Evaluating Value Stream Mapping in Software Testing Context in Automotive Domain: A Case Study

Abhinaya Kasoju
This thesis is submitted to the School of Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:
Author:
Abhinaya Kasoju
E-mail: kasoju.abhinaya@gmail.com

University advisor:
Dr Kai Petersen
Senior Researcher
School of Engineering
E-mail: kai.petersen@bth.se

School of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona
Sweden

Internet : www.bth.se/com
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57
ABSTRACT

Context: Software testing is a crucial area in software development. It has been found that there is little evidence on how testing is performed in software automotive domain. Challenges in this domain have not yet been fully evaluated. Furthermore, interactions between test procedures, and other activities of software development are left unexposed. Assessment of test process is an important part of improvement initiatives.

Objectives: In this study we perform an in-depth investigation to identify challenges which lead to wastes in test process in the context of automotive software domain. To this end, we also identified strengths that add value in test process.

Methods: We performed a Case study at a Sweden based large automotive organization. To this end, we conducted semi-structured interviews and used Grounded Theory method for interview data analysis. We used Value Stream Mapping (VSM), a Lean methodology tool for test process assessment. Furthermore, we conducted Systematic Literature Reviews to look for evidence of similar problems and solutions offered in peer-reviewed literature.

Results: We grouped the identified challenges and strengths into 10 and 4 categories respectively. We also identified 7 kinds of waste and 4 kinds of values based using VSM. Based on systematic literature reviews, we proposed 7 solution proposals, an agile process model with practices.

Conclusions: We conclude that VSM is an efficient tool in eliciting improvement potentials in software testing context using qualitative data. In regard to solution proposals, we propose further empirical work to evaluate the solution proposals themselves and tailor to company needs. However, the results obtained through this study can be compared to test processes at other companies in automotive domain.

Keywords: Automotive software domain Lean methodology, Software Test process, Value Stream Mapping, Embedded Software
Acknowledgement

First and foremost, I would like to express my sincere gratitude to my university supervisor Dr. Kai Petersen for his continuous support, patience, and great efforts he put to explain things clearly and simply. I could not have imagined having a better supervisor for the thesis.

Besides this, the case organization is gratefully acknowledged for providing me this opportunity. Special thanks to all the interview and survey respondents for their patience in providing very informative responses which added great value to the thesis.

I owe my loving thanks to my close friends and dear ones, for helping me get through the difficult times by providing emotional support. I can’t express my gratitude for my parents Dr K.Sudhakar and Prameela Devi K in words, without whose encouragement and understanding it would have been impossible to finish this work. My special gratitude is towards my brother and sister.
# TABLE OF CONTENTS

ABSTRACT .......................................................................................................................... 1

TABLE OF CONTENTS ........................................................................................................ II

1 INTRODUCTION .................................................................................................................. 1

2 BACKGROUND AND RELATED WORK ............................................................................. 5
  2.1 LEAN METHODOLOGY AND VALUE STREAM MAPPING .............................................. 5

3 RESEARCH METHOD ........................................................................................................... 7
  3.1 CASE STUDY DESIGN .................................................................................................... 8
    3.1.1 Research questions .................................................................................................. 8
    3.1.2 Case study and unit of analysis .............................................................................. 9
    3.1.3 Data collection procedures ................................................................................... 9
    3.1.4 Data collection and analysis approach ................................................................. 11
    3.1.5 Threats to validity ............................................................................................... 17

4 RESULTS ............................................................................................................................. 19
  4.1 OVERVIEW OF STUDIED PROJECTS ........................................................................... 19

5 QUALITATIVE DATA ANALYSIS ...................................................................................... 22
  5.1 SOFTWARE DEVELOPMENT PROCESS IN PRACTICE .................................................. 22
    5.1.1 Test process at Case Organization ....................................................................... 24
    5.1.2 Roles and responsibilities within testing ............................................................... 26
  5.2 STRENGTHS IN TEST PROCESS .................................................................................. 27
  5.3 CHALLENGES IN TEST PROCESS .............................................................................. 30
  5.4 PROCESS ASSESSMENT USING VSM ......................................................................... 39

6 SYSTEMATIC LITERATURE REVIEW (SLR) ....................................................................... 47

7 SOLUTION PROPOSALS ...................................................................................................... 53

8 DISCUSSION .......................................................................................................................... 63

9 SYNTHESIS .......................................................................................................................... 67

10 CONCLUSIONS .................................................................................................................... 68
  10.1 OPEN ISSUES AND FURTHER WORK ......................................................................... 68

11 REFERENCES ...................................................................................................................... 69

12 APPENDIX A .......................................................................................................................... 77

13 APPENDIX B .......................................................................................................................... 81

14 APPENDIX C .......................................................................................................................... 82

15 APPENDIX D .......................................................................................................................... 87

16 APPENDIX E .......................................................................................................................... 88

17 APPENDIX F .......................................................................................................................... 89
1 Introduction

Verification & Validation (V&V) is one of the most important and widely used activities in software development. V&V involves a range of activities including test process and testing. According to IEEE 610.12(1990) [1] testing is defined as “an activity in which a system or component is executed under specified conditions, the results are observed or recorded, and an evaluation is made of some aspect of system or component”. Testing plays a central role in quality assurance activities of many organizations. Finding efficient ways to perform testing is a key challenge in testing. Efficient testing is not about writing good tests but it is all about time spent on reporting, debugging and fixing faults [3]. It is observed that an efficient testing process is vital to the quality of the developed product in reducing the overall development expenses [2]. In essence, testing can show if a system does not confirm to the requirements. A challenge in V&V is that testing shows only the presence of defects and not their absence [61]. Quality assurance therefore is responsible to measure the quality of the product that is ready to be delivered.

Constantly changing market demands organisations to focus more on customer needs and value-added by those needs. In order to cope with these changes and gain competitive edge, companies should be able to adapt rapidly to develop software faster and cheaper with high quality. In automotive domain, software has become an enabling technology for almost all safety-critical and comfort functions offered to the customer. Increasing interactions among automotive software functions and the increasing complexity demand new ways of developing and testing automotive systems [31]. Automotive software development and embedded software development in domains such as telecommunications, avionics, etc share similar challenges arising due to system complexity [31]. Some of the challenges can be high degree of scattering of functionality leading to unstable system and corresponding requirements across the system, to manage the growing operational and environmental complexities these systems tend to face, synchronise multiple stances of software development with hardware development [31]. Others are non-functional requirements such as safety, reliability, performance for critical systems, time-to-market goals and daunting budgetary challenges which pose increased competitive pressures driving towards compressed schedules [7].

At present there is a new trend in automotive software development called model-based development. Evolving through several decades automotive embedded devices has changed from electrical and mechanical devices to combination of software and electrical/mechanical devices. The effects of these changes on processes, methods, tools and required competences were found to be very significant. However, these changes did not yet contribute
anything significant to the quality assurance of model-based development, especially testing which is still poorly supported [4]. Verification and Validation of automotive software is only performed to find the presence of improper functionality and lack of fulfilment of requirements in the software and not to introduce quality. Many new programming languages, tools, methods and frameworks like TTCN-3 [38], TPT [4] have been introduced to make software testing easier. To facilitate technology transfer, i.e., to adopt new, emerging technologies in industry, a strong evidence in relation to the specific technology must be offered [37]. Furthermore, large enterprises find it difficult to adapt to new techniques due to the huge and complex processes they possess. For example, according to Zelkowitz [39], technology infusion at NASA took nearly 4 years, including training, pilot studies and tailoring technology to their needs and environment. In addition to this, sudden changes in the processes cannot provide immediate results and takes lot of time for the work force to get acquainted to them and use them effectively. This is same with testing in automotive domain where development requires creation of expensive proprietary testing solutions which are applicable for limited projects [n9]. Due to this, organizations may focus little on testing activity of these systems which can bring adverse effect on software quality [2].

It has been found that little evidence exists on how testing is performed in automotive domain and challenges in this context are not evaluated [5, 6]. Furthermore, interaction between test procedures, methods, tools and techniques with test management, version management is left untold [6]. The need to test as early as possible, on multiple integration levels under real time constraints make high demands on the kind of test process and procedures being used [6]. The need to quantify the quality assurance value of testing activities in automotive context was identified by Sundmark et al [5]. They conducted a detailed study on how system testing is performed in connection to a release process in automotive context and identified several challenges in this regard. Moreover, they observed a need for detailed identification and prioritization of areas with improvement potential. However, there have been no studies with an in-depth focus on strengths and challenges within the whole test process and process improvement initiatives in automotive software context.

Lean software development and its tools gained popularity very recently. Fundamental to lean thinking is the concept of eliminating or reducing waste in a software development process using tools like ‘Value Stream Mapping’ (VSM) [11]. Mujtaba et al. [13] performed an industrial study to perform value stream analysis to identify and analyze waste in a software customization process. They found that VSM is a useful tool for Software Process Improvement (SPI) initiatives. However, the usefulness of VSM has not yet been evaluated fully and empirical studies in a variety of contexts are scarce. Lean thinking was initially implemented in automotive domain at
Toyota [13]. Removing waste in processes helps in focusing on value creating activities [13, 18, 19].

We conducted an industrial case study to investigate problems and challenges in test process in automotive software domain and identify improvement potentials using the concepts of lean thinking. To this end, we conducted 14 semi-structured interviews [22] with 14 representaties of different roles and responsibilities. Apart from these 14 interviews, we conducted one more interview with a representative from an automotive software project that has incorporated agile software development successfully in recent years. A pre-study at the case organization revealed that their biggest challenge lies in managing the drastically increasing complexity of the vehicle electronics system and handling the tested artefact to customer in time which motivated the need for this case study. Thus we performed a qualitative study to assess test process in automotive software domain, with which we identified wastes and improvement potentials in test related activities. Furthermore, we visualized and analyzed Value stream of V & V activities and proposed possible directions for improvement. We performed qualitative data analysis using Grounded Theory [25] research method and conducted a Systematic Literature Review (SLR) [21] to identify solutions from literature for identified challenges in our case study. The contributions of our study are the following:

- Identify and gain an in-depth understanding of the most important issues (problems/challenges) and strengths in test process used at the case organization.
- Illustrate the role of VSM to elicit information of waste and value creating activities in test process. This was done based on qualitative data from interviews and process documentation. The benefits of this include a derivation of current VSM of test process, improvement potential areas and a future VSM.
- Based on a SLR, provide improvement proposals to the problems and challenges identified in our case study.

The outline of this thesis is as follows.

<table>
<thead>
<tr>
<th>Section</th>
<th>Research Contribution</th>
<th>Research Implications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background and Related work</td>
<td>Gathers Information about similar kind of work done in previous studies (Mapped to SLR results) in order to map them to the results of current study.</td>
<td>No study identified which necessarily concentrates on identifying improvement potentials in testing process in automotive context using Value Stream Mapping (VSM).</td>
</tr>
<tr>
<td>Research Method</td>
<td>This section clearly defines the motivation behind choosing each research method and how they are carried out along with the validity threats associated</td>
<td>Case study research methodology with grounded theory for data analysis is used for this study in order to collect more data points. Interviews are used as main</td>
</tr>
<tr>
<td>Part</td>
<td>Description</td>
<td>Details</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>Results</td>
<td>This section presents methods/ tools used in various projects in automotive domain which helps in understanding the phenomenon of testing.</td>
<td>Provides an overview of the projects being developed in automotive domain which helps in understanding the causes of challenges.</td>
</tr>
<tr>
<td>Qualitative Data Analysis (Until section 5.3)</td>
<td>This section presents all the qualitative data collected i.e., testing practices.</td>
<td>The software development methodology, software testing process is clearly illustrated in this section. The testing practices identified are grouped into challenges (10 categories) and strengths (4 categories).</td>
</tr>
<tr>
<td>Process Assessment using VSM (Section 5.4)</td>
<td>Process assessment of the test process using VSM is carried out. It is represented using current state value stream map.</td>
<td>Seven different wastes in 12 subprocess areas within testing are identified. Four (Out of 5) different value-adding activities are also identified.</td>
</tr>
<tr>
<td>Systematic Literature Review</td>
<td>This section aims to identify the existing challenges in testing in the context of automotive domain. Along with this existing solutions available to challenges identified through this study are also identified.</td>
<td>The existing challenges in automotive domain are mapped to challenges identified through this study. However, no study discussed strengths associated with testing in automotive domain.</td>
</tr>
<tr>
<td>Solution Proposals</td>
<td>In this section solutions from previous case studies which are empirically evaluated are identified in order to provide tangible evidence in terms of applicability which were performed with real users in real setting.</td>
<td>Seven different solution proposals were made to address the identified challenges.</td>
</tr>
<tr>
<td>Discussion</td>
<td>This section brings together all the identified results followed by a critical analysis with respect to application of these results. The draft of future state map is also illustrated in this section.</td>
<td>Applicability and limitations of the identified solutions with respect to the challenges is discussed. An evaluation of VSM in this study context is also presented in this section.</td>
</tr>
<tr>
<td>Synthesis and Conclusion</td>
<td>Synthesis of the results with respect to research questions and discussion of open issues which lead to future work are presented in these sections.</td>
<td>These sections conclude that seven kinds of wastes were identified in test process for which improper requirements management, unstructured test process play major role.</td>
</tr>
</tbody>
</table>
2 BACKGROUND AND RELATED WORK

A need for an effective software development methodology for embedded systems was identified by Greene in [8]. In the report [17] Karsai et al noted that the development of software for embedded systems is difficult because these systems are part of a physical environment whose complex dynamics and changing requirements need to be satisfied. In order to achieve this, employing proper methodologies that provide the software engineering team with various ways of integrating software testing into their process in a natural and non-intrusive way is necessary [20]. Software development paradigms will be able to assist organisations in taking proper actions to avoid the risk of budget and schedule over-runs, etc. The most widely used methodologies, which are believed to address the above mentioned challenges are agile and lean software development [16, 33, 35, and 41]. The term agile development evolved in the year 2001 in North America [8]. The major reason for the evolution of this new paradigm of software development is to cope up with constant changes [9]. The ‘Agile Manifesto’ with a set of 4 values forms the base for responding to changes [10]. The lean software paradigm is defined as a set of principles and practices focused on the removal of waste leading to a lean software development process [11].

2.1 Lean methodology and value stream mapping

Waste according to lean philosophy is everything that does not contribute to the value creation. As said by Poppendiecks [11], the basic idea of lean is to focus all development effort on value adding activities and identify non-value adding activities and remove them. Lean thinking eliminates or reduces waste in software development process and improves customer satisfaction [11]. These wastes can be extra features, partially done work, relearning, handoffs, task switching, delays, defects which have been transferred from manufacturing domain to software engineering by Poppendieck et al [11]. Hence, a decision was made to visualize and analyze the value stream of V&V activities and to propose possible measures to optimize it. This further indicated the need to employ VSM as a lean methodological tool for performing the value stream analysis. This tool is used for uncovering and eliminating waste [11]. A value stream is all the actions (both value added and non-value added) currently required to bring a product through the main process steps to the customer a.k.a end-to-end flow of process. The biggest delays or bottlenecks in a value stream provide the biggest opportunity for improving the process capability [11]. Some of the other famous lean tools used in software engineering context are Theory Of Constraints (TOC), Kaizen, Kanban, etc. The motivation behind choosing
VSM is because it is an efficient tool with which we could walk through the testing process to understand workflow and focus explicitly on identifying waste with an end-to-end perspective [15]. It provides managers the ability to step back and rethink the entire process from a value creation perspective [13].

In this study we identified various strengths and challenges in the test process of automotive domain. The challenges found in previous studies can be found in the Systematic Literature Reviews’ (SLRs) results in Section 6 Table 14. However, we could not identify any strength related to automotive testing in the literature suitable for this study context.
3 RESEARCH METHOD

This section illustrates how various research methods were used to identify results for the aims and objectives of this study. Furthermore, the motivation behind choosing every research method is also discussed, followed by a description of the research questions, case and units of analysis, data collection and validity threats. To answer our research questions we used Case Study research methodology and Grounded Theory as our data analysis method. We also performed SLRs to find the solutions available in literature to resolve the issues found in our case study.

Most commonly used research methods in software engineering context are controlled experiments [23], surveys [30], case studies [32], action research [39] and simulation [39].

Controlled Experiments: This research method is used to validate theories in a controlled environment. For this a hypothesis is formulated where cause-effect relationship between one or more independent and outcome variables in studied. This kind of setting have more control which means variables other than independent variables should not affect the outcome.

Surveys: A survey studies the phenomena for a population by surveying a sample in a specific setting. The data collection is performed using questionnaires or interviews. After data collection statistical inference is used to draw conclusions for the overall population.

Case Studies: Case study enables to perform and in-depth investigation of a phenomena focusing on a specific case. The cases are objects in the real world studied in a natural setting, i.e., in real software organizations, software projects with software developers, etc.

Action Research: In action research the main objective is to introduce an intervention in real world setting and observe its effects. In this the researcher needs to be actively involved in introducing and intervention and making observations [27].

Simulations: simulations are executable models of real world phenomena to study their behaviour [39].

Motivation for choice of Research Method: The main research method used for this thesis is case study. The goal of this thesis is to investigate the phenomenon of testing in large scale automotive software development which includes complex environment. This kind of study gives an in-depth understanding of testing in automotive domain with huge amounts of rich qualitative data with little control. The solution obtained through this case study cannot be replicated using lab experiment with students. Also survey is used to obtain an overall view of testing context with regard to population and not understanding of entire test process in practice. Simulations are not considered since this study is not considering measurable characteristics of a
process. Action research concentrates on planning an action and observing its effect because the no action is being taken at the case organisation.

We also conducted SLRs to identify solutions for the issues identified in this case study. However, the SLRs were performed to solve the issues specific to this case study and cannot be generalized for testing in automotive domain. The design of SLRs was largely dependent on interviews. The design of SLRs and how their results relate to our interview results are discussed in Section 6. Studying the effects of implementations of recommendations given through this study can be considered as future work.

3.1 Case study design

This section entails the research design used in this case study.

3.1.1 Research questions

The following are main research questions that should be answered in the case study

<table>
<thead>
<tr>
<th>Research Questions (RQs)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ1: What are the practices in testing that can be considered as strengths within automotive domain?</td>
<td>An inventory of activities that act as strengths in the testing process is provided through this research question. This is extracted from the qualitative data obtained through interviews.</td>
</tr>
<tr>
<td>RQ2: What are the challenges/ bottlenecks identified in testing automotive embedded systems?</td>
<td>Lists of challenges or poorly performed practices that act as barriers to incept quality in testing process are collected to answer this research question.</td>
</tr>
<tr>
<td>RQ3: Is VSM tool useful in eliciting improvement potentials in software testing context in automotive domain?</td>
<td>An evaluation of current test process using Lean methodology to improve the process is conducted. This is done by using VSM which is a lean methodological tool used to create value stream. For this various value adding and non-value adding activities are collected along with improvement potentials in the test process.</td>
</tr>
</tbody>
</table>

The relevance of research questions can be underlined as follows: The related work has shown a number of problems related to software development in automotive domain. However, there is too little empirical evidence available on the value-added and non value-added activities practiced by various practitioners in the software testing context of automotive domain. Thus more data points are needed. Furthermore, the interconnection between different strengths and weaknesses is not addressed in any way so far, making it hard to decide in which way it is most beneficial to improve
software testing, or whether the introduction of a new way of working will help in improving the key challenges experienced in automotive domain.

3.1.2 Case study and unit of analysis

The case being studied is one of the development sites of a large Swedish automotive organization. The case organization is ISO certified organization which uses CMMI for their process assessment. They focus on both soft and hard products involving areas such as telematics, logistics, electronics, mechanics, simulation modelling and systems engineering.

We report on a single-case with multiple units of analysis [32], in which we studied the phenomenon of testing in several projects in one company. The case study helps in performing two types of tasks: (a) generates theoretically similar results, (b) generates contrasting results for predictable reasons. We chose both options (a) and (b) to study, compare and compliment the findings corresponding to challenges and strengths in software testing context. This kind of approach helps in complimenting already existing challenges and strengths of software testing in automotive domain reported by previous studies and also allows us to compare between the testing methodologies, methods and tools being used for different projects at the case organization. This kind of framework enables us to clearly identify the conditions, when a particular phenomenon is likely to be found (a) and when it is not likely to be found (b) [32].

The units of analysis here are different projects developed in automotive domain i.e., case organization which follow different techniques for testing. They are selected in such a way that they have maximum variation in factors such as methodology being used, team size, techniques used for testing. This will enable the results applicable for other projects with similar characteristics in automotive domain. In order to understand the challenges and strengths in their testing process an in-depth analysis of different units of analysis is done using Grounded Theory [25]. This will enable to address a large society which suits any kind of projects being developed in automotive domain i.e., generalization of the results for automotive domain.

3.1.3 Data collection procedures

The data is collected through interviews and various process documentation of testing. However, other sources were not collected due to their unavailability and adequacy of the existing data collected for the study. The motivation behind using several sources of data is to limit the effects of only one interpretation. Interpretation of data using several sources is called Triangulation. Here we have used data point triangulation [27] which helped in validating the identified issue (for ex., a testing practice) by checking it against other data source for confirmation. When same conclusion is drawn from multiple interpretations (i.e., multiple data sources) the conclusion will be stronger [27].
a) Selection of Interviews:

The interviewees’ selection for the case study is done in such a way that they include managers, developers and testers so that overall development life cycle involving testing can be thoroughly studied and understood. The selection of interviewees was done as follows:

The roles were selected as to represent positions that were directly involved with testing related activities or affected by the results of the entire testing process. Roles from both the projects and line organization from three departments alpha, beta and gamma (due to confidentiality reasons, the department names are renamed) were included in our study. Roles selected are given in Table 2.

In departments Alpha and Beta sufficient number of employees were available for the interview but in Gamma due to the unavailability of employees for the interview only one employee was interviewed who had a very vast experience within the case organization.

<table>
<thead>
<tr>
<th>Department</th>
<th>Organization level</th>
<th>Role (No. Of Interviewees)</th>
<th>Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha</td>
<td>Line</td>
<td>Group Manager (1)</td>
<td>Responsible for all test resources such as testing tools. Also responsible to see that the test team has the correct competence level.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project Test Leader (2)</td>
<td>Traditional role responsible for leading all the test activities such as test case design, implementation and reporting defects. Test leader is also responsible for test activities in project and their documentation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Developer (2)</td>
<td>The developer uses the requirements specifications to design and implement the system. This role is also responsible for all the testing (for some projects only).</td>
</tr>
<tr>
<td>Beta</td>
<td>Line</td>
<td>Advanced Engineer (1)</td>
<td>Technical Expert that often works with research projects. (In order to avoid confusion this role is also termed as developer in the later sections)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Domain Expert (1)</td>
<td>As a technical expert this person is responsible for research engineering project who strives to continuously improve testing in their team. (In order to avoid confusion this role is also termed as developer in the later sections)</td>
</tr>
<tr>
<td>Project</td>
<td>Developer (4)</td>
<td>Same as in Department Alpha</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
<td>-----------------------------</td>
<td></td>
</tr>
<tr>
<td>Test/Quality Co-ordinator (1)</td>
<td>Responsible to co-ordinate the all the test activities in the projects and also is responsible for managing the products.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Manager (1)</td>
<td>Responsible for planning, resource allocation, and development and follow-up related to the project. The requirements inflow is also controlled by this role.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The interview process gave an insight into why and how the currently using testing practices emerged, the changes that took place in testing in automotive domain since many years and what must be the level of difficulty for the new practices that will be recommended. A total of 14 employees were interviewed who possess different roles given in Table 2.

Another interview with a scrum master working in similar domain (i.e., automotive) has been conducted to get ideas for the solution proposals to be given through the case study. However, that project is not included in the studied projects as the interview was basically to extract recommendations from agile that work well for automotive embedded systems and not to study the processes in that project in detail.

b) Process documentation

Process documentation such as software development process documents, software test description document, software test plan document and test reports have been studied to gain an in-depth understanding of the test activities carried out in automotive industry. Also organization level process documents which describe the process specifications and hierarchy are studied with which familiarity with respect to the terminology related to testing within the case organization is gained. This in turn helped in understanding the interview data and analyzing it in accordance to the testing practices within the organization. For example some terms such as unit testing is termed as basic engineering tests with the organization. Without studying process documents it would be difficult to understand and perform analysis of interviews. However, quantitative measures regarding test activities are not acquired from the case organization for this study due to certain limitations such as ethical considerations.

3.1.4 Data collection and analysis approach
Grounded theory is chosen as the appropriate method used for data analysis in this case study. Grounded theory research method described by Glaser and Strauss [25] supports theory development about the relevant aspects influencing the specificity of a situation, a group of people or companies. The concepts of Strauss and Corbin [26] were used in analyzing the interview data which can be seen as objectivist approach. Even though the aim of this research was not to develop theories, grounded theory provides valuable tools to carry unbiased analysis of the overwhelming amount of qualitative data [24, 27]. However, there are other analytic strategies that can be used in a case study such as pattern-matching, explanation-building and time-series analysis. But they don’t support to understand the nature and complexity of the processes taking place [32]. This analysis is organized on the basis to describe the general characteristics of testing in automotive domain and relations of phenomenon observed. The concepts applied from grounded theory to this research are: reduction of data through coding, display of data using tables, etc.

Based on our pre understanding of the processes involved in the verification and validation a conceptual model formulated by Giedre et al [28] was used as a guide throughout the study. Data analysis approach used by Petersen [29] was used to further proceed with the research. The practices related to the software testing process at the case organization have been identified conducting the following steps outlined below. Steps A, B, C come into data collection approach and the remaining steps are used to perform data analysis.

A. Definition of the Interview guide:

The interview consists of five themes; the duration of the interviews was set to approximately one hour each. All interviews were recorded in audio format and also notes were taken. The five schemes of the interview were as follows.

1) Warm up and Experience: Questions regarding the interviewee’s background, experience and current activities.

2) Overview of Software testing process: Questions to collect information about criteria required for testing test objects such as obtained specific documents such as requirements traceability matrix or test description document before testing and brief explanation of current testing process being used.

3) Challenges and Strengths in testing process: This theme is aimed to collect information about as many practices as possible. Questions have been asked from two different perspectives: Good practices/Strengths, Challenges/poorly performed practices. The interviewees are supposed to state what kind of practice they used,
what its value contribution is and where is it located in the process with respect to testing context.

4) Improvement potentials in testing process: Questions to collect information about why the challenge must be eliminated or how the test process can be improved.

5) Any other Suggestions/Recommendations: Here suggestions regarding the questionnaire were asked and considered for the further interviews.

B. Interview Planning and Execution:

A semi-structured interview strategy [27] has been used in all the interviews. In a semi-structured interview, questions are planned, but they are not necessarily asked in the same order as they are listed. The development of the conversation in the interview can decide which order the questions will be handled [22]. Therefore the interview guide acted as checklist to make sure all the important topics are covered and thereby enabling a scope for improvisation and exploration of the studied objects. The planning and execution of the Interviews is done in the following way.

1) A complete list of people involved in the testing process irrespective of their role is selected.

2) For the selection process we used cluster sampling [32]. At least two persons from each project are selected from the list. The more persons are available for each project the more persons are selected. However, the final list of employees who participated in the interviews is based on availability.

3) The interviews received an e-mail explaining why they have been considered for the study. Furthermore, the mail contained information of the purpose of the research and an invitation for the interview. There was an excel sheet sent to the interviewees which has different time slots and dates for the interviews from which interviewee can select any time slot depending on his availability.

C. Transcription of Interviews, and division of transcriptions into sections:

In order to ensure comfort of the employees, prior permission of the employee was taken before recording every interview. Each and every interview was transcribed word-to-word into text using a tool called Express Scribe which is used to assist the transcription of audio recordings. On an
average it took 6-8 hours to transcribe each interview. The size of the transcriptions ranged from 3000 to 6000 words per interview on an average.

**D. Definition of proper codes to be assigned to transcribed text**

Initially all the transcribed documents were open coded using an analysis tool called NVivo 9. The result of such coding was around 3000 codes. Most of these codes were found irrelevant for the study hence use of such tools is being discarded for this study. In the next stage, codes from the transcribed text were identified with respect to the research questions to identify some similar patterns. For this manual coding is done for 5 interview transcriptions. All such kind of terms was grouped into sub-categories with which 300 codes were obtained approximately. These codes were clustered into different main categories. With this a coding guide is developed which consists of 15 categories with information on all the testing activities involved in software development. The coding guide also includes 12 categories included which emerge from lean philosophy. In order to validate this coding guide, an interview transcription is manually coded by an employee at the case organization and the results of such coding was compared with the researcher’s interpretation and required modifications were made. However the coding guide was continuously refined throughout the data extraction phase.

We considered three dimensions of test artefacts/activities connected through test processes. For example, all the codes related to challenges in requirements engineering are group together. Thereafter, the statements addressing similar areas within one group (e.g., all areas that would relate to insufficient requirements, communication problems are grouped to their respective group’s requirements and communication) are grouped. The coding guide was influenced by SWEBOK [36], literature and testing related ISO standards [42]. The coding guide was iteratively improved and the transcriptions were repeatedly coded according to the updates version of the coding guide. Three abstraction levels are identified (see Table 3): 1. “What kind of practice” i.e., high level abstraction, 2. “What is the practice” i.e., second level of abstraction and 3. “Where in the process does this activity take place” is the low level of abstraction. The coding guide is provided in the appendix.

<table>
<thead>
<tr>
<th>Abstraction level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>Codes directly related to research questions i.e., testing practices, problems or challenges, strengths, improvement potentials are identified here.</td>
</tr>
<tr>
<td>Medium</td>
<td>Three groups of codes: Challenges – ten categories, Strengths – five categories, Value – five categories, Waste- Seven Categories. See Appendix C.</td>
</tr>
</tbody>
</table>
E. Coding of Interview transcriptions using predefined codes

Fourteen interview transcriptions were coded using the updated version of coding guide. The sample template used during the coding is shown in table 4.

<table>
<thead>
<tr>
<th>Transcribed Text</th>
<th>INT. No.</th>
<th>HIGH</th>
<th>SECOND</th>
<th>THIRD</th>
<th>EXPLANATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>I think the problem is right now we don’t have a structured testing process,</td>
<td>1</td>
<td>Problem/challenge</td>
<td>G1.Organisation, processes</td>
<td>Test process</td>
<td>No defined test process</td>
</tr>
</tbody>
</table>

In the above table the first row indicates the transcribed text where raw data is directly pasted in the first column which relates to the code. “High”, “Second” and “Third” indicates three levels of abstractions which is stated in the next three columns respectively. The raw data which contained detailed explanations is abstracted by deriving statements i.e., explaining them shortly in one or two sentences (can be seen in column ‘EXPLANATION’). The result was a number of problem statements where statements varied in their abstraction level and could be further clustered. Also strengths were coded in the same way.

F. Sorting of coded transcriptions to group them according to codes

The coding template was designed in an excel sheet such that codes belonging to specific category are grouped and sorted from the beginning itself defining validation criteria for each level of abstraction. This made coding easy and manageable.

G. Analysis of the results

The explanation statements were grouped based on their relation to each other and their abstraction level. This was documented in form of mind-map (See Appendix F). Issues with higher abstraction level are closer to the centre of the mind map than issues with lower abstraction level [33]. Mind map indicates the relationship between different activities in the test process. Causes and effects of each issue/strength in testing process were identified and analyzed.

In order to determine which issues/strengths were the most common, the data from the interviews is divided into two parts: Global and Local.
Global ones are stated by interviewees representing more than one role OR also representing not only testing but also other parts of Software Development Life Cycle (SDLC). The local ones are stated by interviewees performing testing OR represents only testing activities.

H. Validation of results by feedback through a survey

In studies of qualitative nature there is always a risk that data is biased by the interpretation of the researcher. Since the transcribed text or the coding sheet (212 rows overall, 15 rows/interview) was very large, this cannot be sent to the interviewee for verification. The employees cannot spent time on validating it apart from their daily activities and also there is a threat of the interviewee misinterpreting codes. Therefore, the issues have been validated in a survey within the case organization. This Survey was conducted within the case organization in order to validate the interpretation of data obtained through interviews. This was done since the interview transcriptions cannot be validated by each interviewee due to lack of time as each transcribed interview consisted of 15 pages on an average. Also a summary of interview data cannot be sent to each interviewee as the interview data was directly extracted into the data analysis sheet which can be understood by the researcher only. This survey helped us in visualizing whether the issue or strength is present in the testing activity practiced within the case organization since the aim of this study is to understand the phenomenon of testing.

I. Weight of Issues

This step is aimed at identifying the most commonly perceived issues/strengths with regard to testing activities. In this way they were prioritized with which we were able to determine which testing activities (issues/strengths) were most commonly perceived. This step is used to ensure that the recommendations come for the issues that need more attention to incorporate quality into testing. However, this process cannot be applied for the solution proposals given as they are treated equally important to be implemented.

After having identified the practices they are prioritized into A-practices, B-practices, C-practices and D-practices as shown in table 5. The actual limits on the classes is based on the results from the interviews i.e., the number of interviews stating it. The main objective of the classification is to systematize and structure the data and not to claim that these classes are optimal or suitable for another study.

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A (General)</td>
<td>The practice is mentioned by more than 1/3 of the respondents</td>
</tr>
<tr>
<td>B (Very Common)</td>
<td>The practice is mentioned by more than 1/5 of the respondents</td>
</tr>
</tbody>
</table>
C (Common) | The practice is mentioned by more than 1/10 of the respondents
D (Others) | The practice is mentioned by less than 1/10 of the respondents or mentioned by one or two roles only.

After classifying the identified practices into the above mentioned classes. The practices are divided into Global and Local issues. *Global issues* are the one which occur in other parts of Software Development Life Cycle (SDLC) as well as testing and indirectly affect the quality of testing. This means that the issue that negatively influences testing is occurring due to other processes in SDLC and not testing process. *Local issues* are the ones that occur within the testing activity and directly affect the quality of testing. These issues occur in testing process and affect testing directly.

### 3.1.5 Threats to validity

A validity threat is a specific way in which you might be wrong [34]. Research based on empirical studies does have threats to consider. Threats to Validity of the outcome of the study are important to consider during the design of the study, which allows taking proper actions to reduce their impact from the early stages. Threats to validity in the case study are reported in [32] and for software engineering context in [23]. The threats relevant to this case study are: Construct validity, external validity and reliability or conclusion validity.

**Construct validity:**
Construct validity is concerned with obtaining right measures for the concept being studied. The following actions were taken to mitigate this threat [29].

**-Selection of people for Interviews:** To obtain the appropriate sample for answering the research questions, the interview list was continuously reviewed by practitioners to give proper feedback on plans as they know the persons and the organization best. Having aid of the people from the organization helped in having a good selection of interviewees for the interviews. The selection of the representatives of the company was done having the following aspects in mind such as process knowledge, roles, distribution across various hierarchies and having a sufficient number of people involved (according to Table 2). Hence proper care is taken to assure variety (across projects and roles) among selected people which aided in not getting a biased result.

**-Reactive Bias:** Furthermore, there is a threat that the presence of research worker influences the outcome of the study. The threat is reduced as the researcher is associated with the organization for this study and not viewed as being external to the organization. Also there has been a contract signed by the research worker and the organization to maintain confidentiality. Due to this fact anonymity of the individuals’ responses is guaranteed. However,
as both the employees and the management were favourable to test process improvement activities the interviews were not biased.

-Correct data Interview: Construct validity also addresses misinterpretation of interview questions. Firstly, a mock-interview was conducted with an employee with the organization in order to ensure the correct interpretation of the questions. Furthermore, the context of the study is clearly explained (through mail/in person) before the interview and also during the interview sometimes if required. However, the results were sent to each interviewee to validate them and also at the end a survey was conducted within the organization to validate the interpretations of the researcher.

External Validity:
External Validity is the ability to generalize the findings to a specific context as well as to general process models [29].

-A specific company: One of the potential threats to validity is that test process at only one company is studied for this case study. It has been impossible to conduct a similar study at another organization since this particular case study is aimed to improve the test processes at the respective organization only. However, this type of in-depth study gave an insight into automotive development in general and the findings have been mapped from the company’s specific processes to general processes. Thus, the context of the study and the situation at case organization are clearly described in detail which supports the generalization of the problems identified which allows others to understand how the results map to another specific context.

Reliability:
This threat is concerned with repetition or replication, and in particular that the same result would be found if re-doing the study in the same setting [29].

-Interpretation of data: There is always a risk that the outcome of the study is affected by the interpretation of the researcher. To mitigate this threat, the study has been designed so that the data is collected from different sources, i.e., to conduct triangulation to ensure the correctness of the findings. However, data related to process documentation is not provided in this report due to ethical considerations. The interviews have been recorded and the correct interpretations have been validated through a survey to improve the traceability of data. The analysis of the researcher is also reviewed by the supervisors continuously to identify potential problems in the analysis and steps of interpretation.
4 **RESULTS**

This section illustrates the results of this case study. First an overview of the projects studied is given where a brief introduction to the methods, methodologies and tools can be found. This is followed by a qualitative analysis part where the interview results are clearly explained.

4.1 **Overview of studied projects**

Overview of the studied projects is given in Table 6. All the projects studied for this research are bespoke as the case organization is the supplier to a specific customer. All the projects here are externally initiated and the organization does not sell any proprietary products/services. Projects within the organization are research oriented which are mostly either maintenance projects or evolution of existing products. It is common within this organization for a role to have multiple responsibilities in more than one project. In this study, we conducted interviews with 14 employees in three departments within case organization. The departments are named as Alpha, Beta and Gamma and the interviewed projects are named as P1-P8 for the sake of confidentiality for this research as shown in Table 6. The Table does not show Group test Manger.

<table>
<thead>
<tr>
<th>Department</th>
<th>Project</th>
<th>Team Type</th>
<th>Nr. of employees interviewed</th>
<th>Roles Interviewed</th>
<th>Type of development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha</td>
<td>P1</td>
<td>Large</td>
<td>1</td>
<td>Test Leader</td>
<td>Embedded system</td>
</tr>
<tr>
<td></td>
<td>P2</td>
<td>Large</td>
<td>2</td>
<td>Test Leader, Developer</td>
<td>Windows application and embedded system</td>
</tr>
<tr>
<td></td>
<td>P3</td>
<td>Small</td>
<td>1</td>
<td>Developer</td>
<td>Embedded systems</td>
</tr>
<tr>
<td>Beta</td>
<td>P4</td>
<td>Large</td>
<td>5</td>
<td>Developers-2, Advanced Engineer, Test co-coordinator, Project Manager</td>
<td>Embedded systems</td>
</tr>
<tr>
<td></td>
<td>P5</td>
<td>Small</td>
<td>1</td>
<td>Developer</td>
<td>Windows application</td>
</tr>
<tr>
<td></td>
<td>P6</td>
<td>Small</td>
<td>1</td>
<td>Developer</td>
<td>Windows application</td>
</tr>
<tr>
<td></td>
<td>P7</td>
<td>Large</td>
<td>1</td>
<td>Domain Expert</td>
<td>Embedded system</td>
</tr>
</tbody>
</table>

Table 6: Overview of projects included in this study
As discussed earlier, the aim of this study was to understand the phenomenon of testing in automotive domain. It is due to this reason the interviews conducted were not concerned with the population of people being interviewed. However, the roles selected and their experience in testing seemed quite sufficient to understand the nature and relations between test activities being conducted at the case organization. Apart from the roles interviewed shown in Table 6, test group manager is also interviewed which is not shown in the table. It is due to the reason that this person is responsible for testing within alpha department and is not directly associated with any of the projects in that department.

Majority of the projects studied develop automotive embedded systems as shown in Table 6. Embedded systems are the one which involves software and hardware parts such as control units, hydraulic parts, etc where as windows applications involve only software. Apart from project P3, P5, P6 other teams are large teams (i.e., medium to large teams but named as large teams for sake of simplicity in this research). Small teams don’t necessarily focus on having a structured development and test process, roles & responsibilities, test methods or tools. Three projects (P3, P6, and P8) did not report any test planning activities. Projects with more number of modules have large teams and these projects are old compared to the projects dealt by small teams and have been evolving over the years. These projects are in turn complex in nature; however they started as small, less complex projects involving less number of testers working for them.

Few projects focus on developing windows applications for the embedded parts, for example Project P2. Different software development methodologies are employed within the organization. However model-based development is the prominent one (used in projects P4, P5, P7 and P8) used with waterfall model concepts such as sequential process involving requirements, design, component development, integration and testing. Others development methodologies such as Agile development (i.e., scrum used in one project) and Waterfall model alone are used. Small teams involving maintenance of previous projects usually have Ad-hoc methodology adopted. Two projects recently introduced some agile practices to incorporate iterative development.

Varieties of tools are employed in the projects for testing such as Test generators which assist in test case generation, Test execution tools which are used to execute test cases in controlled environment, Defect detection & management tools, Debugging tools, Requirements traceability and configuration management tools and also tools for modelling and analyzing Electronic Control Units (ECUs) (For Example see Appendix D). Apart from
these tools customized tools are used in some projects when any other tool cannot serve the specific purpose of the project. These tools are usually meant for test execution which make test environment close to target environment. Small teams, for example P3 does not seem to use more number of tools for test activities, they use spreadsheets instead. Large teams which have several modules use lot of tools for organizing and managing test artefacts. More information regarding testing tools and methods used by the teams is given in Appendix E in detail.

Overview of test levels and techniques used in the projects is given in Figure 1. As can be seen from Figure 1, almost all projects (6 out of 8) had Unit testing in place and in five projects Integration testing was used. Unit/Basic tests used in the projects were similar to smoke tests performed to ensure that the system works normally. However, the Unit tests in this context do not have a well defined scope. Half of the projects studied used automation; however, testing was partially automated. In essence, only test specifications or test cases were automated and the evolved test cases were not always updated into automation builds. From the interview data, it is evident that system integration test is also not performed by many teams. However, most of the teams assume integration test can replace system test. As shown in the figure, other forms of testing like Regression and exploratory testing were found to be less common and are gaining importance recently within the context of this case study.

![Test Methods used](image)

**Figure 1: Test Methods Used**

The above results are intended to give an overview of the projects studied in the case study and does not go into details of testing in the case organization. Further sections present a more clear analysis of the processes, roles and other test activities adopted in practice.
5 Qualitative Data Analysis

This section explains the qualitative data collected from the interviews. Firstly, the development and test processes being used in the case organization to develop automotive software are explained. Furthermore, the identified test practices are grouped into challenges and strengths in the later sections. Finally, the results of the process assessment using VSM tool for the test process is entailed.

5.1 Software development process in practice

This section describes different processes used in the case organization which result in the final software product. The engineering activities performed at the case organization depict model-based development for most of the projects, waterfall model for some projects and agile development model for very few projects.

A model-based development process is for projects which involve hardware, software, and control engineering (for ex: Projects P1, P3, P4, P7, P8). This process starts with a requirements phase, in which the requirements of the functionality to be realized are being specified textually or sometimes agreed between customer and the organization. Following that, the development approach is characterized by integrated deployment of executable models for design and implementation, using commercial modeling and simulation environments such as MATLAB/Simulink, etc (See Appendix D). These tools generally use block diagrams, charts as modeling notations with which helps to analyze the system, visualize results, testing validating and documenting the models. Figure 2 adopted from [31] shows this in detail.

![Figure 2: Brief Structure of Model-Based Software Development](image)

Early in the development procedure an executable model of the control software is developed, which can be simulated as well and tested. This model is used throughout the development process for automatic or manual coding of the embedded software. This is later transformed into C code. As compared to traditional software development, where phases are clearly separate, model-based development shows the phase’s specification, design and implementation to have grown together much more strongly.

A blend of Waterfall model and model-based development which follows an iterative way in order to integrate required functionality into the product
considering requirements priorities is used in some projects (mostly large teams such as P1, P4, P7). The main engineering processes are requirement specification and management, component design, component development, integration and validation. We noticed that testing is performed to verify and validate the functionality of the software being developed during each of the development, integration and validation processes. Errors are detected in each of these processes which are analyzed, fixed and capitalized in the defect management system. The figure 3 depicts the way in which different activities of software development in projects studied are carried out.

![Figure 3: Waterfall model and model-based development methodology](image)

The requirements and design move in a normal way where as component development, integration and testing are carried out in an iterative fashion. Requirements and design are not emphasized much after proceeding to development of functionalities. The team structure at the case organization is development-centric where there is information exchange from other disciplines such as requirements, design and testing to the development team. Disciplines such as requirements, design, testing does not necessarily consist of a dedicated team (except for 2 teams which have dedicated testing team). The information flow between different disciplines to the team can be seen in Figure 4.

![Figure 4: Information flow within various disciplines](image)
5.1.1 Test process at the Case Organization

Majority of the interviewees (9 interviewees) stated that there is lack of clear testing process which can be applied to any project lifecycle. Among 8 projects studied only 3 projects have a clear testing process. A clear testing process defines what functionality or quality is to be achieved; also it must be compatible to the hardware being used. It is observed that each project follows a process very similar to what is shown in Figure 5 below. However, each project doesn’t follow all the activities defined in this identified test process.

Even though for most of the projects there is no proper test strategy planned from the beginning of the project, the tests are planned for every project. A test strategy of an organization describes which type of tests need to be conducted and how they should be used with development projects with minimum risks [40] where as a test plan is a part of test strategy. The test strategy used here is majorly black-box testing and a minor part of white-box testing is also performed. There is a tester’s handbook available within the organization which describes test processes, methods and tools within the organization. However, this study shows that it is not implemented/used by most of the teams. As shown in Figure 5, the test process consists of four major phases. These are: Test Planning, Test analysis and design, Test build, Test Execution and reporting. Among these, test planning is done in advance by five projects (3 large teams -P1, P2, P4 and 2 small teams – P5, P7) although one of these teams has a test strategy to conduct basic tests and not exactly test plans. Most of the small teams did not have any software test plan even though they had a very flexible test strategy/approach to carry on with tests.

**Test Planning:** This activity aims to address what will be tested and why. The entry criteria for this activity is have prioritized requirements ready for the release. The deliveries of this phase are software test plan where estimations and scheduling of all test artefacts and resources required for testing are done. The test plan has all the test techniques, tools and test environment planned for the test process. The roles involved in this phase of testing are customer, project manager and test leader. If there is no test leader available for the project, the developer itself participates in the test planning activities. Delivery of a test plan approved by customer and project management is the exit criteria for this activity.

**Test analysis and design:** This phase of testing aims to define how the tests (by defining test data, test cases and schedule progress for the process or system under test) will be carried out for which software test description document is prepared. Software test description also defines what tests (i.e., test techniques) will be performed during test execution. The other deliveries during this phase are requirements traceability matrix, test cases and test scripts design to fulfil the test cases. Test cases are written and managed using test case management tools which are used in all projects. The
requirements traceability matrix which consists of traceability between requirements and test cases ensures proper requirements coverage. The criterion to enter this phase is to have the software test plan approved by customer and project management. The test plan scheduled in the previous phase is updated with all detailed schedules for every test activity. The roles involved at this stage are test leader or a test co-ordinator who is responsible...
for designing, selecting, prioritizing and reviewing the test cases which are supposed to be tested. Since there are no testers or test analysts in most of the projects the developer himself/herself is responsible to write test cases for his code. The project manager is responsible to supervise the activities being carried out at this phase.

Test Build: In automotive software testing, test build is the most vital part of test process since it involves building a test environment which depicts the target environment. The outcome of this level is having hardware which can be visualized as real time environment, test scripts and all other test data. Since the case organization involves working with control engines and Electronic Control Units (ECUs) [5] for most of the projects, modelling tools such as Simulink along with MATLAB are used to visualize the target environment. Mostly testers or developers are involved in this activity. The project manager is responsible to provide resources such as hardware equipment and test leader to supervise the activity.

Test Execution and reporting: The final stage of test process is to execute tests and report the results to the customer. In order to execute tests, the test leader or project manager will chose an appropriate person to run the test scripts using test cases. After the tests are done the results are recorded in defect management system. These results are later analyzed and evaluated to find if there are any differences in the previous test reports of previous releases. The outcome of this phase is software test report which describes the entire tests carried out and summarize if they have passed or failed. If there are any serious errors they are rectified and tests are repeated again. The project manager is responsible to decide the stopping criteria for test execution.

5.1.2 Roles and responsibilities within testing

Most of the projects (except for two teams) in the case organization did not have any dedicated roles for testing but responsibilities for most of the employees working in a project include testing. The appointment of employees in various departments is shown in Figure 6,
The communication between roles with respect to software testing activities is presented in Figure 7. The main roles involved in test process that directly conduct testing activities are Test leader, project manager, developer/tester and customer. Since some of the projects involved in this case study does not necessarily consist of a test leader/test co-ordinator it is left uncoloured in Figure 7. It means that this position may not exist for every project in the case organisation.

Lines of responsibilities (See Figure 7) indicate that the tasks are being given from one role to another whereas lines of communication indicate that the task being done is reported to or supervised by a specific role. This communication lines may not apply for some small teams and the teams following agile practices such as stand-up meetings, open space etc.

5.2 Strengths in test process

In our case study it has been observed that testing in automotive domain has its own set of strengths. Even though automotive domain is known for its complex functionality development, some of the test practices are considered as advantageous to the test process. These strengths are found to be dependent on team size. Most of the practices considered as strengths in small teams were not perceived as strengths in large teams and vice versa. The identified strengths were found in four different categories. The distribution of various strengths across the studied projects cannot be grouped into the classes predefined in the research method from the interview data available. It is due to the reason that the strengths are different for different projects. However, it is evident from the interviews that the strengths vary with team size. They are discussed below.

S01: Test Process/activities:

In small teams test activities are flexible and don’t necessarily generate test reports. As one interviewee from a small team specifies “we don’t need to spend hours on writing test reports, because we know what we want to test, and we know who needs the results” which motivates the above sentence. However, large teams also adopt flexible test process but for small releases only. Otherwise usually they have a structured approach for testing. Small teams seem to practice continuous integration and iterative development which makes them test for every small iteration. This is well said by an employee “We have some small iterations as well; I mean you can do 10 iterations in a day. For every iteration, we integrate more functionality into the system”. One project (P2) which uses agile development (Scrum) has more strength with respect to testing activities such as continuous testing, sprint planning, etc. This makes
Figure 7: Communication between Roles w.r.t Testing

it easier for them to plan tests for every iteration compatible to requirement specifications having test planning (i.e., scheduling) for all iterations done in the beginning of the project itself. This in turn enables alignment of testing with other activities (such as requirements, design, etc) properly. An employee working in an agile team adds “I think we spend 1/3rd of the product development time on testing which improves test efficiency”. The test cases in large teams are most of the times reused with which they avoid rework and time on testing is spent efficiently working on other test activities.

S02: Communication:

Strengths regarding communication are found in a project having agile practices such as stand-up meetings, regular stakeholder collaboration and working together/open space. Every activity involves a test person which indicates parallel testing effort with development. In addition to this agile approach enhanced the team spirit with which efficient interactions are incorporated making them a cross-functional team. This is well motivated by an employee telling that “There is quick communication between us. We get quick feedback on everything”. Other projects use weekly meetings and other electronic services such as email and communicator for communication with different stakeholders in the project. With respect to this another employee working in same team pointed out that “Good communication with customer and within the team, Quick feedback will make us work on stuff that the customer wants”.

28
S03: Roles and Responsibilities:

Small teams consider having one person to perform tester & developer’s role as strength since this wouldn’t delay the process for having to wait for someone to test the software. An employee quoted as follows with respect to this context “While we are working, since the tester is the same person as the developer, there is no delay in reporting it. So if the Developer/Tester finds out the fault he knows where it is introduced, and instead of blaming someone else, the developer becomes more careful while writing the code”. However, large teams don’t consider this as strength; most of these teams don’t have any dedicated testers (except one large team which has dedicated testing team).

S04: Test techniques, tools and environment:

In small teams less number of testing tools and methods are used to avoid more documentation. These teams generally have less project modules when compared to large teams. In this case the system is well known to the tester/developer (Developing and testing done by one person in small teams) which makes it easy to test using minimum number of tools and methods. Small teams (For Example, projects P3 and P6) generally perform smoke or unit test which tests the basic functionality of the system and then have an integration test. An employee conveys the use of Unit/basic test in the following way: “I think unit testing is strength. With this one goes into details and make sure that each and every subsystem works as it is supposed to”. Tools for testing used here are developed by teams to suit the project requirements. However, these customized tools developed for their specific team are not shared among the teams. The main focus in small teams is to have a test environment that has same hardware and same interface as in target environment. This makes testing maintain internal efficiency.

Contrary to the small teams, large teams use a variety of methods and tools for testing to perform multiple activities. One of the most perceived strength found in large teams is experience-based testing (For eg., projects such as P1, P2, P4, and P8). As the same team members have been working on the same project over the years, they find it easy to use their experience based knowledge in product development and testing which replaces software metrics. An employee responsible for quality co-ordination in a large team says “The metrics used for testing are not very helpful to us as a team as testing is more based on our experience with which we decide what types of test cases we need to run and all”. The other perceived strength is exploratory testing/session-based test management (For e.g., projects P1 and P2). An employee pointed out “Executing charters for session based tests (i.e., exploratory tests) we find critical bugs at a more detailed level”. Hardware in the loop (HIL) is also considered as strengths for one of the large teams since it detects most of the defects during integration test level. HIL used for integration and system level testing is perceived as strength as it detects the most critical defects such as timing issues and other real time issues for large and complex systems. Only one team practices test driven development for which test
cases are designed before the coding starts. Informal code reviews are considered as strength in large teams even though they are also used in small teams. Informal code reviews avoid testing getting biased since it is performed by the person other than the one who is responsible for coding. Coming to the tools, Test case management tools are considered as an advantage in large teams (For e.g., P4) as one employee pointed out “I think test case management tool is a great way to store the test cases and to select the tests that should be performed and also for the tester to provide feedback”. Other tools which are considered useful are defect management tools (for e.g., projects). Test environment in large teams is quite good for testing as it depicts real time environment.

5.3 Challenges in test process

In order to address the research gap, we identified challenges in test process. The identified challenges were found in different process areas. However most of them were found not only in testing but also other process areas related to SDLC. For the reasons of clarity we have categorized the observed process areas for this research which can be found in Appendix B. Also the terms used as challenges here are explained in the coding guide in Appendix C. Table 7 below provides an overview of the challenges identified in different process area with respect to software testing context. For further details on class, type refer to Section 3.1.4, subsection I.

<table>
<thead>
<tr>
<th>ID</th>
<th>Class</th>
<th>Challenges</th>
<th>Type</th>
<th># of Projects</th>
<th>Process Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>C01</td>
<td>General</td>
<td>Organization and its Processes related issues</td>
<td>Global</td>
<td>6</td>
<td>Requirements, Test Process, Test management, Project management</td>
</tr>
<tr>
<td>C02</td>
<td>General</td>
<td>Time and Cost constraints related issues</td>
<td>Global</td>
<td>5</td>
<td>Requirements, Project management, Test Level(Basic/Unit test)</td>
</tr>
<tr>
<td>C03</td>
<td>General</td>
<td>Requirements related issues</td>
<td>Global</td>
<td>3</td>
<td>Requirements, Test Process(Test Planning), Project Management</td>
</tr>
<tr>
<td>C04</td>
<td>General</td>
<td>Resource Constraints related issues</td>
<td>Global</td>
<td>3</td>
<td>Test Process, Project Management</td>
</tr>
<tr>
<td>C05</td>
<td>General</td>
<td>Knowledge Management related issues</td>
<td>Global</td>
<td>5</td>
<td>Test Management, Project Management</td>
</tr>
<tr>
<td>C06</td>
<td>Very Common</td>
<td>Interactions and Communications related issues</td>
<td>Global</td>
<td>3</td>
<td>All Processes in SDLC</td>
</tr>
</tbody>
</table>
a) General Issues:

The most general issues identified in the current test process are as stated below. This challenge is noticed in six projects.

**C01: Organization and Its Processes Related Issues:**

-**No Unified Test Process:** The main challenge automotive software development poses is no unified testing approach. Even though, the tester’s handbook available within the organization has a defined testing approach, most of the teams are not aware of it or don’t feel that it suits their project characteristics. Each project uses different methodologies, testing methods and tools which make it hard to come up with a unified testing process that suits all projects. As interviewees pointed out “It feels there is lack of structured testing process and it is also un-organized always. It works fine for small projects but not for large projects”. The un-structured testing approach can work for small teams but not large teams. This means that testing approach used in the beginning when the project started (like many years ago) does not work now for a huge system due to the evolving complexity of the overall system making it hard to test. With this kind of testing sometimes the quality is compromised. However, another interviewee added that “I think it’s very hard to make good process for the development in this kind of environment (automotive domain) as it is always evolving”. Developing a perfect test process is one of the biggest challenges faced by automotive domain possessing scattered functionality with evolving complexity consisting both software and hardware.

-**Testing is done in haste which is not well planned:** This usually happens when the development gets prolonged and delivery date is not extended which results in less focus on testing and there by low test coverage. With this regard an interviewee pointed out “Sometimes I think the product delivery date cannot be extended which means less time to do testing”. Also the other
interconnecting cause for this issue can be lack of hardware on time from the
customer to test early which usually results in low respect for deadlines. One
interviewee nicely summarized this by saying “The hardware from customer
already contained lot of faults...so they understand that we cannot deliver”.

-Stakeholder’s attitude towards testing: Testing is always not well focused
and may be considered as a less attractive activity by some employees. Some
employees at the life cycle management level complained that change
management for testing is considered as overhead since the improvement
activities always concentrate on development. New approaches for testing
used within the organization do not get much support within the
organization initially which sometimes makes teams to develop their own
methods and tools where greater effort is needed.

-Asynchronous testing activities: Some of the tasks (For Example, Testing)
performed by the case organization don’t get synced with the activities
performed by the other contractor working for the customer. It is due to this
problem that the delivered test artifact sometimes need to be re-structured in
order to synchronize with the artifact supplied by third party i.e., contractor.
This leads to rework with respect to testing.

C02: Time and Cost Constraints:

Challenges regarding time and cost constraints can be due to insufficient
time spent on requirements, testing activities or test process which is
observed in five teams.

-Lack of time and budget for specifying validation requirements: validation
requirements here refer to the requirements which are validated during
testing with the help of test cases. An interviewee adds “Re-write customer
specifications into our own requirements? That is not possible today due to the
reason that customer will not pay for it and we don’t have internal budget for that”.
Even though this saves time and money for the moment, it will lead to lot of
rework in the later stages of the project i.e., testing. The requirements are
often broken down to various types like system requirements, design
requirements, validation requirements, etc. Of these validation requirements
specify various factors like environmental conditions in which the system
needs to be tested. At the moment the validation requirements are not being
provided by the customer. As a result of this the case organization has time
and budget constraints to develop validation requirements and thus the
overall testing approach lack well defined scope and objectives.

-Unavailability of Test equipment on time: The other factors that lead to this
challenge are unavailability of hardware for testing which leads to
implementation of tests incompletely. As one interviewee complains “We are
always late with the delivery we need to do, it is due to this reason we are unable to
perform unit testing”. It is due to this reasons the time spent on testing is not spent efficiently.

C03: Requirements related Issues:

Insufficient requirements for testing, high-level requirements which are hard to understand, and requirements volatility are the challenges that hinder performing proper testing to achieve high quality. These issues generally occur when customer does not specify requirements properly due to lack of time or lack of knowledge which implies poor requirements management. This challenge is identified three teams and specified by 10 interviewees in the current study scenario. Even though it is majorly identified in three teams, it had negative impacts in other teams also.

Lack of requirements clarity: One of the employees specifies “I think it would be better for us in beginning to put greater effort with requirements management to avoid customer complaining about misunderstanding/misinterpreting the requirements specified by them, in order to have fewer issues at the end and save time involved in changing and testing everything repeatedly”. But this is always not happening due to the limited amount of time given for requirements. There are several reasons for this time constraint, for e.g., some of them can be to get the budget sanctioned or to gain competitive edge over other markets. If requirements are not managed in early stages, it consumes so much effort in re-interpreting them in later stages. Due to this process, there is lot of inefficient time spent on specifying and developing requirements, which consumes the quality work planned for testing phase.

Requirements Volatility: Automotive software development is also associated with several critical aspects such as growing complexity and functionality which makes it hard to stop requirements volatility and therefore making it unable to specify certain stopping criteria for testing. During the interviews conducted as part of the case study a software developer states that “It is not easy to write down stable requirements for what we do here (automotive development). So for most of the requirements in software development lifecycle, we just agree them with the customer which is for the testing as well.” Due to this fact, the stopping criteria for testing is always time and budget constraints and not proper test coverage.

Requirements traceability management issues: The testing process must ensure that every output relative to the software requirement is traceable. Traceability between requirements and test cases is the common challenge found in automotive development. With good traceability, it becomes easy to
manage changes in requirements; one interviewee said “Traceability from requirements to test cases must be a bit better so that you can more easily determine which test cases to update when customer changes requirements”. Good traceability also acts as a proper aid to define appropriate test coverage. Traceability in automotive domain is challenging since requirements are volatile due to which they are not always written formally. Sometimes even though requirements are written formally they are written on a high-level of understanding which becomes complex to connect to functionalities and their test cases.

C04: Resource Constraints:

This challenge is majorly identified in three teams and expressed by more than 1/3rd of the employees interviewed. However, the effects of these issues are found in other projects also which didn’t have a dedicated testing team.

-Lack of dedicated testers: With regard to this challenge one interviewee points out “We don’t have a dedicated test person in our project. It’s the developer that when he gets a requirement he/she has to go through the requirements and analyse it so that they understand what are the test cases to be written”. Lack of independent verification and validation team negatively influences the test coverage and therefore the quality of the product. Another interviewee said “Ideally I think those should be two different people implementing the test case and the code so to speak” which means that testing gets biased if the same person is performing both. In consequence, testing scope is reduced as the person’s developing skills may influence the testing activity.

-Unavailability of personnel for testing: Test cases in automotive domain always need experts to write them. So, experienced testers are needed for automotive software testing. An interviewee who manages testing says “It is a difficult to find people with same experience and also they take quite long period to learn and get to know about the product due to its complexity. For this one need to have same knowledge before being able to do testing”. It is also hard to find a replacement for such experienced testers if someone quits or is shifted to another project. For this to be easy one must possess same experience and knowledge on how the system works. Finding the suitable person to perform bug fixes is also hard when he/she is currently not associated with the project.

C05: Knowledge Management Related Issues:

The issues related to knowledge management found in this case studies (identified in five projects) are:

Knowledge Transfer and Knowledge Sharing Issues regarding testing: There are many new testing techniques and tools being developed in automotive development upcoming these days. As one interviewee says “The most
An important drawback about using this new testing (exploratory testing) technique is that the testers need to be quite experienced with using it. It is not easy to have such environment since the testers are always changing and new testers must be trained well before they start testing. This knowledge transfer is not possible right now. In order to use new testing techniques, there is no sufficient information available on how to use them effectively. The knowledge regarding these activities must be transferred to others and shared within the organization which makes it easier to switch roles and responsibilities. One of the interviewee pointed out that “we are looking ahead to work in such a way that the project is not dependent on any single person”.

An important observation here is that even though there is tester’s handbook available within the organization, most of the employees are unaware of it. This makes knowledge management another challenge for automotive industry where there is much emphasis on control engineering, mechatronics and electrical engineering concepts and not software engineering concepts. Knowledge management of software test techniques and tools used in past and present provides decision support to make appropriate plans for future projects.

-Lack of testing fundamentals: Testing is given low priority due to which employees lack knowledge in fundamentals of testing and improper knowledge management which is a direct impact of the above stated issue. With regard to this context an interviewee involved in life cycle management activity stated that “I think there is lack of information on testing fundamentals. Some of us don’t know when to start a test level and when to end it and it feels like grey areas which is not clearly defined anywhere”.

b) Very Common Issues:

These issues are stated by more than 1/5th of the interviewees during the interviews

C06: Interactions, Communications Related Issues:

-Lack of regular interactions with customer regarding requirements: The issue in this context can be lack of regular face-to-face meetings with customer where content of the meeting is clear with proper scope. Another issue in this context is that even though there is an interaction with customer daily, the requirements for testing are not properly understood which is due to the unavailability of the right person to communicate at the customer’s side. The requirements are elicited quite well during the initial stages of the project and in the later stages i.e., during testing there is less customer interaction in some projects which leads to delay in time plans.

-Lack of interactions with other roles within project during testing: Another possible issue in this context can be unable to communicate with employees who have worked on the project previously. One interviewee narrated it in
the following way “I have allocated a person for our team and then he have to communicate with us but it has been sometimes quite tough for the person to find the person since he is working for another team now”. This occurs when a developer is being shifted to other projects after completing the coding part. This makes him unable for the testing activity where he is responsible to fix the bugs that are detected.

-Informal Communication with customer: An important observation that occurred while eliciting the reason behind unclear requirements is sometimes informal communication between customer and the developing organization. It is due to this reason sometimes developing organization cannot ask the customer strictly to have clear requirements. A manager adds “I think it is most critical to maintain the relationship (informal relationship with customer) and demand the customer that we cannot start working before you tell us what you want”. One can start developing even though the customer delivers his requirements with a very vague description but this always lead to rework if the customer is not expecting what is being developed. This always leads to low respect for deadlines.

The above stated issues are found in three teams.

C07: Testing techniques, tools and environment related issues:

The following are the issues related to test techniques, tools and environment.

-Lack of automation leading to rework: Generally used test techniques in automotive domain are scripted tests in the case organization. But the scripted tests are perceived as repetitive and less challenging. Automation of tests such as unit test and regression tests is not being done effectively since there is less support for such activity within the organization. Automated test case generation could avoid most of the rework in testing but unavailability of proper tool for doing that in model-based testing is observed as another challenge. One interviewee pointed out that “Testing is rework as long as it is not properly automated”. Scripted tests are designed to test the basic functionality of the system and don’t allow any creativity. It also consumes more resources to create test scripts for the pre-defined test cases. However, other testing techniques such as exploratory testing (session-based test management) are used in very few projects within the organization which entirely depends on tester’s intuition and skills.

-No unified tool for entire testing activity: One test lead pointed out the need for unified tool which can be used for testing “we have lot of tools for testing but there are some difficulties in deciding which tool to use since there are drawbacks and strengths for every tool being used. Sometime we are forced to develop customized tool because we cannot get any tool from the market that does everything for us”. A tool which does all activities in testing for automotive domain can
be easy to use instead of managing and organizing large number of tools used right now.

**-Improper maintenance of test equipment:** Maintenance of test environment is also an issue because if it is done properly lot of rework and inefficiency can be avoided. One interviewee nicely summarized this as “We have several test environments and test steps to be maintained. They are not always maintained and it takes long time before one can get started with actual testing”. From the interview data there is no specific reason found for this improper maintenance of test equipment. It may be due to the fact that this activity is not mandatory in the process. However proper test environment can be an issue in other case when it is not available from the customer on time which is discussed in the earlier challenges.

**C08: Quality Aspects Related Issues:**

**-Reliability Issues:** Most common quality attribute that is not being satisfied within automotive domain is reliability as one interviewee specifies “It’s hard to achieve several requirement criteria for a system such as working for longer period of time, less resource intensive, ability to work on different platforms, etc”. Also quality cannot be incorporated due to improper test process or faulty hardware components.

**-Quality attributes are not specified well right from the inception of project:** Although some measures are being taken to overcome this challenge as said by one interviewee “we had some quality issues on existing products in the market mostly because of the complex system that needs to be tested but right now we stop delivery if testing is not done well”. Other observation in this regard is that non-functional requirements are not properly specified since the inception of the project which leads to above challenges.

**No quality measurement/assessment:** The software testing done in this context is alpha testing being conducted at developer’s site which is employed as a form of internal acceptance testing. Since beta testing is done by customer, the tested artefact delivered by the developer is not expected have very high levels of quality. In this context, an employee proposes as follows “The quality curve must be better although our customer is satisfied. I think the quality measures should be documented in order to facilitate better analysis of test results”. No proper quality assessment standard is used within the organization which is specific for automotive context which also facilitates use of quality measures for analyzing test results.

**C09: Defect Detection Issues:**

**-Testing late in the process makes it costly to fix defects:** It’s hard to find defects in automotive development due to the evolving complexity of the system when testing is done at the end. Small changes to the existing system
require more effort and time to detect defects. One of the interviewees summarized it well by saying “when we started working with this system it was something really small in the beginning which evolved with time and testing has not been taken care of properly as development. Now if something doesn’t work (during implementation and testing), it takes lot of time to find the defects”. Another reason for this can be that the components being integrated were not continuously tested i.e., usage of big-bang testing left some bugs undetected. This led to a large number of defects undetected at the end which resulted in release with high number of corrections for faults reported by customer.

-Hard to track defects which are not fixed in the previous releases: For development with complex parts (i.e., which involves working with timing issues and other critical issues) the difference in the behaviour of the system between two different releases need to be same. But this is not always happening due to the errors which were not fixed during the previous releases being triggered in the current release. This is because these errors may become serious in the next releases when they become untraceable in such a huge system.

c) Common Issues:

C10: Documentation Related Issues:

- Documentation regarding test artefacts is not updated continuously: The interviewees emphasized that the documentation (such as test cases and other test artefacts) provided was not enough for testing and cannot be trusted for quality work; one interviewee added that “The test documents are not updated continuously, so we find them unreliable”. One of the reasons mentioned was there were small changes being done to the test artefacts which are not always updated to the test document. It can be for e.g., with the test cases as one interviewee pointed out “For every new version of system being developed, we add new test cases. These test cases are not always updated to the existing test cases in the repository”. It can be due to human errors which lead to rework to find out the missing test cases which are not updated.

- No detailed manuals available for some specific test methods and tools: Another observation in this regard was no proper documentation on how tools and methods work which can be used. One interviewee nicely summarized this as “There is support for tools but we always can’t find someone who can fix the problems with them. It could be better documented I guess”. However, it is observed that there are manuals within the organization which serve this purpose. But for some specific tools (such as customized tools) or methods, this doesn’t work. This issue seems to arise when people performing testing could not understand the terminology in manuals or they are not aware of these manuals.
All the above discussed issues vary from project to project depending on their characteristics. The motive of the above results was to not to generalize our findings to all projects rather to give an in-depth understanding of what can be the causes and effects of various challenges. To this end, we also provided challenges with respect to the number of projects they were found in Table 7. This will further enable the case company to address the challenges with respect to their occurrence in various projects.

5.4 Process assessment using VSM

In this section we present the findings of the process assessment carried out using value stream mapping tool. First we present a current state map to discuss how the challenges and strengths identified through interviews lead to non-value added and value added activities respectively. Then we present solution proposals to the wastes identified. Finally, we present a future state map after an analysis of waste and value. However, for this study it was not possible to extract the time line of the processes within testing activity. The data collected from the interviews was sufficient to be used to identify waste and value in the testing process. It was also evident from the interviews that there was no planned time stamp for every activity within the testing. Sometimes, this can be seen as strength as it leads to a flexible process but sometimes this can lead to many issues ending up in delivering a low quality product. Figure 8 gives an outline of steps involved in using the VSM process.

![Figure 8: VSM Process Steps](image)

A. Current Value Stream Map:
We performed a process activity mapping with which we visualized various activities carried out within test process. This section presents the current value stream map which provides an overview of wastes identified in VSM and the interviews. However, value-adding activities are not shown the current value stream here as the study mainly focuses on identifying waste and improvement potentials. Anyways, the values created by various strengths are identified for various team sizes which are presented in Table 8. The value-adding activities V01- V05 as stated in last column of Table 8 are predefined terms whose definitions can be found in Appendix C in the coding guide provided.

<table>
<thead>
<tr>
<th>Strength</th>
<th>Team Size</th>
<th>Description</th>
<th>Value Addition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less Documentation in testing</td>
<td>✓</td>
<td>More time is spent on delivering proper functionality.</td>
<td>V01</td>
</tr>
<tr>
<td>Basic/Unit Test</td>
<td>✓ ✓</td>
<td>Enables to deliver proper functionality.</td>
<td>V01</td>
</tr>
<tr>
<td>Integration test</td>
<td>✓</td>
<td>Incorporates efficient way of testing by detecting more defects in less time.</td>
<td>V01, V03</td>
</tr>
<tr>
<td>Test environment depicts target environment</td>
<td>✓ ✓</td>
<td>Better deployment of test process with test environment (V03).</td>
<td>V03</td>
</tr>
<tr>
<td>Experience Based Testing</td>
<td>✓</td>
<td>Incorporates quality aided by tester’s skill and knowledge.</td>
<td>V02</td>
</tr>
<tr>
<td>Exploratory testing/session-based test management</td>
<td>✓</td>
<td>Compatible to the project requirements and aids defect prevention.</td>
<td>V02, V03</td>
</tr>
<tr>
<td>Testing tools</td>
<td>✓</td>
<td>Better organization of test activities</td>
<td>V03</td>
</tr>
<tr>
<td>Continuous Integration</td>
<td>✓</td>
<td>More functionality</td>
<td>V01</td>
</tr>
<tr>
<td>Iterative development and testing</td>
<td>✓ ✓</td>
<td>Better functionality and quality</td>
<td>V01, V02</td>
</tr>
<tr>
<td>Roles and responsibilities</td>
<td>✓</td>
<td>Flexible roles and responsibilities refers that various stock of skills and knowledge used for testing</td>
<td>V05</td>
</tr>
<tr>
<td>Verification activities (Informal code reviews)</td>
<td>✓</td>
<td>Quality Incorporation</td>
<td>V02</td>
</tr>
<tr>
<td>Reuse</td>
<td>✓</td>
<td>Test artefacts from previous releases are organized and reused</td>
<td>V03</td>
</tr>
</tbody>
</table>
The non-value adding activities are identified in the current value stream of test process as shown in Figure 9.

The current state map of the test process revealed all seven kinds of wastes as described in the Appendix C (Coding guide). The seven kinds of wastes identified are partially done work, extra processing, handoffs, task switching, relearning, delays and defects (numbered as W1-W7). These wastes are identified in different activities within test process which cause rework, increase in waiting times or inefficient time spent within entire test activity which can be found in Figure 9. However, the issues that occur in other activities (for e.g., requirements management, etc) which affect testing are not shown in the current stream map shown here. The reason behind their cause and their negative influence on test activities are discussed in the previous section.

**Wastes in Test Process:**

We identified twelve areas (1-12 as shown in Figure 9) in the test process where wastes occur. Below is a description of the wastes that occur in every sub-process as identified in the current stream map. A mapping of identified challenges that cause wastes along with a description of the issues that lead to the waste is shown in Table 9.

<table>
<thead>
<tr>
<th>Waste ID</th>
<th>Challenge ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1: Partially done work</td>
<td>C01, C03, C08, C09, C10</td>
<td>This waste occurs due to partially done work in terms of test activities such as test plan, requirements, quality incorporation, defect detection and prevention and test documentation.</td>
</tr>
<tr>
<td>W2: Extra features</td>
<td>C03</td>
<td>This waste occurs due to extra features developed due to lack of requirements clarity which are otherwise misinterpreted.</td>
</tr>
<tr>
<td>W3: Handoffs</td>
<td>C02, C03, C04, C05, C07, C10</td>
<td>This waste occurs to lack of availability of test equipment on time, requirements for testing, and test personnel with required competence in testing, knowledge transfer and knowledge sharing within testing, proper documentation on usage of test techniques &amp; tools. This waste also occurs when there is no test maintenance activity to save test artefacts.</td>
</tr>
<tr>
<td>W4: Task switching</td>
<td>C04</td>
<td>This waste occurs due to lack of dedicated testing team or test personnel which is due to unclear roles and responsibilities within the team.</td>
</tr>
<tr>
<td>W5: Relearning</td>
<td>C03, C06, C07</td>
<td>This waste occurs due to rework caused by misinterpreted requirements,</td>
</tr>
<tr>
<td>W6: Delays</td>
<td>C03, C04</td>
<td>Delays in the test process to elicit requirements and allocate resources to perform testing.</td>
</tr>
</tbody>
</table>
The waste related to defects occur when there are no early defect detection and defect prevention activities which indicates that testing is done in the end.

**Waste Identified in Sub-process 1:** The waste that occurs here is due to partially done work with respect to planning and execution of tests. As discussed earlier as challenge C1, this happens due to unplanned tests resulting in an incomplete test activity. In order to perform testing which incorporates all quality constraints, the tests must be planned well since the early stages of project with which a distinct set of requirements can be prioritized and validated during testing. Since this activity in the test process is not well-planned, the tests are conducted in unstructured manner which results in low test coverage and delayed deliveries. It can be observed that these kind of activities result in rework which requires additional time, budget and resources. As observed for some projects, the organization initially spends lot of resources in creating a business strategy being unsure whether the customer rejects or commits to it. However, this work does not serve well when the process starts as the requirements as part of the business strategy are seldom implemented and also are not clear. This partially done work results in waste in the testing activities.

**Wastes Identified in Sub-process 2:** The wastes identified here are test effort involved in ‘extra features’ and ‘handoffs’ due to lack of clear requirements that results in a delay to proceed to the next activity. Extra features are the requirements that are either misinterpreted or new requirements proposed by developers that are at times removed from the system prior to release. However, testing is also performed on such features/functions based on such requirements thus leading to waste called ‘extra features’. This waste occurs in the form of effort that is put in writing the test plan and subsequently scheduling tests and allocating resources for those extra processes. Sometimes customer might refuse the proposal or the requirements may be misinterpreted.

Requirements are volatile and lack clarity which takes time to acquire a set of well-defined requirements for development and testing. Thus the test plan approval which allows to proceed for the next phases is delayed due to the waiting time that occur when the stakeholders (i.e., customer and developing organization) decide what to include in the current release. However, it is observed that requirements keep flowing even after the test plan is finalized and the testing activity proceeds. So during this period, it is evident that so much time, effort and resources are needlessly spent which could be saved otherwise.

**Waste identified in Sub-process 3:** The delay here occurs in allocating appropriate test personnel for performing the testing activity. As identified in the case study, one general issue in case organization is resource
constraints. There is a severe delay in allocating trained testers as the case organization does not have dedicated testers and moreover developers themselves perform entire testing activity handling many other responsibilities. The wastes that occur here are lack of availability of testers (W3: Handoffs) and unclear roles and responsibilities as a part of organization structure which hinders the formation of right teams (W4: Task switching). These waste result in delay in waiting for resources.

**Waste identified in Sub-process 4:** In order to write test cases for the requirements, there must be a stable set of requirements to design and analyze the tests that need to be conducted. In this process, waste occurs in the form of motion of requirements (W1: Partially done work) between customer and developing organization for feedback or clarification. These requirements are the ones that are negotiated to be candidate requirements for current release. It is observed that this process repeats itself numerous times involving several interactions with customer since no one has the same view as others on the requirements. Also the forms of communication associated in this context are e-mails, internal communicator or phone calls. This kind of environment does not necessarily create an environment that is created by face-to-face meetings. This kind of communication also reduces team spirit i.e., working as one team as there is a clear lack of face-to-face meetings involving customer and team.

**Waste identified in Sub-process 5:** The delay here again occurs in form of long waiting times (W06: Delays) for eliciting validation requirements to finalize a checklist of test cases to be performed in the test activity. The test cases from the previous releases are sometimes not updated. This takes away lot of time and effort to be spent in rewriting (W5: relearning) the requirements of the previous version and including those test cases in current release. Lack of automation in test case generation is also a reason for this delay as testing is rework as long as it is not automated.

**Waste identified in Sub-process 6:** Documentation regarding testing is not always maintained as discussed in challenge C10 earlier. The test cases from the previous release are not always updated to the test case repository which means undocumented test artefacts (W1: Partially done work). Some of these missing test artefacts can put the testing activity into critical situation which ends in repeating the entire testing again. In order to proceed for the next activity it is made sure if all the test cases are ready for the test build. It is during this phase in the testing activity it is observed that some test cases from the previous release are missing in the test case database for this needless effort must be spent which could be saved otherwise.

**Waste in Sub-process area 7:** Basic tests such as smoke test which are carried out by developers can be carried out as a parallel activity with test build. Some projects need test equipment to perform these tests. The test equipment from the customer is not available (W3: Handoffs) for tests on time.
However, this waste can be minimized in some cases where the test environment used in the previous releases is saved and maintained for the later versions of the product. As identified in challenge C7, there is no specific reason for this negligence.

**Waste in Sub-process area 8:** All the test activities carried out in case organization are managed using different tools which are usually meant to save time. But in practice these tools does not serve this purpose. Instead management and mapping of test artefacts using these tools consume more resources and sometimes redundancy creating unnecessary complexity. A unified tool which can manage and organize all the test activities for automotive domain is not available which makes it a challenge (C7) as discussed in earlier sections and thus creating a waste called handoffs (W3).

**Waste in Sub-process area 9:** Testing is not done as a parallel activity with development. Tracking defects in the end consumes time and money which appears to be a burden on testers leading to huge delays. Verification activities which support early defect detection such as inspections and code reviews are not used by most of teams. Another kind of waste (W3: Handoffs) that occurs here can be due to lack of availability of testers and training for implementing tests using specific testing techniques such as exploratory tests or experience based testing which is based on tester’s intuition and skills. Even though such testing techniques are considered as strength within the case organization, only a limited number of test personnel who have the competence to perform such activities are available right now. This in turn leads to delays in the testing when such experienced testers quit or shifted to another project. However, documentation on how to use testing techniques and tools are not updated continuously (sometimes not available) which cannot be trusted to perform testing.

**Waste in Sub-process area 10:** The quality attributes that need to be incorporated in the tested artefact are not properly elicited since the inception of the project (W1: Partially done work) which leads to low quality product. Some of the interviewees feel that the testing is being done to ensure basic functionality only and thus one cannot ensure the reliability of the delivered artefact. There is a lack of quality standard that is essential to measure the level of quality attributes that are included in the product which enables to compare test results with previous release. The analysis of test results helps to redefine the quality improvements that need to be implemented in the next versions of the product.

Some employees also reported long delays (W6: Delays) for having to wait for the developers to fix the defects after they are reported. This waiting time is seems to be long when the person responsible for the code is shifted to others projects as soon as he finishes his work in the previous project. This could be easy is the testing is performed parallel to development.
**Figure 9: Current Stream Map**

**Waste in Sub-process area 11:** Due to requirements volatility (C3), the requirements specifications are not documented well which leads to misinterpretations of requirements. These tested artefacts when delivered to
the customer reveal that the effort and resources put in developing and testing the misinterpreted requirements are not useful (W5: Relearning). Then after a series of interactions with customer the necessary requirements are elicited and developed which leads to unnecessary rework. However, the misinterpretation of requirements can be avoided by having regular face-to-face interactions with customer. But the communication with customer for feedback and clarification regarding requirements is done using emails or phone calls which end up in having requirements not specified well.

**Waste in Sub-process area 12:** The defects detected in previous releases are sometimes not fixed (W1: partially done work) which is agreed by the customer. But these defects are difficult to track in the next releases as the system evolves. Lack of verification activities and early defect prevention activities (W7: Defects) creates lot of mess before release with which some of the unfixed defects in the current release are left for the next release. This process repeats by itself so many times during each release that as the functionality grows there are many unfixed defects left behind which are unable to trace in such complex system.
6 SYSTEMATIC LITERATURE REVIEW (SLR)

This section describes the purpose and design of our SLR. The purpose of our SLR is to identify testing related problems in the context of automotive software domain and solutions that have been proposed and applied in industrial context.

Our SLR design consists of several steps as discussed below. Our SLR is based on guidelines provided by Kitchenham [21] with the exception that we did not include study quality assessment. This step is excluded as our main aim is to identify problems and solutions related to automotive software as published in peer-reviewed literature and relate to the problems/challenges as found in our case study.

The following sections describe the steps involved in our SLR.

1. Identification of papers
   In this step we formulated search terms so that they enable unbiased identification of research papers. Search terms were elaborated over several test searches in leading digital libraries. To this end we conducted two distinct SLRs, shown as SLR_1 & 2 in Table 10.

   Table 10: Search string formulation for SLR_1 & 2
<table>
<thead>
<tr>
<th>SLR ID</th>
<th>Search string</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLR_1</td>
<td>automotive AND software AND (test OR verification OR validation)</td>
</tr>
<tr>
<td>SLR_2</td>
<td>automotive AND software AND “model-based” AND tool</td>
</tr>
</tbody>
</table>

2. Search venues (Databases)
   Search string is applied on Titles and Abstracts in the databases IEEEXplore, ACM Digital Library, Springerlink, ScienceDirect and Wiley Interscience. We did not apply search string on full text as it is found that such approach generally yields too many irrelevant results.

3. Paper selection
   To select papers relevant to our goal we formulated inclusion/exclusion criteria. First of all, we excluded papers that are not in English, published before 2000 and were not available in full-text. As our goal was to look for problems and solutions offered in peer-reviewed literature, we excluded editorial notes, comments, and reviews and so on. As we
intended to look for solutions that were applied in industry, we included papers with solutions that have empirical evaluations in industry and in particular automotive software domain. A major criterion to include a study was they present solutions to problems in relation to software testing. By software testing, we mean any of the V&V activities spanning across the whole software development lifecycle (requirements validation, test case generation, unit or regression testing and so on). To ensure these criteria are satisfied, papers were scanned against the checklist.

- Is the paper in English?
- Is the paper available in full text?
- Is the paper published in or after 2000?
- Is the context of research automotive software domain?
- Does the paper talk about any problems and solutions or tools related to any software V&V?
- Does the paper contain an empirical evaluation in industrial context?

4. Execution

As previously mentioned in step 2, the search resulted in 221 papers for SLR_1 and 66 papers for SLR_2. As these are low numbers for an SLR, we were not very strict on step 3, i.e. applying inclusion/exclusion criteria on Title and Abstract. We scanned full text of all papers in the databases for data extraction. Overall, we obtained 29 primary studies for SLR_1 and 5 papers for SLR_2. An overview of the distribution of primary studies across databases is in shown in Table 11.

Table 11: Paper selection for SLR_1 & 2

<table>
<thead>
<tr>
<th>Database</th>
<th>Initial search result</th>
<th>Nr. Primary studies</th>
<th>Full text not available</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SLR_1</td>
<td>SLR_2</td>
<td>SLR_1</td>
</tr>
<tr>
<td>ScienceDirect</td>
<td>35</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>ACM Digital Library</td>
<td>12</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>WileyIntercience</td>
<td>5</td>
<td>4</td>
<td>-</td>
</tr>
<tr>
<td>Springerlink</td>
<td>46</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>IEEE Xplore</td>
<td>123</td>
<td>23</td>
<td>12</td>
</tr>
<tr>
<td>Total</td>
<td>221</td>
<td>66</td>
<td>29</td>
</tr>
</tbody>
</table>

5. Results

We found that SLRs (1 &2) had answers to two of the problems identified in our interviews which are: (1) Lack of automation for test case generation leading to rework, and (2) No unified tool for entire testing activity. Some of the identified problems had roots in other software
development activities mainly in Requirements Engineering. Moreover, a need to tackle continuous inflow of requirements and requirements volatility was observed. Based on this we conducted a few other SLRs to investigate similar problems and solutions in automotive software domain. We followed guidelines, as described before, to conduct SLRs.

- **Identification of papers**

  In this step we formulated search strings based on challenges found in our interviews. In this case, three different areas were investigated, namely Requirements, Configuration Management and Agile/Lean software development. Search strings are given in Table 12.

<table>
<thead>
<tr>
<th>SLR ID</th>
<th>Search string</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLR_3</td>
<td>automotive AND software AND requirements</td>
</tr>
<tr>
<td>SLR_4</td>
<td>automotive AND software AND (agile OR scrum OR “extreme programming” OR lean)</td>
</tr>
<tr>
<td>SLR_5</td>
<td>embedded AND software AND (agile OR scrum OR “extreme programming” OR lean)</td>
</tr>
</tbody>
</table>

- **Search venues**

  Like in our previous SLR, we applied our Search strings are applied on Titles and Abstracts in the databases IEEEXplore, ACM Digital Library, Springerlink, ScienceDirect and Wiley Interscience. We did not apply search string on full text as it is found that such approach generally yields too many irrelevant results [ ].

- **Inclusion/Exclusion criteria**

  The following general criteria are placed on the four search strings.
  - Is the paper in English?
  - Is the paper available in fulltext?
  - Is the paper published in or after 2000?
  - Does the paper contain an empirical evaluation in industrial context?

  In SLR_3 and 4 we included papers only in the context of automotive software domain. In SLR_2, we included papers where the main point of discussion was on RE challenges, requirements definition/specification, requirements clarity, requirements volatility and management. Based on SLR_4, we extended our search to the entire embedded software domain as we could retrieve only few articles on agile in automotive software.
• **Execution**

Search results for each search string are given in Table 13. Search strings are applied on Title and Abstracts.

<table>
<thead>
<tr>
<th>Database</th>
<th>SLR_3</th>
<th>SLR_4</th>
<th>SLR_5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
<td>Selected</td>
<td>Total</td>
</tr>
<tr>
<td>IEEEEXPLORE</td>
<td>163</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>ACM Digital Library</td>
<td>102</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>SpringerLink</td>
<td>31</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>Wiley Interscience</td>
<td>5</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>301</td>
<td>10</td>
<td>12</td>
</tr>
</tbody>
</table>

• **Mapping overall results**

We mapped the identified challenges and solutions offered in our SLRs to the challenges found in our interviews. These are given in Table 14. Based on our SLRs, we proposed seven solution proposals and discussed in the subsequent section. As seen in the table the 5th column shows the mapping to the related work in the literature which shows that the issues identified in this study are also observed in previous studies. The column 4 i.e., sources shown from where the solution proposals in the last column are obtained. Column 5 shows the traceability of the identified issues to literature and column 6 represents the solutions proposed for each issue. However, there cannot be a single solution proposal for each issue. It entirely depends on the type of projects and appropriate strategies (such as resource management, budget management) adopted by teams to implement these solutions.

<table>
<thead>
<tr>
<th>Nr.</th>
<th>ID</th>
<th>Challenge</th>
<th>Sources of Solutions taken</th>
<th>Traceability to other issues in Literature</th>
<th>Solution Proposals Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C01_1</td>
<td>No unified test process/approach</td>
<td>[S18]</td>
<td>-</td>
<td>Test Management</td>
</tr>
<tr>
<td>2</td>
<td>C01_2</td>
<td>Testing done in haste and not well planned</td>
<td>[S43]</td>
<td>[S23]</td>
<td>Agile Incorporation</td>
</tr>
<tr>
<td>3</td>
<td>C01_3</td>
<td>Stakeholders attitude towards testing: low priority</td>
<td>[S43]</td>
<td>-</td>
<td>Agile Incorporation</td>
</tr>
<tr>
<td>4</td>
<td>C01_4</td>
<td>Asynchronous test activities</td>
<td>[S43]</td>
<td>-</td>
<td>Open/Agile Incorporation / Test</td>
</tr>
<tr>
<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>
</tr>
<tr>
<td>5</td>
<td>C02_1</td>
<td>No time and budget allocated for specifying validation requirements</td>
<td>-</td>
<td>[S7], [S8], [S9]</td>
<td>Open Management</td>
</tr>
<tr>
<td>6</td>
<td>C02_2</td>
<td>Unavailability of test equipment on time</td>
<td>[S43]</td>
<td>-</td>
<td>Test Management</td>
</tr>
<tr>
<td>7</td>
<td>C03_1</td>
<td>Lack of requirements clarity</td>
<td>[S19, S30, 31, 32, 33, 35, 36, 38, 39]</td>
<td>[S5], [S8], [S9], [S14], [S21]</td>
<td>Requirements Management</td>
</tr>
<tr>
<td>8</td>
<td>C03_2</td>
<td>Requirements volatility</td>
<td>[S19, S30, S37]</td>
<td>[S5], [S7], [S9]</td>
<td>Requirements Management</td>
</tr>
<tr>
<td>9</td>
<td>C03_3</td>
<td>Requirements traceability</td>
<td>[S15, S19, S27, S28, S30, S33, S37, S38]</td>
<td>[S5], [S8], [S9]</td>
<td>Requirements Management</td>
</tr>
<tr>
<td>10</td>
<td>C04_1</td>
<td>Lack of dedicated testers</td>
<td>[S32]</td>
<td>[S11]</td>
<td>Competence Management</td>
</tr>
<tr>
<td>11</td>
<td>C04_2</td>
<td>Unavailability of personnel for testing</td>
<td>[S32]</td>
<td>[S10], [S15]</td>
<td>Competence Management</td>
</tr>
<tr>
<td>12</td>
<td>C05_1</td>
<td>Knowledge transfer and sharing issues regarding testing</td>
<td>-</td>
<td>[S25], [S26]</td>
<td>Test/Competence management</td>
</tr>
<tr>
<td>13</td>
<td>C05_2</td>
<td>Lack of testing fundamentals</td>
<td>-</td>
<td>[S10], [S15]</td>
<td>Test/Competence Management</td>
</tr>
<tr>
<td>14</td>
<td>C06_1</td>
<td>Lack of regular interactions with customer regarding requirements</td>
<td>-</td>
<td>[S7]</td>
<td>Requirements Management/Agile Incorporation</td>
</tr>
<tr>
<td>15</td>
<td>C06_2</td>
<td>Lack of interactions with other roles within the project during testing</td>
<td>-</td>
<td>[S10], [S15]</td>
<td>Test/Competence Management</td>
</tr>
<tr>
<td>16</td>
<td>C06_3</td>
<td>Informal communication with the customer</td>
<td>-</td>
<td>-</td>
<td>Requirements Management</td>
</tr>
<tr>
<td>17</td>
<td>C07_1</td>
<td>Lack of automation for test case generation leading to rework</td>
<td>[S1, S2, S4, S5, S6, S7, S9, S11, S13, S21, S25, S26, S29, S40, S43]</td>
<td>-</td>
<td>Test Automation/Test Tools Introduction</td>
</tr>
<tr>
<td>18</td>
<td>C07_2</td>
<td>No unified tool for entire testing activity</td>
<td>[S5, S19]</td>
<td>-</td>
<td>Tool – Open</td>
</tr>
<tr>
<td>19</td>
<td>C07_3</td>
<td>Improper maintenance of test equipment</td>
<td>[S17]</td>
<td>-</td>
<td>Test Management</td>
</tr>
<tr>
<td>20</td>
<td>C08_1</td>
<td>Reliability issues</td>
<td>[S1, S3, S40]</td>
<td>-</td>
<td>Open/Test Automation</td>
</tr>
<tr>
<td>21</td>
<td>C08_2</td>
<td>Quality attributes are not specified well</td>
<td>[S17, S33, S34, S35]</td>
<td>[S7]</td>
<td>Requirements Management</td>
</tr>
<tr>
<td>22</td>
<td>C08_3</td>
<td>Lack of quality measurement/assessmen</td>
<td>-</td>
<td>-</td>
<td>Quality Assurance</td>
</tr>
<tr>
<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>
</tr>
<tr>
<td>23</td>
<td>C09_1</td>
<td>Testing late in the process makes it costly to fix defects</td>
<td>-</td>
<td>[S2]</td>
<td>Agile Incorporation / Test Management</td>
</tr>
<tr>
<td>24</td>
<td>C09_2</td>
<td>Hard to track defects which are not fixed in previous releases</td>
<td>-</td>
<td>-</td>
<td>Agile Incorporation / Test Management</td>
</tr>
<tr>
<td>25</td>
<td>C10_1</td>
<td>Documentation regarding test artifacts is not updated continuously</td>
<td>-</td>
<td>-</td>
<td>Agile Incorporation</td>
</tr>
<tr>
<td>26</td>
<td>C10_2</td>
<td>No detailed manuals available for some specific test methods and tools used</td>
<td>-</td>
<td>[S12]</td>
<td>Test Management</td>
</tr>
</tbody>
</table>
SP1: Requirements Management (RM)

Requirements related issues C03_1, C03_2, C03_3, C06_1, and C06_3 can be tackled through better RM. We have identified similar issues in automotive software domain in our SLR. For example, Grimm in his experience report from DaimlerChrysler recommends early simulations of requirements and derivation of test cases from specification and also suggests tracing and administering requirements across entire software development [S19]. An efficient acquisition management was recommended as a way to tackle cooperation and communication with customer/supplier. Bühler et al. [S30] in their work with DaimlerChrysler proposed abstraction levels for requirements specification. These levels are Software level, Function level, System level and Vehicle level. Each requirement on each abstraction level is in turn linked to goals and scenarios. The use of abstraction levels is also supported by Braun et al. [S39]. The introduction of goals, scenarios and requirements on abstraction levels facilitated in a comprehensive documentation of requirements (C03_1), implementation of requirements changes (C03_2) and traceability (C03_3). To provide a comprehensive understanding of automotive software requirements (C03_1) Liu et al. [S31] defined three kinds of information in requirements documents: (1) Structural information which defines interface information and the decomposition of a functionality, (2) Behavioral information that describes how a functionality reacts to the stimulus coming from the environment and (3) Communication information that specifies the properties of communication media like CAN bus. Weber and Weisbrod [S33] presented RE challenges and experiences at DaimlerChrysler which fall under clarity (C03_3) and clarity (C03_1). They suggest to document high-level requirements and with rationales. To this end, they find a need for database-based RE tools which can retain the functionalities of word-processors and adding capabilities for traceability, version control handling and requirements handshake between customer and supplier. They recommend well-structured requirements and design decisions at several levels of abstraction using description techniques like goal trees, feature lists and use cases. To this end, Puschnig and Kolgari [S32] also support the use of use cases as an efficient means of modeling requirements. It is also recommended that quality attributes or non-functional requirements should be refined until they can be refined to functional requirements (C08_2).

Requirements specifications reviews are highly recommended for agreement between customer and supplier. To this end, we believe frequent role specific reviews should be conducted and decision rationale is documented. Requirements reviews [S32] can be performed as a kind of ‘pair working’ where a requirements engineering works collaboratively with an expert in
developing use cases, reviewing and brainstorming issues with requirements. This technique was found to improve knowledge sharing (C05_01). It is also recommended to develop prototypes of parts of systems describing goals and scenarios. Formal reviews of requirements documents and meetings/brainstorming as a communication forum were found to lift requirements specifications to reliable and mature levels. Roles that participate in reviews are Authors (responsible for writing requirements), Inspectors (experts in the domain), Project manager and Moderator specializing in RE and conducting reviews. To this end, we believe inviting customer to participate in regular review meetings (face-to-face meetings) where requirements negotiations are kept more open. Puschnig and Kolgari [S32] recommend documenting review comments in RE tools like DOORS.

Islam and Omasreiter [S36] presented an evaluated an approach to elicit and specify user requirements for automotive software. Text-based use cases are elicited in interviews where various stakeholders participate. Requirements are collected to represent goal like scenarios in the system using a use case template. After conducting the interview, a formal review process takes place where experts participate, links between use cases are established and missing information is identified. Requirements generated thus could be linked to software test specification/plan using a traceability matrix [S38].

Post et al. [S35] presented a case study on linking functional requirements and verification. Their work aims at improving requirements documents by translating goal like statements into a formal language, which is used in identifying consistency between requirements and code implementations. To this end, Lee et al. [S15] presented an approach to combine Hardware in the loop simulations and Requirements based testing to achieve full coverage of requirements and check if it aligned with development.

To deal with changing requirements or requirements volatility (C03_2), Heumesser and Houdek [S37] recommends incremental development activities with defined vehicle prototypes. They use the concept of Feature lists which itemizes all essential characterizes under development, hardware and architectural properties. To keep track of requirement changes, requirements state models are used, which describe history of modifications to requirements. This was found as an efficient practice particularly when different developers work on the same requirements and suppliers need to know the status of the requirements. As it is common for developers to do the testing tasks, pairing up with experts at regular intervals will benefit them with testing tasks.

**SP2: Competence Management**

We identified a need for Competence Management based on the issues C04_1, C04_2, C05_1 and C05_2. To improve competence, we believe it is valuable to extend the concept of Experts as proposed by Puschnig and Kolgari [S32] to testing. They state that project management should provide
efficient means for involvement of experts in sharing knowledge and expertise with less experienced testers in projects. To this end, workshops are recommended where test team communicate with experts to communicate information on testing, tools as a means for informal training. The concept of ‘pair working’ (pair programming, a practice in agile software development) can be implemented in small teams. This can also be conducted as workshops where developers pair up experts (in testing) as a means for knowledge sharing. Moreover, to improve knowledge on testing and tools (C05_1, C05_2), we believe a central repository on testing tasks should be maintained. The test network available within the case organization can be useful to improve competencies related to testing.

**SP3: Quality Assurance**

AUTOSAR is industry open and standardized automotive software architecture. Cha and Lim [S44] report that AUTOSAR does not support processes and guidelines for developers. Based on Waterfall lifecycle, they proposed a process model for automotive software domain, describing work outputs in each phase. To perform quality assurance, they recommend peer reviews on artifacts produced in design, implementation and testing phases. Towards this end, DaimlerChrysler developed their own Software quality management handbook for their automotive projects [S33]. Although ISO 9000:2000 has been implemented in automotive sector, its applicability for software development has not been empirically exposed yet [S45]. A Draft International Standard (DIS) ISO 26262 (“Road vehicles – Functional safety) is expected to release [S46]. The standard defines the artifacts and activities for requirements specification, architectural design, implementation and testing, system integration and verification. The standard also prescribes the use of formal methods for requirements verification, notations for design and control flow analysis, the use of test case generation and in-the-loop (for e.g. hardware in the loop, software in the loop) verification mechanisms [S46]. ISO 26262 is adapted from IEC 61508. Other standards for software safety include DO-178B and DO-178C. ‘Controller Style Guidelines For Production Intent Using MATLAB, Simulink and Stateglow’ is a modeling catalogue for Simulink models in the context of automotive systems developed by The MathWorks Automotive Advisory Board (MAAB) [S22]. MAAB is an association of leading automotive manufacturers such as Ford, Toyota and DaimlerChrysler.

**SP4: Test Automation**

Automation is clearly one of the most important issues in industry and there is a considerable amount of research describing the state of the practice. Conrad et al. [S1] proposed a Model-based black box testing (M B³T) for the design of test scenarios and its integration in model-based development. Logical test scenarios are derived from textual requirements specifications and executable test scenarios are derived from functional model using
Classification trees. In the next step, consistency between requirements-based test scenarios and model-based test scenarios is evaluated. This is recommended especially for systems that have high safety and reliability requirements (C08_1). Lochau and Goltz [S2] describe a model-based approach for test-case generation taking feature interactions into account. This approach aims at identifying potentially undesired behavior from interface shared by different features. Based on functional architectural specification test cases are generated using STATEFLOW automata. Buhler and Wegener [S4] dealt with the challenge of automating functional testing. They proposed and evaluated Evolutionary testing to automate functional testing. A key element in evolutionary testing is to define a suitable objective function which represents the test objective, representative individuals and their transformation into valid test scenarios. Test objective is transformed into an optimization task. Kruse et al. [S7] also describe the use of evolutionary testing for automating black-box testing. They used the tool MESSINA which allows the implementation of hardware and software independent test sequences specified in Java or UML. Evolutionary testing has been successfully implemented at DaimlerChrysler [S9]. A tool named AUSTIN for evolutionary testing has been developed within Daimler for testing modules written in C [S26] (C07_2). Lindlar et al. [S24] have also presented a case study on the use of Evolutionary testing in Model-based development.

Classification-Tree method [6] and Classification-Tree editor CTE [43] have been developed at DaimlerChrysler for a systematic approach to the design of functional test cases [S5]. Pfaller et al. [S6] presented an approach to derive test cases from along different levels of abstraction during the test phase. Services on the abstraction level called ‘service level’ are used as test specification, while models on other levels are used as test models from which test cases are generated. The authors state that generating test cases from design level abstraction levels focuses on requirements overage and also on error-prone design decisions.

The tool called MTTest (C7_02) [44] is being used at DaimlerChrysler for testing on the level of system models [S5]. MTTest is based on modeling tool MATLAB/Simulink. The test system TESSY provides computer support for C modules and covers testing activities from design to execution, and test evaluation to documentation (C7_02).

Awedikian and Yannou [S11] describe an approach to automatically generate test cases based on compromise between structural and functional formal coverage and the cost of generate test cases. This approach uses simulation models that describe automotive requirements and tests are generated based on quality indicators such as Structural and functional Coverage. Brillout et al. [S13] described an application of the bounded model checking to generate test suites for Simulink models with high mutation coverage. Bringmann and
Krämmer [S21] presented Time Partitioning Testing (TPT) approach for model-based automotive testing. TPT uses a graphical test modeling language which are portable and reusable on different platforms like Model in the loop, Software in the loop etc. and can be executed in real-time.

Schwarzl and Peischl [S25] proposed test case generation based on communicating UML state chart models (SCMs). They describe the rules for model transformation allowing creation of Symbolic transition system (STS) from which test cases are generated using probabilistic approach. However, this approach does not support all elements of UML like timed transitions. In regard to the use of UML, Chimisliu [S29] presented a semantics-preserving model transformation from UML State charts to the specification language LOTOS. Since the UML states are mapped to LOTOS processes, this transformation allows for coverage-based test case generation.

Park et al. [S18] presented test methods for the validation of AUTOSAR software components created by model-based design. They recommend Walkthroughs, i.e. inspecting algorithms and source codes by following paths. They also recommend the use of C coding guidelines and standard for vehicle embedded software. In regard to test case generation, they utilized Statement plug-in program and T-VEC Tester, 3rd party tools for models designed in Telelogic’s Statemate and MATLAB/Simulink respectively (C07_2) [45]. To unveil errors in the model, they stated the use of Model Checker/Certifier and Safety-Checker Blockset for Statemate and MATLAB (C07_2) respectively [45].

**SP5: Test tool introduction**

In regard to tools in automotive software context, we have identified several papers. As discussed before, some of these tools were developed/used in the context of test case generation (for e.g. Statement plug-in for Telelogic’s Statemate and T-VEC Tester for MATLAB/Simulink). AUSTIN was developed to give tool support for Search-based testing for automotive components developed in C within Daimler [S26]. The tool called MTest, based on MATLAB/Simulink is being used at DaimlerChrysler for testing on the level of system models [S5]. Following a strong tradition in automotive software research, DaimlerChrysler developed TPT tool covering major activities from test design, test case generation and execution [S21]. The resulting test cases are independent of underlying software architecture, technology of system under test and test platform. Seo et al. [S3] developed and performed an industrial case study on test automation tool called Analytic Master of System (AMOS) v2.0 for performance analysis (C08_1) in automotive context. Gross et al. [S23] developed and performed case studies on EvoTest Framework (ETF) Structural test tool that comes as a plug-in for Eclipse for C/C++. The tool can be used to generate test cases (C7_01) for ANSI C modules and aims at achieving 100% branch coverage of the function under test.
Cleaveland [S47] mentions the use of the tool Reactis [46] for automated support when using Simulink or Stateflow. Ferdinand et al. [S48] present their report on integration of tool aIT/StackAnalyzer into the code generator called ASCET [47]. Stackanalyzer can be used as a tool for resource usage and aIT for timing analysis (worst case execution time) (C08_1). The same authors also point out integration of aIT and StackAnalyzer in a model-based tool called SCADE from ESTEREL [S49], [48]. Grossmann et al. [S18] describe a test exchange language called TestML which can be used as a common test specification language across Model in the loop, Software in the loop and Hardware in the loop tests. They also provide an example of TestML implementation using Simulink/Stateflow in their work. They also mention the use of test automation tool called AutomationDesk from dSpace [49].

Papadopoulose and Grante at Volvo Cars [S50] propose a design process semi-automatic safety and reliability analysis. Their process relies on the use of tools for the incorporation of safety and reliability concerns (C08_2). A tool has been developed to assists the search of design concepts early in the design phase/process. The tool has been validated in a case study by Volvo involving 52 active safety functions [50]. A second tool, Automated Safety Analysis Tool, has been developed for safety and reliability analysis that generates system Fault Trees and Failure Modes and Effects Analysis (FMA) from Matlab Simulink or Simulation X models. Case studies on the use of tool were performed in collaboration with DaimlerChrysler [51] and Germanisher Lloyd [52].

**SP6: Agile incorporation**

We identified the use of agile approach as one of the biggest strengths/advantages in one of the projects. However, in this context we found it interesting to know what other practices related to agile could be used within the case organization in other projects to overcome above challenges.

With respect to this, an interview with a Scrum Master working in similar domain in another organization helped to elicit advantages of agile such as adapting to changing requirements, developing projects in short increments thus giving better visibility throughout the project. The team headed by this interviewee uses Scrum methodology [53] using 4-week sprints. Practices adopted in the project are Test-driven development (TDD) where unit tests are written before the implementation. Test cases thus developed are used to verify the implementation. In cases when implementation is changed, test cases are re-run aiding in automation of regression testing [54]. The practice Kanban (Kanban boards) [55] is used for progress visualization. The use of TDD enabled continuous testing throughout each sprint. Features were implemented in short iterations (sprints). Features were selected from Product backlog containing prioritized lists of requirements stacks. Requirements within a sprint are kept frozen and are not allowed to change.
Features that are implemented within a sprint are always added to the overall product (Continuous Integration) after the sprint. In this manner, Continuous Integration facilitates a progressive and continuous manner for integration testing. The tool ‘Jama Contour’ is being used in the team which is used by testers to manage implemented requirements and test artifacts.

Based on our interviews, we identify a need for change in the software development process used in the case organization to cope with requirements changes. This is also particularly relevant for defect management issues (See C09_1 and C09_2) and the overall importance given to testing. One of the projects interviewed within case organization has implemented some of the agile practices such as scrum, stand-up meetings, TDD as discussed in the Section 5.2.

The agile project, as described before, gives a lot of emphasis on testing (C01_1 and C01_2) by enforcing continuous testing. It is also known that agile encourages interactions with customer and among team members [56]. As the software is developed in short cycles (iterations/sprints), it can be continuously reviewed. With TDD in place, there will be an increase in collaboration between developers and testers, thus facilitating knowledge transfer (C05_1). To this end, we propose a process model and set of agile practices based on our interview with the Scrum Master. The proposed agile model is shown in Figure 10 and practices are discussed below:

- **Product Backlog**: Requirements collected are maintained in a requirements stack. Requirements are packaged based on their priorities. That is, requirements with top most priority go in for development in the first iteration and so on. To this end, we recommend RM should be extra cautious to define requirements priorities and estimate the size of requirements as suitable for the length of iteration. Also, requirements in a package should be defined in a way that requirements fit together like a small project. Product Owner is responsible for maintaining the Product Backlog [53].

- **Sprint**: Requirements packages are developed in iterations known as sprints. Scrum methodology advocates sprint length of 30 days [53]. However, requirements in a sprint can be time-boxed with a defined deadline in a ‘Sprint Planning’ meeting. In Sprint planning meeting, the Scrum Master and team focus on how sprint is implemented. There are various tools to monitor the progress in a sprint such as Kanban-pull systems [55] and Burn-down charts [57]. The work package produced at the end of each sprint reviewed by customer in a ‘Sprint Demo’ as shown by ‘Package Demo’ in Figure 10.

Scrum also advocates Daily Scrum meetings/stand-ups to discuss the progress of the scrum team [53]. The Scrum master conducts stand-ups. These are generally 15 minutes in length focusing on: (1) What has been
accomplished since yesterday?, (2) What needs to be accomplished? and (3) What are the impediments in achieving goals?

- **TDD**: Development within sprint is test-driven. Unit test are written before code implementation. This way TDD contributes to quality and provides benefit to testing [58].

- **Continuous Integration**: After a work package is tested in each sprint, it added to the project code base (See ‘Project Build’, Figure 10) and tested. Any change in the project build is tested immediately. All the defects detected and corrected should be communicate to the Test Management. Test Management verifies if the developed Project code base so far is good enough to proceed with the next sprint. This step is crucial to solve defected related issues found in our interviews (See C09_1 and C09_2).

![Figure 10: Agile Development Model](image)

Agile development has been implemented at DaimlerChrysler [S42]. It has been observed that Agile offers flexibility, high-speed development and high quality. Automation of Unit testing and TDD were found very useful. Mueller and Borzuchowski [S43] report experiences in using Extreme Programming (XP), an Agile software development method on an embedded legacy product. They report that TDD and automation of Unit tests were the
essential ingredients for success. They used JUnit as the test framework for implementations in Java.

**SP7: Test Management:**

From the above identified solution proposals it can found that most of the activities are concerned with organization of testing and its artefacts. It was also evident from the data analysis done from the interviews that most of the challenges identified in the case study were more or less related to management of test activities. There is no study that necessarily concentrates on test management activities related to automotive domain. However, very few articles were found in literature which describes the activities that test management must concentrate in order to improve testing which suits this study context. So this solution proposal was formulated to co-ordinate the above proposed solutions.

Software test management according to Gao [59] is to enable software in accordance with pre-test project cost, schedule, quality and successful completion of cost personnel and schedule, quality risk analysis and management activities. He also identified that that test management must attend 3Ps: Processes, Products and People related to testing activities. Test management is a part of project management which is intended to manage everything related to testing and quality assurance. In essence, test management is responsible to manage resources and artefacts related to testing. The other disciplines related to test management are requirements management, defect management, configuration management, and competence management, testing tools and automation and quality assurance activities [60]. Test management activities as observed in our study can incorporate the following activities.

1. **Test Process management:** Manages various activities within test process such as test planning, test analysis, test build and test execution [59]. This activity is also applicable when agile practices are introduced.

2. **Test artefacts and assets organization:** Reuse and maintenance of test artefacts such as test cases, test versions, test tools, test environment, test results and test documentation. This activity can also be termed as test configuration management with which change throughout the life cycle of test activities can be managed [59, 60].

3. **Requirements management in accordance to testing:** Responsible to analyze and determine requirements change which facilitate reasonable adjustment in test schedule and test strategy and thus improve test cases to fulfil new requirements [59, 60].
4. *Competence management:* Responsible to allocate test personnel with required stock of skills and knowledge necessary to perform the specific testing activity [59, 60].

5. *Defect management:* Responsible for early detection of defects that need to be effectively managed and supported through various stages by different people working together [59].
The lessons learned can be generalized to other companies working in similar domain. The draft of future state VSM depicts the flow of test processes after implementation of the given solution proposals.

It is apparent from the results of this case study that other processes especially requirements impact testing in a negative manner and lead to many unwanted wastes. We found that most commonly perceived wastes i.e., W3: handoffs and W1: partially done work were occurring due to long delays in eliciting clear and stable requirements for testing i.e., to perform test planning, write test cases which requires prioritized requirements. The identified challenges in the test process report that continuous inflow of requirements lead to reduction in test coverage and increase in the amount of faults due to late testing. The faults that arise in the current release are sometimes left behind and delivered due to which the same faults repeat in the next releases but becomes hard and costly to trace and fix them. In this way the time allocated for testing is not spent on reporting, debugging and fixing faults which can deliver high quality. Hence the testing approach currently used does not suit the flow of requirements which indicates a necessity in shifting to new approach which can manage and organize changes at the same add quality. In this regard we recommend use of agile practices (SP6) and proper defect management (SP7) which makes time of testers to be used more efficiently with parallelization of development and testing incepting early fault detection and short ways of communication. Agile can also help in achieving high transparency in terms of requirements for testers since the test planning is done for all iterations in the beginning itself. However, test plans can be updated in detail for every iteration. The future state VSM is shown in Figure 11 which is agile in nature. The given process can be considered as one of the iteration in and agile environment.

Flexible test process which enables to integrate more functionality is found to be most significant strength in the projects especially small teams. Most of time in testing is concentrated to deliver more functionality (Value: V1) rather than quality. However, some of the test techniques such as exploratory and experience based testing which totally rely on tester’s abilities and skills are found to add quality to the test process implemented in automotive domain. This study also implies that challenges with respect to resource constraints such as difficulty in finding practitioners with right competence in testing who have expertise and experience in performing testing specific to automotive domain act as a barrier to quality incorporation. The wastes identified in this context can be long delays or lack of resources to perform testing activity (W3, W4). Almost 6 out of 8 studied projects lack dedicated
testing team which cause the above waste and lead to low test coverage. In

this case, use of quality standards/measures (SP3), verification activities or
iteration demos (Sprint demos in SP6) would serve as a proper replacement
as explained in solution proposal quality assurance. A variable test approach
may not involve quality incorporation but with proper agile practices (SP6)

**Figure 11: Draft of Future State VSM**
in place this can be possible as found in SLR. The recommendations given by the scrum master working from automotive industry clearly indicate that when properly employed agile methods not only provide flexibility and agility but also quality which adds value to the testing being performed in automotive domain.

The challenges related to time and cost constraints and testing techniques (C02), tools and environment (C07) make it obvious that writing good tests which consume time and effort is more emphasized over defect prevention activities. Automating tests could save time and improve value and benefits in testing. Coming to the test levels, unit test and integration are the most common strengths. Integration test is perceived as strength since it saves time detecting most of the strengths. An appropriate problem found in this context is reliability issues. That is, these tests work well in assuring proper functionality of the system under specific conditions for a specific time period only and hence are prone to failures easily. In order to avoid this situation teams can try to implement other testing techniques such as exploratory testing which is already used in some projects and can find defects at a more detailed level. Automation of unit tests and regression tests (SP4, SP5, SP6) can facilitate reuse of test cases effectively and also add value to the end product. For this tools which are already used in industry were identified and suggested from SLR.

From this study it is reasonable to say that testing is not as emphasized as development in automotive domain as said in [6]. Testing is given low priority which does not facilitate knowledge sharing and knowledge transfer in testing as observed from the interviews. In this regard, competence management can be considered essential to testing with activities which can improve skills and knowledge with respect to testing involving knowledge transfer and sharing within testing (SP2). Also it would be better if the required resources in terms of test personnel can be estimated in the beginning of the project and allocated. This would also help them improve the competence level for every iteration and hence improve testing. Recruiting new person in the middle of a project would take long time for the person to understand the system and testing performed on it.

Solution proposals for the identified opportunities were given from literature and an interview. The validation of the solution proposals was not possible in the current situation in order to limit the scope of this thesis. However, the suggested proposals were taken from peer-reviewed literatures which were validated in industry and also from an interviewee working as a scrum master implementing agile practices successfully thereby improving testing in automotive domain.

Value Stream Analysis: The application of VSM tool in software testing context was useful in identifying all the value adding and non-value adding
activities. We identified seven kinds of non-value adding activities and four types of value adding activities in the test process. Since the study scope of this research was limited to testing, it was possible to identify each and every waste that occurs in the step-by-step test process. All the wastes/challenges identified through this study may not be of major relevance for every project studied and so are the solution proposals recommended. As this study is primarily concerned with studying the phenomenon of testing, it did not necessarily concentrate on only one project. With this, generalization of results for testing in automotive domain can be possible.

The evaluation of VSM used in this study does not take quantitative data (i.e., historical data related to test activities such as lead times documented in test results) as done in [13]. It was not possible due to the flexible test approach employed which was an important observation made in this context. However, qualitative VSM can be considered as a good compromise since it aided in identifying the value adding ad non-value adding activities within test process in a detailed level. This approach is quite effective in identifying VSM driven improvement opportunities. After identification of all wastes in a detailed manner it is evident that having an unstructured test approach is no more an advantage. The assessment of test process using VSM (lean) made it visible that the lack of a systematic test process acts as a barrier in achieving process value (V04).
9 SYNTHESIS

The synthesis draws together the obtained results in order to provide answers to the research questions formulated in this thesis.

RQ1: What are the practices in testing that can be considered as strengths within automotive domain?
In the in-depth investigation conducted through the case study it was shown that testing in automotive domain have many strengths which has been evolving over years with significant developments. The most perceived practice which is considered as strength is ability to test in an environment which depicts real time environment which is essential for critical systems which has safety, performance, reliability, etc as primary non-functional requirements. Other strengths regarding test techniques are exploratory and experience based testing which play major role in testing with automotive embedded systems. Apart from these there are other practices which add quality to testing in automotive domain such as informal code reviews, continuously integrating, reuse of test artefacts, testing tools used in organizing and managing test activities. Majorly used test levels such as unit testing and integration test seems to strengthen and increase functionality of the system. The strengths identified were not observed in all projects but scattered among various projects which differ by project and team size. A more detailed explanation for the strengths can be found in Section 5.2.

RQ2: What are the challenges/bottlenecks identified in testing automotive embedded systems?
The challenges identified with testing in automotive domain can be found in Section 5.3 where challenges are C01 to C10. Among them the most general ones are organization and its processes related issues, time and cost constraints and requirements related issues. Even though the identified issues are less common within small teams, the same teams may evolve as large teams for upcoming releases. Also these challenges were mapped with the existing challenges found in literature.

See Appendix F for the mapping of challenges and strengths in the testing.

RQ3: Is VSM tool useful in eliciting improvement potentials in software testing context in automotive domain?
We found VSM tool is powerful in eliciting various strengths, wastes and improvement potentials in testing. A clear explanation of identified activities can be found in Section 5.4 ‘Process Assessment Using Lean‘.
10  **CONCLUSIONS**

In this study we used Lean methodology (VSM) to identify the wastes in the software test process, with a goal of eliciting improvements potentials. The wastes detected were delays or waiting times that occurred due to partially done work, extra features, handoffs, task switching, relearning, delays and defects. A detailed analysis revealed that major delays were due to lack of requirements management (i.e., lack of clarity, requirements volatility) which was the main reason for low test coverage also. The proposed solution proposals were elicited through SLR and an interview with a successful agile practitioner. With this we were able to reduce the validity threat for the solution proposals as these solutions were already practiced and validated in similar domain (i.e., automotive embedded systems). The VSM maps will be useful in visualizing problems in the testing process and possible improvements.

The main lessons learned about the testing in automotive domain are improper requirements management, unstructured test process, testing late in the process consumes more time & effort, incorporating low quality & value to the testing activity which is undesirable. Improvement initiates such as use of quality standards to build quality, agile incorporation to cope with changes, structured testing process should be employed to improve lead time and avoid wastes. Test management must co-ordinate all these activities to organize and improve testing activities.

10.1  **Open issues and further work**

**VSM:**
A more detailed quantitative analysis would be helpful to identify time taken for each activity or delays that exist between two process steps in a detailed manner i.e., in terms of queue time or processing time.

**Other Improvements:**
Some of the issues found in this study were not adequately addressed in the papers collected through the SLR conducted. They are Time and cost constraints in writing proper requirements, reliability issues in the end product and a unified tool for testing. With the use of agile practices time and effort to be spent on every activity in software development gets clearer. A unified tool for testing is not available for automotive domain since the techniques used to test such systems vary from project to project. However, there is a room for improvement to achieve more significant contributions for these issues.
11 REFERENCES


[34] Jingyue Li, Nils Brede Moe, Tore Dybå: Transition from a plan-driven process to Scrum: a longitudinal case study on software quality. ESEM 2010


11.1 Systematic Literature Review References


[S22] Rule Checking within the Model-Based Development of Safety-Critical Systems and Embedded Automotive Software.


12 Appendix A

Interview Study
In this appendix description of the interviewees, the questions asked during the interview are presented.

a) Description of Interviewees
Below is the brief description of the 14 employees (14 out of 15 were considered for the study) following with their background to increase the understanding of what they have based their answers on the interview questions.

Interviewee 1
Date of Interview: 2011-03-21
Responsibilities: Developing and testing software components along with a global off-site partner.

The interviewee is a Masters in Electrical engineering and has several years of working experience with process control engineering and radar related signal processing in automotive domain.

Interviewee 2
Date of Interview: 2011-03-29
Responsibilities: Developer and control strategist in research projects.

The interviewee is a Masters in Automation engineering and has several years of working experience in product development and testing. Currently working for research projects in developing and implementing different concepts.

Interviewee 3
Date of Interview: 2011-03-24
Responsibilities: Involved in research projects performing all activities.

The interviewee is a Masters in Computer science and Engineering and has several years of experience working with various methodologies, methods involved in software testing. Currently leading a competence team in software testing.

Interviewee 4
Date of Interview: 2011-03-15
Responsibilities: Responsible for defining proper requirements and coordinating tests.

The interviewee has a computer engineering background and has several years of work experience associated with different kind of roles such as project launcher, technical contact, etc. He is currently responsible to define requirements and coordinate traceability to different tests being conducted.
Interviewee 5
Date of Interview: 2011-03-14
Responsibilities: Responsible for developing various applications and performing unit tests on them.

The interviewee has a computer science and engineering background and has several years of work experience as technical leader. Currently working to develop several Windows applications used in embedded systems.

Interviewee 6
Date of Interview: 2011-03-10
Responsibilities: Software test leader and scrum master.

The interviewee has a computer science background, working for testing from several years. Currently working on a project using agile methods to design tests, execute and report them. Also involved in several process improvement activities in order to improve test processes.

Interviewee 7
Date of Interview: 2011-03-16
Responsibilities: Implementation and testing of different functions for automotive systems.

The interviewee has a Masters in Electrical Engineering and has been working to develop the applications for the requirements from the customer and also testing the functions.

Interviewee 8
Date of Interview: 2011-03-29
Responsibilities: Software developer and tester (unit testing)

The interviewee has a Electrical engineering background and has been working with source control systems.

Interviewee 9
Date of Interview: 2011-04-04
Responsibilities: Managing test activities for a group of projects

The interviewee has a Masters in computer science and engineering and has been working for several in software development projects initiating software process improvement activities. Currently working as a test group manager steering towards improving software test processes.

Interviewee 10
Date of Interview: 2011-03-30
Responsibilities: Involved in developing, managing and testing software for Vehicle Electronics
The interviewee has a Masters in computer science and engineering and is working to manage development and testing of the software components being delivered to customer.

**Interviewee 11**

Date of Interview: 2011-03-09

Responsibilities: Include functional development and testing

The interviewee has a Masters in Electrical engineering and is working with requirements to develop proper functionalities and test them to assure quality.

**Interviewee 12**

Date of Interview: 2011-03-24

Responsibilities: Managing vehicle installations and test vehicle responsible

The interviewee is a Masters in computer science and engineering and has worked for several years as test leader for embedded software. Currently responsible for a project which involves vehicle installations and testing.

**Interviewee 13**

Date of Interview: 2011-03-08

Responsibilities: System developer and tester

The interviewee is a Masters in computer science and is working to develop different applications and frameworks related to vehicle communications software.

**Interviewee 14**

Date of Interview: 2011-03-16

Responsibilities: Verification of several ECUs using different tools

The interviewee is a Masters in computer science and is working on a maintenance project to test ECUs in different parts of cars.

b) **Interview Guide**

The interview guide is as follows

**Phase I: Warm Up and Experience**

The goals and objectives of the thesis are explained in brief to the interviewee. However, the interview request mail had the purpose of the thesis described in detail along with the interviewee questionnaire sent prior to the interview schedule date.

1. What is your professional background (Education and experience)?

2. What is your role within software development life cycle?

3. How long have you been working with software testing in automotive domain?
4. What is the current software development paradigm you and using and your experience in it?

5. Can you briefly describe the recent most project you worked with?

**Phase II: Testing Concepts**
1. What are the goals of the current project you are working for?
2. What are the requirement criteria for testing in your project?
3. Can you briefly explain the testing process you are using?
4. What are the test objects you test?
5. In which phase of this process do you think most critical defects are detected?

**Phase III: Test Process Assessment**
1. Can you identify any three strengths in your test process that in your opinion work well and others could learn/adapt them?
   a) Why are they strengths?
   b) Where in the testing process do they occur?
   c) How do they affect the overall testing?
2. Can you identify three challenges/problems in your test process that hinder in achieving high-quality?
   a) Why are they perceived as a challenge?
   b) Where in the test process do they occur?
   c) How do they affect the overall testing?
3. In which parts of the process do you see highest improvement potential in order to further improve with respect to value?

**Phase IV: Conclusion**
1. Is there anything else you would like to add that you think is interesting in this context, but not covered by the questions asked in the interview?

The interview is concluded by saying that the interview recorded will be transcribed, summarized and communicated to the interviewee for validation of the interpretations done by the researcher.
13 Appendix B

We identified different process areas for the practices being implemented within the automotive domain. However most of them were found not only in the testing process but also in other process areas. For the reasons of clarity we have categorized the observed process areas for this research. They are as shown in table below.

Table B.1: Process areas and descriptions

<table>
<thead>
<tr>
<th>Process Area</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>Activities which are related to requirements engineering such as elicitation, specification, management, etc.</td>
</tr>
<tr>
<td>Project Management</td>
<td>All about planning, managing and organizing all activities in SDLC to reach specific goals and objectives of the project.</td>
</tr>
<tr>
<td>Test Process</td>
<td>Process carried with an intention of finding errors in the system. It includes planning, designing, executing and evaluating tests.</td>
</tr>
<tr>
<td>Test Levels</td>
<td>It includes different stages that are done as a part of test activity such as Basic/Unit test, Integration test, System test, etc.</td>
</tr>
<tr>
<td>Test Management</td>
<td>Activities that involve organizing, controlling the test process and monitoring progress.</td>
</tr>
<tr>
<td>All Processes in SDLC</td>
<td>This category refers to all the process areas (i.e., activities carried out in that process area) in software development life cycle.</td>
</tr>
</tbody>
</table>
14 **APPENDIX C**

**Coding guide:**
The coding guide used to code transcribed interview text is as follows.
Transcriptions are coded using three levels of code. They are:

1. **High level abstraction** which is directly connected to the research questions. They were grouped into five divisions. They are Challenges, Strengths, Waste, Value and Improvement potential.

2. **Medium level abstraction** consists of different categories of practices used for testing. They were grouped according to above stated divisions.

3. **Third level of abstraction** states the process area where the practice is adopted/ or to be adopted (For improvement potential).

However, there are other two columns in the coding guide, one is to directly copy the text from the interview guide which is directly applicable to the research question and the other column where we write comments on why the practice comes into one of the divisions as stated in high abstraction level. All these columns must be mentioned mandatorily.

**High Level Codes: Research Questions**

<table>
<thead>
<tr>
<th>Division</th>
<th>Description</th>
<th>Research Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strength</td>
<td>Activities with test process which are considered as good practice is termed as strength</td>
<td>RQ1</td>
</tr>
<tr>
<td>Challenge</td>
<td>Problems and challenges related to current test process which needs to be eliminated</td>
<td>RQ2</td>
</tr>
<tr>
<td>Waste</td>
<td>Activities within test process or within SDLC which lead to rework or inefficient time spent in testing activity can be considered as waste.</td>
<td>RQ3</td>
</tr>
<tr>
<td>Value</td>
<td>Activities within test process which add value to the test process is considered as value which is considered as lean thinking</td>
<td>RQ3</td>
</tr>
<tr>
<td>Expected Improvements</td>
<td>Suggestions on the way to improve test process and expected benefits to be achieved by it</td>
<td>RQ3</td>
</tr>
</tbody>
</table>

**Medium Level Codes: Categories**

Medium level codes are designed in such a way that every division in the high level code has its own set of categories. More than one code can also be applicable for a text. In order to avoid confusion, each division from high
level code is given a set of categories which are adopted and defined using literature [28], [36] for this study.
Categories for challenges adopted from [28] and [36]:

<table>
<thead>
<tr>
<th>ID</th>
<th>Category Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C01</td>
<td>Organization, its processes</td>
<td>Poorly performed practices related to organization and its test processes such as change management, lack of structured test process, etc. Also includes stakeholder’s attitude towards testing (If testing is given low priority).</td>
</tr>
<tr>
<td>C02</td>
<td>Time and Cost Constraints</td>
<td>Problems in practices related to time, schedule and budget constraints.</td>
</tr>
<tr>
<td>C03</td>
<td>Requirements</td>
<td>Problems in practices related to requirements such as stable requirements, requirements traceability, requirements clarity, requirements management, etc.</td>
</tr>
<tr>
<td>C04</td>
<td>Resource Constraints</td>
<td>Includes problems related to management of resources (test personnel) for testing team in the project such as lack of dedicated testers, lack of expertise in testing, employee turnover.</td>
</tr>
<tr>
<td>C05</td>
<td>Knowledge Management</td>
<td>Poorly performed practices related to knowledge capture, knowledge creation, knowledge sharing and knowledge transfer related to testing.</td>
</tr>
<tr>
<td>C06</td>
<td>Interactions, Communications</td>
<td>Problems in practices related to communication between different stakeholders in involved in testing. Also includes improper form of communication such as lack of regular face-to-face meeting, lack communication between customer and test personnel.</td>
</tr>
<tr>
<td>C07</td>
<td>Testing techniques, tools and environment</td>
<td>Problems related to usage of current test techniques, environment and tools.</td>
</tr>
<tr>
<td>C08</td>
<td>Quality Aspects</td>
<td>Problems related to incorporating quality attributes of testing such as reliability, maintainability, correctness, efficiency, effectiveness, testability, flexibility, reusability, etc. Involves tradeoffs between quality and other activities.</td>
</tr>
<tr>
<td>C09</td>
<td>Defect Detection</td>
<td>Problems related to practices which disable the tester to trace the defect or the root cause of defect creation, also includes problems related defect prevention.</td>
</tr>
</tbody>
</table>
Poorly performed practices related to test documentation such as insufficient documentation, no documentation or too much documentation that doesn’t give proper support for maintaining quality in test process.

Categories for Strengths:

Table C.3: Description of categories for Strengths

<table>
<thead>
<tr>
<th>ID</th>
<th>Strength Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S01</td>
<td>Testing Process/activities</td>
<td>Any testing practice (within test process and test levels) that is perceived as strength in current testing is performed in automotive domain. Alignment of testing with other activities in SDLC, reuse of test artefacts for the next release is also considered as strength in this context. Practices related to usage of test methods, tools or environment is excluded in this category.</td>
</tr>
<tr>
<td>S02</td>
<td>Communication</td>
<td>Practices which improve communication with respect to testing process.</td>
</tr>
<tr>
<td>S03</td>
<td>Roles and responsibilities related to testing</td>
<td>Includes practices which result in clear understanding of roles and responsibilities related to test team.</td>
</tr>
<tr>
<td>S04</td>
<td>Testing techniques, tools and environment</td>
<td>Includes testing techniques, tools which are currently used perceived as strength. Also includes test environment when considered close to target environment.</td>
</tr>
<tr>
<td>S05</td>
<td>Testing related measures</td>
<td>Activities related to test measurement that are usually considered instrumental to quality analysis come under this category. Measurement may also be used to optimize the planning and execution of the tests. Test management can use several process measures to monitor progress.</td>
</tr>
</tbody>
</table>

Categories related to Assessment of current test process using lean methodology i.e., Value Stream Mapping tool are directly taken from literature related to Lean methodology [11], [13] and [15].

Table C.4: Description of categories for value adding activities

<table>
<thead>
<tr>
<th>ID</th>
<th>Value-adding activities</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>V01</td>
<td>Functionality</td>
<td>The capability of the tested product/service to provide functions which meet stated and</td>
</tr>
</tbody>
</table>
implied needs when the software is used under specific conditions.

<table>
<thead>
<tr>
<th>ID</th>
<th>Non-value adding activities (or) Waste</th>
<th>Description : (Practices related to)</th>
</tr>
</thead>
<tbody>
<tr>
<td>W01</td>
<td>Partially done work</td>
<td>Test activities which are not completely done such as unfixed defects, undocumented test artefacts or not testing at all.</td>
</tr>
<tr>
<td>W02</td>
<td>Extra features</td>
<td>Testing features/functionalities that are not required by the customer.</td>
</tr>
<tr>
<td>W03</td>
<td>Relearning</td>
<td>Misinterpretation caused due to no documentation of any activity that negatively affect testing. Ex: misinterpreted requirements.</td>
</tr>
<tr>
<td>W04</td>
<td>Handoffs</td>
<td>Lack of availability, knowledge or training in adopting compatible test techniques, data, tools or environment.</td>
</tr>
<tr>
<td>W05</td>
<td>Task Switching</td>
<td>Unclear roles and responsibilities as a part of organization structure with respect to testing which doesn’t result in forming right teams.</td>
</tr>
<tr>
<td>W06</td>
<td>Delays</td>
<td>Delays that occur to elicit clear validation requirements, approvals and other resources to perform test activities.</td>
</tr>
<tr>
<td>W07</td>
<td>Defects</td>
<td>Testing in the end, no early defect detection or prevention activities and lack of verification activities such as code reviews, inspections.</td>
</tr>
</tbody>
</table>

Furthermore during coding, the applicable text from interview transcript must be assigned to the suitable category and also comments must be mentioned to make it clear on why that text falls into the specified category.
By doing this the interpretations made during coding will be clear for carrying out analysis.
15 APPENDIX D

Examples of modelling using Simulink adopted from [61]

Figure D.1: Modeling using Simulink

Figure D.2: Visualizing results in Simulink
# APPENDIX E

## Table E.1: Testing tools and techniques used in projects studied

<table>
<thead>
<tr>
<th>Project</th>
<th>Test Methods Used</th>
<th>Software Development Methodology used</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>Basic/Unit(Smoke) test, System test, Integration test, Session-based test management, Script based testing, Code reviews</td>
<td>Waterfall model used with some agile practices is used.</td>
</tr>
<tr>
<td>P2</td>
<td>Basic/Unit test, System test, Integration test, Regression test, Exploratory test.</td>
<td>Agile software development using Scrum.</td>
</tr>
<tr>
<td>P3</td>
<td>Basic/Unit(Smoke) test, Integration test.</td>
<td>Waterfall development methodology</td>
</tr>
<tr>
<td>P4</td>
<td>Basic/Unit(Smoke) test, System test, Regression test, Semi-automated testing, Script based testing, Code reviews.</td>
<td>Model-based development with waterfall development methodology</td>
</tr>
<tr>
<td>P5</td>
<td>Basic/Unit test, Script based testing, Automated testing</td>
<td>Waterfall methodology using some agile practices such as Open Space/Working together.</td>
</tr>
<tr>
<td>P6</td>
<td>Integration test</td>
<td>Maintenance project, Ad-hoc development.</td>
</tr>
<tr>
<td>P7</td>
<td>Basic/Unit test, System test, Integration test, Regression test, Script based testing, Automated tests</td>
<td>Waterfall with Model based development</td>
</tr>
<tr>
<td>P8</td>
<td>Basic/Unit test, Integration test</td>
<td>Waterfall with Model based development methodology</td>
</tr>
</tbody>
</table>
Figure F.1: Mind map for testing process studied in case organization.