Introducing Domain Specific Language for Modeling Scrum Projects

ZHOU, CE and ZHANG, YANPENG
This thesis is submitted to the Faculty of Computing 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(s):

CE ZHOU
E-mail: programer_ce@163.com

YANPENG ZHANG
E-mail: zypqdu@gmail.com

University advisor:

Dr Ludwik Kuzniarz
School of Computing
E-mail: ludwik.kuzniarz@bth.se

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden

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

Context. A clear software process definition is important because it can help developers to share a common understanding and improve the development effectiveness. However, if the misconceptions or misunderstandings are introduced to the team during the process definition, it will bring numerous uncertain problems to the projects and reduce the productivity. Scrum is one of the most popular Agile development processes. It has been frequently used in software development. But the misunderstanding of usage of the Scrum method always leads to situations where teams cannot achieve the hyper-productivity even failure. Therefore, introducing a reasonable graphical language for describing the Scrum process may help learners to gain a correct and common understanding of the Scrum method.

Objectives. In this study, we introduce a graphical Domain Specific Language for modeling the Scrum process and specific Scrum projects. Further, we evaluated the proposed language to figure out if and how this language can help developers learn Scrum method and understand the specific Scrum projects. For the first, we decide to extract the essential elements and their relative relationships of the Scrum process, and based on that, we define and specify the graphical language. After that, we evaluate the proposed graphical language to validate whether this language can be considered as useful to help developers to learn Scrum method and understand the specific Scrum projects.

Methods. In order to define the graphical language, we studied and reviewed the literature to extract the essential elements and their relationships for describing the Scrum process. Based on that, we defined and specified the graphical DSL. With the aim of evaluating the proposed graphical language, we performed the experiment and survey method. This experiment was conducted in an educational environment. The subjects were selected from the undergraduate and master students. At the same time, we carried out a survey to capture the developers’ opinions and suggestions towards the proposed language in order to validate its feasibility.

Results. By studying the literature, we listed and specified the essential elements for describing the Scrum process. By executing the experiment, we evaluated the efficiency and effectiveness of learning Scrum in using the proposed language and the natural language. The result indicates that the graphical language is better than the natural language in training Scrum method and understanding specific Scrum projects. The result shows that the proposed language improved the understandability of the Scrum process and specific Scrum projects by more than 30%.

We also performed a survey to investigate the potential use of the proposed graphical DSL in industry. The Survey results show that participants think the proposed graphical language can help them to better understand the Scrum method and specific Scrum projects. Moreover, we noticed that the developers who have less Scrum development experience show more interests in this proposed graphical language.

Conclusions. To conclude, the obtained results of this study indicate that a graphical DSL can improve the understandability of Scrum method and specific Scrum projects. Especially in managing the specific Scrum project, subjects can easily understand and capture the detailed information of the project described in the proposed language. This study also specified the merits and demerits of using the graphical language and textual language in describing the Scrum process.

From the survey, the result indicates that the proposed graphical language is able to help developers to understand Scrum method and specific Scrum projects in industry. Participants of this survey show positive opinion toward the proposed graphical language. However, it is still a rather long way to applying such a graphical language in Scrum projects development because companies have to consider the extra learning effort of the graphical DSL.

Keywords: Scrum process, Domain Specific Language, Software development process
ACKNOWLEDGEMENT

First and foremost, we express our sincere gratitude to our supervisor Dr. Ludwik Kuzniarz for his invaluable support, guidance, and patience on our thesis work. His guidance helped us to figure out our research direction and improve our master thesis quality. We are very lucky to have a chance to work with him.

Besides this, we would like to thank our examiner Dr. Jürgen Börstler for his essential advice on our thesis topic and plan.

We also express our thanks to all people who participated in our initial expert interview, experiment and survey for their valuable time and contributions to our thesis project.

Last but not least, we would like to thank our families and friends who support and encourage us. Without their support and help, it was impossible to accomplish this research work.
# CONTENTS

ABSTRACT ................................................................................................................................. I
ACKNOWLEDGEMENT ............................................................................................................... II
CONTENTS ................................................................................................................................ III
LIST OF FIGURE ....................................................................................................................... IV
LIST OF TABLE ........................................................................................................................ V

1 INTRODUCTION ....................................................................................................................... 1
   1.1 AIMS AND OBJECTIVES ............................................................................................ 3
   1.2 RESEARCH QUESTIONS ............................................................................................ 3
   1.3 EXPECTED OUTCOMES ............................................................................................. 4
   1.4 TERMINOLOGY ........................................................................................................... 4

2 BACKGROUND ....................................................................................................................... 5
   2.1 OVERVIEW OF SCRUM ........................................................................................... 5
   2.2 SCRUM TEAM ORGANIZATION ............................................................................. 6
   2.3 SCRUM PROCESS ...................................................................................................... 6

3 MOTIVATION ........................................................................................................................ 9

4 RESEARCH METHODOLOGY ................................................................................................ 11
   4.1 OVERVIEW ................................................................................................................ 11
   4.1.1 Chosen research methods .................................................................................... 11
   4.1.2 Discussion of alternatives ..................................................................................... 12
   4.2 LITERATURE STUDY ............................................................................................... 13
   4.3 DSL DEVELOPMENT METHOD ............................................................................. 15
   4.4 EXPERIMENT ............................................................................................................. 16
   4.5 SURVEY ..................................................................................................................... 22

5 THE PROPOSED LANGUAGE DEVELOPMENT .................................................................. 25
   5.1 INTRODUCTION OF DSL ....................................................................................... 25
   5.2 THE DSL DECISION PHASE ................................................................................... 25
   5.3 THE DSL ANALYSIS PHASE .................................................................................... 26
      5.3.1 Domain analysis ................................................................................................. 26
      5.3.2 Extracted elements .............................................................................................. 27
   5.4 THE DSL DESIGN PHASE ......................................................................................... 31
      5.4.1 Language structure ............................................................................................. 31
      5.4.2 Terminology and element design ....................................................................... 34
      5.4.3 Scrum model ........................................................................................................ 39
      5.4.4 Design of the syntax and semantics ................................................................... 40
   5.5 EXAMPLE PROJECT DESCRIBED IN THE GRAPHICAL LANGUAGE ......................... 46

6 THE PROPOSED LANGUAGE EVALUATION ...................................................................... 52
   6.1 EXPERIMENT EXECUTION ......................................................................................... 52
      6.1.1 Validation design of the experiment– pilot experiment ....................................... 52
      6.1.2 Experiment operation ......................................................................................... 52
   6.2 RESULTS OF EXPERIMENT ...................................................................................... 53
      6.2.1 Analysis of experiment results ........................................................................... 53
      6.2.2 Hypotheses testing ............................................................................................. 55
      6.2.3 Validity threats .................................................................................................. 57
   6.3 SURVEY EXECUTION AND RESULT ANALYSIS ....................................................... 58

7 CONCLUSION AND FUTURE WORK ..................................................................................... 63
   7.1 ESSENTIAL ELEMENTS OF SCRUM METHOD ....................................................... 63
   7.2 DEFINITION OF THE PROPOSED GRAPHICAL DSL ............................................. 63
   7.3 EVALUATION RESULTS OF THE PROPOSED GRAPHICAL LANGUAGE ................ 64
      7.3.1 The experiment result ......................................................................................... 64
      7.3.2 The survey result ............................................................................................... 64
   7.4 FUTURE WORK ........................................................................................................... 65

REFERENCES ........................................................................................................................... 66

APPENDIX A. TRAINING MATERIALS ..................................................................................... 68
APPENDIX B. SURVEY QUESTIONNAIRE ............................................................................... 85
LIST OF FIGURE

Figure 1. Scrum method ................................................................. 5
Figure 2. Research methods and steps ........................................... 12
Figure 3. DSL development approach ........................................... 16
Figure 4. DSL feature structure ................................................... 31
Figure 5. Scrum roles structure ..................................................... 32
Figure 6. Scrum event structure .................................................... 32
Figure 7. Scrum artifact structure ................................................ 33
Figure 8. Scrum model structure .................................................. 34
Figure 9. Scrum project structure ................................................ 34
Figure 10. General structure of Scrum model ............................... 39
Figure 11. Relationship definition of participant ......................... 40
Figure 12. Relationship definition of Meeting ............................. 41
Figure 13. Relationship definition of Update ............................... 41
Figure 14. Relationship definition of Product Backlog ................. 42
Figure 15. Relationship definition of Sprint ............................... 42
Figure 16. Relationship definition of Task ................................... 43
Figure 17. Relationship definition of Sprint Backlog ................... 43
Figure 18. Example overview ..................................................... 47
Figure 19. Scrum team example .................................................. 48
Figure 20. Product Backlog example ......................................... 49
Figure 21. Sprint example ......................................................... 50
Figure 22. Updated Product Backlog ......................................... 51
Figure 23. Numbers of subjects that correctly answered each question ......................................................... 54
Figure 24. Time spent (in minutes) for completing the training materials ................................................. 54
Figure 25. Time spent (in minutes) for a correct answer .......... 55
Figure 26. Survey participants experience distribution ............... 59
Figure 27. Answer distribution of question 2 ............................ 60
Figure 28. Answer distribution of question 3 ............................ 60
Figure 29. Answer distribution of question 4 ............................ 61
Figure 30. Answer distribution of question 5 ............................ 61
LIST OF TABLE

Table 1. Scrum process.............................................................................................................7
Table 2. Keywords and result for searching literature............................................................14
Table 3. Literature results and selected reasons ................................................................14
Table 4. Variables...................................................................................................................20
Table 5. Survey questions....................................................................................................23
Table 6. Decisions made for DSL development.................................................................26
Table 7. Product Backlog Item attribute list.........................................................................27
Table 8. Sprint Backlog Item attribute list...........................................................................28
Table 9. Task attribute list....................................................................................................28
Table 10. Sprint attribute list...............................................................................................29
Table 11. Meeting attribute list...........................................................................................29
Table 12. Participant attribute list........................................................................................30
Table 13. Designed element for the proposed language.......................................................35
Table 14. Structure elements for building Scrum project....................................................38
Table 15. Semantics definition.............................................................................................43
Table 16. Result of experiment............................................................................................55
Table 17. Result of the statistical significance testing.........................................................56
Table 18. Results of the improvement/ deterioration............................................................57
Table 19. Questionnaire validation criteria and result..........................................................59
1 INTRODUCTION

A sound and reasonable software development method plays a significant role in software engineering by offering formal specification, quality assurance, and verification techniques for completing projects [1]. In recent years, Agile methods have been gaining their popularity at an amazing rate [2]. Agile methods such as Scrum, Extreme Programming (XP), Lean Development, and OpenUP are frequently used in industry. Especially Scrum. Scrum has been widely applied for software projects with rapidly changing requirements and frequently delivering artifacts [3]. However, there are some problems detected when introducing and applying Scrum in companies. Experiences showed that the introduction of Agile methods in various types of companies highlighted the difficulty of establishing a clear and shared understanding of the Agile process among all team members [4].

In theory, Scrum is an incremental and iterative software development framework with a small number of simple rules, which is hard to be applied [5]. Some stakeholders and developer do not have an accurate understanding or do not adapt to Scrum principles, such as collaboration between Product Owner and Development Team, self-organization and incremental development process. Hurtado [6] claimed that Scrum teams often experience trouble in working with a Product Owner when putting the explicit user stories from the product backlog into a Sprint backlog. In addition, Scrum does not provide a complete and detailed description of how to organize specific works in Scrum projects [7]. It can be considered as a disadvantage. Schwaber [8] found that Scrum teams always face difficulties to accurately organize works when using Scrum in projects. This problem can be a reason of delay to deliver artifacts at the end of a sprint. It also increases the risk of reworking and leads to a poor performance of whole Scrum team.

Moreover, Scrum doesn’t provide a detailed specification of the implementation process. That is why Scrum is often used with other Agile methods, such like XP [3, 7, 9], Rational Unified Process (RUP) [10, 11]. According to Hurtado [6] and Schwaber [8], Scrum puts emphasis on the management of software project, but it does not include some technical descriptions, such as how to elicit requirements, design, and implement software artifacts. The incomplete description of the rapidly changing requirements caused the problems sharing a clear vision of the Sprint goal among team members. Additionally, according to the self-organization principle, Scrum team allows team members to select work and make decisions on their own. However, this principle is always misunderstood as work in isolation and lead to the lack of properly monitoring for their work progress [12]. The misunderstanding of Self-organization also causes other problems, such as the insufficient collaboration of information sharing and the lack of responsibility for the technical solution [13].

During our initial work, we also conducted interviews with two experts in order to get further knowledge of the existing problems that are encountered when understating and applying Scrum framework in real projects. The interviewees provided us detailed explanations of examples of the problems that existed in real projects.

The first interviewee works in a Swedish engineering and consulting company. He mentioned that they always encounter the information sharing and work management problems, such as tasks synchronization among different developers.

―Sprint tasks are assigned to individual developers, I am always suffering from troubles of arranging works, because my tasks always have some dependencies on others tasks. Moreover, in some situation, two or more developers are modifying the same code, what’s worse, we have no idea about it at all. We have to spend much more time on fixing the conflict of the code.‖

― Interviewee

The second interviewee comes from a Chinese internet company; he provided his own opinion toward the documentation usage for training new team members.

―In fact, our company wants to record the development process in documents to accumulate experience on Scrum projects. Those data, such like time estimation and tasks
allocation, can be used to train other teams or help a new team member to understand how to applying Scrum framework in projects.”

—— Interviewee

Summarizing the problems reported in the literature and the interviews, it indicates that those problems are caused by the incomplete description of the Scrum project, which always leads to the misunderstanding of the application of the Scrum framework. There were also some indications from the interviewees that graphical language for expressing the Scrum method and specific Scrum project may help the Scrum team to correctly understand the projects’ process and apply the Scrum framework.

“In my opinion, the graphical language would be helpful to improve the Scrum team performance. Because this language will enable us to describe the implementation process, track the project’s progress, and properly allocate the tasks.”

—— Interviewee from Chinese company

Generally, the graphical language can be used for describing the Scrum method and specific Scrum project to help Scrum team to understand and apply Scrum framework properly, and also be used to training team members. In this thesis, we mainly focused on investigating whether and how the graphical language for describing the Scrum process could help developers learn Scrum or contribute to training new team members in applying Scrum method. With the demand of correctly expressing the Scrum process, the desired graphical language should have the following properties:

**Enable to describe a particular Scrum project in general.**

The Scrum team can use the proposed language to formulate the blueprints of Scrum projects. And the use of the language enables the Scrum team to figure out the overall project’s process and learn how to manage and control the Scrum project.

**Enable to describe a particular Scrum project in details.**

This language can be used to describe the detailed information of each step of project realization. This can help the Scrum team to learn how to arrange the works and apply the Scrum method. The language allows the Scrum team to build different kinds of diagrams that will be used to express different aspects of the Scrum process, such like the time estimation, resource allocation and implement progress. The explicit Scrum process description will guide the team to correctly learn and apply the Scrum framework in projects.

In order to model the development process, some languages are suggested, such as UML Activity Diagram, Event Process Chains (EPCs), Business Process Model Notation (BPMN) and Domain Specific Language (DSL). Among languages, BPMN and DSL are wildly used in industry.

The BPMN was suggested as the standard notations by Object Management Group (OMG). It provides a graphical notation for specifying business processes in a Business Process Diagram (BPD), based on a flowcharting technique. In recent years, BPMN has become the most popular business process notation. Despite its popularity, the problem behind using BPMN is that BPMN is a generic language which does not focus on a specific domain. Many of the modeling tools use custom elements with dependence from the specific domain instead of the standard languages or notations [14]. Normally, this approach is more adapted to the specific domain, and it makes the modeling of the process be simpler and easy to understand for users. A DSL is such a specification language dedicated to a particular problem domain with customized building blocks. Hence, compared to the generic language BPMN, a DSL is more suitable to be applied for modeling the Scrum process. As a consequence, the desired language can be considered as a kind of graphical DSL that will be used to express the detailed information of the Scrum process for Development Teams to learn the Scrum framework accurately.

For the purpose of designing a proper graphical DSL for modeling the Scrum process, the following steps were performed:

**Analysis:** Gathering relevant knowledge of the Scrum method.

In order to accurately model the Scrum process, a deep understanding of the Scrum principles, such like the definition of development phases, team member roles, product
releases is required. We investigated the usage of the Scrum method in the literature and real projects to further understanding how to model the Scrum process.

**Design:** Providing basic building blocks of the graphical language and rules for combining them into a description of the Scrum process. Moreover, we selected a reasonable DSL development method to define the proposed language.

For the evaluation of the proposed language, we conducted both quantitative and qualitative approaches to assess the proposed language:

**Evaluation:** For quantitative research, we performed an experiment with students in the academic environment. We took a Scrum project as an example and produce two types of description of this project which were used as the training materials to introduce the Scrum process. The first one is described in a traditional way, such as textual documents. The other one includes the description of the Scrum process in the developed language. These two types of training materials of the Scrum process were used to evaluate if and what extend the proposed language is considered as useful to help the students to learn and understand Scrum method.

Besides, we also conducted an industrial survey by questionnaire with several open-ended questions to collect the Scrum users’ opinions toward the proposed language. The data collected was handled in a qualitative manner in order to assess the potential use of the proposed language in industry.

As discussed above, this thesis aims at developing and evaluating a graphical DSL for modeling Scrum process. By evaluating the proposed language, we are able to validate whether a graphical language is considered as useful in learning Scrum process.

### 1.1 Aims and objectives

The aim of this thesis project is to **develop a graphical language for modeling Scrum process and evaluate its usefulness**.

In order to meet the aim, the following objectives should be achieved:

**Objective 1.** To find out the essential artifacts used in modeling the Scrum process, the building components, their properties, and relations that are used to express the artifacts. These elements and their properties presented in the description of the Scrum process are studied and listed. The results help us to design the basic building blocks of the proposed Scrum modeling language.

**Objective 2.** To design a graphical language that is able to model Scrum process. The proposed graphical language is defined as a kind of graphical DSL that can be used to model the Scrum process.

**Objective 3.** To examine whether the proposed language is considered as useful to help developers to learn the Scrum method and understand specific Scrum projects. The result of the proposed language evaluation allows us to find out how the proposed language for modeling the Scrum process contributes to Scrum training.

**Objective 4.** To investigate the potential use of the proposed language in industry. The result of the investigation allows us to validate whether the proposed language is able to improve the understandability of the Scrum method from the Scrum users’ perspective.

### 1.2 Research questions

To achieve the objectives of this thesis project, we defined following RQs:

**RQ1:** What are the basic concepts, elements and their properties that are present in the description of Scrum process? *(Objective 1)*

**RQ2:** What is the definition of the proposed graphical language? *(Objective 2)*

**RQ3:** How the proposed Scrum modeling language contributes to Scrum training in an academic environment? *(Objective 3)*

**RQ3.1:** How the proposed Scrum modeling language contributes to learning Scrum method?

**RQ3.2:** How the proposed Scrum modeling language contributes to understanding specific Scrum projects?
RQ4: How the proposed Scrum modeling language is assessed by the Scrum users as an aid for understanding and applying Scrum method in the industry? (Objective 4)

In this study, we extracted the elements and rules used in Scrum process (RQ1) firstly. Based on the result of RQ1, we designed the proposed language (RQ2) and evaluated it (RQ3, RQ4).

1.3 Expected outcomes
The expected deliverables of this thesis project are the followings:
1. The list of the elements and their properties that are used in modeling the Scrum process. (RQ1)
2. A graphical DSL language for modeling Scrum process. Presentation of the proposed language with detailed specification, basic building blocks, rules, and syntax. (RQ2)
3. The training materials of the Scrum process. (RQ3)
4. The result of the proposed language evaluation in an academic environment. (RQ3)
5. The result of the proposed language evaluation from the industrial Scrum users. (RQ4)

1.4 Terminology

<table>
<thead>
<tr>
<th>Terms</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>The proposed (graphical) language / The proposed Scrum modeling language/ The proposed DSL</td>
<td>A graphical Domain Specific Language for describing the Scrum process, which is developed by the authors.</td>
</tr>
<tr>
<td>Scrum process</td>
<td>The flow and related activities of the Scrum projects. In this thesis, describing the Scrum process also includes the representation of the Scrum team member roles.</td>
</tr>
<tr>
<td>Domain specific language</td>
<td>Domain-specific language (DSL) is designed to express statements in a particular problem space, or specific domain.</td>
</tr>
</tbody>
</table>
2 BACKGROUND

2.1 Overview of Scrum

Scrum is an Agile process framework that has been used to manage complex product development since the early 1990s [15]. In 1993, Jeff Sutherland with John Scummiotales and Jeff McKenna introduced Scrum approach at Easel Corporation and were the first to refer to it using the single word Scrum. And then, Ken Schwaber and Jeff Sutherland jointly presented a paper elaborating on the Scrum concept at Object-Oriented Programming, Systems, Languages & Applications’ 95 (OOPSLA’ 95) conference held in 1995 in Austin, Texas. Since then, several Scrum users, experts, and authors have continued to merge their experiences and industry best practices into the Scrum methodology.

Scrum employs an adaptive, iterative, incremental, and simple approach to productively deliver projects of the highest possible value. The Scrum framework consists of Scrum teams and their related roles, events, artifacts, and rules [15]. Each component within the framework plays a specific role to guarantee Scrum’s success. The Scrum construct works in iterations called sprints, which contain the sprint planning, daily Scrums, development work, the sprint review, and the sprint retrospective. Figure 1 illustrates an overview of a Scrum project’s flow.

A Scrum project’s development cycle begins with a stakeholder meeting, and then the project vision statement will be created. The Product Owner then creates the product backlog which contains a prioritized list of product requirements written in the form of user stories. Each sprint has a sprint planning meeting during which the Product Owner and team plan about what to be done for the next sprint; the high priority user stories are considered for inclusion in the sprint backlog. A sprint generally lasts between 1-4 weeks in length. During the execution of each sprint, the Scrum team conducts a short, highly focused meeting every day to track and discuss the work progress; the meeting is called "Daily Scrum Meeting". At the end of the sprint, the Development Team holds a sprint review meeting during which they demonstrate the deliverables to the Product Owner and relevant stakeholders. The result of the review meeting is a revised product backlog that contains the adapted items for the next sprint. The sprint cycle ends with a sprint retrospective meeting, where the
Development Team reviews their performance and discusses the ways to improve their effectiveness. Then, the Scrum team moves forward into the next sprint until they accomplish the whole project.

2.2 Scrum team organization

In this section, the Scrum team organization is discussed to understand the team members’ roles and responsibilities in the context of a Scrum project. It is important to understand the defined roles and responsibilities to accurately implement the Scrum projects. Scrum roles can be broadly divided into two categories, core roles, and non-core roles.

- Core Roles

  In Scrum, there are three core roles which are committed to the project in the Scrum process. The three core roles are the Product Owner, Scrum Master, and the Development Team. A clear understanding of the Scrum core team roles is essential to ensure the successful implementation of the Scrum projects.

  **Product Owner** – The Product Owner is responsible for maximizing the value of the product and the work of the Development Team [15]. The Product Owner is one person, not a committee. The Product Owner represents the voice of the customers and manages the product backlog. The Product Owner should order the items in the product backlog and ensure the product backlog is visible and clear to all. No one is allowed to take the work from other source or requirements, and the Development Team should follow the Product Owner’s decisions in the product backlog.

  **Scrum Master** – The Scrum Master is a facilitator who is responsible for ensuring the Development Team understands the Scrum theory, practices, and rules. The Scrum Master guides, facilitates and helps the team members involved in the project to figure out which of their interactions with the Scrum team can lead to maximized the value and improve them. The Scrum Master is a servant-leader for the Scrum Team. The Scrum Master serves the Product Owner in several aspects, including: managing the Scrum meetings with the Development Team, understanding how to arrange the product backlog to maximize value, and helping the Development Team understand the Product Owner's decisions in the product backlog. The Scrum Master serves the Development Team with coaching Scrum theory and removing the obstacles in the project progress.

  **Development Team** – The Development Team consists of a group of professionals who are responsible for creating the project deliverables according to the requirements specified by the Product Owner. The Development Team is self-organizing and cross-functional, which has all of the necessary skills to create a product.

- Non-core roles

  Non-core Roles are those roles which are not mandatorily required for the Scrum project. However, the non-core roles may also play an important part in some particular Scrum projects.

  **Customer** – The customer is the individual or organization that provides the requirements and acquires the product or service. In some Scrum projects, the customer may directly interact with the Product Owner to specify the user stories. They are also involved in the Sprint Review meeting to validate the deliverables demonstrated by the Development Team.

  **User** – Users are the people who directly use the project’s product or service. They may be present at the end of the project process and show their opinions for the product to help the Scrum team to improve it.

2.3 Scrum process

Scrum process addresses the flow and its related activities of a Scrum project. According to SBOK™ Guide [16], there are 5 phases and 19 processes which will be conducted in a Scrum project. These phases and activities are shown in Table 1.
Table 1. Scrum process

<table>
<thead>
<tr>
<th>Phases</th>
<th>Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiate</td>
<td>Create Project Vision</td>
</tr>
<tr>
<td></td>
<td>Identify Scrum Master</td>
</tr>
<tr>
<td></td>
<td>Form Development Team</td>
</tr>
<tr>
<td></td>
<td>Develop Epic(s)</td>
</tr>
<tr>
<td></td>
<td>Create Prioritized Product backlog</td>
</tr>
<tr>
<td></td>
<td>Conduct Sprint Planning</td>
</tr>
<tr>
<td>Plan and Estimate</td>
<td>Create User Stories</td>
</tr>
<tr>
<td></td>
<td>Approve, Estimate, and Commit User Stories</td>
</tr>
<tr>
<td></td>
<td>Create Tasks</td>
</tr>
<tr>
<td></td>
<td>Estimate Tasks</td>
</tr>
<tr>
<td></td>
<td>Create Sprint Backlog</td>
</tr>
<tr>
<td>Implement</td>
<td>Create Deliverables</td>
</tr>
<tr>
<td></td>
<td>Conduct Daily Scrum Meeting</td>
</tr>
<tr>
<td></td>
<td>Groom Prioritized Product Backlog</td>
</tr>
<tr>
<td>Review and Retrospect</td>
<td>Convene Scrum of Scrums</td>
</tr>
<tr>
<td></td>
<td>Demonstrate and Validate Sprint</td>
</tr>
<tr>
<td></td>
<td>Retrospect Sprint</td>
</tr>
<tr>
<td>Release</td>
<td>Ship Deliverables</td>
</tr>
<tr>
<td></td>
<td>Retrospect Project</td>
</tr>
</tbody>
</table>

**Initiate** – As the initiation of a project, this phase includes the processes of founding a team, defining the product, and creating the Product Backlog.

- Create Project Vision: In this process, the Product Owner will be identified and to communicate with the customers. The business case is discussed to create a Project Vision Statement for the Scrum Team to understand the project.
- Identify Scrum Master: The Scrum Master is identified using the company's specific selection criteria.
- Form Development Team: Development Team members are selected to create the product in this process. Normally, the Product Owner should collaborate with the Scrum Master to select the appropriate developers for the specific project.
- Develop Epic(s): Epics are developed at this stage when most user stories are written in high-level functionalities description and requirements are roughly defined.
- Create Prioritized Product backlog: In this process, the defined Epics are broken down into smaller and prioritized by the Product Owner to create the Prioritized Product Backlog.
- Conduct Sprint Planning: The Scrum core team discusses the requirements in the Product Backlog to create Sprint Planning Schedule, which is a phased deployment schedule for guiding the team to create the artifacts. The length of the Sprint will be also determined in this process.

**Plan and Estimate** – In this phase, the user stories listed in the Product Backlog are described in detail by the Product Owner. The user stories are broken down into specific tasks for the Development Team to accomplish. The Scrum team estimates the effort required to develop the functionality described in the user stories.

- Create User Stories: User stories are usually described in details by the Product Owner to ensure that the customer’s requirements are clearly depicted and can be understood accurately by the Development Team.
- Approve, Estimate, and Commit User Stories: In this process, the Product Owner selects and approves the user stories for a Sprint. Then, the Scrum team discusses and estimates the effort required to implement the functionalities described in each user stories. Finally, the Development Team commits to creating and delivering the products depicted in the user stories in the upcoming Sprint.
• Create Tasks: The approved, estimated, and committed user stories are broken into specific tasks and created a task list.
• Estimate Tasks: The Scrum core team discusses and estimates the effort required to complete each task in the task list.
• Create Sprint Backlog: In this process, the Scrum team holds the Sprint Planning Meeting during which the Development Team collaborates with the Product Owner to create a Sprint Backlog containing the selected tasks to be completed in the upcoming Sprint.

**Implement** – The Development Team executes the tasks and produces the project’s product at this stage. In order to promote the tasks completion, Scrum Master should ensure the Daily Scrum Meeting be held with the Development Team to inspect the problems in the work and track the progress towards the Sprint Goal. And the Product Backlog is maintained and adapted continuously to meet the Product Goal along with the project progresses.

• Create Deliverables: The Development Team executes the tasks listed in the Sprint Backlog to produce the product. A Scrum board is often used to facilitate the team tracks the progress of the work. Tasks are grouped into three partitions: “Not Check out”, “Check Out” and “Done”, on the Scrum board guiding the Development Team to complete the tasks.
• Conduct Daily Scrum Meeting: The Development Team uses the Daily Scrum Meeting to inspect the impediment, track the progress, and arrange the work. This meeting promotes the team members to communicate and collaborate with each other.
• Groom Prioritized Product Backlog: The Prioritized Product Backlog is continuously updated and adjusted. The Sprint Review Meeting may be held, during which the changes will be discussed by the Scrum core team to update the Product Backlog.

**Review and Retrospect** – This phase focuses on reviewing the deliverables that have been done in the current Sprint, and discussing the ways to improve the performance and productivity of the Development Team. In large organizations, this phase also includes holding Scrum of Scrums Meetings.

• Convene Scrum of Scrums: In this process, Scrum teams hold the Scrum of Scrums Meeting to collaborate and check the dependencies, track their respective progress, and inspect the impediments across teams. This case is only for large projects where multiple Scrum Teams are involved.
• Demonstrate and Validate Sprint: During the Sprint Review Meeting, the Development Team demonstrates the Sprint Deliverables to the Product Owner and customers who will determine whether or not to accept the deliverables.
• Retrospect Sprint: In this process, the Development Team has the opportunity to inspect itself and create an improvement plan to be carried out in the upcoming Sprint. The Scrum Master encourages the Development Team to improve, in using the Scrum framework, its process, and practices to make it more effective and productive for the next Sprint.

**Release** – The release phase is to deliver the accepted deliverables to the customers and identify relevant documents of the project.

• Ship Deliverables: In this process, the accepted deliverables are transitioned to the customers. And then the Scrum Team ends this Sprint.
• Retrospect Project: The Serum Team collaborates with customers to retrospect this Sprint and identify relevant documents and internalize the lessons learned.
3  **Motivation**

There are several meta-modelling languages have been proposed to refine the development process, such as Unified Modelling Language (UML, O.M.G., 2007), Business Process Modelling Notation (BPMN, O.M.G., 2005), ECore (Budinsky, 2003) [17] and Graph-Object-Property-Role-Relationship (GOPRR, Kelly 1997) [18]. Alegria [19] et al. suggests using Systems Process Engineering Meta-model (SPEM) to describe the roadmap of the Agile development process. SPEM was suggested as a standard by Object Management Group (O.M.G., 2008) to model engineering process. Most of these languages are generic. Although these languages are very popular, the problem behind them is the generic languages are not focused on a specific domain such like a specific software development method. Different software development method has different design philosophy and its own particular features, which always cannot be expressed accurately in using a generic language. Many of the modeling tools are not based on the standard languages, because they cannot model the custom elements with dependence from the specific domain [14]. A general language, such as UML, does not provide a mechanism for defining the elements using user viewpoints that can help developers to understand and model a complex and specific system process.

However, in recent years, Domain Specific Language (DSL) is becoming increasingly popular in to capture the powerful abstractions of well-studied application domains [20]. DSL enables the specification of software from a customized perspective. It raises the level of the abstraction and bridges the implementation closer to the vocabulary understood by the developers, and end-users [21]. It can also make the modeling of the processes simpler to be understood by customers, and help them to track the progress.

DSL notations can be classified into two groups: graphical notations and textual notations. Comparing to the textual notations, graphical notations are easier to understand and use by non-technical people, such as the new learner or lower grade students. And also, according to the study by Lu and Sadiq [22], the graphical language enables the description of most of the business process with simpler semantics and a more abstract syntax. Therefore, in this thesis, we proposed to define a graphical DSL for modeling Scrum process.

In fact, DSL are always be used to model a specific application, but for modeling software development process has seldom been seen in literature. However, the software development process definition is very important for developers to execute proper techniques to implement the software. Scrum as the most popular method has been applied widely to development products. But the 2015 CHAOS Report released by the Standish Group, only about 39% Agile projects can be called successful (The resolution of all Agile projects from FY2011 to 2015 within the new CHAOS database) \(^1 \). Project’s success means the Development Team reaches their goals with a satisfactory result, within the planned budget and on time. Despite Agile approaches resulted in more successful projects and less outright failures than traditional development methods, companies still face drastic risks and challenges to take Agile methods. During our initial interviews, the interviewee also pointed out that team members do not have a common understanding of how to perform the Scrum method at beginning, and the emergency situations were always be handled improperly. Scrum also has an inherent problem, the lack of the description of the project implementation process. Development Teams use the Burn-Chart to manage the tasks and track the progress of the projects. Nonetheless, the product and the implementation are always misunderstood by developers. Scrum does not provide a tool for users to check the details of the development progress, what has been done, the details of the deliverables. These problems are solved by enhancing the communication. Developers use the daily meeting to explain their finished and on-going tasks and also need to know what others had done. But the fifteen minutes meeting is not enough to help them to capture all the details of the progress, and the

conflicts of the codes happen due to the lack of description of the project. Therefore, in this project, we suggest that use the graphical DSL to modeling the software development process and help the developers to understand the Scrum method and its related projects. We believe that introducing a well-defined DSL for Scrum method can not only help the inexperienced users to learn and apply the Scrum method, but also facilitate the Development Team to solve the problems existed in the implementation progress, and mitigate the failure risks.
4 RESEARCH METHODOLOGY

4.1 Overview

In order to answer the research questions, we conducted the literature study, controlled experiment, and survey.

Literatures study is an approach used for identifying, evaluating, and interpreting the available research relevant to a particular research question or topic area. In this thesis, literature study is selected to gather the knowledge regarding the Scrum method (RQ1) and the DSL (RQ2). Experiment is a systematic and scientific approach which involving the manipulation of one or more variables and the measurement of the effects of this manipulation on variables. An experiment is conducted to enable us to evaluate the proposed language in an academic environment and answer RQ3. The survey method can provide a quantitative description of the information to compare and explain the knowledge [23]. It can be used by designing a questionnaire and providing it to the participants online. In general, survey costs less time for authors to create and for participants to answer. The survey is used to collect the Scrum users’ opinions toward the proposed language for modeling Scrum process. The result of the survey enables us to investigate the potential use of the proposed language in an industrial environment (RQ4).

4.1.1 Chosen research methods

In order to find out the elements, their properties, and rules which are used to represent the Scrum process, the literature study was conducted. As a result of conducting the literature study, a set of Scrum elements and their relationships identified in the literature were obtained to provide an answer to RQ1.

Before defining the foreseen Scrum description language we investigated commonly used methods for defining graphical languages in order to pick the method appropriate for the definition of the language. Later, based on the result of RQ1, we extracted the basic building blocks and rules of the proposed language for describing the Scrum process. Then, we defined the syntax and semantics of the proposed language according to the reported DSL development method and answered RQ2.

Next, we conducted both quantitative and qualitative methods to evaluate the usefulness of the proposed language. We conducted an experiment in an academic environment and a survey in industry. For the quantitative study, the controlled experiment can provide a direct result of whether the proposed language is considered as useful in learning the Scrum method and understanding the specific Scrum projects. The type of this experiment design is a comparison design. The subjects were college students. The treatments were the description methods, the proposed language, and non-proposed language. The instruments of this experiment were the Scrum training materials which have two parts, the Scrum method introduction, and a specific Scrum project description. We produced two kinds of descriptions of these training materials. One described in a traditional way used textual documents. The other described using the developed Scrum modeling language. Before conducting the experiment, we carried out a pilot experiment to test and improve the design of the experiment to ensure the correctness of the experiment result. All subjects with no experience on Scrum were randomly divided into two groups. One group studied the Scrum framework and example projects using the training materials described in the proposed language, and the other one used the textual training materials. After learning the Scrum training materials, all subjects were required to answer several questions to measure to what extent the subjects learn the Scrum method (RQ3.1) and understand the specific Scrum projects (RQ3.2). By analyzing the result of the experiment, we validated whether the proposed language is helpful in Scrum training and answered RQ3.

Meanwhile, the survey was also conducted in the industry to assess the potential use of the proposed language. Participants were asked to compare these two kinds of descriptions and evaluate whether the proposed language helps them in understanding the Scrum process.
Participants in the survey are the Scrum users from the industrial Scrum teams. The data collected was handled in a qualitative manner in order to identify the benefits and limitations of the proposed language from the Scrum user’s perspective (RQ4).

The overall research methods and steps are presented in Figure 2. The figure should be read from left to right and from top to bottom.

4.1.2 Discussion of alternatives

With the aim of identifying the building blocks of the Scrum process, the Systematic Literature Review (SLR) and survey can be performed. The result of the SLR can provide a complete list of the elements of the Scrum process. Similarly, conducting the survey in the industry can also collect the Scrum process building blocks used in Scrum projects. However, both of these two methods require lots of time and efforts to carry out. Due to the limitation of time and resource, these two methods are unreasonable to be used to answer RQ1. In addition, Scrum is an iterative method with a small number of simple rules and elements. We can sufficiently collect the building blocks and rules by studying literature and without the
need for reviewing all the related papers. Moreover, we also considered using industrial expert interviews to collect the relevant knowledge. However, Scrum is an incremental and iterative software development framework with a small number of simple rules, conducting literature study is sufficient for extracting Scrum element, whereas, using industrial expert interview requires more extra resources. Besides, holding industrial interviews needs the industrial experts to pay more time efforts, which are not available for us.

For selecting the proper DSL development methods (RQ2), we studied two papers [37, 38], which are clearly reported the common DSL development process: decision, analysis, design, implement, and deployment. For the main development phases, analysis and design phases, there are several approaches have been proposed by researchers.

For analysis phase, Mernik [37] introduced two common ways to do the domain analysis: FODA (Feature-Oriented Domain Analysis) and FAST (Family-Oriented Abstractions, Specification, and Translation). Also, Solís-Martínez [14] reported a kind of expert-opinion based approach to conducting analysis phase. The FAST method requires collecting related DSLs or applications to do a commonality analysis. However, according to our initial investigation, there are few related DSLs or applications can be referred. Therefore, the FAST is an improper approach in this project. For expert-opinion based approach, it requires lots of Scrum expert resources and time efforts. But, it is impossible for us to access to sufficient Scrum and DSL experts. Hence, this approach is also excluded. In this study, we selected FODA to do a DSL analysis; FODA requires a set of structured features, which can be captured in Scrum literature. It is a feasible and reasonable method for us to accomplish the DSL analysis part.

For the design phase, Sylvain [25] and Karsai [38] proposed two approaches to design a DSL: from scratch or customize an existing general-purpose language. It is obvious that designing a DSL from scratch requires more efforts and experience comparing with reusing some existing language. In fact, there is a similar general-purpose language can be used for modeling activities, BPMN. Reusing BPMN can reduce our design effort and the existing language can also provide lots of valuable definitions or concepts for us to refer. Therefore, we decided to reuse some concepts from BPMN in our research to develop the proposed language.

To evaluate the usefulness of the proposed language in Scrum training (RQ3), the case study could be performed. The result of the case study can exhibit how the proposed language contributes to the Scrum training. However, for the purpose of assessing the usefulness of the proposed language, a case study is not a proper way in this study; because case study is the method that is composed of the observation of a real word situation within a natural setting. It can illustrate how the proposed language works but cannot provide a result of whether this proposed language is useful or not compared to the existing modeling languages.

In order to investigate the potential use of the proposed language in the industry (RQ4), the experiment could be conducted. The experiment can provide a result of whether the proposed language is reasonable to be applied in industry. However, to provide a useful data for further analysis, the experiment should be conducted a Scrum project in an industrial environment, but it was not available for us. Moreover, for the Scrum team, using new modeling technique requires extra resources. After an investigation, we found this option as not feasible. Performing such experiment can be considered as a follow-up research.

4.2 Literature study

In order to gather the domain knowledge and specify the domain problem, we conducted a literature study. The purposes of this literature study are shown as follow:

- To deeply understand the Scrum artifacts, events, and roles.
- To extract the elements in Scrum methods.
We used ―Scopus‖ and ―BH library‖ as our literature data source. The basic keyword we used is ―Scrum‖, and the publish date is set from 2005 to now. However, we found over 10,000 results. Then we set more keyword to filter the papers we really want. More specific keywords are shown in Table 2.

Table 2. Keywords and result for searching literature

<table>
<thead>
<tr>
<th>Keyword and restrictions</th>
<th>Paper Left</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyword: “Scrum” + “Empirical—project”</td>
<td>367</td>
</tr>
<tr>
<td>Publish date: 2008 – now</td>
<td></td>
</tr>
<tr>
<td>Keyword: “Scrum” + “development project—study”</td>
<td>174</td>
</tr>
<tr>
<td>Publish date: 2010 – now</td>
<td></td>
</tr>
<tr>
<td>Keyword: “Scrum” + “industry—project”</td>
<td>107</td>
</tr>
<tr>
<td>Publish date: 2010 – now</td>
<td></td>
</tr>
<tr>
<td>Keyword: “Scrum” + “Learning—”</td>
<td>102</td>
</tr>
<tr>
<td>Publish date: 2011 – now</td>
<td></td>
</tr>
<tr>
<td>Keyword: “Scrum” + “teach—”</td>
<td>15</td>
</tr>
<tr>
<td>Publish date: 2012 – now</td>
<td></td>
</tr>
</tbody>
</table>

Based on the searching results, we selected 8 papers which are about industry and development project and 5 papers related to reporting the learning style of Scrum and how to improve the Scrum teaching approach. Table 3 shows the reason for why we select the articles and what elements extracted from the article.

Table 3. Literature results and selected reasons

<table>
<thead>
<tr>
<th>Article reference</th>
<th>Extracted elements</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>[24, 25, 26, 27, 28, 29, 30, 3132]</td>
<td>Product backlog</td>
<td>As one of the most important elements in Scrum method, Product backlog is mentioned in all selected papers. We refer the first five papers because these articles analyze the characters of product backlog and the relationships between other Scrum elements. The paper [15] specifically describes what product backlog is and how it is used in a Scrum project. The papers [24, 26, 27, 28] illustrate the relationship with Product Owner and Scrum Master. Moreover, the papers [27, 30] also report that in some real project situation, the Product Backlog can be influenced by other people who do not belong to Scrum team. The papers [31, 32] report some ideas that how to introduce the concept ―Product Backlog‖ to students who are learning Scrum first time.</td>
</tr>
<tr>
<td>[24, 15, 27, 28, 33, 34, 35]</td>
<td>Sprint backlog</td>
<td>The papers [15, 27] explain the difference between ―Product Backlog‖ and ―Sprint Backlog‖. And the papers [24, 15, 28, 33] explain the relationship between Product Owner, Scrum Master, and Development Team. For the papers [34, 35], the authors explain the importance of Sprint Backlog in teaching and learning. Moreover, the papers provide us some basic conception of how to design the Backlog part of our language.</td>
</tr>
<tr>
<td>[15]</td>
<td>Increment</td>
<td>The official guide of Scrum [15] reported that the increment is a set of all the Product Backlog items completed in the last Sprint. A new Increment must be in useable condition.</td>
</tr>
</tbody>
</table>
The paper [15] defines the Sprint, it provides us a basic concept of Sprint such as what should be done in a sprint and how long time should be planned for a sprint. The papers [15, 27] explain the relationships between other Scrum events. The papers [26, 28] report how a Sprint is conducted in a real project. Finally, the five papers [30, 34, 31, 35, 32] describe several approaches for teaching the concept of Sprint to students.

These four types of meeting constitute the base of Scrum meeting structure. The paper [15] specifically defines these four kinds of meeting, such as what should be done in each type of meeting, how long time duration for each type of meeting. Moreover, the papers [24, 26, 27, 28, 30] report the implementation of these meetings in real projects and the problems might occur during these meetings. The papers [34, 31, 32] explain the difference introducing an approach for each type of meeting.

The three different roles constitute the whole Scrum team. The papers [24, 15, 26, 27, 28, 33, 29, 34, 31, 35, 36] define and explain these three roles. To be more specific, the paper [24, 15] defines the roles and duty of the roles. The papers [26, 33, 29] illustrate features of the roles in projects and teamwork. In the papers [34, 31, 35, 36], the authors introduce several approaches to teaching the concepts of the roles, such as role cards or simulate a Scrum team.

### 4.3 DSL development method

According to the Mernik [37], a complete DSL development process contains five phases: decision, analysis, design, implement, and deployment. Developing a DSL is not a sequential process, it means that these five phases always influence each other and depend on each other. For example, the design of DSL is often influenced by implementation considerations [37]. In this section, the different phases are separately introduced.

- **Decision.**
  
  In decision phase, the DSL developers have to make several critical decisions, the examples are what type of the DSL (graphical or textual) should be, what the DSL problem domain is, what existing DSL should be selected for adopting for the new DSL [37] and so on. Gabor [38] suggested to DSL developers that “Decide carefully whether to use graphical or textual realization”. Developers have to weigh the advantages and disadvantages for each decision because a wrong decision can waste a lot of investment effort and lead to rework in other phases [38]. However, there are still a lot of decisions that cannot be made at the beginning, because the decisions have to rely on other conditions in other phases.

- **Analysis.**
  
  In the analysis phase, the problem domain should be identified and the domain knowledge should be gathered [37]. The inputs of this phase are different knowledge sources, such as literature database, and the decisions from the decision phases, such as the problem domain of the DSL. The outputs in analysis phase are the knowledge gathered including literature about existing related DSL, basic domain-specific terminology and perform of semantics [37, 38].

- **Design.**
  
  As the name of this phase, the DSL should be completely defined and explained in design phase [38]. The input of this phase is the output of analysis phase, and outputs is a complete domain model including: a domain definition, complete domain terminology, detailed domain concepts descriptions, and feature models representing “the commonalities and variabilities of domain concepts [37] “.
There are two scales for classifying the methods of designing DSL [37]: 1) the formal nature of the design description, 2) the degree of inheriting from existing DSLs. To be more specific, in an informal DSL design description, the specification is usually explained in natural language and a series of diagrams. By contrast, a formal DSL design is written in effective semantic definition methods [37]. For the second dimension, there are two ways to develop a new DSL: Composing and reusing existing DSLs or inventing a completely new DSL [37, 38]. It is obvious that development of a completely new language requires a lot of effort. However, reusing existing DSLs to develop a new DSL is much easier. However, the selected languages need to meet the seamlessness principle to be composed [38].

- **Implementation**

  For executable DSL, after design phase, the implementation technique should be decided (back to decision phase). In the implementation phase, the executable DSL running environment or application generator should be created. The inputs of this phase are complete DSL definition from the design phase and the implementation approach decided in the decision phase. The output is the executable DSL compiler or application generator or language interpreter [37, 39].

- **Deployment**

  The deployment phase also called use phase [39], in the deployment phase, the implemented compiler or application generator should receive a DSL instance as the inputs and execute or generate the corresponding application [37].

  In our DSL development work, due to Scrum is a type of software development method instead of an executable application, we decided to develop a non-executable DSL. Therefore, the implement and deployment phases are not included in the proposed DSL’s development process. Figure 3 illustrates the process and approaches we used to develop the proposed DSL.

![Figure 3. DSL development approach](image)

### 4.4 Experiment

This section describes the steps in the experiment process, which consists of the scoping and planning. First of all, we scope the experiment according to the problem, objective, and goals. The next step is the planning, where we determine the design of the experiment and consider the instrumentations.

- **Scoping**

  Scoping is the first activity of the experiment process. It explains why the experiment is conducted. In the scoping phase, the foundation of the experiment is determined. A proper experiment scope ensures that the intention of the experiment can be fulfilled during the experiment conducting process.

  The aim of the scoping phase is to define the objective and goals of the experiment. The experiment’s scope is determined by defining its goals. And the goal is formulated based on the problem that we want to solve. In this thesis project, the main problem is to investigate if and how the proposed language for describing the Scrum project contributes to learning and understanding the Scrum process. Therefore, the purpose of this experiment is to evaluate the usefulness of the proposed Scrum modeling language compared to other describing languages.

  Textual language as the general method is always used for describing and introducing the Scrum process. In this experiment, we planned to compare the learning efficiency and
effectiveness between these two different languages, the proposed Scrum modeling language and the textual language, in introducing Scrum process to the software developers.

In order to clearly define the experiment goal, we used a goal template proposed in [40]. By using the goal template, we are able to ensure that essential aspects of the experiment are accurately defined before the planning and operation. The goal of this experiment is summarized as:

- Analyze the textual language and the proposed Scrum modeling language for the purpose of evaluation with respect to efficiency and effectiveness from the point of view of the students in the context of undergraduate and master software engineering students learning the Scrum process.

**Object of study.** The object of study is the methods used for describing Scrum process and their performance on helping students to understand and learn Scrum process. The methods evaluated in this experiment are the textual way and the proposed Scrum modeling language.

**Purposed.** The purpose of this experiment is to evaluate the impact of these two different methods for understanding and learning the Scrum process, in particular with respect to the usefulness of the proposed language. By comparing the result of the evaluation, we are able to draw the conclusions about how and if the proposed language contributes to the learning of the Scrum process.

**Quality focus.** Efficiency and effectiveness are the main qualities focus of the describing methods studied in this experiment. They are the primary effect of these two describing languages on helping developers to learn and understand the Scrum process. We measure and quantify the learning efficiency and effectiveness when subjects learn the Scrum process in different describing methods.

**Perspective.** This experiment is studied from the students' perspective. It focuses on evaluating the capacity of these two languages for helping the students to understand and learn the Scrum process. The viewpoints of the students are different from the researchers. Researchers are always concerned about the principle or philosophy of the Scrum rather than its application. Students focus more on the application of the Scrum and how to understand and apply it. Therefore, the evaluation results of the describing language are interpreted from the students' point of view.

**Context.** The context of this experiment is that the subjects learning the Scrum process in different languages. The subjects are students who major in software engineering.

- **Planning**

  After scoping the experiment, we need to make a plan to guide the execution of the experiment to ensure the experiment can be carried out properly. The planning prepares that how the experiment should be conducted. The planning phase has 6 activities, and the input to this phase is the experiment goal definition. Based on the goal definition, we should determine the experiment's context firstly. The context is the environment in which the experiment we carried out. After that, the hypothesis should be formulated and the variables of independent and dependent should be selected. Then the subject selection is conducted. The type of the experiment design is determined based on the hypothesis and the selected variables. Finally, the instrumentation provides the practical implementation of the experiment.

  The planning phase prepares a detailed and complete design for the experiment execution.

**Context:** This experiment is supposed to be conducted at the university. It is a run off-line experiment (not in an industrial environment). It is focused on the Scrum learning in an educational environment. The idea of the experiment is that the subjects learn the Scrum firstly, and then answer several questions based on the training materials. We will measure the time they spend on learning the Scrum and the number of questions they answered correctly to evaluate the efficiency and the effectiveness of different treatments.
The experiment is run with the students. Using the students as the subject is cheaper and easier to control than professionals. Besides, introducing the Scrum for students means they can learn a new development method from this experiment. It might be more interesting to the subjects, and be easier to inspire them to participate in this experiment. Another reason for conducting the experiment in an education environment is that the previous development experience of the professionals from industry may affect them when they learn and understand the Scrum process described in different languages. Less development experience can make the students focus more on the training material (created in different describing languages) itself to learn the Scrum. Thus, compared to the industry case, educational environment is easier to provide a direct result of the evaluation of the describing languages and with less confounding factors.

Hypothesis and data analysis: Hypothesis testing is an important aspect of the experiment to know what is going to be evaluated in the experiment. A hypothesis should be stated formally and clearly. If the hypothesis can be rejected by the data collected from the experiment, then the conclusions can be drawn based on the hypothesis testing.

As the goal definition expressed that we are going to compare both efficiency and effectiveness when it comes to learning the Scrum process in using two different describing languages to evaluate the usefulness of the proposed language. The first method is the textual language and the second one is the proposed Scrum modeling language. The textual language is the natural language. The proposed language is a graphical DSL created for specifically modeling the Scrum process.

The hypothesis can be expressed as following situations:

1. Subjects use the prepared training materials to learn the Scrum process and answer the questions based on the materials to test how much they learn about the Scrum. If subjects both from the group using the textual training material and the group using the graphical training material have similar learning efficiency and effectiveness, it means that whichever type of the training materials they used has no influence on learning and understanding the Scrum process.

2. In another possible case, subjects using the graphical training materials perform a better or worse understanding of Scrum process than the subjects using the textual materials. It is expected that they have higher or lower efficiency and effectiveness than students using the textual materials. In other words, subjects using the graphical materials spend more or less time to learn the Scrum process (efficiency) and also can correctly answer more or fewer questions (effectiveness) than the other group. If the proposed language has a positive result, it means that the proposed language can improve the understandability of the Scrum and help users to learn the Scrum process, and it can be considered as useful to introduce to developers. If the proposed language has a negative result, we should conclude that the proposed language is useless.

Based on these possible situations of the hypotheses, the formal hypotheses can be stated as:

Efficiency:

Null hypothesis \( H_0 \): There is no difference in efficiency (measured as the learning time and the time spend per correct answer) between students learning Scrum process described in textual language and the proposed language.

\( H_0: \text{Efficiency (Graphical)} = \text{Efficiency (Textual)} \)

Alternative hypothesis \( H_1 \): There are differences (better or worse) in efficiency (measured as the learning time and the time spend per correct answer) between students learning Scrum process described in textual language and the proposed language.

\( H_1: \text{Efficiency (Graphical)} \neq \text{Efficiency (Textual)} \)

Measures needed: Describing method (textual or graphical) and efficiency (learning time and the time spend per correct answer).
Effectiveness:

Null hypothesis

\( H_0: \) There is no difference in effectiveness (measured as the number of the correct answers) between students learning Scrum process described in textual language and the proposed language.  
\( H_0: \) Effectiveness (Graphical) = Effectiveness (Textual)

Alternative hypothesis

\( H_1: \) There are differences in effectiveness (measured as the number of the correct answers) between students learning Scrum process described in textual language and the proposed language.  
\( H_1: \) Effectiveness (Graphical) < > Effectiveness (Textual)

Measures needed: Describing method (textual or graphical) and effectiveness (the number of the correct answers)

The hypotheses mean that we should test and verify the statistical significance that the two describing languages have different efficiency and effectiveness in learning Scrum process. If we want to conclude that the proposed language is useful for learning Scrum process, we need to reject the null hypothesis, and then to validate the proposed language has a positive result on helping users to learn Scrum process.

In order to validate the statistical significance difference of the languages, the following data need to be collected to evaluate the outcomes of the experiment.

- **TMEC**: The time, measured in minutes, which was required to read the materials and answer questions.
- **NRESP**: The number of correct answers for each subject when asked questions about the Scrum.
- **TMEC/NRESP**: The time, measured in minutes per answer, which is required to produce one correct answer.

Efficiency is always measured in terms of task time, which subjects complete to read and learn the training materials. Effectiveness can be calculated by counting the number of the correct answers the subjects made when they attempt to accomplish a task. For the different question, there is a different design and purpose of evaluation. Therefore, counting the number of the correct answers can help us to figure out to what extent the subjects learned from the materials. Besides, this data can be also used in calculating the efficiency (TMEC/NRESP). Actually, the accuracy rate is also a measurement which can be used to calculate the effectiveness. However, in this experiment, these tree types of data are sufficient to perform the statistical significance difference testing, so we did not use the accuracy rate of the problems.

The hypotheses and the measures determine what type of statistical test should be used. For each measure (TMEC, NRESP, and TMEC/NRESP), there are two treatments (textual language and graphical language) should be compared. Firstly, we use the Shapiro-Wilk test to calculate each measure to verify whether it obeys the normal distribution. If the distribution could be classified as normal, the t-test is be used. The t-test is one of the most often used parametric tests. The test is used to compare two independent sample means and the design should be one factor with two treatments [40]. It can be used to test whether there is a significant difference between two treatments. On the other hand, if the test indicates that the measure is not distributed as normal, the Mann-Whitney test can be used to analyze the data. Mann-Whitney test is a non-parametric alternative to the t-test. It is always be used if the assumption made by the t-test are uncertain [40].

If the test results indicate that subjects working with the textual and graphical materials have a significant difference in learning and understanding Scrum, and moreover, the graphical language can improve the efficiency and effectiveness of Scrum learning, we can be able to conclude that the proposed language is useful to be introduced to Scrum process. On the contrary, if the test results imply there is no difference between these two Scrum describing languages and the graphical materials have a negative result, we know the proposed language is useless.

**Independent and dependent variables**: The independent variables are those variables that are manipulated and changed in the experiment. They are also the treatments what we want to investigate. The purpose of this experiment is to investigate different Scrum describing
methods, applied to two objects (training materials). Hence, there is one independent variable in this experiment, the Scrum describing language, with two values: textual language and the graphical language (the proposed language).

Different treatments may generate various results. The effect of the treatments is measured by the dependent variables. The dependent variables of this experiment are the learning efficiency and effectiveness. Efficiency is measured by TMEC and TMEC/NRESP. Effectiveness is measured by NRESP. It means that we have to ensure that the subjects can clearly write down the training time they spent and correctly mark the answers of the questions.

There are some main confounding factors should be controlled in this experiment such as the knowledge background and development experience of the subjects, their study programs, reading styles and so on.

The variables in this experiment are summarized in Table 4:

<table>
<thead>
<tr>
<th>Table 4. Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Independent variables</strong></td>
</tr>
<tr>
<td>Scrum describing method</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Dependent variables</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Main Confounding variables</strong></td>
</tr>
</tbody>
</table>

Selection of subjects: This study was conducted in an academic environment. We used the convenience and judgment sampling strategy to select subjects to conduct this experiment. Convenience sampling is a non-probability sampling technique to invite the nearest and most convenient persons to take part in the experiment [40]. In this project, it is convenient to select and access to students in the university. In order for this experiment to be representative of the entire student population, we should invite students with various education backgrounds. Therefore, the subjects selected in this experiment were from both the undergraduate and master students who major in software engineering (and related). It is important that the subjects should feel free to refuse the invitation, without any compulsion. The participations have to promise that they will carry out this experiment seriously.

Judgment sampling is another non-probability sampling strategy for selecting subjects from a population. It is a biased method that is used to select subjects when some persons of a population make better subjects than others. The reason for performing a judgment sampling is because that the differences in Scrum experience level of the subjects may affect the experiment result seriously. In this experiment, subjects were required to perform some tasks of Scrum, and we measured the effect of the treatments on them. If the subjects had a good knowledge of Scrum, they will spend less time on learning the materials and easily answer the questions based on the training materials, no matter what type of training materials they used. Hence, the Scrum development experience or knowledge level of the subject is a predominant confounding factor in this experiment. To avoid being affected by
this factor, we used the judgment sampling method to select the students who have no experience or knowledge of Scrum.

Actually, the aim of this study is to help developers to have a better understanding of Scrum, it would be better to use professional developers to carry out this experiment. However, as we have discussed in Section 4.1.2, it is impossible to arrange enough professional developers to participate in our experiment. Besides, there are many benefits of using students as the subject in an experiment. According to the study of Svahnberg [48], students are able to express their opinion about the way is works in the industry. In addition, as students, their behavior and perceptions are inclined to be more influenced by what they have read and what they have been taught. It is very important for this experiment to test how much do subjects learn from the training materials. However, professional developers tend to think about more industry realistic, which will affect the result of the experiment and make it be inaccurate to evaluate the performance of the different treatments.

**Experiment design:** In order to design the experiment, we have to consider the hypothesis to see which statistical analysis we sued to reject the null hypothesis. Based on the statistical assumptions, to evaluate the significance difference between the two different describing languages, we made the experiment design as randomization and balancing.

In this experiment, we want to investigate if the proposed language for describing the Scrum process with higher learning efficiency and effectiveness than the textual way to describe the Scrum process. The factor is the describing method and the treatments are the proposed language and the textual language. According to the purpose of the experiment, we decided to apply the randomized design to this experiment. We use the same objects for both treatments and randomly assigned the subjects to each treatment. Each subject should use only one treatment on one object [40]. It means that each subject uses one type of training materials, and the two types of training materials have the same context (be applied to one object).

Randomization is also used to average out the effect of a factor that may occur [40]. For example, we cannot measure how much the subjects know about the Scrum, however, the knowledge background of the subject is a critical confounding factor in this experiment. It is difficult to control this factor. Someone may know a little about Scrum, someone may not. Hence, the randomization design can be used to mitigate the influence of the experience level differences of the subjects. Besides randomization, we have to balance the number of the subjects in each treatment group. Because we use the t-test to analyze the data, the t-test is used to process two datasets with the same number of data. Therefore, it is necessary to make sure that each treatment has an equal number of subjects.

**Instrumentation:** The instruments for this experiment are the training materials. Before execution, we developed two types of training materials, the textual version, and the graphical version (The proposed language).

The training material has two parts, the introduction of the Scrum process and a project example of the Scrum application. At the end of each part, there is a quiz with several questions to test how much the subject learned after studying the provided material. By collecting and analyzing the data from the quiz, we could measure the learnability of each training materials. The training materials as the instruments provide a way to monitor the treatment’s result of this experiment.

The first part of the training material is a brief introduction to the essential ideas of Scrum. It aims to provide subjects a basic knowledge and understanding of Scrum. This part included a description of the Scrum process, the artifacts of Scrum and the Scrum team roles. In order to provide an accurate knowledge introduction, this stuff is edited and managed based on *The Scrum Master Training Manual* [41]. The training manual is a guide to help the Scrum learner to pass the Professional Scrum Master Exam (PSM). Compared to other literature, the introducing pattern and context of this training manual are easier to be understood and learned. As a professional guide for PSM, it is more focused on the application rather than the pure Scrum philosophy or theory; hence, it is closer to the
guidance of real development conditions. The quiz’s questions of the first part are also selected from the training manual. The little quiz contains 5 questions. These questions are designed to evaluate whether subjects understand the essential elements, relationship and execution process of the Scrum method. There were two types of the questions asked in the quiz:

1. Checking the understanding of the construction and the responsibility of the Scrum team roles;
2. Asking for how to arrange the activities during Scrum iteration cycle.

The result of the quiz would allow us to measure how much do the subjects learned and understood about the Scrum. It also indirectly reflected that which type of training material performs better on Scrum training.

The second part contains a project example of Scrum process that subjects can learn how to apply the Scrum method in practice. The example project can be considered as the experiment object, which revised from the specific Scrum project document. It is important to select an appropriate experiment object which can be used as a medium to evaluate the learnability of the two Scrum modeling languages. We selected a specific student Scrum project as the experiment object and then described this example project in two ways, the textual, and the graphical way. The reason we selected a student project is that the industrial Scrum project is too complicated to understand for new learners. Besides, it is difficult to access to a specific Scrum project in the industry due to some commercial reasons. The student project has more advantages over the industrial project. In general, student project’s scale is relatively small, that will be easier for learners to gain the details of the project within in a limited time.

In consideration of the limited experiment execution time, we just introduce the overview of the project and illustrate two Sprints as examples for participants to learn and understand the Scrum process. We also intentionally miss some important elements of the Scrum process to detect whether subjects understand the Scrum and find the mistakes in application. The quiz attached to the second part has 7 questions, which focused on testing the subject’s comprehension on Scrum, the learnability of the training materials, and the ease of use of finding the details information of the project. The product of the project was iteratively produced, for the dynamic changing part, whether the subject can easily find the correct information in the training materials. The types of the questions can be summarized as:

1. Asking for the how many days in a specific Sprint;
2. Asking for the content of the updating tasks;
3. Checking whether the process or activities were complete according to the Scrum method definition.

4.5 Survey

As another important research method in our approach, a survey is used to investigate the industrial Scrum users’ opinions toward the proposed DSL for describing Scrum.

In order to make the survey process has a better reliability, we decided to design our survey following Linaker’s survey guidelines [42]. In this section, details of the survey plan were described based on the guidelines.

- **Survey Objectives**
  The objective of the survey is to collect the Scrum users’ opinions towards the proposed DSL, and then to investigate the potential use of the proposed language in the industry (RQ3.2).

- **Target Population**
  The Scrum users who have Scrum experience are the target population. Since the objective of this survey is to evaluate the proposed DSL and provides opinions, only experienced Scrum users can provide the reliable response for the survey.
Sampling Plan

As the target population is set to Scrum users who have Scrum experience, the random sampling method is obviously not suitable for this survey. In the guideline [42], the writer suggested four types of non-probabilistic sampling methods: accidental sampling, quota sampling, judgment sampling, and snowballing sampling.

Accidental sampling is the easiest way to select the survey samples, the only criterion is convenience [42]. However, as students, we do not have too much resource to contact with industrial experts. Moreover, to improve the reliability of the survey result, we require the selected Scrum users have at least one-year experience for Scrum. Hence, this sampling method is excluded.

Quota sampling is to divide the population into different subset basing on their different characteristics, then chose a number of suitable units in different subsets [42]. However, again, the problem is we do not have a big number of populations to conduct it. Besides, the only characteristic we concern is the Scrum users' experience of Scrum; hence we also do not have enough characteristics to classify the Scrum users. Finally, this sampling method is also excluded.

Snowballing sampling is a type of extends accidental sampling, we cannot control the requirement for Scrum experience. Then the snowballing sampling method is excluded.

Finally, we selected judgment sampling. The main reason is we have a clear selection criterion, which is the experience of using Scrum should be more than 1 year.

Pilot and Expert Review

In order to make the survey have a better quality and get more accurate result. We performed a pilot survey with experienced Scrum users, who have more than three years Scrum experience. In the pilot survey, the pilot survey questionnaire is sent to the selected Scrum users via email. Then we collected the answers from them. The reviews of the survey questionnaire were also received at the same time. After that, we improved the questionnaire according to the received answers and reviews.

Survey Questionnaire

The questionnaire contains two parts. The first part is the example project described by both textual and graphical ways. The second part is a set of questions about the opinions of comparing the two expression types.

For the question part, we designed 10 questions, including four close-ended questions and six open-ended questions. Table 5 shows the questions and reason why we ask those questions.

<table>
<thead>
<tr>
<th>Question</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>How long time have you been using Scrum?</td>
<td>This question is used to distinguish the experience level of the developer.</td>
</tr>
<tr>
<td>How did you learn Scrum?</td>
<td>This question is used analyze the most common Scrum learning method nowadays.</td>
</tr>
<tr>
<td>Do you think a graphical language can help you to understand Scrum process or project?</td>
<td>This question is used to measure graphical language’s value for experienced Scrum users.</td>
</tr>
<tr>
<td>Do you think a graphical language can help the new learner to understand Scrum process in a better way?</td>
<td>This question is used to measure graphical language’s value for beginners of Scrum.</td>
</tr>
<tr>
<td>Which way (textual/graphical) you would like to use in learning and teaching Scrum process.</td>
<td>This question is used to judge which way (textual or graphical) the developer prefers to use when learning and teaching Scrum.</td>
</tr>
<tr>
<td>In your opinion, what are the advantages and disadvantages of teaching the Scrum process in using this graphical language?</td>
<td>This question is used to collect the Scrum users’ opinions towards graphical language.</td>
</tr>
<tr>
<td>In your opinion, what are the advantages and disadvantages of teaching the Scrum process in using the textual materials?</td>
<td>This question is used to collect the Scrum users’ opinions towards textual language.</td>
</tr>
</tbody>
</table>
What are the problems or obstacles when you learn the Scrum process?  
This question is used to collect the problems happened in traditional Scrum learning.

Do you think a graphical language can help you to solve or mitigate the above problems, and why?  
This question is used to collect the potential solutions for above problems by using graphical language.

Will you use this graphical language in the future work? Why?  
This question is used to analyze the willing of introducing a graphical language into an industrial environment.

- Implementation Plan
  
The survey would continue four weeks. Due to the lack of industrial Scrum users' resource, and the time limitation of our master thesis, we considered four weeks is a proper time length for our survey.

  There are two main sources to have contact with Scrum users: contacting software companies for asking if there is any experienced developer interested, and contacting Scrum users via social networks applications, such as Facebook and LinkedIn. The survey questionnaire is sent by email.

- Data Analysis Strategy
  
  We performed the qualitative content method to analyze the data because most of the survey questions are subjective. As we mentioned before, the purpose of this survey is to collect industrial Scrum users' opinions toward the proposed language in order to investigate its potential use in industry. By conducting the qualitative analysis, we identified the benefits and limitations of the proposed language from the Scrum user’s perspective, as well as assessed the usefulness of the proposed language.
5 THE PROPOSED LANGUAGE DEVELOPMENT

5.1 Introduction of DSL

Domain-specific language (DSL) is a type of languages designed for a specific domain. DSL provides "substantial gains in expressiveness and ease of use compared with general-purpose language in their domain" [43, 37]. Considering the features of a particular domain, DSL improves the knowledge communication between domain experts and developers [44]. DSLs are designed for a limited domain. Therefore, the notations and constructs can be concisely expressed without ambiguous general languages.

A well-designed designed DSL should fulfill the following principles [44]:

- A DSL provides a direct mapping to the elements, relationships, and attributes of the specific domain.
- The DSL must use the normal terms of the specific domain. The terms must be easy to understand during communication between developers and domain experts.
- The DSL cannot contain any ambiguous expression.

Regarding the executability can be classified into four scales [37]. The first type of DSL has well-defined executable semantics, such as HTML and SQL. These kinds of DSLs can be executed directly. Then the second type is the application generator. Users input the formulated language (DSL instance) and the DSL generator generates the corresponding application. The third classic DSL type is not primarily designed for execution, but they can be useful for application generation. For example, BNF is not designed for direct execution or generating an application, however, it can be used to define other DSLs or as an input language for a parser generator [37]. The last kind of DSLs is designed as a non-executable language, for instance, some DSLs just meant to represent a domain-specific data structure [37, 45]. This kind of DSLs can benefit from the different tool types such as tailored editors.

In this project, we suggest introducing the graphical DSL for modeling the Scrum process. Since Scrum is a development method instead of an executable application. Therefore, it is reasonable to develop a non-executable DSL for describing the Scrum process. As we have discussed in section 4.3, the graphical DSL should be developed in three steps: the decision, analysis, and design phases. The development and deployment phases have been removed from the DSL development process in this paper. The following sections illustrated the detailed development process of the proposed DSL.

5.2 The DSL decision phase

In the decision phase, we decided a set of critical problems such as the problem domain of the DSL, the type of the DSL and so on. These problems decided out our later development direction.

After the initial interviews, the results showed that all of the interviewees are all holding a positive opinion toward the idea that introducing a DSL for Scrum process. They think introducing a DSL may help them to understand Scrum method and specific Scrum projects. Therefore, we did the first decision, which is introducing a DSL for representing Scrum process and concepts.

For the DSL representation type, we compared graphical DSL and textual DSL, the result was obvious. Gabor [38] listed the advantage and disadvantage of these two type of representation. Textual representations for example usually have the advantage of faster development and are platform and tool independent. [38]" On the other hand, graphical models provide a better overview and easier understood models. [38]" We finally decided to develop a graphical DSL because better overview and easy understanding are exactly what we want.

Business Process Modeling Notation (BPMN) is a standard notation for modeling process [46, 47]. This is a very popular graphical notation in representing process domain. And there are thousands of documents and articles talking about BPMN. Considering the
The proposed DSL has the similar purpose: illustrating process. The only difference is BPMN is for all business process, in contrary, the proposed DSL is only focused on Scrum process. Therefore, we decided to reuse some basic concept from BPMN. Of course, the proposed DSL should be represented in graphical.

In addition, the proposed DSL is designed for helping Scrum learner to better understand the Scrum process and concepts. There is no executable application can be generated. Therefore we decided to develop a non-executable DSL. Then the future work can modify the proposed DSL to an executable DSL to fulfill requirements in other domain problems.

Table 6 shows the decisions we made during this phase:

<table>
<thead>
<tr>
<th>Decision</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduce a DSL for representing Scrum process and concepts</td>
<td>There are some domain gaps and needs requiring a representation approach to mitigate.</td>
</tr>
<tr>
<td>Reuse concepts from BPMN</td>
<td>There are two advantages if we reuse some elements of BPMN. First, it can reduce the development effort, because we do not have to design a new language element and it related rules. Second, it will be easier to be accepted by the people who have learned BPMN before, because they do not have to learn new concepts for using this language.</td>
</tr>
<tr>
<td>Graphical DSL</td>
<td>Graphical DSL is easier to understand for beginners.</td>
</tr>
<tr>
<td>Non-executable DSL</td>
<td>Unnecessary to implement and deploy at the current stage.</td>
</tr>
</tbody>
</table>

5.3 The DSL analysis phase

5.3.1 Domain analysis

In this phase, we conducted a formal domain analysis. According to Mernik [37], a formal domain analysis output contains four parts: 1) domain definition, 2) domain terminology, 3) descriptions of domain concepts, and 4) feature models. Hence, we represented our work fulfilling this formal structure.

In this section, first, we identified that how we analyzed and classified the selected papers. Then based the papers, we specifically explained the domain scope and domain terminology. Furthermore, for each term, we described its concepts and why we extracted it from literature. Finally, we demonstrate the feature model for completing this formal analysis.

At the beginning of the analysis phase, first, we conducted a literature study to select literature we need. The result of the literature study is shown in section 4.2, then we got a set of literature. In addition, in the decision phase, we already decided the DSL’s purpose, which is helping Scrum learner to get a better understanding during studying process. Therefore, the articles are classified into two categories: using Scrum in projects and teaching Scrum for students. The articles in the first category were used to extract the Scrum elements and the papers in the second category were used to elicit more effect approaches for teaching and training Scrum. Then we composed the education-based features and the proposed DSL.

- The proposed DSL scope

The proposed DSL provides a graphical notation and model for introducing Scrum process to Scrum learners and improving learners’ understanding of Scrum concepts and specific Scrum project. Based on this description, the proposed DSL is a non-executable domain specific language. Hence, an application generator or compiler is not included in the DSL scope.
The domain terminology and description of domain concepts

We extracted the Scrum elements from those papers. To make the outputs more consistent, we combine the domain terminology and descriptions for domain concepts together. For each term extracted, we described the domain concepts and the attributes for each element together.

5.3.2 Extracted elements

The section demonstrates the extracted elements and their attributes.

- Product backlog
  
  As the base of the Scrum framework, all of the requirements should be selected from the Product Backlog. A product backlog is a list of requirements [24]. Each requirement should have a property to illustrate its importance and the sequence of development. Additionally, each requirement has a brief explanation about what the requirement is. The product backlog is maintained by Product Owner [15]. The requirements listed in product backlog are called —user story” [24, 26, 28] or —backlog item” [24, 27, 28].

  The reason for selecting product backlog as an element is very obvious. Each Scrum project must have a product backlog. This is the basic conception of Scrum method.

- Sprint Backlog
  
  Sprint Backlog is also a set of requirements. The requirements are selected from Product Backlog in Sprint Planning Meeting. However, Sprint backlog not only is a subset of product backlog but also provides much more specific details for each requirement [27], such as what is —done” for each functionality [15]. Product Owner should specifically describe the selected Product Backlog Items to Development Team. Sprint Backlog lists everything should be done in the sprint. In theory, once a sprint starts, all new update should be added in the next sprint. In practice, during a Sprint, some little but urgent requirements also can be added in Sprint Backlog. Moreover, there can be several teams sharing the same Product Backlog, but only one team takes a specific Sprint Backlog [15].

- Product Backlog Item
  
  This element is an attribute of Product Backlog. In Product Backlog story list, user stories are briefly described and the development time is generally estimated.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Reason and description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A short, descriptive story name can briefly explain the purpose of the product backlog item [28].</td>
</tr>
<tr>
<td>Importance</td>
<td>Importance is used to order the backlog story items. Only the Product Owner can decide this attribute. This attribute is essential because all of the ordering work is based on this attribute [24].</td>
</tr>
<tr>
<td>Initial time estimate</td>
<td>This attribute is designed for roughly estimate a user requirement on an initial level. This attribute can help Product Owner and Scrum Master to arrange the scrum activities [28]. This number should be given by the Development Team who are going to develop this item. The estimation doesn’t have to be very accurate because it is often used to compare with each other story items. For example, if a story item is estimated for one-time point and another story item gets two time points. It only means the second user story requires more time to complete. It doesn’t mean the second user story has to spend time as twice as the first one [28].</td>
</tr>
</tbody>
</table>

- Sprint backlog Item
  
  Sprint Backlog Item is similar with the Product Backlog Item. However, the description of user stories in a sprint should be more specific and well-estimated. Sprint backlog lists all activities that have to be completed in this sprint [27]. In a sprint planning meeting, the Product Owner specifically explains the user requirements. Then the Development Team
should estimate the time effort carefully. These data should be recorded and will be used in the whole sprint.

Table 8. Sprint Backlog Item attribute list.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Reason and description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A short, descriptive story name can briefly explain the purpose of the product backlog item [28].</td>
</tr>
<tr>
<td>Importance</td>
<td>This attribute is used to measure the development priority in a sprint. The value of this attribute should be settled at the sprint planning meeting by the Product Owner. During a sprint, the Product Owner can update this value to adjust and control the development progress [27].</td>
</tr>
<tr>
<td>Time estimate</td>
<td>This attribute is similar to the attribute in “Product backlog story list”. However, this attribute is a more accurate time estimation. In a Sprint Planning Meeting, the Development Team should carefully give a time estimation for each sprint backlog story [28].</td>
</tr>
<tr>
<td>Scenario</td>
<td>This attribute is a high-level description of how the story can be demonstrated at the end of sprint [3].</td>
</tr>
<tr>
<td>Criteria</td>
<td>There are a lot of literature that emphasize the importance of defining “Done” for a story [24, 15, 27, 28, 33]. Actually, “Done” is a kind of criteria. When a software artifact meets a criterion, this story will be regarded as “Done”.</td>
</tr>
<tr>
<td>Category</td>
<td>This attribute indicates what kind of requirement this item is. For example, this attribute can be “testing” “design” “optimization” and so on [28].</td>
</tr>
<tr>
<td>Requestor</td>
<td>On the Product Owner side, he/she may want to keep contacting with customer or stakeholder to get more details about requirements. On the development side, the team can use this attribute to trace back the person who really need this item and get more specific and clear feedback via face to face discussing [28].</td>
</tr>
<tr>
<td>Bug Tracking ID</td>
<td>If a project has a separate bug tracking system, this attribute will be useful to keep track of any correspondence between an item and bugs [28].</td>
</tr>
</tbody>
</table>

- Task

Scrum Development Team usually defines tasks to accomplish the Sprint backlog stories. According to Quaglia [26], all tasks should be defined within the sprint scope. Then the team members decide who is going to do what considering the task they have defined. Although this element is not defined in Scrum formal documents, it is still a good reason to extract this element because there is a lot of reported usage of the task in practice [26]. Usually, a task is assign to only one developer.

Table 9. Task attribute list.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Reason and description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A short, descriptive story name can briefly explain the purpose of the task.</td>
</tr>
<tr>
<td>Estimated time</td>
<td>This attribute is used to measure the time spent on the task.</td>
</tr>
<tr>
<td>Actual time spent</td>
<td>Actual time is used to record the actual time spent on this task. So that these data can be selected and used to measure the similar task time effort.</td>
</tr>
</tbody>
</table>

- Increment

There is not too much discussion in papers and official guide. The only definition is “Increment is the sum of all the Product Backlog Items completed in a Sprint [15]”. Moreover, a new increment has to be completely finished. Product Owner can decide to
release it or not. Basically, Increment is a set of Product Backlog Item. This list is used to show the “done” item.

- **Sprint**

The Sprint is a period of time the Scrum team works after having conducted the Sprint Planning meeting [27]. It is the heart of Scrum, and all development jobs will be done in Sprints [27].

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Reason and description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal</td>
<td>The sprint goal is an objective set for the sprint that can be accomplished through the implementation of the sprint. During the sprint planning meeting, the sprint goal is created by the Product Owner. It provides guidance to the Development Team and gives the team more flexibility to conduct their work [15].</td>
</tr>
<tr>
<td>Sprint backlog</td>
<td>This attribute is the core of the sprint. All selected user stories and related information will be presented in the Sprint backlog.</td>
</tr>
<tr>
<td>Time duration</td>
<td>This attribute is used to record the time which is going to spend on this Sprint.</td>
</tr>
</tbody>
</table>

- **Meeting**

In Scrum method, meeting is a very important way to communicate within a Scrum team and control the progress of development [15]. Therefore we extracted this element to indicate all meetings. A specific kind of meeting category should be filled in Category attribute.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>This attribute is used to illustrate the main subject for this meeting.</td>
</tr>
<tr>
<td>Start time</td>
<td>This attribute is used to record the start time of a meeting.</td>
</tr>
<tr>
<td>Duration Length</td>
<td>Different kind of meetings have different time duration, for example, Sprint planning meeting always holds for about 2 hours [28]. However, a daily scrum meeting only needs 15 minutes [27, 28]. Therefore, it is necessary to specify the length of a meeting.</td>
</tr>
<tr>
<td>Category</td>
<td>In Scrum theory, there are three kinds of basic meeting, which are Sprint planning meeting, daily scrum meeting, sprint review meeting, and sprint retrospective [15]. Considering the practical situation, there might be more different kinds of meeting; we design this attribute to specify the category of the meeting.</td>
</tr>
<tr>
<td>Participants</td>
<td>This attribute refers all the people who are going to participate in the meeting. Also, the roles of participants should be specified in this attribute.</td>
</tr>
<tr>
<td>Resources/Input</td>
<td>This attribute specifies all resources or input materials needed in this meeting. For instance, in sprint review meeting, the input of review meeting is the software artifact developed in this sprint [28, 33].</td>
</tr>
<tr>
<td>Result/Output</td>
<td>This attribute specifies all result or output materials needed in this meeting. For example, a sprint goal could be a result of Sprint planning meeting [27].</td>
</tr>
</tbody>
</table>

- **Participant**

This element represents a single person who is involved in this Scrum project. The normal Scrum Roles consist of Product Owner, Scrum Master, and Development Team. Product Owner has only ability to maintain the product backlog, for instance, adding a new user story into the Product Backlog, and setting the importance of all user stories [24] [15, 28].
Scrum Master is one or a group of people who help Scrum Owner and Development Team to arrange and conduct Scrum activities [15].

In theory, Development Team should be a cross-functional and self-organizing team [24] [15]. It means that the Development Team should have the ability to handle all technical problems. However, in practice, sometimes, scrum teams are semi-functional teams [28, 33]. Therefore, some outside participants might be involved to help the development to complete the sprint.

Hence, this participant can be Product Owner, Scrum Master, Development Team, or other special roles. In practice, the three roles suggested by Scrum guide [15] are not sufficient for a real scrum project. For example, sometimes, customers need to directly communicate with developers [33], this kind of activities is not included Scrum activities. Scrum Master has no responsibility and experience to arrange this kind of activities. That is why some companies introduce more roles to solve this kind problem. The most typical role is a product manager, which comes from traditional development models [29, 30]. In some situation, customers need to directly communicate with the Development Team. Thus, a customer is involved in this project [27, 33]. Therefore, the element “participant” should have an ability to represent these different roles.

### Table 12. Participant attribute list.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Reason and description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role</td>
<td>This attribute is used to specify the role of a Scrum project, such as Scrum Master, Product Owner, and Project manager [29, 30].</td>
</tr>
<tr>
<td>Contact</td>
<td>In scrum projects, stockholders always need to communicate with each other. Therefore an attribute for recording how to contact a stockholder is necessary.</td>
</tr>
</tbody>
</table>

- The feature model

As we explained before, the proposed DSL’s purpose is describing Scrum for beginners of Scrum. Consequently, the proposed DSL’s features can be simplified to demonstrating different knowledge. There are three categories of knowledge that the proposed DSL should represent to learners.

The first category is terminology. In this part, The DSL will explain the meaning all of Scrum elements. Also, the corresponding element legends will be demonstrated in this part. The second part is Scrum Model. By illustrating the Scrum conceptual prototype, this part mainly describes the Scrum theory, Scrum principles, and the relationship between Scrum elements. The last part of the proposed DSL is Scrum project section. This part is used to combine the above two parts into a specific project. The language user can easily build up a Scrum project process model according to syntax and semantics of the proposed DSL. The syntax is about how to construct a legal language expression, while, the semantics is about the meaning of constructed expression. Based on this part, the DSL provides a convenient way to manage and express specific Scrum projects.

**Figure 4** demonstrates the DSL feature structure.
5.4 The DSL design phase

This section describes the complete DSL design details. According to Mernik [37], Damyanov [44], a formal DSL definition should consist of the regular expression; attribute grammars, syntax specifications, and semantic specification. The following subchapters demonstrate these requirements in different aspects.

5.4.1 Language structure

In section 5.3 we already demonstrated the overall feature structure. The language structure can be divided into the following three parts: Terminology, Scrum Model, and specific Scrum project. In this section, the more detailed structure will be involved and explained.

In the ―Terminology‖ part, all of Scrum elements are classified into three different categories: role, event, and artifact, according to Scrum official guideline [15]. There are only three roles in Scrum official definition, which are Development Team, Scrum Master, and Product Owner. However, in order to enable the proposed DSL to illustrate the Development Team structure and team configuration, such as how many developers and how much time of each developer can contribute to the team, we added a new role element ―developer‖ at the next level of the Development Team. Figure 5 illustrates the structure of ―role‖ in the proposed DSL.
In the Scrum official guide [15], there are five types of event, which are Sprint, Sprint Retrospective, Sprint Review Meeting, Daily Scrum Meeting, and Sprint Planning. Whereas, Sprint Retrospective, Sprint Review Meeting, Daily Scrum Meeting, and Sprint Planning Meeting can be classified to "Meeting", because they are all held as meeting style, having the same element attributes. The only differences are the purpose and length of time box. Hence, we added a new level "meeting" between "Event" and the meetings to make the structure more reasonable and understandable. Figure 6 shows the structure of "Event".
The "artifac" contains Product Backlog, Sprint Backlog, and Increment, each of them is a set of "user story", and the only difference is that the Product Backlog and the Increment consist of Product Backlog Items and the Sprint Backlog only contain Sprint Backlog Item [15]. The difference is that Sprint Backlog Items have more detailed descriptions than Product Backlog Items.

As we introduced before, Scrum only consists of several principles and concepts. It doesn’t provide too much detail. Whereas, the purpose of the proposed DSL is to specifically demonstrate Scrum projects. In order to mitigate this gap, we added a new artifact "task" to illustrate more specific information about how a Sprint Backlog Item is divided and solved. On the other hand, as we all know, Scrum is one of Agile methodology. The biggest advantage of Agile is agility. In order to highlight the agility, again, we added another new element "update". The element can be used to express Product or Sprint updates in a Sprint or between two Sprints.

**Figure 7** demonstrates the structure of "artifac".

![Figure 7. Scrum artifact structure](image)

The second part named "Scrum Model", actually, this part is only used to demonstrate the Scrum theory on an abstract level. Designing this part has an educational purpose. If we only want to introduce a graphical DSL for modeling specific Scrum project, this part is completely useless at all, because this part doesn’t belong to any DSL essential part. In another word, even this part is removed; the proposed DSL is still complete. The "Scrum Model" part does not define any term, syntax, or semantics. It is a constant model for showing Scrum concepts. Nevertheless, on the other hand, it is necessary to demonstrate the Scrum Model for Scrum beginners who are not familiar with Scrum. We have to consider that most of the users are beginners in Scrum. Even if we introduce all terms in Scrum, all syntax, and semantics to the users, they still do not know how Scrum project works. Hence, "Scrum Model" part should be introduced to improve users’ understanding of Scrum theory. In "Scrum Model", the proposed DSL explains the Scrum theory, Scrum principles and the relationship between them. **Figure 8** shows the structure of "Scrum Model".
The last part is "Scrum project". This part provides an approach to manage and illustrate specific Scrum projects. Moreover, the syntax and semantics are also defined in this part. Besides, this part can be regarded as a composition of "Terminology" and "Scrum Model". First, users learn Scrum elements and terms from "Terminology" part. Then, the Scrum theory and principles are used to build up an abstract prototype of Scrum project in users' mind. Finally, users use the syntax and semantics to manage or create the specific Scrum project in the proposed DSL. Figure 9 illustrates the structure of "Scrum project" part.

5.4.2 Terminology and element design

In this section, we described the details of the elements and their attributes selected from the extracted elements in the analysis section 5.3. Moreover, the legend designed for each element was also illustrated in Table 13. Actually, the attributes of each element were designed to be sufficient and flexible enough to allow users to select the necessary items to describe an element. Users do not need to use all of the attributes.
<table>
<thead>
<tr>
<th>Element</th>
<th>Description &amp; Attribute</th>
<th>Legends</th>
</tr>
</thead>
</table>
| Participant                   | **Description:** This element is used to indicate Product Owner, Scrum Master, and Developer.  
                                | **Attribute:**  
                                |         |
|                               | **ID and Name:** Using a unique ID or Name to specify a participant.  
                                |         |
|                               | **Role:** This attribute is used to specify the role in a team; it can be the Product Owner, Scrum Master, or Developer.  
                                |         |
|                               | **Contact Information:** This attribute is used to specify the participant's contact information, such as phone number or working address.  
                                |         |
|                               | **Note:** This attribute is used to specify the information that is necessary but not included in this element.  
                                |         |
| Development Team              | **Description:** This element is used to show the structure of developers in a Scrum Development team.  
                                |         |
|                               | **Attribute:**  
                                |         |
|                               | **ID and Name:** Using a unique ID or Name to specify a Development Team.  
                                |         |
| Sprint                        | **Description:** Sprint element is used to start a new Sprint in the proposed DSL.  
                                |         |
|                               | **Attribute:**  
                                |         |
|                               | **ID:** Using a unique ID to specify a Sprint.  
                                |         |
|                               | **Time length:** This attribute is used to specify the time supposed for completing a Sprint.  
                                |         |
| Meeting                       | **Description:** We reused BPMN’s Activity Box as the meeting element in the proposed DSL to specify the details of a meeting.  
                                |         |
|                               | **Attribute:**  
                                |         |
|                               | **Meeting:** Only can be one of the four candidate options [Daily Scrum meeting, Sprint Planning Meeting, Sprint Review Meeting, Sprint Retrospective]  
                                |         |
|                               | **Time length:** This attribute is used to specify the time spent for completing a Meeting.  
                                |         |
|                               | **Subject:** The attribute specifies the topic of Meeting.  
                                |         |
|                               | **Participant:** The attribute specifies the participant which value can be developer or Development Team.  
                                |         |
| **Product Backlog Item** | **Description:** This notation is reused by BPMN's Data Object. This element has to be placed in element Product Backlog. Using this element singly or connecting this element to any other element is meaningless.

**Attribute:**

- **ID:** Using a unique ID to specify a Product Backlog Item.
- **Importance:** This attribute is decided by Product Owner, can only be a number, to indicate the importance of this Product Backlog Item.
- **Estimated Time:** This attribute indicates the estimated time effort for completing this Product Backlog Item. The time unit can be hour or day. The value has to be a number.
- **Description:** This attribute is used to generally describe the requirement. |

| **Product Backlog** | **Description:** This element is used to show the Product Backlog Items. The arrow indicates the sequences of importance.

**Attribute:**

- **Product Owner:** This attribute is used to specify the Product Owner who creates this Product Backlog. The value can be the Product Owner's ID or name.
- **Scrum Master:** This attribute is used to specify the Scrum Masters who are in charge of this Product Backlog. The value can be the Scrum Masters' ID or name. |

| **Sprint Backlog Item** | **Description:** Same with Product Backlog Item, this element notation partly reuses from BPMN's Data Object. This element has to be placed in element Sprint Backlog. Using this element singly or connecting this element to any other element is meaningless.

**Attribute:**

- **ID:** Using a unique ID to specify a Sprint Backlog Item.
- **Category:** This attribute is used to classify the work type of the Sprint Backlog Item, such as test, development, and design.
- **Importance:** This attribute is decided by Product Owner, can only be a number, to indicate the importance of this Sprint Backlog Item.
- **Requestor:** This attribute indicates the person who comes up with this Sprint Backlog Item. Most of the time, this person is the Product Owner, however, in some situation, the |
| Development Team will have technical requirements. |
| Time estimation: This attribute indicates the estimated time effort for completing this Sprint Backlog Item. The time unit can be hour or day. The value has to be a number. |
| Bug Tracking ID: Using a unique ID to identify the bug position in a Sprint. The ID can be number and characters. |
| Story scenario: This is a specific user story or technical requirement. |
| "Done" criteria: The criteria for finish this Sprint Backlog Item. |

| Description: This element is used to show the Sprint Backlog Items. The arrow indicates the sequences of importance. The Section 5.5 will show an example for using Sprint Backlog to represent user requirements. |
| Attribute: |
| Goal: The goal of this Sprint should be expressed clear and simple. It explains the work direction of a Sprint. |
| Development Team: This attribute is used to specify the Development Team which is responsible for this Sprint Backlog. In a Scrum project, more likely, there are several Scrum team cooperating with each other. But one Sprint has to assign to only one Development Team. The value has to be a Development Team's ID. |

| Description: Task is not an element which has been clearly defined in Scrum principles. However, introducing this extra element can help developers describe a Scrum project more accurately and specifically. |
| Attribute: |
| ID: Using a unique ID to specify a Task. |
| Assigned developer: This attribute is used to specify the developer who is responsible for this Task. |
| Time estimation: This attribute indicates the estimated time effort for completing this Task. The time unit is often hourly. The value has to be a number. |
| Actual time effort: This attribute indicates the real time effort for completing this Task. The time unit should be hourly. The value has to be a number. |
| Description: This attribute is used to describe task details. |
Task Board

**Description:** This element is used to show the Tasks.

Update

**Description:** This element is another extra element. This element can be used to illustrate update information of a Sprint or Product Backlog during different phases.

**Attribute:**
- **ID:** Using a unique ID to specify an Update.
- **Update Type:** This attribute indicate the type of update, it can be: Product Backlog Update, or Sprint Update.
- **Action:** This attribute indicate the action of the update, it can be: "insert", "remove", "modification".
- **Description:** This attribute is used to describe Update details.

Increment

**Description:** This element is used to specify the end of a Sprint which consists of Product Backlog Items.

**Attribute:**
- "Done" Product Backlog Items: This attribute is used to show the "done" Product Backlog Items during a Sprint.

In the proposed DSL, there is a type of element named —Structure Element”. Just like prepositions of English, this type of elements does not have real meaning without using with other normal elements.

Table 14. Structure elements for building Scrum project.

<table>
<thead>
<tr>
<th>Structure Element</th>
<th>Description</th>
<th>Legends</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Time Line</td>
<td><strong>The Product Time Line has to be vertical.</strong> The basic concept comes from BPMN, In BPMN, this arrow means sequence flow, and it is used to show activity orders [47]. In the proposed DSL, the arrow has the similar meaning. We designed the Product Time Line to show the order of Sprint and Updates.</td>
<td>![Vertical Arrow]</td>
</tr>
<tr>
<td>Sprint Time Line</td>
<td><strong>The Sprint Time Line has to be horizontal.</strong> Same with the Product Time, the concept of Sprint Time Line is also reused from BPMN. Sprint Time Line shows the order of all type of meetings.</td>
<td>![Horizontal Arrow]</td>
</tr>
<tr>
<td>Connection</td>
<td>The Connection is used to connect two normal elements, or a normal element and Connection Point.</td>
<td>![Connection Line]</td>
</tr>
</tbody>
</table>
5.4.3 Scrum model

This Scrum model is used to construct the Scrum method by using the basic abstract syntax of the proposed DSL. It can help Scrum users to have a better understanding of the Scrum process instead of just displaying the Scrum method itself. Figure 10 shows the Scrum Model.

A Sprint starts from Sprint Planning Meeting, the length of the Sprint Planning Meeting usually is 4 to 8 hour [15]. In this meeting, the following questions have to be decided:

- What could be done in the following Sprint?
- How to do the work decided in the first question?

Several Product Backlog Items is selected during the planning meeting and the Product Owner should specify the details of those items. Finally, those Product Backlog Items become to Sprint Backlog Items. Moreover, keep in mind that in Scrum principles, only Product Owner has the power to operate the Product Backlog and Sprint Backlog. In some situation, other Scrum team members request to operate Product Backlog and Sprint Backlog, including add, remove, and modification. But the product Owner decides to approve that or not.

After the Sprint Planning Meeting, the Sprint is officially started. Sprint is a time box, which is usually 2 to 4 weeks. During a Sprint, the Daily Scrum meeting is held every day. Usually, a Daily Scrum Meeting continues 30 minutes. In a Daily Scrum Meeting, each of the Development Team members should answer the following questions:
What did he/she do yesterday for achieving the Sprint Goal?
What does he/she will do today for achieving the Sprint Goal?
Is there any impediment that prevents him/her or the Development Team from achieving the Sprint Goal?

Product Owner is usually absent in Daily Scrum Meetings. And during a Sprint, only the Development Team can change the Sprint Backlog. After 2 to 4 weeks, the Sprint is ended and the done Product Backlog Items is selected into Increment. A Sprint review Meeting will be held to inspect the increment and adapt the Product Backlog. A Sprint Review Meeting is always held four hours (for one-month Sprint). The Product Backlog Items that were not done should put back to Product Backlog and be reordered. After Sprint Review Meeting, a three-hour (for one-month Sprint) Sprint Retrospective meeting will be held to inspect the team itself. The goal of Sprint Retrospective is:
- To inspect the last Sprint went, regarding people, relationships, process, and tools.
- To summarize the Sprint Backlog Items went well and provide potential improvement
- To plan the improvement.

5.4.4 Design of the syntax and semantics

5.4.4.1 Relationships between elements and Abstract Syntax Model

The syntax is the grammar of a language. It decides the elements’ order and expression structure. We defined the Syntax by using Class diagram from Unified Modeling Language (UML). In this section, first, we showed the inheritance relationship among the normal elements of the proposed DSL. Moreover, we need to specify that, the UML version is before 2.5, therefore, there is no overlap between to child classes. For example, a Developer cannot be a Product Owner anymore. Figure 11 demonstrates the relationships among different roles.

![Figure 11. Relationship definition of participant](image)

As we can see in the diagram, the participant is a superclass, and it has three child classes: Developer, Product Owner, and Scrum Master. The child classes do not have any new attribute, but they represent different roles in Scrum. Several developers compose the Development Team, which is one of the three Scrum roles.

Figure 12 shows the UML diagram of Meetings.
The super class “Meeting” has four child classes; they also have no more attribute. But all of these child classes take different parts in Scrum project. For example, the Sprint Planning Meeting should only be used at the first day of a Sprint; in contrary, Sprint Review Meeting and Sprint Retrospective usually are held on the last day of a Sprint. **Figure 12** shows the relationships of Meeting.

The super class is a new concept introduced by the proposed DSL and has two child classes. Child classes do not provide new attribute, but they have a different meaning in Scrum project. Sprint Update is an operation on Sprint Time Line. It happens during a Sprint duration and is usually raised by Development Team. A Sprint update can modify two type of element: “Task” and “Sprint Backlog Item”. Product Backlog Update is usually raised by Product Owner during Sprint Review Meeting. **Figure 14** demonstrates the abstract syntax of Product Backlog.
In this diagram, we can notice that a Product Backlog consists of a number of Product Backlog Items. On the other hand, a Product Backlog only can connect to one Product Owner and one Scrum Master. Besides, several Sprints can share one Product Backlog. Moreover, similar with Product Backlog, Increment also contains a set of Product Backlog Items. The difference is the Product Backlog Items in Increment are all "done".

Figure 15 demonstrates the abstract syntax of Sprint.

In the proposed DSL, Sprint is the most complex element. There are night different elements that compose a Sprint, and Sprint also refers a Product Backlog. To be more specific, a sprint has the composition relationship with the four meetings, because without a Sprint, the meetings a meaningless and will not be held. Also, a Sprint has aggregation connection with Increment, Sprint Backlog, and Task Board. Except for Task Board, other elements are necessary for a Sprint, but they are artifacts and even a Sprint is canceled, these artifacts still exist. Furthermore, except Daily Scrum Meeting and Sprint Update, a Sprint only can have a connection with one of each element.
**Figure 16** demonstrates the abstract syntax of Task. It illustrates that a task only can be assigned to one Developer. Besides, A Task only can belong to one Task Board, but a Task Board can contain more than one Task.

![Figure 16. Relationship definition of Task.](image)

**Figure 17** demonstrates the abstract syntax of Sprint Backlog Item. It displays that a Sprint Backlog Item have a composition connection with Sprint Backlog, it means that if a Sprint Backlog is canceled, all Sprint Backlog Item in this Sprint Backlog does not exist. Moreover, for a Sprint Backlog Item, only one Participant can be assigned as a requestor. On the other hand, a Spring Backlog only can be assigned to one Development Team.

![Figure 17. Relationship definition of Sprint Backlog.](image)

5.4.4.2 Semantics definition.

Semantics is the meaning of language element and expression. We already introduced the meaning of elements in (Section 5.3.2). In this section, we are going to explain all meaningful element combinations and the meaning of those expressions.

<table>
<thead>
<tr>
<th>Combination</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Backlog &amp; Product Backlog Item</td>
<td>Product Backlog is a container of Product Backlog Items. In the Product Backlog, the importance arrow indicates the item importance order. The Product Backlog Items, which are in the same row, have the same importance. The Product Backlog Items in lower rows have lower importance.</td>
<td>![Example Diagram]</td>
</tr>
</tbody>
</table>

43
**Sprint Backlog & Sprint Backlog Item**

Sprint Backlog is a container of Sprint Backlog Items. In the Product Backlog, the importance arrow indicates the item importance order. The Sprint Backlog Items, which are in the same row, have the same importance. The Sprint Backlog Items in lower rows have lower importance.

**Sprint Backlog & Connect & Connection Point & Sprint Time Line**

A Sprint Backlog can only connect to a day of Sprint timeline. This combination indicates that the Sprint Backlog and its items are defined on that day.

**Development Team & Connection & Developer**

The Developer belongs to the Development Team. According to the Scrum principles, the Development Team members are completely equal.

**Product Backlog & Connect & Connection Point & Product Time Line**

This expression indicates that the Product Backlog becomes the current Product Backlog, all following Sprint should start basing on this Product Backlog. And all Product Backlog Updates before the next Product Backlog are based on this Product Backlog. The date of the time point can be written on the left of the Connection Point, if necessary. The Product Backlog should be used before each Sprint to specify the state of the Product Backlog.

**Sprint & Connect & Connection Point & Product Time Line**

This expression means that starting a Sprint, at the date of the Connection Point. The date of the time point can be written on the left of the Connection Point, if necessary.
<table>
<thead>
<tr>
<th>Event Description</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product Backlog Update &amp; Connect &amp; Connection Point &amp; Product Time Line</strong></td>
<td>This expression means that the Product Backlog is updated at that moment.</td>
</tr>
<tr>
<td><strong>Sprint &amp; Sprint Time Line</strong></td>
<td>This expression indicates that a Sprint has been started; the Sprint Time Line can be extended if needed. The Sprint Time Line can only be connected through a Connection Point. All Connection Point on the Sprint Time Line indicates working days of the Sprint.</td>
</tr>
<tr>
<td><strong>Meeting &amp; Connect &amp; Connection Point &amp; Sprint Time Line</strong></td>
<td>The meeting is held on the day of the Connection Point. The day number can be written above the Connection Point. Moreover, according to Scrum principle, the Daily Scrum meeting should be held every day. In order to prevent duplicating of Daily Scrum Meetings, a rule is introduced that if a Daily Scrum Meeting is same with last day. Then this Daily Scrum meeting element can be ignored on the diagram.</td>
</tr>
<tr>
<td><strong>Sprint Update &amp; Connect &amp; Connection Point &amp; Sprint Time Line</strong></td>
<td>The current Sprint is updated; the updated object can be the Sprint Backlog or the Task Board. If the Sprint Backlog is updated, this Sprint Update should connect with one or more Sprint Backlog Items. In contrary, if the Task Board is updated, one or more tasks should be attached with this Sprint Update.</td>
</tr>
<tr>
<td><strong>Increment &amp; Connect &amp; Sprint Review Meeting</strong></td>
<td>This expression means the Increment is inspected on that Sprint Review Meeting.</td>
</tr>
</tbody>
</table>
The expression indicates that the specified task is updated. The new information should be written in the Task element.

The expression indicates that the specified Sprint Backlog Item is updated. The new information should be written in the Sprint Backlog Item element.

---

5.5 **Example project described in the graphical language**

In this section, we showed an example project, in order to simplify the project, we only show on Sprint as the example. At the beginning of the example, we demonstrate an overview of the whole project by **Figure 18**.
Figure 18. Example overview.
In Figure 19, we demonstrated the structure of the Scrum team. As we can see, there are one Scrum Master and one Product Owner in the Scrum team. And there are 6 developers in the Development team.

Figure 19. Scrum team example

Figure 20 shows the details of the Product Backlog. The product Backlog and its items are specified before a Sprint. There are 9 Product Backlog Items in this Product Backlog. The items are ordered by their importance value from top to bottom. The items with the same
importance are listed on the same line. The three items which have the importance value as 100 are listed on the first line. It means they have same priority to be developed. The following two items with importance value as 80 are listed on the lower line. The other items are ordered by the same rule.

Figure 20. Product Backlog example

Figure 21 shows the specific Sprint. At the first day of the Sprint, the Sprint Backlog items are determined in Scrum planning meeting and the tasks are also set. Then the Daily Scrum Meetings are held every day and at the day 8, the developer number 3 is absent and comes back at day 13. The tasks number 10 and number 11 are updated at day 10. At the end
of this Sprint, a Sprint retrospective and a Sprint Review Meeting are held. Moreover, during the Sprint Review Meeting is used to review the Increment developed in the Sprint. After this Sprint, the Product Backlog is updated.

Figure 21. Sprint example

Figure 22 demonstrates the Product Backlog status after the Sprint. Comparing with the Product Backlog before, the first three items are developed, and the item number 7 is removed after the Sprint.
Figure 22. Updated Product Backlog

Product Backlog
Product Owner: P001
Scrum Master: S001

PH001
200 hours
View restaurant
As a User, I want to be able to see the complete Restaurants list.

PH002
180 hours
Register
As a User, I want to be able to register myself to the website with my Email and a chosen password to access to more features.

PH003
90 hours
User Profile Management
As a Registered User, I want to be able to update my profile. Or delete it.

PH004
256 hours
Create restaurant page
As a Registered User, I want to be able to make my own Restaurant page.

PH005
160 hours
Search
As a User, I want to be able to search for a particular Restaurant based on location, name or category.

PH006
100 hours
Rate restaurant
As a Registered User, I want to be able to rate a restaurant I visited.

PH007
350 hours
Comments
As a Registered User, I want to be able to leave a comment about a restaurant I've visited.

PH008
280 hours
Administrator management
As an Administrator I want to be able to search for a User, and delete it if needed.
6 THE PROPOSED LANGUAGE EVALUATION

This section described the experiment and survey’s execution process and their corresponding results and discussion.

6.1 Experiment execution

6.1.1 Validation design of the experiment– pilot experiment

For the purpose of validating the design of the experiment, we conducted a pilot experiment which helped us to identify the potential problems in the experiment and detect the flaws in the training materials and questionnaire.

The pilot experiment was run with two master students, who were chosen based on their experience of Scrum project development. They worked in a group to develop a Scrum project in a course of during their study program. Therefore, these two subjects have similar knowledge and understanding about Scrum method and project. The prepared training materials, textual version, and graphical version were assigned to these two subjects separately. And we allocated them into different rooms in order to avoid mutual influence.

After they finished the experiment, we assessed their questionnaire and discussed the training materials with them. The result from the pilot experiment showed that their previous experience affected their understanding and made mistakes or wrong choice. However, their understandings of Scrum were always mixed in their experiences and personal viewpoints. When they reading the training materials they always skip some information because they supposed they have known. As a matter of fact, both of them have inaccurate understandings about the Scrum method. The traditional describing way cannot give them a visual development process of the Scrum project. That leads to a different understanding about the Scrum method for the new learner. On the other hand, the graphical language can provide a clear project process that helped the subject to understand the materials and solve the questions.

The subjects also suggested us to redesign the layout of the graphical materials. Due to the limitation of the paper’s scale, some information was dense and unreadable. That makes them felt tired of reading the stuff. Besides, several questions focused on the projects were too complex to be answered; subjects were confused about the questions content.

According to the result of the pilot experiment, the layout of the graphical version was changed and the complexity of the questions was balanced. The result also indicated that subjects’ previous Scrum experience or knowledge will significant impact the experiment result.

6.1.2 Experiment operation

Preparation. First of all, we invited the students to participate in this experiment. We issued an inspiration letter on the web page of the course (PA1415, Software Design, BTH) to motivate the undergraduate students. We also sent the invitation letter to students who are studying in the master program by emails. The motivation for students taking part in this experiment is that they can acquire some extra knowledge on Scrum which will be beneficial for their future study and education. In order to manipulate the confounding factor (previous experience or knowledge of Scrum), the candidate subjects we selected were the students who had little knowledge about Scrum. All subject carried out this experiment voluntarily rather than by coercion.

We made a time plan and booked several rooms for subjects, and informed them when and where they can perform the experiment. They can pick one available time point and carry out the experiment in the assigned rooms. In this experiment, time is an important measurement in this experiment; we prepared subjects a digital clock that they can easily access to the time.
Execution. In this study, the experiments were executed in different ways. The experiment for the undergraduate students was run within a lecture. The undergraduate students were randomly assigned into two groups. In different groups, subjects used different training materials to learn Scrum. On the other hand, the master student performed the experiment in an individual way because they could not be gathered together in a common time. The treatments were also randomly assigned to the subjects.

The task for subjects was to learn and answer the training materials. The training material includes two parts, the introduction of Scrum method, and a specific Scrum project example. For the graphical version, Scrum method was described in the proposed language. Due to the limitation of ability, we did not provide a lecture of our proposed language for subjects. It means subjects need to learn the proposed language by using the materials as well. Learning the Scrum method and the specific project should be performed together. We did not provide any explanation of the knowledge in the training materials. In this experiment, the only way that subjects can learn the Scrum method was using the experiment instruments. Subjects could have breaks during the experiment as long as they marked the time.

Data Validation. The data of this experiment was collected from 16 undergraduate students and 18 master students. However, after assessing the training materials they finished, data from four students should be removed, due to that, the data was regarded as invalid or extreme. These data cannot represent the normal performance of the population. The four students' results were removed due to:
- Data from one student was not filled completely; the questions of the second part do not have answers.
- One student did not record the time.
- Data from two students was removed because they spent too much time to finish the experiment. The data can be regard as the outlier that should be removed. The mean of the execution time is about 30 minutes, but they spent more than one hour to complete the experiment.

After filtering the data, we left 14 undergraduate students and 16 master students for statistical analysis and interpretation of the results.

6.2 Results of experiment

6.2.1 Analysis of experiment results

The experiment was performed on 34 students, but it was found that only 30 answers could be taken into the analysis (See section 6.1.2). The results of the experiment indicate that using the graphical language improves the understandability of Scrum process. We would like to examine the result from different perspectives: effectiveness and efficiency of Scrum training.

The effectiveness of Scrum learning was measured by the numbers of the correct answers. The accurate rate of answers shows to what extent the subjects understood the Scrum process after learning the provided materials. We designed 13 questions for subjects to assess their understanding, and each question focused on one concept or usage of Scrum method. Each question has four or five choices. Subjects should validate the correct or wrong of each choice; the question can be considered as answered correctly when all the four choices are correctly judged. It could make sure that subjects cannot pass a question with wrong or ambiguous understanding. It is necessary to explain that the sixth question was removed due to the improper design since the question was not consistent with the information of the materials. Hence, we skipped this question when we collect the corresponding data. For each question, a bar plot is used as a presentation of the number of subjects that answered this question correctly in different types of materials in Figure 23.

The Figure 23 shows that subjects used the graphical version have a better understanding over the textual version users. Question NO.1 to NO.5 are used to test the understanding of the essential idea of Scrum. For question 4 and 5, the subjects in the
The graphical version performed much better compared to the textual. These two questions are raised to test the understanding of the usage of Product Backlog and Scrum team structure. It indicates that graphical language performs better on the aspect of explaining the Scrum organization structure and usage of artifacts. For question No. 7 to 13, graphical language's users exhibit an enormous advantage than the textual version. These questions are focused on the detailed information and process of the Scrum example project. Obviously, graphical language is easier to be used to improve the understanding of the Scrum projects process. Subjects are able to accurately catch the information of the project in using the graphical language materials. Graphical language indeed improved the understandability of Scrum process. Graphical notations are more visible than charts or tables to explain the concepts and demonstrate the information of the Scrum process.

![Figure 23. Numbers of subjects that correctly answered each question](image)

There is no doubt that time is an important measurement to evaluate the efficiency of learning Scrum. Figure 24 shows that for subjects working with the graphical materials, the time required on average for completing the training materials is less than the subjects working with the textual materials. The graphical material is easier to be understood in learning the Scrum process.

![Figure 24. Time spent (in minutes) for completing the training materials](image)
In addition, subjects working with the textual materials spent twice as much time as the subjects working with the graphical materials in giving a correct answer (see Figure 25). It means that subjects could quickly comprehend and apply the basic knowledge of Scrum process. The result indicates that graphical language is a more efficient way in helping subjects to learn the Scrum method and understand specific Scrum project.

![Mean of Time for generate one correct answer](image)

**Figure 25. Time spent (in minutes) for a correct answer.**

6.2.2 Hypotheses testing

The detailed descriptive statistics for the experiment are illustrated in Table 16. The result indicates that the null hypotheses regarding the learning effectiveness and efficiency could be rejected, which is supported by the statistical significance testing.

According to the experiment design, the statistics should be analyzed from the learning Scrum method and the applying Scrum process in a specific project example. We collected and summarized the data, and analyzed the result of the graphical and textual language on both Scrum knowledge (Scrum method, RQ3.1) and applying Scrum in projects (specific Project RQ3.2). For each aspect, there are three variables NRESP, TMEC and TMEC/NRESP to be used to evaluate the training effect. The differences between the graphical materials and textual materials are also presented as values in Table 16.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mean</th>
<th>Difference: Graphical-Textual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scrum method part</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NRESP</td>
<td>4.40</td>
<td>3.13</td>
</tr>
<tr>
<td>TMEC</td>
<td>11.53</td>
<td>10.40</td>
</tr>
<tr>
<td>TMEC/NRESP</td>
<td>2.67</td>
<td>3.80</td>
</tr>
<tr>
<td>Scrum project example part</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NRESP</td>
<td>5.73</td>
<td>3.33</td>
</tr>
<tr>
<td>TMEC</td>
<td>13.13</td>
<td>18.6</td>
</tr>
<tr>
<td>TMEC/NRESP</td>
<td>2.27</td>
<td>6.60</td>
</tr>
</tbody>
</table>

The columns in Table 17 contain the name of the statistical significance test used, the understandability improvement, and the hypotheses testing. The usage of the statistical significance test for each variable is determined by the result of the normality. Therefore, we used the Shapiro-Wilk² test to verify the normality of each variable, and the borderline is chosen to be 0.01(If the result value is more than 0.83, the dataset should be considered as

² The data were calculated by the Shapiro-Wilk Normality Test Calculator.
http://sdttami.altervista.org/shapirotest/ShapiroTest.html
normality). For each variable, if both the datasets of textual and graphical are normality, t-test should be used to test the significance level; otherwise, we used the Mann-Whitney test to process the data. The improvement/ deterioration are presented as value and percentage in Table 18. For the calculation of the percentages, the value of the textual language is taken as the basis.

After conducting the statistical significance tests for each variable, the results show that the observed language, the textual and graphical language has a significant difference in understanding the specific project except learning the Scrum basic knowledge.

For learning the essential idea of Scrum method, the null hypotheses for variables NRESP and TMEC/NRESP can be rejected. As can be seen in Table 18, compared to the textual language, the learning efficiency (TMEC/NRESP) and effectiveness (NRESP) have been increased 41% and 30% in using graphical language to train subjects. However, in learning basic knowledge of Scrum method, graphical language had a significant deterioration than the textual version. It means that subjects working with the graphical materials spent more time in learning the Scrum method. Actually, there is a confounding factor, the graphical language learning cost, which may result in the deterioration of the graphical language result. Due to the resource limitation, we did not provide a lecture to introduce the graphical language used in the materials. Therefore, subjects need to learn the graphical language first, and then learn the Scrum based on these notations. That is why subjects working with the graphical materials need spend more time in learning Scrum method. However, once they had learned the graphical language, the extra learning cost should be eliminated. Hence, in the second part of the experiment, subjects working with the graphical materials showed a significant improvement in TMEC, they only spend 70% of the time that the subjects using the textual materials required for understanding the Scrum project. Besides, the total time spent for completing the whole experiment, subjects worked with the graphical materials is also less than using the textual materials. It means that, in general, graphical language indeed improves the efficiency and effectiveness in learning the basic knowledge of Scrum method.

For understanding the Scrum method in specific Scrum projects, graphical language also has a significant improvement in learning efficiency and effectiveness compared to textual language. Especially for variable NRESP and TMEC/NRESP, graphical language group has more than 60% increments over the textual language. The null hypotheses for all variables can be rejected. It means that graphical language can affect the understandability of the specific Scrum project.

Table 17. Result of the statistical significance testing

<table>
<thead>
<tr>
<th>Variable</th>
<th>Shapiro–Wilk test (Textual/Graphical)</th>
<th>Significance level (p≤0.05)</th>
<th>H0-accepted H0-efficiency &amp; H0-efficiency</th>
<th>H1-Polarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scrum method part</td>
<td>NRESP 0.85/0.76</td>
<td>0.00152 (Mann-Whitney)</td>
<td>No</td>
<td>Positive</td>
</tr>
<tr>
<td></td>
<td>TMEC 0.73/0.95</td>
<td>0.1141 (Mann-Whitney)</td>
<td>Yes</td>
<td>Negative</td>
</tr>
<tr>
<td></td>
<td>TMEC/NRESP 0.88/0.74</td>
<td>0.03662 (Mann-Whitney)</td>
<td>No</td>
<td>Positive</td>
</tr>
<tr>
<td>Scrum project example part</td>
<td>NRESP 0.92/0.85</td>
<td>&lt;0.0001 (t-test3)</td>
<td>No</td>
<td>Positive</td>
</tr>
<tr>
<td></td>
<td>TMEC 0.95/0.92</td>
<td>0.000795 (t-test)</td>
<td>No</td>
<td>Positive</td>
</tr>
<tr>
<td></td>
<td>TMEC/NRESP 0.77/0.75</td>
<td>0 (Mann-Whitney)</td>
<td>No</td>
<td>Positive</td>
</tr>
</tbody>
</table>

3 The data were calculated by the T- Test Calculator.
Table 18. Results of the improvement/deterioration

<table>
<thead>
<tr>
<th>Variable</th>
<th>Improvement (value)</th>
<th>Improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scrum method part</td>
<td>NRESP 1.27</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>TMEC 1.13</td>
<td>-11 (Deterioration)</td>
</tr>
<tr>
<td></td>
<td>TMEC/NRESP -1.13</td>
<td>30</td>
</tr>
<tr>
<td>Scrum project example part</td>
<td>NRESP 2.40</td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>TMEC -5.47</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>TMEC/NRESP -4.33</td>
<td>66</td>
</tr>
</tbody>
</table>

According to the combination of Table 17 and Table 18, we can see clearly that graphical language has improved the understandability of Scrum process and project, and the null hypotheses of the learning efficiency and effectiveness should be rejected. The result can be demonstrated as follow:

\[ H1: \text{Efficiency (Graphical)} > \text{Efficiency (Textual)} \]
\[ H1: \text{Effectiveness (Graphical)} > \text{Effectiveness (Textual)} \]

As described in section 4.4, if the graphical materials have a significant difference with the textual materials, moreover, it can improve the learning efficiency and effectiveness, we should lead to the conclusion that the proposed language is useful for helping subjects to learn Scrum method and understand the specific Scrum projects. The proposed language can improve the understandability of the Scrum process and projects.

6.2.3 Validity threats

In this experiment, there are four types of validity threats that should be analyzed: conclusion validity, construct validity, internal validity, and external validity. The conclusion validity is concerned with the ability to draw the correct conclusion and the relations between the treatments and the outcomes of the experiment. Threats to construct validity are concerned with the result of the experiment to the theory behind the experiment. Threats to internal validity are the issues that can affect the independent variable with respect to causality, without the researcher’s knowledge [40]. External validity concerns generalization of the result of the experiment to other environments, such as the industry environment.

6.2.3.1 Conclusion validity

Conclusion validity focuses on the statistical analysis of the result and the composition of subjects.

In this experiment, we applied the reasonable statistical techniques to process the data and generate the result. The selection of the statistical techniques for each variable was reliable. In the first place, we tested the property of the dataset with the well-known method. After that, we applied the relative techniques to calculating the data based on the test result. The data processing is exacting and the methods are robust to valid the assumptions.

The major threat regarding the conclusion validity is the low number of samples, which may reduce the ability to generalize the correct conclusion. Only fifteen students participated in each experiment group, which reduced the power of the statistical significance tests used. However, we seriously selected the subjects and filtered the data, which could ensure the quality and accuracy of the data and result.

6.2.3.2 Construct validity

The main construct validity threat is that there was no lecture given before the experiment for subjects to introduce the proposed language. That resulted in a confounding result that some subjects in graphical language group spent more time in learning the basic knowledge of Scrum method. However, the total time the subjects working with the graphical materials spent in finishing the experiment was less than the subjects using the textual materials.

Due to the resource limitation, we could not provide a lecture to introduce our proposed language; that indeed caused some obstacles for subjects to understand the proposed language and the materials. But the cost of learning the proposed language is also an important factor that should be considered to evaluate the usefulness of the proposed
language. The result indicates that even subjects need to spend extra time in learning the graphical notations; they also performed better than the subjects working with the textual materials in general.

6.2.3.3 Internal validity

The internal validity includes three major threats. The first threat is that the design of the instruments may not be appropriate to test the understanding of subjects. The complexity of the training materials and quiz are not the same as the complexity of real-world design documents. For instance, some questions focused on the project’s details were too difficult to give a complete answer. The instruction of the training material was not clear; some subjects did not know how to correctly use the training materials, that lead to them made mistakes on filling the questions’ answers and we cannot precisely examine the performance of the subjects.

The second threat to the internal validity is that some undergraduate students might be difficult to understand the content of the training materials due to lack of the development experience. Some of the subjects also lack sufficient software development knowledge. Hence, it is difficult for them to figure out the precise meaning of the technical terms in a specific context. This problem seriously affected their learning and understanding.

The last threat of the internal validity is the lack of diversity in the selected subject population. We just selected the volunteers who have less knowledge about Scrum; because the result of the pilot experiment indicates that the Scrum experience of the subjects was a confounding factor that would affect the experiment result. However, it also eliminated the diversity of knowledge and experience background of the subjects. The subjects with less Scrum experience may just represent the new learners; the sample of the subject might not be able to cover the whole population of Scrum users.

6.2.3.4 External validity

The external validity is the selection of the subjects. This experiment was conducted in an educational environment, therefore, subjects are all students, and most of them are new learners of Scrum and lack of practical development experience. That is the main difference between the subjects and the industrial developers who are familiar with the software development process.

However, the purpose of this experiment is to evaluate how the proposed language improves the understandability of the Scrum process. The development experience of the subjects is a confounding factor that should be strictly controlled. In addition, the influence of the experience background on this experiment is indistinct.

6.3 Survey execution and result analysis

We conducted an industrial survey by questionnaire with several questions to collect the Scrum users’ opinions toward the proposed language. We used the qualitative method to handle the data and evaluate the proposed language.

Nevertheless, in this study, due to the limitation of the time and contact resource, the survey method was not performed sufficiently to evaluate the proposed language in the industry. The author had no adequate industrial contact resource to invite enough Scrum users to evaluate the proposed language. Besides, most of the participants did not study and assess the details of the proposed language carefully because of the time limitation. Indeed, the result of the survey may not adequately represent the attitude of full industry Scrum user population. But, in general, it still helped us to identify the benefits and limitations of the proposed language according to industrial Scrum users’ opinions. This survey also investigated the potential use of the proposed language in the industry.

Survey execution: This survey was conducted during 4th July to 31st July. We got the Scrum users’ contact information from our friends and published blogs.

- Demographics

We contacted 25 Scrum users who fulfill our sampling plan (section 4.5), and finally, we received 21 answers after four weeks. Five participants are working in Swedish companies and the other participants are working in China.
Questionnaire Validation

To exclude the questionnaires which cannot provide valuable information, we made four criteria for filter the valueless questionnaires. Table 19 shows the details of the four criteria and related reasons.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Reason</th>
<th>Validated Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answered the first question.</td>
<td>The first question is used to get participants’ working experience and classify questionnaires. It is an important information in this survey.</td>
<td>21</td>
</tr>
<tr>
<td>Answered more than five questions.</td>
<td>There are 10 questions in our questionnaire. The participants who did answer more than half of all questions are regarded as ―unwilling to answer this survey‖. Therefore, the answers they provided will be excluded.</td>
<td>21</td>
</tr>
<tr>
<td>Provided opinions towards graphical language.</td>
<td>We want to collect the industrial opinions about graphical DSL to describe Scrum method and specific Scrum projects. Questionnaires‘ answer without opinions towards graphical cannot provide us valid information.</td>
<td>19</td>
</tr>
<tr>
<td>Compared textual and graphical languages.</td>
<td>The purpose of the survey is to validate if the proposed graphical DSL can help industrial Scrum users to understand specific Scrum projects. Therefore, questionnaire’s answers without comparing textual and graphical language are not useful.</td>
<td>17</td>
</tr>
</tbody>
</table>

Result Analysis: We mainly used the qualitative content method to analyze the data. The purpose of this survey is to investigate if the proposed DSL can possibly be accepted by the industrial developers and their opinions toward the proposed language. By conducting the qualitative analysis, we summarized the advantages and disadvantages of the proposed language from the Scrum user’s perspective.

Question 1: How long time have you been using Scrum?

After validating the data, we finally selected 17 valid questionnaires to analyze. In these Scrum users, the average Scrum experience is about 2 years. There are three Scrum users have 3 years of experience. They are all working in Sweden. Figure 26 illustrates the Scrum experience distribution.

Figure 26. Survey participants experience distribution.

Question 2: How did you learn Scrum?

In the second question, we found out that all of the participants tried to learn Scrum via the Internet. Ten participants learned Scrum through book and literature. Only 8 participants
learn Scrum from school. All Scrum users had experience on learning Scrum in projects. It means that modeling specific projects is a potential solution for improving Scrum learning. **Figure 27** illustrates the result of question 2.

**Figure 27.** Answer distribution of question 2.

**Question 3: Do you think a graphical language can help you to understand Scrum process or project?**

For the question 3, **Figure 28** demonstrates that 14 of 17 people are holding positive opinion towards graphical language; moreover, 5 people strongly agree that a graphical language can help them to understand Scrum. Three people do not know whether or not a graphical is helpful in learning Scrum. In the chart, we can notice that nobody chooses negative opinions. It supports our idea to some extent that a graphical can help developers to learn Scrum.

**Figure 28.** Answer distribution of question 3.

**Question 4: Do you think a graphical language can help the new learner to understand Scrum process in a better way?**

The result of Question 4 is pretty much as same as question 3. The **Figure 29** shows that compared to the answers of question 3, more people think a graphical language is more useful for Scrum beginners. Only two participants still hold a neutral opinion towards the capacity of graphical language. Ten Scrum users think that a graphical language can provide
some improvement for Scrum learners. And the five people still strongly believe that a graphical language can help them to understand Scrum method.

![Figure 29. Answer distribution of question 4.](image)

**Question 5: Which way (textual/graphical) you would like to use in learning and teaching Scrum process, and why?**

The result of question 5 is basically in line with above questions results. However, noticeably, although at least 15 people think that graphical language is helpful for learning Scrum (data from question 3, 4, and 5), there still almost 2/3 people think textual language is necessary for learning Scrum. In Figure 30, 2 people prefer to learn via textual way, and 9 people choose both. They think the combining textual and graphical ways is probably better than one single method. It indicates that there are still some drawbacks that the provided graphical language does not overcome.

![Figure 30. Answer distribution of question 5.](image)

**Question 6: In your opinion, what are the advantages and disadvantages of teaching the Scrum process in using the proposed graphical language?**

This question is an open-ended question. We summarized the main ideas from the participants based on the question 5.

61
Advantage

All 17 participants are holding a viewpoint that graphical language extremely increased the learning efficiency. Moreover, 13 participants think the proposed graphical language can provide an easier manner to get an overview of Scrum concepts and specific projects. Nine people mentioned that the proposed language makes Scrum concepts more clear and understandable. A participant wrote a positive comment about teaching: "if the content to be taught is really boring, then the graphical way could be a better option, for keeping the students away from sleeping in the class".

Disadvantage

For the disadvantage, all of the participants also have the same opinion: it is still too general to develop a specific project. They think many conceptual details are missed. Besides, the extendibility is not as good as textual language; they have to spend more efforts if they want to modify some parts of Scrum for their own projects. Moreover, one participant also thinks that there is too much redundancy information, for experienced Scrum users.

Question 7: In your opinion, what are the advantages and disadvantages of teaching the Scrum process in using the proposed textual materials?

This question is also an open-ended question.

Advantage

The textual language is more rigorous to define Scrum. Besides, the textual language provides more details of Scrum concepts.

Disadvantage

The textual language is hard to get an overview of the specific project at the beginning; it will be harder for Scrum beginner. Furthermore, textual language is using natural language; it means there are language barriers. For example, if the textual language is written in English, it is difficult to ask someone who does not know English to read the textual language. Comparing English, learning a graphical language is much easier.

Question 8: What are the problems or obstacles when you learn the Scrum process?

In this question, there are two types of answers. For Scrum users who have more than 2.5 years of experience, they think managing and distribute tasks is the main obstacles for them. For Scrum users who used Scrum for 1 – 2 years, the main problem is still a lack of practical experience. In some cases, they do not know how to implement the Scrum concepts to actual projects.

Question 9: Do you think a graphical language can help you to solve or mitigate the above problems, and why?

For the Scrum users who have more Scrum experience, they think the proposed language can mitigate the problems to some extent, but the effect is still limited because the proposed graphical language is mainly focused on managing Scrum project. But the existing problems are more related to technical solutions.

For the Scrum users who only have 1 – 2 years of Scrum experience, they think the proposed graphical language definitely can help them in understanding Scrum and applying Scrum concepts to specific Scrum project.

Question 10: Will you use this graphical language in the future work? Why?

All of the participants are holding a positive answer on this question. The main reason is due to the high efficiency of the proposed language. For Scrum users who have 3 years of experience, they think it is good to have such a language, especially when they join to a new project or at the beginning of learning Scrum method. On the other hand, they also pointed out the learning and maintaining cost of this language. They claimed that they will balance the cost and efficiency of the proposed language carefully if it is introduced to their project in future.
7 Conclusion and Future Work

In this study, we introduced a graphical DSL for describing Scrum method and specific Scrum projects. And we conducted an experiment and a survey to evaluate the proposed graphical DSL. The results of the evaluation indicate that using the proposed graphical DSL improves the learnability of Scrum method and understandability of specific Scrum projects.

As the result, the outcomes and conclusions can be summarized in three branches.

7.1 Essential elements of Scrum method

**RQ1**: What are the basic concepts, elements and their properties that are present in the description of Scrum process?

In order to extract the Scrum concepts and building elements, we conducted a literature study at the beginning of this research. In the literature study, we mainly focused on collecting knowledge about Scrum project modeling and Scrum teaching. The study result shows that there are three types of concept in Scrum: roles, events, and artifacts. Specifically, there are three roles in Scrum: Scrum Master, Product Owner and Development Team. Although the developer is not the main role in Scrum, it is still an important role in Scrum role concept, because it makes up the Development Team. Therefore, we added Developer as a role in the proposed DSL.

Scrum events are: Sprint, Sprint Planning meeting, Daily Scrum meeting, Sprint Review meeting and Sprint Retrospective meeting. A Sprint consists of different Scrum meetings and other development events. The Scrum planning meeting is held at the beginning of a Sprint and continues about 6 to 8 hours for a 4-weeks Sprint. Daily Scrum meeting should be held at the beginning at every Sprint day, the time length is around 30 minutes. At the end of a Sprint, two meetings should be held, which are Sprint Review meeting and Sprint Retrospective meeting. The Sprint Review meeting is used to review and demonstrate the requirements developed in last Sprint. And the Sprint retrospective meeting is used to inspect the Development Team and to solve technical and organizational problems. The last type of concept is Scrum artifacts, Product Backlog, Sprint Backlog and Increment. The two types of Backlog are sets of project requirements that are going to be done. Only Product Owner has the power to manage the Backlogs. The Increment is a set of “done” requirements. During Sprint Review meeting, the Increment should be demonstrated.

7.2 Definition of the proposed graphical DSL

**RQ2**: What is the definition of the proposed graphical language?

After extracting Scrum elements, we found that it is insufficient to describe Scrum project only with above elements. Therefore, in order to facilitate the Scrum teaching and Scrum modeling, we introduce several extra elements: Product Backlog item, Sprint Backlog item, and Task (Section 5.3.2). Product Backlog item and Sprint Backlog item are the units for specifying the requirements. The Task is the distributed Sprint Backlog item which is assigned to a specific developer. Moreover, we also introduced two types of the timeline to build up the structure of the proposed graphical DSL.

In order to formally design and define the proposed language, we followed a reported DSL design approach and design principle (Section 5.1.2 and Section 5.4). Since Scrum is a development method instead of an executable application, it is reasonable to develop a non-executable DSL for describing the Scrum process. Therefore, we developed the proposed DSL in three steps: the decision, analysis, and design phases. The development and deployment phases have been removed from the DSL development process in this study. The decision phase is conducted to set the scope of the DSL (Section 5.1). Then in the analysis phase, all domain knowledge is accumulated and domain problems are specified (Section 5.2). And in the design phase, the formal definitions of the DSL are explained (Section 5.3).
The proposed graphical DSL contains three parts. The first part is “Terminology”, all Scrum elements and language legends are shown in this part (Section 5.4.2). The second part is “Scrum Model”, this part is used to simply demonstrate the Scrum basic concepts and principles (Section 5.4.3). The last part is “Scrum project” (Section 5.5); the language users can construct specific Scrum projects using the proposed language.

In order to design and define a DSL formally, we reused several concepts from BPMN. And we used UML to define the language syntax and the relationships among the elements (Section 5.4.4). This syntax reflexes the Scrum principles and relations among Scrum roles, events, and artifacts. According to this syntax definition, the language users can build up a Scrum project described in the proposed graphical DSL.

For the semantic specification, we used a table to list all possible element combinations and element variants, and then we explained the meaning for each specific combination and variants (Section 5.4.4).

7.3 Evaluation results of the proposed graphical language

In this evaluation phase, we conducted two research approaches to evaluate the proposed graphical in both educational and industrial field.

7.3.1 The experiment result

**RQ3: How the proposed Scrum modeling language contributes to Scrum training in an academic environment?**

In order to evaluate the usefulness of the proposed language, we conducted a controlled experiment in the educational environment to examine whether the proposed language can improve the efficiency and effectiveness of in learning Scrum method.

The experiment results indicate that the proposed language can help the new Scrum learners more accurately gain the knowledge of the Scrum method (RQ3.1), and capture the important and detailed information about the specific Scrum project (RQ3.2). By applying the statistical significance tests for each dataset collected from the subjects, we got a positive result which indicates that the null hypotheses should be rejected and the proposed language significantly improve the understandability of the Scrum method and its relative projects. Compared to the textual language, the proposed language improved the subject’s learning effectiveness and efficiency in understanding the Scrum method by 41% (NRESP) and 30% (TMEC/NRESP) (see section 6.2.6). However, there was only one negative result show that subjects need to spend more time on understanding the Scrum method in the proposed language (RQ3.1). But when they were familiar with the proposed notations, the learning effectiveness can be improved prominently. The Scrum project learning effectiveness and efficiency of the subjects have been improved by 72% (NRESP), 30% (TMEC) and 66% (TMEC/NRESP) after understanding the notations (see section 6.2.6) (RQ3.2). Thus it can be seen that the extra time spent on understanding the Scrum method was caused by the learning cost of the new proposed language (RQ3.1).

7.3.2 The survey result

**RQ4: How the proposed Scrum modeling language helps the Scrum users to understand and apply the Scrum method in the industry?**

In order to investigate the potential use of the proposed language in the industrial area, we did a survey with experienced developers who are working in Swedish and Chinese companies. In fact, due to the limitation of the resource, the survey method was conducted insufficiently in this study. However, it still helped us to identify the advantages and disadvantages of the proposed language from the industrial Scrum user’s perspective.

We received 21 questionnaires and, then, there are 17 valid questionnaires. All participants think that the proposed graphical DSL is useful for improving the understandability of the Scrum method. Moreover, they think the proposed DSL provides an easier way to get the overview of specific Scrum project and makes the project data more
readable and understandable. Of course, these advantages can significantly improve the learning efficiency of Scrum concepts and specific Scrum projects.

Also, there are several disadvantages were raised by the participants. First, the extendibility is not as good as the textual language. For example, if a Scrum team wants to modify traditional Scrum and add a new type of event. In textual language, it is very simple to add a paragraph and describe this event. But in the graphical language, the new elements definition and legends need to be designed. In worse cases, the structure of this language may need to be refactored to fit the new element. Second, some concept definitions are missed. As formally defining the Scrum concepts without textual description is not easy, therefore, to define Scrum concepts with the mix of the proposed DSL and textual language can help the language users to better understand Scrum concepts.

Noteworthy, the developers who have less working experience show more interest in the proposed language. It means that, so far, the proposed DSL is more suitable for Scrum teaching.

7.4 Future work

In this study, we developed and evaluated a graphical DSL for modeling Scrum process. However, due to the limitation of the time and resource, this thesis project just provided an exploratory research rather than a longitudinal study. It is just an initial work of the proposed language evaluation and the research findings validation. More intensive and concrete work should be conducted in the follow-up research.

- Validation of the proposed language
  The definition of the proposed language is created by the authors based on the learning and analysis of the literature study. There is a need to validate the accuracy of the language’s definition to make sure it is reasonable and correct for modeling Scrum projects. Therefore, it would be necessary to introduce this language to Scrum experts and DSL experts to revise this language in order to improve its capacity to precisely describe diversity Scrum projects.

- Validation of research findings in industry environment
  In this research, the experiment was conducted in the educational environment with students who have no Scrum experience. There is no doubt that students as subjects have differences from the industrial developers, such like development experience. Validation of the research findings in an industry setting experiment is needed to check the validity of the findings with the experienced developers. It would be interesting to conduct an experiment with a real Scrum team in the industry environment to validate the usefulness of the proposed language.

  Due to the resource limitation, the survey was not performed sufficiently in this study. The execution time and the number of participants were limited; hence, the result may not adequately represent the full industrial users. It would be necessary to conduct a further investigation of the proposed language in the industrial environment.

- Introduce the proposed language for managing Scrum projects
  In this research, we introduced a new Scrum modeling DSL for helping Scrum users to understand Scrum method and apply it. However, this language can be not only used for Scrum training, but also managing the projects.

  From the result of the experiment, it indicates that this language significantly improved the subjects’ understanding of the project. It is easy for subjects to find the project’s detailed information in the proposed language. Therefore, it would be interesting to investigate that apply a Scrum modeling DSL to the projects to help the developers manage the development process.
REFERENCES


APPENDIX A. TRAINING MATERIALS

Experiment (Textual Version)

Instructions
The aim of this experiment is to evaluate how good the training material is to help the participants to understand the Scrum development processes.
This experiment has two parts:
   I. Learning Scrum
   II. Testing the level of understanding Scrum

In the first part you will get a short description of the basics of Scrum – which you are supposed to read – to get the knowledge and understanding of essential ideas of Scrum. In the second part you will be asked some questions testing how much did you manage to learn after studying the provided training material.

It is very important that you fill in the time at which you start and then when you finish reading each part of the training materials, and answering the questions, so please write down the time as well.

At last but not least, there is a short survey for you to evaluate this experiment at the end of this document. Before going further please remember to fill in personal data on the first page.

Scrum Training Material
Start time:

<table>
<thead>
<tr>
<th>Hours</th>
<th>Minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This training material illustrates the essential elements of the Scrum framework and how to apply the Scrum method.

1. **TYPICAL SCRUM PROCESS**
   This section will give you a basic idea of how the Scrum method works. The following diagram shows the complete Scrum process.
Scrum employs an iterative, incremental approach to developing projects. Each Scrum project has a set of Sprints. A sprint is a time-boxed iteration of one to four weeks, during which the Scrum team will create and deliver the products.

1.1 Sprint Activities:

a) Sprint Planning meeting is the first activity of a Sprint, it is held to plan what will be implemented into a Sprint (a fixed period of time used to deliver parts of the final product). The requirements from the customers are formulated as user stories to be delivered. The Product Owner selects the user stories with high importance from the Product Backlog to make up the Sprint Backlog.

b) The Development Team breaks down (expands) these user stories into tasks. The Development Team then takes 30 days or so to deliver the selected user stories. The Development Team must hold a Daily Scrum meeting (normally 15 minutes) each day to collaborate with each other. Scrum Master ensures The Development Team holds the Daily Scrum meeting.

c) At the end of the Sprint, the Development Team demonstrates the completed parts of the product to the customer in a Sprint Review meeting.

d) The last activity is the Scrum Retrospective meeting, where the Development Team reviews the Sprint and looks for ways of improving (lessons learned).

1.2 Scrum Roles

There are three roles in a Scrum project.

Development Team

The Development Team consists of a group of professionals (3 to 9 people, full-time) who are responsible for creating the project deliverables according to the user stories specified by the Product Owner.

Product Owner

The Product Owner is one person. The Product Owner should order the user stories in the Product Backlog. The Development Team should follow the Product Owner's decisions in the product backlog.

Scrum Master

The Scrum Master is a facilitator who serves the Product Owner in managing the Scrum meetings with the Development Team, understanding how to arrange the product backlog to maximize business value, and helping the Development Team to understand the Product Owner's decisions in the Product Backlog. The Scrum Master serves the Development Team with coaching Scrum theory and removing the obstacles in the project progress.
1.3 Scrum Artifacts

Product Backlog: An ordered list of user stories that are needed in the final product. The Product Backlog user stories are ordered based on their business value (importance) by the Product Owner. Each user story has ID, estimate hours, importance, and tasks description. The Product Backlog is dynamically changing and improving. When the Product Owner detects the new user stories, he/she should add it immediately. The Development Team does not wait until the Product Backlog is complete to start delivering the user stories; the first Sprint can be started as soon as the Product Backlog has a sufficient number of user stories defined.

Sprint Backlog: Selected user stories from the Product Backlog to be delivered through a Sprint.

Finish time:

<table>
<thead>
<tr>
<th>Hours</th>
<th>Minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Questionnaire**

Start time:

<table>
<thead>
<tr>
<th>Hours</th>
<th>Minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:** Please answer the following questions based on the training materials you just read.

Please fill → T ” in the square bracket behind the choice when you think it is Correct, such as [ T ]

Please fill → F ” in the square bracket behind the choice when you think it is Wrong, such as [ F ]

If you not sure the choice is correct or wrong, please just leave it as [  ].

**Example:**

Q: Who has the authority to cancel a task?

[ T ] Product Owner   (This choice is Correct, so fill T)
[ F ] Scrum Master    (This choice is Wrong, so fill F)
[  ] Customer          (I am not sure about this choice, so I just leave it as blank)
[ T ] Development Team (This choice is Correct, so fill T)

**Questions for Scrum**

Q1. Who helps the Product Owner decide on right actions?

[  ] Another Product Owner
[  ] Scrum Master
[  ] Development Team
[  ] There’s no specific role for helping the Product Owner

Q2. We are going to start the first Sprint. What’s the first activity?

[  ] Sprint Initiation
[  ] Sprint Startup
[  ] Sprint Planning
[  ] Daily Scrum
Q3. Team Members decided to cancel Daily Scrum meetings for the rest of one Sprint, to save time and get things done faster. What do you think of this decision?
[ ] Acceptable, because delivery of the products is the first priority
[ ] Not right, but acceptable since they've reached this decision and it's their own responsibility to manage their own efforts
[ ] Not acceptable, because the Daily Scrum is required in Scrum

Q4. How should the Product Backlog be ordered based on?
[ ] the effort of the user stories
[ ] the float of the user stories
[ ] the importance of the user stories
[ ] the relationship among user stories

Q5. What are the main roles in Scrum Team?
[ ] Scrum Master
[ ] Product Owner
[ ] Development Team
[ ] Developers

Finish time:

<table>
<thead>
<tr>
<th>Hours</th>
<th>Minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. SCRUM PROJECT EXAMPLE

Start time:

<table>
<thead>
<tr>
<th>Hours</th>
<th>Minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This section will illustrate you a real project used the Scrum method. This project has 4 sprints, due to the time limitation, we only illustrate two Sprints description as an example to help you understand the Scrum process.

This project developed and delivered a new website named “Karlskrona Restaurant Network” (KRN). This website is aiming to collect and exhibit the information of the restaurants in Karlskrona.

2.1 Information on the Scrum Team

This Scrum team has 6 team members, one Product Owner, one Scrum Master and four developers. Their roles are specified as follows:
- Product Owner: Hamid
- Scrum Master: Helen
- Development Team members: Alex, David, Philip and Jeffrey.

2.2 Product Backlog

This is the initial Product Backlog created by the Product Owner at the beginning of the project.
2.3 Sprint Backlog (Sprint one)

After the Sprint Planning meeting, the Development Team selected two user stories with highest importance value, K1 and K2, to be delivered in the first Sprint. These user stories were specified in the Sprint Backlog. The Sprint one cost 4 weeks (20 days).

<table>
<thead>
<tr>
<th>Importance</th>
<th>ID</th>
<th>User story</th>
<th>Description</th>
<th>Initial Size Estimate (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>K1</td>
<td>View restaurant</td>
<td>As a User, I want to be able to see the complete Restaurants list.</td>
<td>75</td>
</tr>
<tr>
<td>80</td>
<td>K2</td>
<td>Register</td>
<td>As a User, I want to be able to register myself to the website with my Email and a chosen password to access to more features</td>
<td>20</td>
</tr>
<tr>
<td>70</td>
<td>K5</td>
<td>Manage profile</td>
<td>As a Registered User, I want to be able to update my profile. Or delete it.</td>
<td>20</td>
</tr>
<tr>
<td>66</td>
<td>K3</td>
<td>Create restaurant page</td>
<td>As a Registered User, I want to be able to make my own Restaurant page.</td>
<td>20</td>
</tr>
<tr>
<td>50</td>
<td>K6</td>
<td>Search</td>
<td>As a User, I want to be able to search for a particular Restaurant, based on location, name or category.</td>
<td>35</td>
</tr>
<tr>
<td>45</td>
<td>K4</td>
<td>Rate restaurant</td>
<td>As a Registered User, I want to be able to rate a restaurant I visited.</td>
<td>45</td>
</tr>
<tr>
<td>20</td>
<td>K7</td>
<td>Comments</td>
<td>As a Registered User, I want to be able to leave a comment about a restaurant I've visited.</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td>K8</td>
<td>Share on social media</td>
<td>As a Registered User, I want to be able to share a Restaurant's information with my Facebook Friends.</td>
<td>30</td>
</tr>
<tr>
<td>10</td>
<td>K9</td>
<td>Administrator management</td>
<td>As an Administrator I want to be able to search for a User, and delete it if needed.</td>
<td>30</td>
</tr>
</tbody>
</table>

Table 1. Product Backlog
During the Sprint 1, the Product Owner decided to add two new user story points into the Sprint 1. He added the task T2.5 at the 5th day and added the T2.6 at the 11th day during the Sprint 1 and updated the Sprint backlog immediately. The changes are specified as follows:

### Table2. Sprint one Backlog

<table>
<thead>
<tr>
<th>ID</th>
<th>User stories</th>
<th>Description</th>
<th>Sprint Tasks</th>
<th>Assigned developer</th>
<th>Initial Size (story points)</th>
<th>Actual Effort at Sprints (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Task ID: Tasks (story points)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K1</td>
<td>View restaurant</td>
<td>As a User, I want to be able to see the complete Restaurants list.</td>
<td>T1.1 Website architecture design</td>
<td>Alex</td>
<td>75</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T1.2 Website layout design and implementation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T1.3 Database tables design</td>
<td>David</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T1.4 Query to database asking for restaurants list</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T1.5 Display the restaurant list</td>
<td>Philip</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T1.6 Display the restaurant's webpages</td>
<td>Jeffrey</td>
<td></td>
<td></td>
</tr>
<tr>
<td>K2</td>
<td>Register</td>
<td>As a User, I want to be able to register myself to the website with my Email and a chosen password to access to more features</td>
<td>T2.1 Make registration forms</td>
<td>Alex</td>
<td>20</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T2.2 Submit with confirmation emails</td>
<td>Philip</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T2.3 Insert user's information into database</td>
<td>David</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T2.4 Display registration success page to users</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.4 Changes during the Sprint one

During the Sprint 1, the Product Owner decided to add two new user story points into the Sprint 1. He added the task T2.5 at the 5th day and added the T2.6 at the 11th day during the Sprint 1 and updated the Sprint backlog immediately. The changes are specified as follows:
Table 3. Sprint 1 Changes

<table>
<thead>
<tr>
<th>ID</th>
<th>User stories</th>
<th>Description</th>
<th>Sprint Tasks</th>
<th>Assigned developer</th>
<th>Initial Size Estimate (hours)</th>
<th>Actual Effort at Sprints (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Task ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Tasks (story points)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K2</td>
<td>Register</td>
<td>As a User, I want to be able to register myself to the website with my Email and a chosen password to access to more features</td>
<td>T2.5 Forget password</td>
<td>Jeffrey</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T2.6 Change password</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.5 Scrum Review and Updated Product Backlog

During the Scrum Review meeting, the Product Owner changed some user stories’ importance and added one user story K10 to the Product Backlog according to the new requirements from the customer. The updated Product Backlog is shown as follow:

Table 4. Updated Product Backlog

<table>
<thead>
<tr>
<th>Importance</th>
<th>ID</th>
<th>User stories</th>
<th>Description</th>
<th>Initial Size Estimate (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>K1</td>
<td>View restaurant</td>
<td>As a User, I want to be able to see the complete Restaurants list.</td>
<td>Done</td>
</tr>
<tr>
<td>80</td>
<td>K2</td>
<td>Register</td>
<td>As a User, I want to be able to register myself to the website with my Email and a chosen password to access to more features</td>
<td>Done</td>
</tr>
<tr>
<td>75</td>
<td>K6</td>
<td>Search</td>
<td>As a User, I want to be able to search for a particular Restaurant, based on location, name or category.</td>
<td>35</td>
</tr>
<tr>
<td>70</td>
<td>K3</td>
<td>Create restaurant page</td>
<td>As a Registered User, I want to be able to make my own Restaurant page.</td>
<td>20</td>
</tr>
<tr>
<td>65</td>
<td>K7</td>
<td>Comments</td>
<td>As a Registered User, I want to be able to leave a comment about a restaurant I've visited.</td>
<td>20</td>
</tr>
<tr>
<td>63</td>
<td>K4</td>
<td>Rate restaurant</td>
<td>As a Registered User, I want to be able to rate a restaurant I visited.</td>
<td>20</td>
</tr>
<tr>
<td>50</td>
<td>K5</td>
<td>Manage profile</td>
<td>As a Registered User, I want to be able to update my profile. Or delete it.</td>
<td>20</td>
</tr>
<tr>
<td>40</td>
<td>K8</td>
<td>Share on social media</td>
<td>As a Registered User, I want to be able to share a Restaurant's information with my Facebook Friends.</td>
<td>30</td>
</tr>
<tr>
<td>30</td>
<td>K10</td>
<td>Edit Administrator account</td>
<td>As an Administrator I want to be able to edit my account, revised or update my information.</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>K9</td>
<td>Administrator management</td>
<td>As an Administrator I want to be able to search for a User, and delete it if needed.</td>
<td>30</td>
</tr>
</tbody>
</table>

2.6 Sprint Backlog (Sprint Two)
<table>
<thead>
<tr>
<th>ID</th>
<th>User stories</th>
<th>Description</th>
<th>Sprint Tasks</th>
<th>Assigned developer</th>
<th>Initial Size Estimate (hours)</th>
<th>Actual Effort at Sprints (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Task ID</td>
<td>Tasks (story points)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K6</td>
<td>Search</td>
<td></td>
<td>T6.1 Query to database with User's criteria. (location)</td>
<td>Alex</td>
<td>35</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T6.2 Query to database with User's criteria. (name and category)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T6.3 Display the result list</td>
<td>David</td>
<td></td>
<td></td>
</tr>
<tr>
<td>K3</td>
<td>Create</td>
<td></td>
<td>T3.1 Design restaurant making page form</td>
<td>Philip</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>restaurant</td>
<td></td>
<td>T3.2 Store new restaurant information into database</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>page</td>
<td></td>
<td>T3.3 Generate restaurant page from the making page</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T3.4 Display restaurant page generation success page to users</td>
<td>David</td>
<td></td>
<td></td>
</tr>
<tr>
<td>K7</td>
<td>Comments</td>
<td></td>
<td>T7.1 Make the commenting form on the Restaurant page.</td>
<td>Jeffrey</td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T7.2 Insert the comments of corresponding restaurants to the database</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K4</td>
<td>Rate</td>
<td></td>
<td>T4.1 Make the rating form on the Restaurant page.</td>
<td>Jeffrey</td>
<td>20</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>restaurant</td>
<td></td>
<td>T4.2 Store the rating value of corresponding restaurants to the database</td>
<td>David</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T4.3 Display the rate on the Restaurant page.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
After the Sprint two Planning meeting, the Scrum team selected four user stories to be delivered according to their importance. Each user story was broken down into several tasks to be completed.

Table 5 Sprint 2 Backlog

2.6 Changes during the Sprint Two

During the Sprint two, Product Owner decided to add one new user story points into the Sprint 2 on the 14th day. The changes are specified as follows

Table 6. Sprint 2 Changes

<table>
<thead>
<tr>
<th>ID</th>
<th>User stories</th>
<th>Description</th>
<th>Sprint Tasks</th>
<th>Assigned developer</th>
<th>Initial Size Estimate (hours)</th>
<th>New Estimate at Sprints (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>K4</td>
<td>Rate restaurant</td>
<td>As a Registered User, I want to be able to rate a restaurant I visited.</td>
<td>T4.4 Update the average rating value and show in the restaurant page</td>
<td>Philip</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

2.7 Scrum Review and Updated Product Backlog

During the Scrum Review meeting, the Development Team demonstrated the completed parts of the product to the customers, and got a good result. The customer accepted the artifacts of the Sprint two. The customer also required to remove the user story K8, because it has low business value for this project. And the updated Product Backlog is shown as follow:
Table 7. Updated Product Backlog

<table>
<thead>
<tr>
<th>Importance</th>
<th>ID</th>
<th>User stories</th>
<th>Description</th>
<th>Initial Size Estimate (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>K1</td>
<td>View restaurant</td>
<td>As a User, I want to be able to see the complete Restaurants list.</td>
<td>Done</td>
</tr>
<tr>
<td>80</td>
<td>K2</td>
<td>Register</td>
<td>As a User, I want to be able to register myself to the website with my Email and a chosen password to access to more features</td>
<td>Done</td>
</tr>
<tr>
<td>75</td>
<td>K6</td>
<td>Search</td>
<td>As a User, I want to be able to search for a particular Restaurant, based on location, name or category.</td>
<td>Done</td>
</tr>
<tr>
<td>70</td>
<td>K3</td>
<td>Create restaurant page</td>
<td>As a Registered User, I want to be able to make my own Restaurant page.</td>
<td>Done</td>
</tr>
<tr>
<td>65</td>
<td>K7</td>
<td>Comments</td>
<td>As a Registered User, I want to be able to leave a comment about a restaurant I've visited.</td>
<td>Done</td>
</tr>
<tr>
<td>63</td>
<td>K4</td>
<td>Rate restaurant</td>
<td>As a Registered User, I want to be able to rate a restaurant I visited.</td>
<td>Done</td>
</tr>
<tr>
<td>50</td>
<td>K5</td>
<td>Manage profile</td>
<td>As a Registered User, I want to be able to update my profile. Or delete it.</td>
<td>20</td>
</tr>
<tr>
<td>0</td>
<td>K8</td>
<td>Share on social media</td>
<td>As a Registered User, I want to be able to share a Restaurant's information with my Facebook Friends.</td>
<td>30 (Delete)</td>
</tr>
<tr>
<td>30</td>
<td>K10</td>
<td>Edit Administrator account</td>
<td>As an Administrator I want to be able to edit my account, revised or update my information.</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>K9</td>
<td>Administrator management</td>
<td>As an Administrator I want to be able to search for a User, and delete it if needed.</td>
<td>30</td>
</tr>
</tbody>
</table>

2.8 Sprint Retrospective

During the Sprint Retrospective meeting, the Development Team summarized the performance of the Sprint 2, and suggested one improvement tip:

- Team members should enhance the collaboration with each other. Especially the tasks have dependency on other tasks, the responsible developers should share their progress and implementation details with their counterparts.

Finish time:

<table>
<thead>
<tr>
<th>Hours</th>
<th>Minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Start time:

<table>
<thead>
<tr>
<th>Hours</th>
<th>Minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Questions about Scrum process (Based on the example project)

Q6. How many Sprints are planned in the project?
[ ] One
[ ] Two
[ ] Four
[ ] It’s not determined

Q7. What was the totally initial estimate effort of the Sprint one needed before changing for the example project?
[ ] 100 hours
[ ] 75 hours
[ ] 95 hours

Q8. Who should attend the Daily Scrum meeting in the example project?
[ ] Helen
[ ] Hamid
[ ] Philip
[ ] Alex
[ ] Jeffrey

Q9. How many Daily Scrum meetings in Sprint 1?

Q10. In which day the Product Owner update the task T2.6? (T2.6 is the ID of the task)

Q11. What is the most important user story in Product backlog after Sprint 1 Review meeting? (Please write the ID of the user story)

Q12. What activity (ies) is (are) missing in Sprint one?

Q13. What items have been updated in Product Backlog at Sprint 1 Review meeting?

Finish time:

<table>
<thead>
<tr>
<th>Hours</th>
<th>Minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
This section illustrates the essential elements of the Scrum framework and how to apply the Scrum method.

Scrum employs an iterative, incremental approach to develop projects. Each Scrum project has a set of Sprints. A sprint is a time-boxed iteration of one to four weeks, during which the Scrum team will create and deliver the products.

Sprint Activities:

a) Sprint Planning meeting is the first activity of a Sprint, it is held to plan what will be implemented into a Sprint (a fixed period of time used to deliver parts of the final product). The requirements from the customers are formulated as user stories to be delivered. The Product Owner selects the user stories with high importance from the Product Backlog to make up the Sprint Backlog.

b) The Development Team breaks down (expands) these user stories into tasks. The Development Team then takes 30 days or so to deliver the selected user stories. The Development Team must hold a Daily Scrum meeting (normally 15 minutes) each day to collaborate with each other. The Scrum Master ensures the Development Team holds the Daily Scrum meeting.

c) At the end of the Sprint, the Development Team demonstrates the completed parts of the product to the customer in a Sprint Review meeting.

d) The last activity is the Scrum Retrospective meeting, where the Development Team reviews the Sprint and looks for ways of improving (lessons learned).

Scrum roles:

- **Development Team**: The Development Team consists of a group of professionals (3 to 9 people, full-time) who are responsible for creating the project deliverables according to the user stories specified by the Product Owner.
  - Service:
    - Helping the Development team to understand the Product Owner’s decisions in the Product Backlog.
    - The Scrum Master serves the Development Team with coaching Scrum theory and removing the obstacles in the project progress.

- **Product Owner**: The Product Owner is one person. The Product Owner should order user stories in the Product Backlog. The Development Team should follow the Product Owner’s decisions in the product backlog.
  - Service:
    - Understanding how to arrange the product backlog to maximize business value.

Scrum Master:

- **The Scrum Master** is a facilitator who serves the Product Owner in managing the Scrum meetings with the Development Team.

Developer:

- **Developer** is not a mandatory role defined in Scrum framework. We introduce this role to specify the Development Team member.
### Basic Scrum Events

- **Sprint**
  - Time Length
  - ID
  - Sprint is a fixed period of time used to deliver parts of the final product.

- **Meeting**
  - Sprint Planning meeting
  - Daily meeting
  - Sprint review meeting
  - Retrospective meeting

There are four types of typical meetings in Scrum framework:
- Sprint planning meeting
- Daily Scrum meeting
- Sprint review meeting
- Retrospective meeting.

### Software artifact

#### Product Backlog

<table>
<thead>
<tr>
<th>Importance</th>
<th>ID</th>
<th>User story</th>
<th>Initial Time estimate</th>
<th>Short Description</th>
</tr>
</thead>
</table>

- **Product Owner:**
- **Scrum master:**

#### Sprint Backlog

<table>
<thead>
<tr>
<th>Importance</th>
<th>User story ID</th>
<th>User story</th>
<th>Time estimate</th>
</tr>
</thead>
</table>

The Sprint Backlog consists of the selected user stories from the Product Backlog to be delivered through a Sprint.

#### Task

<table>
<thead>
<tr>
<th>Task ID</th>
<th>User story</th>
<th>Assigned developer</th>
<th>Related Component</th>
</tr>
</thead>
</table>

The Development Team breaks down (expands) the user stories into tasks. However, task is not a mandatory artifact defined in Scrum framework. We introduce this artifact to specify development details for each user story.

#### Update

<table>
<thead>
<tr>
<th>ID</th>
<th>User story ID</th>
<th>Description</th>
</tr>
</thead>
</table>

Update is not a mandatory element defined in Scrum framework. We introduce those elements to illustrate any change of the user stories or others during the process of project.

#### Product Time Line

Product Time Line is not a mandatory element defined in Scrum framework. We introduce the product time line to guide the iterative, incremental progress of the whole Scrum project.

#### Sprint Time Line

Sprint Time Line is not a mandatory element defined in Scrum framework. We introduce the Sprint time line to guide progress of a sprint.

### Finish Time

<table>
<thead>
<tr>
<th>Hours</th>
<th>Minutes</th>
</tr>
</thead>
</table>

80
Questionnaire

Notes: Please answer the following questions based on the training materials you just read.
Please fill "T" in the square bracket behind the choice when you think it is Correct, such as [ T ]
Please fill "F" in the square bracket behind the choice when you think it is Wrong, such as [ F ]
If you are unsure if the choice is correct or wrong, please just leave it as [  ]

Example:
Q: Who has the authority to cancel a task?
[ T ] Product Owner (This choice is Correct, so fill T)
[ F ] Scrum Master (This choice is Wrong, so fill F)
[  ] Customer (I am not sure about this choice, so I just leave it as blank)
[ T ] Development Team (This choice is Correct, so fill T)

Part I: Questions for Scrum

Q1. Who helps the Product Owner decide on right actions?
[  ] Another Product Owner
[  ] Scrum Master
[  ] Development Team
[  ] There's no specific role for helping the Product Owner

Q2. We are going to start the first Sprint. What’s the first activity?
[  ] Sprint Initiation
[  ] Sprint Startup
[  ] Sprint Planning
[  ] Daily Scrum

Q3. Team Members decided to cancel Daily Scrum meetings for the rest of one Sprint, to save time and get things done faster. What do you think of this decision?
[  ] Acceptable, because delivery of the products is the first priority
[  ] Not right, but acceptable since they’ve reached this decision and it’s their own responsibility to manage their own efforts
[  ] Not acceptable, because the Daily Scrum is required in Scrum

Q4. How the Product Backlog should be ordered based on?
[  ] the effort of the user stories
[  ] the float of the user stories
[  ] the importance of the user stories
[  ] the relationship among user stories

Q5. What are the main roles in Scrum Team?
[  ] Scrum Master
[  ] Product Owner
[  ] Development Team
[  ] Developers

Section II. Scrum Project Example

This section will illustrate you a real project used the Scrum method. This project has 4 sprints, due to the time limitation, we only illustrate two Sprints description as an example to help you to understand the Scrum process.

This project will developed and delivered a new website named “Karlskrona Restaurant Network” (KRN). This website was aiming to collect and exhibit the information of the restaurants in Karlskrona.

This Scrum team has 6 team members, one Product Owner, one Scrum Master and four developers. Their roles are specified as follows:
This is the initial Product Backlog created by the Product owner at the beginning of the project.

After the Sprint Planning meeting, the Development Team selected two user stories with the highest importance value, K1 and K2, to be delivered in the first Sprint. These user stories were specified in the Sprint Backlog.

Notes: Due to the space limitation, some contents of the Daily meeting have been neglected. If there is no special specification of the day, you can just imagine it as a “normal day” as the same as the previous day.
During the Scrum Review meeting, the Product Owner changed some user stories' importance and added one user story K10 to the Product Backlog according to the new requirements from the customer.

After the Sprint two Planning meeting, the Scrum team selected four user stories to be delivered according to their importance. Each user story was broken down into several tasks to be completed.

### Product Backlog

<table>
<thead>
<tr>
<th>Importance</th>
<th>ID</th>
<th>User Story</th>
<th>Initial Time estimate</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>75</td>
<td>K6</td>
<td>Search</td>
<td>35 hours</td>
<td>As a User, I want to be able to search for a particular Restaurant, based on location, name or category.</td>
</tr>
<tr>
<td>70</td>
<td>K3</td>
<td>Create restaurant page</td>
<td>20 hours</td>
<td>As a Registered User, I want to be able to make my own Restaurant page.</td>
</tr>
<tr>
<td>65</td>
<td>K7</td>
<td>Comments</td>
<td>20 hours</td>
<td>As a Registered User, I want to be able to leave a comment about a Restaurant I've visited.</td>
</tr>
<tr>
<td>63</td>
<td>K4</td>
<td>Rate restaurant</td>
<td>20 hours</td>
<td>As a Registered User, I want to be able to rate a Restaurant I've visited.</td>
</tr>
<tr>
<td>50</td>
<td>K5</td>
<td>Manage profile</td>
<td>20 hours</td>
<td>As a Registered User, I want to be able to update my profile. Or delete it.</td>
</tr>
<tr>
<td>40</td>
<td>K8</td>
<td>Share on social media</td>
<td>30 hours</td>
<td>As a Registered User, I want to be able to share a Restaurant's information with my Facebook friends.</td>
</tr>
<tr>
<td>30</td>
<td>K10</td>
<td>Edit Administrator account</td>
<td>15 hours</td>
<td>As an Administrator I want to be able to edit my account, revised or update my information.</td>
</tr>
<tr>
<td>10</td>
<td>K9</td>
<td>Administrator management</td>
<td>30 hours</td>
<td>As an Administrator I want to be able to search for a User, and delete it if needed.</td>
</tr>
</tbody>
</table>

### Sprint 2 Backlog

<table>
<thead>
<tr>
<th>User story ID</th>
<th>User story</th>
<th>Time estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>75</td>
<td>K6 Search</td>
<td>35 hours</td>
</tr>
<tr>
<td>70</td>
<td>K3 Create restaurant page</td>
<td>20 hours</td>
</tr>
<tr>
<td>65</td>
<td>K7 Comments</td>
<td>20 hours</td>
</tr>
<tr>
<td>63</td>
<td>K4 Rate restaurant</td>
<td>20 hours</td>
</tr>
</tbody>
</table>

### Sprint 2

21 days

**Sprint planning**

<table>
<thead>
<tr>
<th>Day</th>
<th>Daily Meeting Day 1</th>
<th>Daily Meeting Day 2</th>
<th>Daily Meeting Day 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Length: 3 hours</td>
<td>Length: 3 hours</td>
<td>Length: 3 hours</td>
</tr>
</tbody>
</table>

**Sprint planning**

<table>
<thead>
<tr>
<th>Day</th>
<th>Daily Meeting Day 4</th>
<th>Daily Meeting Day 5</th>
<th>Daily Meeting Day 6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Length: 3 hours</td>
<td>Length: 3 hours</td>
<td>Length: 3 hours</td>
</tr>
</tbody>
</table>

**Sprint planning**

<table>
<thead>
<tr>
<th>Day</th>
<th>Daily Meeting Day 7</th>
<th>Daily Meeting Day 8</th>
<th>Daily Meeting Day 9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Length: 3 hours</td>
<td>Length: 3 hours</td>
<td>Length: 3 hours</td>
</tr>
</tbody>
</table>

**Sprint planning**

<table>
<thead>
<tr>
<th>Day</th>
<th>Daily Meeting Day 10</th>
<th>Daily Meeting Day 11</th>
<th>Daily Meeting Day 12</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Length: 3 hours</td>
<td>Length: 3 hours</td>
<td>Length: 3 hours</td>
</tr>
</tbody>
</table>

**Sprint planning**

<table>
<thead>
<tr>
<th>Day</th>
<th>Daily Meeting Day 13</th>
<th>Daily Meeting Day 14</th>
<th>Daily Meeting Day 15</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Length: 3 hours</td>
<td>Length: 3 hours</td>
<td>Length: 3 hours</td>
</tr>
</tbody>
</table>

**Sprint planning**

<table>
<thead>
<tr>
<th>Day</th>
<th>Daily Meeting Day 16</th>
<th>Daily Meeting Day 17</th>
<th>Daily Meeting Day 18</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Length: 3 hours</td>
<td>Length: 3 hours</td>
<td>Length: 3 hours</td>
</tr>
</tbody>
</table>

**Sprint planning**

<table>
<thead>
<tr>
<th>Day</th>
<th>Daily Meeting Day 19</th>
<th>Daily Meeting Day 20</th>
<th>Daily Meeting Day 21</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Length: 3 hours</td>
<td>Length: 3 hours</td>
<td>Length: 3 hours</td>
</tr>
</tbody>
</table>

### Task 10.4

**ID:** T4.4

**Tasks:**

- Display the rating of the Restaurant page

**Assigned Developer:** Philip

**Related User Story:** K4

**Description:**

The Product Owner deleted user story K3 from Product Backlog.
Questions for Scrum process (Based on the example project)

Q6. How many Sprints are planned in the project?
- [ ] One
- [ ] Two
- [ ] Four
- [ ] It’s not determined

Q7. What was the totally initial estimate effort of the Sprint one needed before changing for the example project?
- [ ] 100 hours
- [ ] 75 hours
- [ ] 95 hours

Q8. Who should attend the Daily Scrum meeting in the example project?
- [ ] Helen
- [ ] Hamid
- [ ] Philip
- [ ] Alex
- [ ] Jeffrey

Q9. How many Daily Scrum meetings in Sprint 1?

Q10. In which day the Product Owner update the task T2.6? (T2.6 is the ID of the task)

Q11. What is the most important user story in Product backlog after Sprint 1 Review meeting? (Please write the ID of the user story)

Q12. What activity (ies) is (are) missing in Sprint one?

Q13. What items have been updated in Product Backlog at Sprint 1 Review meeting?

Finish Time:  

Start Time:  

Minutes
APPENDIX B. SURVEY QUESTIONNAIRE

Survey

1. How long time have you been using Scrum? __________

2. How did you learn Scrum process?
   ● Reading books
   ● Searching Internet
   ● Taking lectures
   ● Via actual project

3. Do you think a graphical language can help you to understand Scrum process?
   ● Strongly agree
   ● Agree
   ● Neutral
   ● Disagree
   ● Strongly disagree

4. Do you think a graphical language can help the new learner to understand Scrum process in a better way?
   ● Strongly agree
   ● Agree
   ● Neutral
   ● Disagree
   ● Strongly disagree

5. Which way (textual/graphical) you would like to use in learning Scrum process.
   ● Textual
   ● Graphical
   ● Both

6. In your opinion, what are the advantages and disadvantages of teaching the Scrum process in using the proposed graphical language?

7. In your opinion, what are the advantages and disadvantages of teaching the Scrum process in using the proposed textual materials?

8. What are the problems or obstacles when you learn the Scrum process?

9. Do you think a graphical language can help you to solve or mitigate the above problems, and why?

10. Will you use this graphical language in the future work? Why?

   Thanks for your participation!
   Please send this questionnaire to zyphth@gmail.com