data being true as compared to inferential statistics that provides the probability of the
calculation of the result being true. In real life we are more close to Bayesian thinking. For example, we think about the probability of an event to be true instead of the probability of the computation being true.

Bayesian meta-analysis has been implemented in health research [150]. The Bayesian synthesis method proposed in this chapter is inspired from the Bayesian meta-analysis approach. The Bayesian meta-analysis method considers the subjective opinions and qualitative evidence together in the prior probabilities. In order words, the individuals know the information from qualitative studies before assigning prior probabilities. However, as we want unbiased opinions, we separate the qualitative evidence from the prior probability. The previous methods implemented in health research do not differentiate between the types of evidence [44, 150, 182]. It leaves it up to the researchers/practitioners to decide how they want to evaluate the evidence. However, we recommend providing detailed description in terms of summaries, context and quality information. This guides the researchers/practitioners to make informed decisions. Unlike previous Bayesian methods, no mathematical equations are used to compute the posterior probability. Instead, the individuals themselves refine the probabilities based on their experience. The four approaches for formulating posterior probabilities of SE research is also another novel contribution of the Bayesian synthesis proposed in this chapter.

6 CONCLUSIONS

We conclude by discussing the contributions and limitations of Bayesian synthesis. The synthesis methods analyze the results from multiple studies and supports KT. Bayesian synthesis goes beyond synthesis of only research evidence by actually guiding how the results can be applied or used in a particular context.

SLR is one of the research methods where multiple studies are synthesized. Often SLRs follow a rigorous approach to search, select and extract information from the primary studies. However, if the primary studies are weak or lack necessary descriptions then, it is difficult to make conclusions on a phenomenon and provide recommendations based on the outcome of an SLR. However, Bayesian synthesis, due to its ability to incorporate subjective opinions, is particularly useful when the data from primary studies is insufficient to draw generally valid conclusions [141]. Bayesian synthesis can be regarded as a KT process to interpret and translate outcomes of SLRs, particularly when
there are few primary studies or when there are differences in findings in the primary studies. In addition, Bayesian synthesis can be particularly useful for research problems related to decision-making where individual judgements and opinion can be combined with research evidence thereby making evidence-informed decisions. Developing detailed guidelines and evaluating Bayesian synthesis in practice (real context) is considered as future work.

**Limitations:** Capturing subjective opinions in the analysis allows better synthesis. However, it is a known fact that people are not good probability accessors, validity threats in eliciting opinions have been summarized by Kandane and Wolfson [92]:

1. **Availability:** Recent or easily recalled events might be given higher probability, and vice versa.

2. **Adjustment and anchoring:** The opinions are anchored at some starting point and tend to exert an inertia. The subsequent opinions might not be adjusted sufficiently and might be too close to the first probability.

3. **Conjunction fallacy:** A higher probability might be assigned to a parameter that is a subset of a parameter that has lower probability.

4. **Hindsight bias:** The opinion might be biased if the data is available before elicitation of the opinions.

These validity threats can be mitigated by using elicitation techniques such as interactive feedback with a structured interview. In case of any issues such as a conjunction fallacy, the person providing the probability can be asked to reflect more on the probability. Also by eliciting priors before providing evidence from qualitative and quantitative studies, unbiased probabilities can be obtained.

In this chapter, we have described the use of Bayesian synthesis for KT in SE. The use of Bayesian synthesis is illustrated using an example to illustrate the working and flexible use of the Bayesian synthesis for KT.
GUIDELINES FOR KNOWLEDGE TRANSLATION IN SOFTWARE ENGINEERING

Abstract

**Context** - Transfer of software engineering research into industrial practice is a challenge; academic articles are not the best enabler for industrial adoption of research. Furthermore, synthesis of knowledge from multiple research studies is needed to provide evidence-based decision-support for industry. However, translation of research outcomes (for e.g., results of a systematic review) to provide recommendations for practitioners is seldom practiced.

**Objective** - The objective of this chapter is to provide a knowledge translation framework in software engineering research, in particular to translate research evidence into practice by combing contextualized expert opinions with research evidence.

**Method** - We adopted the framework of knowledge translation from healthcare research, adapted and combined it with the Bayesian synthesis method for use in software engineering research and practice. We evaluated the outcome of the knowledge translation framework along with the effectiveness of the interventions undertaken as part of knowledge translation in two cases.

**Results** - The framework provided in this chapter includes a detailed description of each step of the knowledge translation in software engineering. In the evaluation of the knowledge translation framework we found that, in comparison to the prior opinions practitioners become more consensual after the integration of opinions and knowledge from research studies.

**Conclusions** - Knowledge translation using Bayesian synthesis provides a systematic approach towards contextualized, collaborative and consensus-driven application of research results. Dissemination of research results, transfer of research outcomes and sharing tacit knowledge has been done to some extent to provide explicit knowledge. However, there is limited research on how to translate research outcomes to make the research more contextualized and hence more easy, accessible and acceptable by industry. In conclusion, this chapter contributes towards the application of knowledge translation in software engineering through its framework and a first evaluation of its use.
1 INTRODUCTION

The knowledge produced by researchers should be applied in practice [25, 96]. It should be translated beyond the dissemination and towards the application of the knowledge in practice. Budgen et al. [25] suggest that knowledge should be presented in the form of recommendations that enable and support evidence-informed decision-making in software engineering (SE) practice. This is reflected in the fourth step of the five-step process for adapting the practices of evidence-based software engineering (EBSE) which is referred to as knowledge translation (KT) [25, 96]. A large survey at Microsoft indicates that the developers rely on and are more influenced by their “personal experience” than evidence from research [50]. Devanbu and Zimmerman identify this as a problem and suggest that greater efforts should be made to translate knowledge so that practitioners are informed and rely on verified evidence, rather than prior opinions which could be “biased, error-prone and spotty” [50].

Greenhalgh and Wieringa [68] argue that “objective, impersonal research findings” are unhelpful. In addition, research might not always be straightforward applicable, relevant and suitable for practice according to Cartaxo et al. [30]. Therefore, KT should not be viewed as just supplying the research outcomes to professionals rather aligning the knowledge to a specific context.

In addition, Devanbu and Zimmerman [50] point out that by not accepting research results software practitioners might be at risk of relying too much on their personal opinion and experience. They regard subjective and personal recollection to be notoriously error-prone [50].

It is clear that both objective and impersonal knowledge from research studies and subjective and personal opinions of practitioners together should be considered for impactful implementation of research results. Therefore, knowledge translation should be an integration of knowledge from research studies and practitioners’ opinions. However, it is not often practiced [30]. Kitchenham et al. consider KT as a research activity involving researchers, subjective opinions of practitioners and policymakers/decision-makers to make evidence-informed decisions [96].

What is KT? - KT in SE is defined as “the exchange, synthesis and ethically sound application of knowledge - within a complex system of interactions between researchers and users - to accelerate the capture of the benefits of research through better quality software and software development processes” [96]. The three main aspects of KT are “exchange”, “synthesis” and “application”. The Bayesian synthesis
method proposed in Chapter 6 supports KT as it synthesizes data and provides interpretation of the research outcome in the application context by incorporating knowledge from research studies and experience of intended users.

**What is not KT?** - Different processes may be undertaken after the completion of a research study, for example, dissemination, knowledge transfer and KT. Dissemination is spreading the knowledge in different scientific channels. Whereas, knowledge transfer is an activity of making knowledge explicitly available in a form that allows users to use explicit information that applies to them in their specific context. The key aspect that differentiates KT is that it focusses on the application of contextualized knowledge in practice. Therefore, it is beyond dissemination and making knowledge available (knowledge transfer) to practice. It is also different from technology transfer as it involves the subjective opinions of practitioners and it involves the process of actual use of knowledge in practice.

**Why KT?** - Based on the classic two-community theory, the two communities of academia and practice are not always synced due to differences in perspectives and cultures [66]. A survey indicates that developers’ knowledge is formed based on personal experience and opinion and far less on research results [50]. This is also identified by Ayala et al., they found a gap between decision-making processes proposed in research and the processes used in industry. [9]. As a result, knowledge produced in research is too rarely implemented in practice. This is even more problematic if the studies do not provide explicit recommendations or guidelines to practice. Very few secondary studies in SE provide recommendations for practitioners [24, 25].

Due to the differences in software engineering research and practice, the translation of knowledge into practice is a complex research activity and not always straightforward. Practitioners are often busy which makes it difficult for them to keep up with research. However, even if practitioners are presented with strong and reliable evidence, it is often difficult for practitioners to accept the evidence and agree that it applies to them [145]. The need to transform evidence to local opinion [145] and the need to put more effort and systematize the dissemination of research findings is identified [50]. Budgen et al. [25] state that KT in SE is done in an ad-hoc manner and lacks adequate documentation. In addition, they highlight that “KT should itself be systematic and repeatable as possible, and it should also reflect the needs and mores of practitioners as well as of the different forms of organizational context within which they work” [25, 96]. The need to develop guidelines for undertaking KT
in SE has been identified [25]. The aim of this chapter is to provide guidelines in the form of a knowledge translation framework using Bayesian synthesis.

The remainder of the chapter is structured as follows. Section 2 describes the background and the related work. We then provide the description of the KT framework using Bayesian synthesis in the context of SE in Section 3. The evaluation and reflections from application of KT in a case study are presented in Section 4. Finally in Section 5 we discuss our findings in relation to previous research and conclude in Section 6.

2 BACKGROUND AND RELATED WORK

Different approaches to support software practice have been proposed. We discuss different approaches from knowledge sharing to knowledge translation in this section.

2.1 Sharing tacit knowledge

Frameworks like the quality improvement paradigm (QIP) [12] and the experience factory [13] have been proposed to utilize the knowledge and experience of software practitioners. The QIP framework [12] consists of two cycles - control cycle and capitalization cycle. The control cycle is about project learning to provide immediate solutions to the problems for on-going projects. The capitalization cycle is about capturing experience and transfer learning from one project to other applicable projects. The five steps in the capitalization cycle are - analysis of project results, packaging and storing practitioners’ experience, characterizing and understanding the experience, setting goals for improvement and finally choosing processes, methods, techniques and tools for implementing the process improvement.

The experience factory [13] is a repository to store experiences and lessons learned from working in projects and providing the experience to applicable projects on demand.

2.2 Academia-industry collaborations

Studies point out that opinions based on personal experience might be biased, subjective and error-prone [145] and [50]. Therefore, relying just on personal opinions and experience might not be the best approach for practitioners. In addition, the importance of interaction between re-
searchers and practitioners to integrate evidence from research results (knowledge) and practitioners’ opinion and experience has been identified. Evidence-based software engineering guidelines emphasize such interactions [96].

Academia and industry collaborations are important for better utilization of research results and for guiding future research directions [61] and [133]. Initiation from both researchers and practitioners is required for effective research collaboration [61]. According to a study [133], regular and continuous collaboration among researchers and practitioners with major contribution from researchers leads to better implementation of research in practice.

Case studies have been identified as an effective research method for implementation of research in practice [157]. According to an experiment conducted in Microsoft [117], 71% of practitioners had a positive opinion on software engineering research studies. An updated study [33] on research published in the empirical software engineering and measurement conference, indicates that practitioners are receptive towards research. However, even though the research results are sound, it still might be difficult to convince practitioners to implement the results [145], [50].

2.3 Technology transfer

Technology transfer in software engineering has been recognized as an important activity [140]. Over the years it has been improved and refined to make it more collaborative and a technology pull rather than a technology push activity [64]. Gorschek et al. proposed a model that validates a candidate solution in academia followed by a static and dynamic validation in practice before releasing as a solution. A need to make the technology transfer activity more efficient and effective has been identified [123]. Mikkonen et al. emphasize the importance of co-creation and co-learning instead of a one-way transfer from research to practice. The KT framework proposed in this chapter is a complementary solution to the technology transfer model proposed in [123]. However, KT framework focuses on adaptation and translation of research results in general and not only transfer of a technology or a candidate solution.
2.4 Knowledge transfer and translation

To aid the implementation of research in practice, an approach to capture and share the tacit knowledge to make it explicit and widely available has been proposed by Cartaxo et al. [30]. They present and evaluate a template called evidence briefings to be used to summarize the findings from systematic reviews. Evidence briefings transfer the research results in an attractive format for practitioners. However, it does not translate the knowledge into the context and does not focus on application of the knowledge in practice. Although, evidence briefings can be used to facilitate knowledge translation. We have depicted the use of knowledge transfer (evidence briefings) in knowledge translation in Figure 47. The key difference in knowledge translation is that a relevance jury consisting of both researchers and practitioners discuss the relevance of the primary studies to identify relevant studies. The findings from these relevant studies is then aligned to the local context.

![Figure 47: The use of knowledge transfer (evidence briefings) in knowledge translation.](image-url)
In healthcare Bayesian approaches to integrate qualitative and quantitative evidence with opinions and knowledge have been proposed \([141]\) and \([161]\) and applied with some customization in \([140], [44], [150]\) and \([182]\). A customization and application of Bayesian synthesis to align external knowledge to local context by integrating opinions and evidence in software engineering has been proposed [Chapter 6]. Bayesian synthesis consists of three steps, in the first step the prior opinions of practitioners is collected. In the second step, the evidence from the research results is presented to the practitioners. This evidence could be presented in the format of evidence briefings. In the last step, the practitioners are asked about their opinion in the light of the knowledge from the research results. We have used Bayesian synthesis in our KT framework to align the external knowledge from research studies to practitioners local context.

3 Description of the KT Framework

3.1 Overview of the KT framework

In this section, we describe the KT framework which is adopted from healthcare research \([66]\). Figure 48, represents the overview of the knowledge to action activity. It consists of two main parts: knowledge creation (represented as knowledge creation funnel) and knowledge application (represented as knowledge translation cycle). The knowledge creation funnel shown in Figure 48 consists of the following steps -

- Knowledge inquiry - It is done through primary studies.
- Knowledge synthesis - It is done through secondary studies.
- Knowledge tools/products - It presents the knowledge in end-user friendly format based on the needs of the end-user (e.g. - evidence briefings \([30]\)).

As the knowledge passes through the knowledge creation funnel it gets more tailored to the end-users’ needs. The tailoring in the knowledge creation funnel is done by filtering the knowledge that could be applicable to the end-users. For example, the tailoring is done by filtering information that is not relevant to the users such as methodological details. It is different from the tailoring based on the context of knowledge application which is part of knowledge translation.

The knowledge creation funnel not only provides input to the KT cycle, it also accepts input from the KT cycle hence, depicted by the
Figure 48: Knowledge translation framework (Adapted from Graham et al. [66]).

double ended arrow (<==>) and labeled as “interface” between the knowledge creation funnel and KT cycle. This is further elaborated in Section 3.2 and Figure 49. The main purpose of the KT cycle is to implement knowledge in practice. As mentioned earlier, KT is a research activity that goes beyond dissemination and towards the application of the knowledge.

The KT cycle is related to the application of the knowledge/evidence. In Figure 48, the KT cycle is represented in relation to Bayesian synthesis [Chapter 6] and generic improvement paradigms such as the PDCA cycle (Plan-Do-Check-Act) [12]. Due to the uncertainty involved in software engineering practice, it is important to adapt the knowledge to the local context before it is implemented in practice. Therefore, it is important to customize the external knowledge and adapt it to the local context, this is mainly done through the Bayesian synthesis part of the KT cycle (discussed in detail in Section 3.2). Once, the contextualized knowledge is identified, the next step is to implement the contextual-
ized knowledge using the PDCA cycle. The implementation is done by assessing the current situation and identifying interventions that are necessary for knowledge implementation and then monitor and evaluate before sustaining it as an industrial practice.

3.2 Detailed working of the KT framework

![Knowledge Creation Cycle](image)

Figure 49: Knowledge creation cycle (Adapted from Straus et al. [169]).

The steps involved in producing primary and secondary studies or tools (within knowledge creation) are cyclic in nature. Therefore, the knowledge creation funnel in Figure 48 is represented as a cycle together with the knowledge translation cycle in Figure 49. As seen in Figure 49 the knowledge creation cycle and KT cycle are connected and provide input to each other. The entry point for the cycles depends on the need for knowledge creation or implementation indicated by an arrow head with a circle and labeled as “entry point”. Apart from these entry points the execution of the cycles can continue from the previous cycles as indicated by the arrows with a line and labeled as “switch between labels”. The arrows indicate the execution flow in the knowledge creation cycle and the knowledge translation cycle. Each cycle also has exit points indicated by a “square” which marks the end of the cycle. For example, publishing research is the exit step of the knowledge creation cycle.
Steps i-vi in Figure 50 are part of the knowledge creation cycle, Step vii is a switch which points either towards application of knowledge (Step viia) or a continuation in the knowledge creation cycle (Step viib). Since the focus on this chapter is not on knowledge creation we will focus on the KT cycle i.e. Steps 1-9 in Figure 51.

These steps can be further categorized into steps (1-4) to make knowledge contextualized using Bayesian synthesis and steps (5-9) to carry out the knowledge implementation. Step 9 is a switch, which points either towards continued application of knowledge (Step 9a) or towards the creation of the next generation research based on the lessons learned from implementing knowledge in practice (Step 9b).

Figure 51, depicts the detailed working of KT in connection with the use of Bayesian synthesis [Chapter 6].

**Entry point for knowledge translation - Need for knowledge implementation:** It is done by identifying the need for improvement in practice for which relevant knowledge from literature exists or has been created that could be used in practice. The identification of the need should be done together by researchers and practitioners. The root-cause analysis of the problems should be done to identify the real need for improvement. In order to implement evidence in practice, it is useful if there is a local champion and there is support from management. Once the topic is identified then the required knowledge is identified, reviewed and selected.
The description of Steps 1 to 9 in Figure 51 are as follows:

1. **Identify, review and select knowledge**: In this step, the knowledge that could be used in practice to address the need for improvement is identified, reviewed and selected. In addition, the quality, strength and applicability of the knowledge is determined so that the knowledge is correctly interpreted. The quality of the research studies can be evaluated using the rigor and relevance criteria [85] and the strength of knowledge/evidence can be determined using guidelines in [54] and [189]. It is important to determine if the knowledge is based on a general context or a specific context. A simple literature review or a systematic literature review can be conducted to identify, review and select knowledge.

2. **Elicit prior (Prior probability)**: In this step, the current state-of-practice in terms of probabilities are elicited from the relevant stakeholders. It involves the following two sub-steps:
• Selecting individuals - In SE, the subjective opinions of practitioners, decision-makers/policy-makers are relevant. The selection of individuals who will be the users’ of the knowledge is important. For example, if a decision needs to be made in a software project then, all the practitioner roles that should be involved in making the decision should be selected to elicit their subjective opinions.

• Eliciting opinions - Opinions and experiences are elicited to collect prior probability. Opinions and experiences of practitioners related to the area of improvement should be elicited. For example, if the test automation process needs to be improved then the practitioners’ experiences and opinions on working with test automation are to be elicited. Prior probability can be captured in terms of percentage ranging from 0 to 100 % or in terms of absence/presence (0/1) of a parameter value.

3. **Assess gaps:** In this step, the size and nature of the gap between the current and desired knowledge, skills, and outcomes are assessed. Quality indicators can be used to assess the gap. For example, what is the current and desired level to achieve the goal?

4. **Align external knowledge to local context:** In this step, individuals or groups of decision-makers go through the knowledge identified in Step 1 and determine the value, usefulness and validity of the knowledge in their context. Before knowledge from research studies is provided to the practitioners, the terminology used in the literature should be adapted to the language and terminology followed in the company. This adaptation can be done with the help of the local champion.

The knowledge can be summarized in the form of likelihood calculations. The likelihood is the representation of what is known. In other words, it is the summary of all the research studies within a specific research objective. The likelihood is calculated as the percentage of the research studies reporting a finding. For example, the likelihood of 60 % indicates that 6 out of 10 research studies are reporting a particular finding. More examples on how the likelihood can be presented is mentioned in our previous work [Chapter 6]. The likelihood calculation is not always the indication of the importance of a finding. For example, if the likelihood of Finding F1 is 80% and Finding F2 is 70%, it does not mean that F1 is more important. It only means that Finding...
F1 is more often researched. Therefore, the likelihood does not always indicate importance. The importance of a research finding needs to be judged by the practitioners, and hence findings from research may be seen as a checklist for contextualizing the knowledge related to different research findings.

5. **Assess barriers and facilitators to knowledge use:** In this step the barriers and facilitators that restrict or help implement knowledge are identified. The barriers and facilitators are useful in understanding the intentions of the practitioners. Barriers for knowledge use could be related to knowledge, attitudes, skills, habits or the like of the potential adopters according to Graham et al. [66]. The possible interventions for such barriers could be interactive educational workshops or training and documenting the knowledge in the form of company standards.

6. **Select, tailor and implement KT intervention:** Based on the identified barriers and facilitators the KT intervention for example training is selected, tailored and implemented in this step.

<table>
<thead>
<tr>
<th><strong>Knowledge use</strong></th>
<th><strong>Description</strong></th>
<th><strong>Examples of measures</strong></th>
<th><strong>Strategy for data collection</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Conceptual</td>
<td>Use of knowledge to change the levels of knowledge, understanding or opinions of the software practitioners.</td>
<td>Practitioners’ knowledge, attitudes and/or intentions to change.</td>
<td>Survey and/or interviews</td>
</tr>
<tr>
<td>Instrumental</td>
<td>Use of knowledge to change a software practice or practitioners’ behavior.</td>
<td>Adherence to recommendations (e.g. change in the testing process or adoption of an innersource strategy ).</td>
<td>Archival or document analysis</td>
</tr>
<tr>
<td>Persuasive</td>
<td>Use of knowledge as ammunition to attain specific power or profit goals. For example, use of knowledge to justify a decision.</td>
<td>Decision-makers’ or policy-makers knowledge</td>
<td>Survey and/or interviews</td>
</tr>
</tbody>
</table>
7. **Monitor use:** Once the knowledge is in use or has been applied, it should be monitored. To be able to monitor the knowledge use, it is necessary to know the type of knowledge use. Using the classification scheme by Straus et al. [169] we identify the use and monitoring of use in software engineering as shown in Table 57.

Persuasive knowledge use is related to the use of knowledge as a ammunition. Conceptual and instrumental use of knowledge refers to the use of knowledge to make changes in practice. The key difference between conceptual and instrumental use is that in conceptual use, knowledge can be used to make an informed decision but not change practice unlike instrumental use where the process is changed and standards are defined. Monitoring knowledge use is important to understand if the desired level of knowledge use is attained. If the knowledge use is not adequate then, the barriers and facilitators of knowledge use need to reassessed to understand the intentions of the practitioners’ to use the knowledge.

8. **Evaluate outcomes:** The evaluation is based on the impact of the knowledge (outcome evaluation) and the evaluation of the KT intervention used to make the knowledge available to all the potential users of the knowledge.

- **Evaluation of knowledge use outcomes:** The outcome of knowledge use could be on the process/organization level, practitioner lever or customer level. These outcomes are presented in Table 58.

- **Evaluation of KT intervention:** The effectiveness of the KT intervention selected in Step 6 could be conducted using quantitative and qualitative evaluation methods. Quantitative measure can be such as randomized trials or a interrupted time series method that indicate the effectiveness of the KT intervention for example, the testing process improved after all the involved stakeholders participated in the training workshop. At the same time, qualitative information such as feedback from workshop participants is important in evaluating the KT intervention. For example, the participants might have different level of knowledge, experienced participants might not benefit in the same way as participants with less experience. Such cost-benefit analysis can be done through qualitative evaluation of the KT inter-
### Table 58: Outcome of knowledge use

<table>
<thead>
<tr>
<th>Outcome</th>
<th>Description</th>
<th>Examples of measure</th>
<th>Strategy for data collection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process or organization level</td>
<td>Impact on process or organization of using or applying the knowledge.</td>
<td>Effectiveness and efficiency of the process (e.g., testing).</td>
<td>Project outcomes/artifacts, document analysis, survey and/or interviews</td>
</tr>
<tr>
<td>Practitioner level</td>
<td>Impact on practitioners of using or applying the knowledge.</td>
<td>Practitioner satisfaction, confidence or knowledge.</td>
<td>Survey and/or interviews</td>
</tr>
<tr>
<td>Customer level</td>
<td>Impact on customers of using or applying the knowledge.</td>
<td>Customer satisfaction</td>
<td>Survey and/or interviews</td>
</tr>
</tbody>
</table>

Sustain use: Sustainability of knowledge is continuation of knowledge use after the initial adoption. In order to sustain knowledge use, sustainability-oriented action plans need to be developed. There should be consensus on the implementation need and benefits of the knowledge use. Once, the knowledge is used in an on-going implementation, the long-term implementation of the knowledge use needs to be evaluated. For example, can the changes be implemented organization wide? What support is required from the management for continued implementation? It is also important to respond to the shift in context while spreading the knowledge implementation. Along with ensuring sustained use of knowledge, it is also important to allow integration of new ideas that emerge from knowledge use.
Case description - The KT framework was used to translate evidence from research studies related to three areas: when to begin automation, what are the testability requirements and what is waste in test automation? In particular, the unit of analysis were the criteria for when to automate, testability requirements and factors associated with waste. From here onwards we refer the criteria, requirements and factors as variables. The outcome of the knowledge translation with respect to the three variables is depicted in Figures 52, 53 and 54 and elaborated in the steps below.

**Entry point - Need for knowledge implementation:** The need for improvement in practitioners’ opinion and knowledge related to GUI test automation was identified together with the local champion in a expert panel discussion. The expert panel consisted of a test expert (local champion), a professor, a PhD student and a master student. The test expert was the local champion in the case organization who was interested to know the perceptions of the team members as well as about the evidence in terms of the research outcomes. The main purpose of the discussion was to delimit the scope.

The steps followed in the KT cycle were as follows:

1. **Identify review and select knowledge (Likelihood):** A literature review was conducted to identify relevant knowledge from research studies. Overall 19 studies related to the test automation areas were found. We extracted the frequency and qualitative information related to the variables from the identified studies. The frequencies in the form of percentages of the research studies that mention any one of the areas were calculated. For example, if a criterion for when to automate was mentioned in 5 out of 19 identified studies then, the likelihood was calculated as 26%. The likelihood is indicated by dashed lines (---) in Figures 52, 53 and 54. In addition, qualitative information about each variable was extracted. Qualitative information includes the name, description and references of the papers that mention the criterion, testability requirements and factors associated with waste.

**Observations on the Likelihood** - The likelihood calculation might be based on the percentage of papers that discuss a particular variable. The papers might focus on a particular subset of criteria and not all possible criteria. Therefore, 10% likelihood does not necessarily mean that only 10% of the papers found the criterion.
It merely indicates the research focus which might indicate the significance of the criteria but not necessarily.

2. **Elicit prior (Prior probabilities):**
   - Selecting individuals: The practitioners related to the testing activity were selected. The roles investigated were test lead, test architect, developer, tester, design lead and product owner. In this particular case, all roles involved in the project were selected to understand all different perceptions. In total, 12 interviews were conducted.
   - Elicit opinions: The state-of-practice in terms of prior probabilities are elicited through interviews from the above mentioned stakeholders. We asked the practitioners about their testing process. In particular, the following questions were asked: what criteria do you use to decide when to automate? What testability requirements do you consider? What are the factors that you associate with waste? The information on the research outcomes (likelihood) was not provided to the interviewees while eliciting prior probabilities. Similar to the likelihood calculations, the percentage of the interviewees mentioning the factor was calculated. The prior probability is indicated by dotted lines (....) in Figures 52, 53 and 54.

   **Observations on the prior probability:** Not all interviewees mentioned the same variables. For example, only F2 in Figure 7 was mentioned by all interviewees. In most cases only a subset of the interviewees identified the factors. Out of the total number of possible responses (12 interviewees x 10 factors = 120) obtained from interviewees only 42% of the interviewees’ responses accounted for the identified criteria in Figure 7 for example. This indicates that all stakeholders are not equally informed or are not able to articulate or communicate the same variables when they are asked. This further implies that when decisions are made or processes are defined, some of the variables might not be considered or communicated.

3. **Assess the gap:** The local champion wanted to know the perceptions and also about the evidence in research studies. In addition, the desired state was that all team members shared the same knowledge.

4. **Align external knowledge to local context:** The knowledge identified, reviewed and selected in Step 1 is presented to the inter-
viewees and their opinions (posterior probabilities) were elicited again in the light of knowledge that was presented to them. The posterior probability is indicated by solid lines (—) in Figures 52, 53 and 54.

**Observations on the posterior probability:**

- Posterior probability is not always 100%, though the results show that the respondents became more consensual, it still shows that complete consensus is not achieved. In some cases, the likelihood is not applicable or aligned to the con-
text hence, there is no difference between posterior probability and prior probability.

- The average posterior probability of the variables that were mentioned by a subset of the interviewees and also mentioned in the literature was 81% whereas the average prior probability was only 39%. Which indicates that after the knowledge translation the interviewees became more consensual and more informed by having the variables available from literature.

- Even when the interviewees did not mention the variables in their prior probability, the final probability increased drastically. The average posterior probability of such factors was 73% even when the likelihood percentage was rather low i.e. 26% on an average. If the research result is applicable to the context then the interviewees revised their opinions i.e. posterior probability increased however, the research that is applicable to a specific context might not be heavily researched. Therefore, it is not surprising that there was a significant increase in posterior probability in spite of the likelihood being low.

5. **Assess barriers and facilitators**: The possible barrier anticipated with the one-on-one interviewees was that the interviewees might tend to agree more to avoid discussions or due to time constraints. On the other hand, the interviewees might not agree or not know about certain aspects that are related to other roles or tasks. In
order words, the interviewees’ opinions and knowledge might not be aligned. The main facilitator was the local champion who identified the need for improvement and wanted to change and improve the overall understanding of the practitioners.

6. **Select, tailor and implement KT interventions**: Due to the identified barriers and the need to have a joint discussion, we decided to conduct a workshop with the interviewees. We decided to tailor the workshop by only using aggregated results and not asking individual questions on why they did not mention certain aspects. Rather than discussing what they did not mention in the interview we discuss what other companies and researchers think and if they agree or disagree to others. This format of the workshop encourages discussion rather than focusing on individual responses. In addition, the intention of conducting a workshop was to facilitate discussions among workshop participants that had different roles in order to understand the big picture of the end-to-end test automation process.

7. **Monitor use**: Since, the identified need for improvement was to improve the practitioners’ knowledge and opinions, the knowledge use was on the conceptual level. Therefore, in the workshop we collected information about the levels of knowledge and attitudes before and after being informed about the research results. We asked about the variables that were not mentioned by all the interviewees. Based on the response of the workshop participants we conclude that the practitioners’ opinions before knowledge translation are driven by a number of factors: the tasks practitioners are involved in at the time of the interview, their role and the background such as education and work experience. If practitioners miss out on mentioning an aspect, then it might be due to the fact that it is not a priority for the practitioner working on a particular task at that time of the interview. For example, as a tester, the task was to automate everything, so when asked about criteria for when to automate, the tester is likely to mention the criteria that are relevant to his/her task. Criterion such as feasibility assessment of test automation might be a criterion that is considered by the architect and most likely the tester will not mention it right away.

In addition, the question regarding the intention to change based on the knowledge from research results were asked. The workshop participants were already following most of the practices
mentioned in the literature however, we found some new insights which were not considered by the practitioners prior to the KT process. However, the practitioners did not perceive it to have much impact. Although, this is a reflection on the results from research studies and not so much on the KT process itself.

8. **Evaluate outcomes:**

- Evaluating knowledge use outcomes: The need for improvement was on practitioner level, therefore, we collected the practitioner’s satisfaction, confidence and knowledge in the workshop. The evaluation of the KT outcome is based on the time, usefulness and value of KT as perceived by the workshop participants.

  *Time:* Practitioners hardly have time to find answers in research studies. Therefore, it was appreciated that the research results were summarized to them and it was considered to be a time efficient process. In addition, scientific results were valued more than the external information practitioners access (usually technical blogs).

  *Usefulness:* The KT process helped in aligning and streamlining the opinions of the workshop participants. They appreciated the external input and the knowledge as sometimes they can get too influenced by their own tasks and environment. They also identified the most use of KT process in supporting decision-making. Particularly when they do not have all the information needed to evaluate all the alternatives.

  *Value:* The workshop participants expressed an increase in confidence as they got a confirmation that they are following good practices and that they are not missing any crucial knowledge from the research.

- Evaluation of KT intervention: The interventions used in the KT process were interviewees and a workshop. The combination of interviews and workshop was appreciated. The interviews allowed the practitioners to express their individual opinions without being influenced by others. At the same time, after expressing their opinions, it was good to discuss with the other practitioners and reflect more on it. In addition to the conversation between the researchers and practitioners during the workshop, the practitioners asked questions to other practitioners and discussed among themselves.
as well. The practitioners did more reflecting and discussed the aspects that they did not mention in the interview.

5 DISCUSSION

In this section we discuss the adaptation to the KT framework based on the evaluation. In addition, we compare the KT evaluation result with the related work.

Updates in the KT framework based on the challenges in the pilot study:

The difference in terminology: The terminology used in the research studies was used straightaway, therefore, it created problem in communicating the knowledge and created problems for practitioners in interpreting the results. Hence, after the information is retrieved from the research studies in Step 1, it is important to discuss the results with the local champion. The results summarized in the terminology that is used in the company will be effective in aligning it to the local context in Step 4.

Allowing time between collecting prior and posterior opinions: After collecting the prior opinions, the practitioners wanted to get the research results beforehand so that they can try to better understand the research results and discuss more concretely. The practitioners did not get enough time to think and discuss the research results as they were presented in the same interview. The practitioners were asked to give their opinion immediately after knowing the research results. Getting prior access to the research results will give practitioners time to think and reflect on it. In addition, the results tailored to the terminology that practitioners are used to, will make it clearer. We agree that allowing time between collecting prior and posterior probability will help in capturing true opinions and reflections. Although, booking two different interviews might be a challenge. In particular, it might be difficult if the benefits of the participation are not convincing or clear to the practitioners.

Participating researchers: In order to facilitate good discussions, the researchers participating in the knowledge translation process should be well informed about the topic. Thus, it may not be suitable to use students, unless being PhD students focusing on the area being translated to industry.

Discussion on the evaluation results in related to the related work: The practitioners involved in the evaluation value results from research studies. This is consistent with the study conducted in Microsoft where
practitioners were positive towards results from software research studies [117].

The evaluation results indicate that the practitioners have different opinions and knowledge. The practitioners might not always be aware of each others’ opinions and knowledge. This is evident by the prior probability line (....) in Figures 52, 53 and 54. This creates misalignment which might be a problem in cases where alignment and shared knowledge is necessary. This is supporting the problem identified in related works [145] and [50] that suggest replying solely on practitioners’ opinion and knowledge might be error-prone.

In the previous study, Devanbu et al. mention that practitioners are heavily influenced by their prior beliefs which impacts their response to new evidence [50]. Although, through the several interactions in the KT interviews and workshops we noticed that in the posterior probability a broader view is often taken than when discussing the prior probabilities. Therefore, the practitioners were not biased and heavily influenced by their prior beliefs. Another finding in the previous study was on the relation between level of agreement with the strength of evidence. They conclude that “the level of agreement didn’t always correspond very well with the strength of evidence in regards to the claim” [50]. In the previous study [50], the practitioners were asked to respond to the new evidence through a survey, which might explain why practitioners were more influenced by personal opinions than evidence. However, in our evaluation of the KT process we found that the practitioners did considerably revise their opinions although, only when the results from research studies are valid to their context.

6 CONCLUSIONS

KT in software engineering research is an important activity for application of research results in practice. This chapter outlines the steps in the KT framework and the evaluation results indicate that KT can be valuable in addressing the practice-research gap.

In conclusion, KT framework presented and evaluated in this chapter is:

- Systematic,
- Repeatable,
- Reflective of the needs and mores of the practitioners,
- Iterative,
• Collaborative and consensus-driven.

Beecham et al. [14] discuss the practice-research paradox where practitioners do not regularly look at academic literature even though they perceive it to be valuable. We discuss the results of the KT evaluation in relation to the three reasons identified by Beecham et al. [14] for the practice-research paradox: accessibility, credibility and relevance.

The KT framework starts by identifying and summarizing the research results (Step 1) in order to make it accessible to the practitioners. These results are presented to the practitioners by the researchers in one of the step in KT framework (Step 4) therefore making it accessible. It incorporates expert opinions of the researchers as well as elicits practitioners’ knowledge and opinion (Step 2) thereby making the results more credible for the practitioners who prefer advice from people with “skin in the game” [14]. In addition to making the research results accessible, the KT framework also facilitates aligning the external knowledge from research studies to local context, thereby making the research relevant to the context.
REFERENCES


ABSTRACT

Context: The amount of software in solutions provided in various domains is continuously growing. These solutions are a mix of hardware and software solutions, often referred to as software-intensive systems. Companies seek alternatives to improve the software development process to avoid delays or cost overruns related to software development. Component origins such as in-house, outsourcing, COTS or OSS are gaining popularity, leading to the decision to choose among component origins.

Objective: The overall goal of this thesis is to support decision-making for selecting component origins. Following a decision-making process including all the key decision-making activities is crucial in making decisions. Therefore, the objective of the thesis is to support the decision-makers to create a decision-making process based on their context. In addition, the objective is to improve the decision-making process by incorporating research results and decision-makers’ opinion and knowledge in practice.

Method: We identified the factors that influence the choice to select among different component origins through a systematic literature review using an SB strategy and a DB search. We extended the investigation and conducted a case survey of 22 cases. Using design science, we developed solutions including a process-line to support decision-makers, a Bayesian synthesis process to integrate the evidence from literature into practice and a KT framework to facilitate the implementation of research results in practice.

Results: In-house development and alternative component origins (outsourcing, COTS, and OSS) are being used for software development. Several factors such as time, cost and license implications influence the selection of component origins. Solutions have been proposed to support the decision-making. However, these solutions consider only a subset of factors identified in the literature. According to the case survey, the solutions proposed in literature are not aligned with practice. In practice, the decisions are mostly based on opinions. The design objective to support decision-makers with the decision-making process is identified. Therefore, we propose a process-line to address the design objective. In addition, to make the decision-making more informed we propose a KT framework incorporating Bayesian synthesis to help decision-makers make evidence-informed decisions.

Conclusions: The decision to choose among component origins is case dependent. To support the decision-making process, the flexibility and customization of the solution based on the context are important. Therefore, the process-line proposed in the thesis is not prescriptive rather it is customizable to the context. In addition, to facilitate evidence-based decision-making, we provide an application of the KT framework that allows decision-makers to consider research results in addition to their own opinions and knowledge.