Thesis no: MSSE-2016-17

Assessment of Agile Maturity Model

A survey

Rahul Deekonda

Prithvi Raj Sirigudi

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

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):
Rahul Deekonda
E-mail: rade15@student.bth.se

Author(s):
Prithvi Raj Sirigudi
E-mail: prsi15@student.bth.se

University advisor:
Indira Nurdiani
Department of Software Engineering

Faculty of Computing	Internet	:	www.bth.se
Blekinge Institute of Technology	Phone	:	+46 455 38 50 00
SE–371 79 Karlskrona, Sweden	Fax	:	+46 455 38 50 57

Abstract

Context. In recent years Agile has gained lots of importance in the field of software development. Many organization and software practitioners has already adopted agile practice due to its flexibility in nature. Hence, agile development methodologies have been replaced to traditional development methods. Agile is a family of several methodologies namely Scrum. eXtreme programming (XP) and several others. These several methods are embedded with different set of agile practices for the organizations to adopt and implement for their development process. But there is still a need for empirical research to understand the benefits of implementing the Agile practices which contributes to the overall success of accomplishment of the software project. Several agile maturity models have been published over a decade but not all of the models have been empirically validated. Hence, additional research in the context of agile maturity is essential and needed.
Objectives. This study focus on providing a comprehensive knowledge on the Agile Maturity Models which help in guiding the organizations regarding the implementation of Agile practices. There are several maturity models published with different set of Agile practices that are recommended to the industries. The primary aim is to compare the agile maturity maturity models and to investigate how the agile practices are implemented in the industry Later the benefits and limitations faced by the software practitioners due to implementation of agile practices are identified.
Methods. For this particular research an industrial survey was conducted to identify the agile practices that are implemented in the industry. In addition, this survey aims at identifying the benefits and limitations of implementing the agile practices. A literature review is conducted to identify the order of agile practices recommended from the literature in agile Maturity Models.
Results. From the available literature nine Maturity Models have been extracted with their set of recommended agile practices. Then the results from the survey and literature are compared and analyzed to see if there exist any commonalities or differences regarding the implementation of agile practices in a certain order. From the results of the survey the benefits and limitations of implementing the Agile practices in a particular order are identified and reported.
Conclusions The findings from the literature review and the survey results in evaluating the agile maturity models regarding the implementation of agile practices.

Keywords: agile maturity models, agile maturity frameworks, agile assessment model.

Acknowledgments

Foremost, we would like to express our sincere gratitude to our supervisor Indira Nurdiani who supported us continuously with a great patience in a professional manner throughout this scientific journey. Undoubtedly she is the person who has shared may ideas and framed a basis for presenting this Master Thesis. We both sincerely thank our supervisor for the encouragement and guidance provided by her.

Secondly we thank all the participants who are involved in the survey for taking time and sharing their personal experiences. Thereby giving us an essential input for conducting this research.

Finally we are truly lucky to have the most caring and loving families who are supporting us all the time even in the hard times of epic journey of life. We are fortunate enough to have some trustful friends who always make us smile and stay happy even in sad times. A word of thanks is not sufficient to express our gratitude to Sirigudi family, Deekonda family and for the loving friends which we have made in our lives.

Thank you all

Rahul Deekonda,
Prithvi Raj Sirigudi.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Agile Development and Methodology

Agile is first introduced in 2001 publishing the agile manifesto with the leading visionaries in the software field. Agile focuses on frequent delivery of high quality, working software with the demand of high business-valued functionality [13]. The rise of the “agile” era made “agile” a buzz word and opened the door for parallel and dynamic service development, maintenance and support [17]. “Agility is the ability of to both create and respond to change in order to profit in a turbulent business environment” [32]. Agile methodology is characterized by extensive planning and light-weight process [39], [47]. Agile is a family of several methodologies namely Scrum, Extreme Programming, Lean development, Feature Driven Development (FDD), Dynamic System Development Method (DSDM), Crystal method, Lean Software Development (LSD) and Adaptive Software Development (ASD) , [3], [15]. The main advantage over the traditional methodologies is that agile methodologies are adaptive in managing the change on requirements throughout the development lifecycle process [2], [13].

Back in February 2001, the agile software development alliance was formed with 17 software developers to discuss light-weight development methods [13]. These software developers signed and published a manifesto for agile software development uncovering better ways of developing software, the key values stated in the manifesto for agile software development were [13] [2].

- Individuals and iterations over process and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.

Agile mainly focuses on the execution of the project rather than extensive planning[2]. Agile is characterized by self-organizing teams, value driven, fast execution and business oriented [2]. The essential features provided by the agile development methodology as stated in the article [2] are

- Iterative and incremental style of development that dynamically adjusts to changing requirements.
- It is people oriented with simple design, oriented with 2 to 4 weeks of development life cycle.
- It provides regular testing and frequent releases with collective code ownership.
- It enables better risk management with the focus on code refactoring and the product standards.

Several organizations had already adopted agile practices. But the study conducted by Patel and Ramachandran states that the software process improvement models have not yet shown a clear mechanism for aligning Software Process Improvement (SPI) activities with business objectivities [32]. As agile is also a matter of organizational culture, agile maturity needs to deal with cultural issues like fixed mindset vs. growth mindset, power distance, and uncertainty avoidance [45].According to the study in the article [45] approximately there are about 40 agile maturity models published. These maturity models guide the organizations in a systematic development process to accomplish the project successfully with desired capabilities [45]. But not all of these models are validated through proper empirical research and also not all of the agile maturity models are available. Moreover, “these models differ in their underlying structure prescribing different possible paths to maturity in agile software development, neglecting the fact that agile teams struggle to follow prescribed process and practices” [32][11]. Hence, organizations might possess a greater challenge in adopting a suitable maturity model for their own development process. They might also face some difficulties in improvising their development process, as not every software project is similar to each other. Furthermore, these several different agile maturity models implement different agile practices in different order [47].

So, how would organizations adopt these agile practices in a particular order according to their project domain and scope? This itself can possess a greater challenge and can impact the organization’s overall performance and reputation. These several challenging factors in the agile development process which are been faced by the current IT industries motivated the authors in performing a scientific research in revealing the actual benefits and limitations of implementing the agile practices given a certain order. With a more concern on how extent these agile maturity models are implemented in the current IT industry? Also focusing on the order of agile practices implemented in their development process.

1.2 Research Questions

The main aim of this research is to evaluate the benefits and limitations of implementing agile practices given a certain order. This research entails examining of current literature in agile maturity models and validate the findings from the literature through an empirical study. The study focus on comparing the order of implementation of agile practices in the industries with the scientific literature. The research questions are formulated as follows:

- **RQ1:** What are the order of practices recommended in agile maturity models?
- **RQ2:** Which order of agile practices are implemented in the industry?
- **RQ3:** What are the benefits and limitations of implementing agile practices?

1.3 Expected Outcomes

By conducting an empirical study, we aim to explore the benefits and limitations of agile maturity models given by a certain order. Through this study we expect the following outcomes:

- **EO1:** The order of practices in agile maturity models are extracted through conducting a literature review. For instance, the order of agile practices can be like initially story cards (requirements engineering) is performed then TDD is implemented with pair programming and so on.
- **EO2:** The agile practices implemented in the industry are identified then the benefits and limitations of the implementing agile practices are derived through conducting a survey.
- **EO3:** By synthesizing and analyzing the collected data the differences and commonalities of the agile practices are discussed. The benefits and limitations of implementing the agile practices are identified through conducting a survey.

1.4 Structure of Thesis

This section describes the structure of the research and gives an overview of each chapter presented sequentially in this document. The following is a small description of the chapters included in this document:
Chapter 2: This chapter discusses the background and related work of the research and it is further divided into four sections. In the first section, a brief

overview of the agile methodologies is discussed. In the second section agile practices and its benefits are discussed followed by introduction of agile maturity models in the third section. Coming to the fourth section it comprises of related work.

Chapter 3: This chapter discusses the research methodology. In which this chapter is further divided into five sections. In the first section, how the literature review is conducted is discussed and it also includes the steps involved in conducting the literature review. Coming to the second section, how the survey is carried out to achieve the objectives of the thesis is discussed. Next the third section comprises of mapping the research questions to the research methodology followed by the fourth section with the data analysis methods.

Chapter 4: This chapter gives an account of the results and analysis. It includes the synthesized results and the method followed for analyzing the results achieved through conducting the literature review and survey.

Chapter 5: The first section of this chapter includes the validity threats for the study. The second section of this chapter entails the discussions of the findings. It discusses about answering the research questions. This chapter also describes the threats to validity involved in the process of executing thesis.

Chapter 6: This chapter finally concludes the findings of the study and the contribution of the study are reported with future work.

Chapter 2

Background and Related Work

2.1 Agile Methods

As mentioned earlier the field of agile development involves several methods with a pre-defined set of practices developed by the experienced expert software practitioners for an enhanced process of software development. Based on the published scientific articles a deep analysis of these methods is performed. Here author's present a brief summary of these methods involved with agile development in order to provide a clear understanding of the methods even for the readers who are not aware of the agile methodologies.

2.1.1 eXtreme Programming (XP)

The XP methodology focuses on user satisfaction with five key values i.e. communication, simplicity, feedback, respect and courage [2]. Extreme programming is based on 12 practices planning game, collective code ownership, coding standards, small releases, simple design, pair programming/ TDD, metaphor, refactoring, continuous integration, user acceptance tests, and sustainable pace [2]. The testing process is done with unit test (UT) and user acceptance test (UAT).

2.1.2 Feature Driven Development (FDD)

The FDD aims at conceptualizing a model of feature and their priority, its main focus is towards delivering the working features to the end user for use [2]. This approach will help both the user and developers to prioritize the features thereby achieving high priority features as needed, it also saves time and costs for the user. FDD uses eight essential practices in delivering the working software, they are domain object modeling, component/ class ownership, feature teams, configuration management, regular builds, visibility of progress, developing by feature, inspections and results [2]. Comparing both the FDD and XP, it results that XP is suitable for large teams with the cooperative environment because XP requires effective communication so this can be more complex with the large teams and cooperate projects [2].

2.1.3 Scrum

Scrum, due to its simplicity and proven productivity in the software industry it has gained increasing popularity over the last decade, moreover a research survey conducted by Cao et.al showed that this is the most common used model in the software industry even with the multiple teams across large firms [2]. Scrum involves five principles; they are [2].

- Teams are divided into small, cross functional through encouraging self-organizing teams.
- Product backlog splits the work tasks which contain small and well defined features. For backlog, each feature is prioritized based on its importance and estimated effort for accomplishment.
- These tasks are split into iterations with time span of 3-4 weeks [2] [1]. These iterations are named as sprints.
- The priorities for the release plan are planned with the collaboration of the customer.
- In the end of each sprint, the working features are presented and delivered to the customer.
- From the past iterations, the lessons for improving the process are learned and the process is optimized.

The scrum methodology introduces three vital roles namely product owner, scrum master and scrum teams. The product owner is almost a replica of the end user and is responsible for the product specifications. The product owner is responsible for assuring complete satisfaction of working software as per user needs. Scrum master collaborates the work with the product owner and facilitates the team. The team size is consistent to be small around 7 (plus/minus 2) members [2].

2.1.4 Crystal Family of Methodologies

Crystal methodology is an adaptable approach which includes several family of agile methodologies. Each methodology is assigned with different color and each methodology possesses unique characteristics driven by variable factors like team size, system criticality, and project priorities. The different methodologies are crystal clear, crystal yellow, crystal orange and crystal red. The darker the color, the heavier is the methodology. Crystal methodology is suggestible for a project having less criticality and for collocated teams. Clear development has seven characteristics: frequent delivery, osmotic communication, personal safety, reflective improvement, focus, easy access to expert users and requirements for the technical environment [7].

These different methods of agile are embedded with different agile practices with a focus on frequent delivery of high quality, working software with the demand of high business-valued functionality [13]. Implementation of these agile methods will help the organizations in improving the agile methods through obtaining an adaptive and iterative process with an enhanced process in development process [21]. Several agile practices have been identified from the above mentioned different agile methods. In the coming section, a brief description of each agile practice is mentioned corresponding to it.

2.2 Agile Practices

These existing agile software development methods and models formulate several agile practices following all the 12 agile principles. “Agile practices are the concrete activities and practical techniques used to develop and manage software projects in a consistent manner with the agile principles” [47]. For reducing costs and responding to changes in dynamic market conditions agile practices have been recognized in many software companies as a mechanism to enhance their development process [36]. To create an adaptive product which is less expensive and easier to develop the agile practices are implemented at both the project and organizational level with an advantage to quickly respond to the changes in their project ecosystem [36]. In order to achieve this, several agile practices are adopted with the focus on frequent delivery of working software through collaborative practices such as face-to-face communication, daily standups, etc. The following are the agile practices extracted from different scientific articles [30], [28], [51], [7], [26], [34]. These practices were considered since they were related to agile methodologies. And also from the recent studies included in the related work shows that these are the most used practices in the industry and thus we have considered these 17 agile practices for our study. These agile practices are further incorporated into the questionnaire for conducting the survey. Here a brief description of each practice with an ID is provided for better understandability of the agile practices with its usage in the table 2.1.

Table 2.1: Agile practices

ID	Agile Practice	Description	Agile Process
P1	Face-to-face meeting	The interaction between the teammates in the same location or video conferencing if the team is distributed [26].	Scrum, XP, FDD and crystal

P2	Self-organizing cross functional teams	It is a practice where the team organizes themselves, assign tasks and responsible for their own task [28].	Scrum
P3	On-site customer	The customer is available at any time during the project execution for explaining in detail about the user stories [53], [51].	XP
P4	Pair programming	Pair programming is a way of programming were two programmers or developers work at one workstation which enables them to work together on same code, same algorithm and same test cases [53].	XP
P5	Sprint Planning/ Planning game	Planning game is to design the project plan implementing in iterations by the product owner [30], [53].	Scrum and XP
P6	Tracking progress	Used to track the progress of the project with the help of burndown charts, burnup charts and others [53].	Scrum
P7	Refactoring	It is a practice where the code can be restructured from the existing codes but not the behavior [30].	XP
P8	Iteration Reviews/ Retrospectives	A meeting is conducted after each iteration to discuss the work done [53].	Scrum and XP
P9	Short Iterations and Frequent Releases	Frequent releases of the software, early and continuous delivery of partial but fully functional software [53], [26].	Scrum and XP
P10	Simple design	A goal to design simplest solution [51], [26].	XP
P11	Time Boxing/ Sprint/ Iterations	A fixed deadlines are created for each cycle to stay on schedule [34].	Scrum and XP

P12	Metaphors and stories	This is a high level requirement outlining the purpose of the system and involves breaking the requirements into user stories and maintaining the stories in a backlog. This acts as a communication medium between the product owner, customers, and developers [30], [26].	Scrum, FDD and XP
P13	Test driven/ Test first Development (TDD)	Test cases are written before the implementation of the function code [51].	XP
P14	Continuous Integration	Integrating the work frequently done by the team members at least once a day [51].	XP
P15	Coding standards	All the team members or developers follow the same coding rules and standards [30].	XP
P16	Collective ownership	A team member or a developer can change the code at any time without approaching the code owner for improving the code quality like bug fixing [53], [30].	XP
P17	Daily standup meetings	A short meeting of 10-15 minutes is conducted every day to know and check the status of the developed product [51], [53].	Scrum

2.3 Agile Maturity Models

A maturity model presents “an evolutionary progress in the demonstration of a specific ability or in the accomplishment of a target from an initial to a desired or normally occurring end stage” [22]. Maturity models are also defined as “The development of an entity over time and this entity can be anything of interest, a human being, an organization function, etc.” [52] Whereas Fontana et al. states “Maturity models are the instruments used to rate capabilities and based on this rating, initiatives can be implemented to improve the maturity of an element- a person, an object or a social system” [10]. However, these different definitions provide a logic behind on how these maturity models work but none explains the actual meaning of maturity neither the elements of the model [52]. These agile maturity models are based on the agile software development values, principles, and practices [32]. They provide a clear path at every stage of the development process and guides the organization in an appropriate way to complete the project successfully. The maturity models link the agile software development practices to the maturity levels to keep the representation clear, understandable and usable, but it is not an exhaustive representation of agile software development process [32]. These agile levels contain a set of agile practices and when adopted collectively it makes significant improvements to the development process so that core value of agility can be achieved [47]. In order to understand the maturity models, it is necessary to have a deeper understanding of the agile maturity models from three perspectives as suggested in the article [52] they are,

- An understanding of basic terms like 'maturity' and 'capability'.
- Purpose, application and benefits.
- Structure and components.

It is important to understand the agile maturity models as these provide a structured development process for a defined purpose with several applications. Thereby it is necessary for the practitioners to have a clear idea on the development process so as to develop in an efficient way achieving the agile benefits.There are several maturity models proposed in recent years but not all the maturity models are validated in the scientific research and development. The issue here is that models are still initial, few approaches have been scientifically tested and there is some evidence that agile practitioners do not realize benefits in having prescriptive maturity models [12]. These concerns and confessions over the maturity models pushed authors forward in understanding the benefits and limitations of implementing the agile maturity models. Authors performed and managed to retrieve the most distinct maturity models from the scientific databases through the access provided by Blekinge Institute of Technology, Sweden. After performing a detailed analysis and synthesis an overview of each different agile maturity model published in recent years is presented with complete reference to the respective researchers and is reported in the results section.

2.4 Related Work

Begel et.al conducted an empirical study at Microsoft to know the usage of agile development and perception of people involved in developing, testing and training [5]. A survey was conducted to know the practitioners' perception who are involved in product development with the agile usage and implementation of its practices. This study also finds out the benefits and limitations of implementing agile development methods and its practices. The findings of the study showed that one-third of the respondents use the agile software development methods, especially scrum as their development methodology. This study result shows that the test-driven development and pair programming are the least adopted practices in the industry.

A study conducted by R. Vijayasarathy et.al aims to find out the factors that drive them to adopt and use the agile practices [50]. For this study authors have chosen survey as the research methodology and conducted the survey with 98 highly experienced software professionals. The survey results showed that the test-first (TDD) and XP practices are mostly used followed by pair programming, scrum practices, agile modeling, agile unified process, continuous integration, Feature Driven Development (FDD). Practices like frequent releases and refactoring are less used. Finally, in this article authors conclude that personal interest drives the practitioners to adopt the agile practices.

A survey conducted by Santos and et.al [43] aims at finding the perception of practitioners regarding the implementation of agile practices that reduces the cost, scope and improve the product efficiency. Initially, a literature review is conducted to gather the agile practices later they conducted an online survey to improve the performance criteria for a software project with respect to cost and scope [43]. An exploratory factor analysis was conducted to analyze the results of the survey. By analyzing the results, it addressed that the adoption of agile practices can be represented in factors with respect to the application to improve the cost deduction and scope [43]. The exploratory factor analysis conducted in the study shows that the agile practices can be implemented together in improving the efficiency on cost and scope in four different aspects team abilities, management of requirements, quality of the code developed, delivery of the software on budget and on time [43]. But the author does not describe the challenges faced during the adoption of the agile practices.

Rodriguez et.al conducted an empirical study to find out the usage of agile/ lean methodologies and its practices in the software industries [41]. A survey was conducted to collect the data from 200 software industries across the globe. The results of the study show that 58% are implementing the agile/ lean practices. The survey results also tell the benefits of adopting agile/ lean practices. In this study authors conclude that the capability of working on the distributed development is a very big challenge, and also participants who are not implementing the agile/ lean practices are not showing any preferences towards implementing the agile/ lean practices.

An exploratory study conducted by Petersen et.al [49] to understand which of the practices are adopted together in the industry and how these practices relate to the agile benefits and limitations. For this particular study authors conducted a survey, first objective is to identify the development models based on rigid development (RD) and agile practice usage by practitioners. Secondly authors identified the use of agile practices overtime by providing the time indicating sliders. At last in relation to the development models and usage of agile practices authors investigated and prioritized the agile benefits and limitations. Survey was conducted with 45 practitioners to identify the agile benefits and limitations and the data is analyzed by hierarchical cluster analysis and voting analysis. Author identified agile practice adoption scenarios based on eliciting practice usage over time. Author also prioritized the agile benefits and limitations in relation to development models and agile practice adoption scenarios.

Kent Beck [4] discussed 12 of the agile practices that are related to XP. In this book, author invents a new discipline software development with a set of agile practices. The new discipline is invented based on the agile principles and activities like coding, testing, listening and designing which will help in improving the economic performance in software development [4]. Author states that one practice does not stand well but it requires another practices to keep them in balance [4]. Authors summarized the twelve practices and links two practices which reinforce each other. The figure 2.1 published by Beck summarizes and links the 12 XP practices. In this study author is particular to the XP practices but not all of the agile practices.

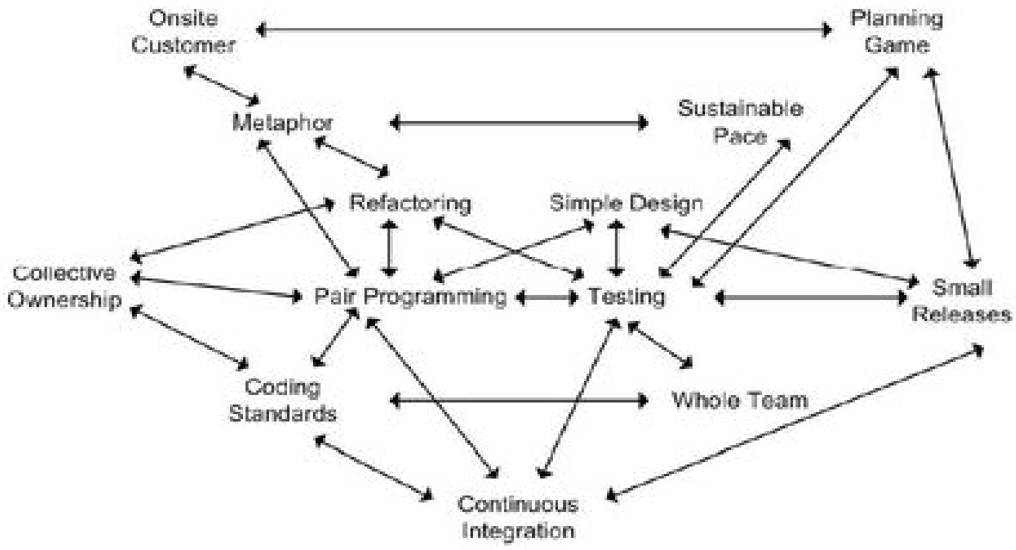


Figure 2.1: XP practices [4].

Scheweigert et. al had performed a study on Agile maturity models. Authors describes the status of the agile maturity models and conveys that the available maturity models are structured in a top level compilation [45]. In their article they provided an approach to analyze the agile maturity models in terms of extracting the content, mapping it to a reference models and finding the real agile maturity issues through synthesis [45]. They did not make any attempt in proposing a new maturity model within their research but they concluded that there is a need for scientific research in this particular topic i.e. Agile maturity models.

Demirors et al had made a study in assessing the agile maturity models to figure out the strengths and weakness of the agile maturity models/ frameworks. They have considered five maturity models available in the literature to know how sufficient these models can provide insights about an organization's agile capability [31]. They made an assessment on each maturity models with an assessment criterion in terms of fitness for purpose, completeness, definition of agile levels, objective and correctness through conducting a case study [31]. They also figured out the strengths and weakness in each model and concluded that "there is a need to improve the maturity models for better guidance in agile process adoption, process improvement and process assessment" [31].

A wide range of investigation is going on agile since decades. Most of the articles [27], [37], [46] focuses only on the agile practices adoption and usage of the agile practices. For implementing these agile practices there needs to be a structured process and the agile maturity models helps the organizations to implement the agile practices in an order. Several maturity models have been published in the recent years which guides the organizations in a structured manner in implementing the agile practices for a better development process. But not all of these maturity models defined with particular order of implementing of the agile practices are empirically validated [45], [32], [31]. This motivated authors to conduct this research.

Chapter 3

Research Method

3.1 Literature Review

Hart defined literature review as “The use of ideas in the literature to justify the particular approach to the topic, the selection of methods, and demonstration that this research contributed something new” [23]. For conducting this literature review authors followed the guidelines provided by Levy and Dr. Rowley in the articles [23] and [42]. Literature review process is performed in three sequential steps Input, Processing and Output to know the existing knowledge on agile maturity models. Stages of the literature review process are performed in sequential steps to collect, know, comprehend, apply, analyze, synthesize and evaluation of the literature [23], [24]. Initially input step comprises of gathering the manuscripts required for conducting the literature review. Next in the processing stage a detailed examination of the literature is carried out with identifying, summarizing, illustrating, comparing, connecting and generalizing the valid literature. Finally output step entails of documenting the results of the literature review by following the guidelines mentioned in the article [23], [24].

Literature review helps in finding out the existing body of knowledge related to the subject area. It also helps the authors to know what has done and what is needed to be done [23]. A literature review is conducted in this study to retrieve and understand different agile maturity models with the process involved in agile development. This resulted in gathering the agile maturity models and the order of practices implemented in each maturity model. This framed the basis for performing the research.

After getting finalized with the topic “Agile Maturity Models” the initial step to perform a background and related study, which has been performed and reported in the previous chapter 2 of the document. The next preliminary step is to perform a literature study to identify the work that has already been done in this field of research which enables the authors in providing insights about different agile maturity models. This provided the authors in gaining immense comprehensive knowledge about the software development involved with agile. Later on, literature searches are performed to extract all the published maturity models available in the scientific databases. The main aim of the literature review is to identify different maturity models with the identification of the order of practices recommended for the agile development.. As mentioned earlier, the literature review is performed in three sequential steps namely input, processing and output.

3.1.1 Input

A literature review is conducted in this study for gathering agile maturity models from the existing literature, the objective of the literature review is to summarize the state of the art in this subject field [36]. Initially, authors framed the keywords as agile maturity models, agile maturity frameworks, agile assessment models and software process improvement for conducting the search. Using these keywords database search is performed in the scientific databases for retrieving the articles related to study. Initially, a search string was framed using Boolean AND/OR operations and is used in several scientific databases namely Google scholar, Engineering village, Scopus and BTH-Summon. According to the context and design of the databases the search string was modified to retrieve the articles. The search string is presented in the table 3.1.

Table 3.1: Search string

Search string: ((((agile maturity model) OR agile maturity framework) AND software process improvement) OR agile assessment model)

From the refined searches, all titles and abstracts were read thoroughly and the papers which describe about the agile maturity models were only considered. To find an effective literature it makes sense to look into conference papers, journals, and scientific articles [35], [37]. To the best of the authors knowledge initially very less scientific articles were found which are related to the research topic. A start set of 8 research articles were considered. The search strategy used for this study is presented in the figure 3.1.

Thereby reading the full text of the articles authors decided to perform forward literature and backward literature search as per the guidelines indicated in the article [35]. As the keyword search process is associated with the use of technology specific terms, keywords appears and disappear from the literature overtime. Therefore, backward and forward approaches are implemented for the ease of authors to follow the models, theories, theoretical constructs and research streams [35]. The forward and backward literature searches are performed to build a solid theoretical foundation for the study through extraction of additional essential information from the literature.

Backward Literature search: It is performed in three specific sub-steps backward reference search, backward author search and previously used keywords

[23]. Backward reference search is performed through references of references as mentioned in the guidelines to possess a deeper knowledge of the evolution of agile maturity models [23]. Backward author search is referred as the search conducted based on the authors names. This helped in finding out the author's previous work and gathering the papers related to the field of study. The final step in this process is performed by using previously used keywords from the relevant papers This helped the authors in finding a total of 9 articles out of 75 that are related to the problem domain.

Forward Literature search: This is carried out in two ways: forward reference search and forward author search [23]. Forward reference search is performed through reviewing the articles that had cited the paper which further helped in finding the new literature. Forward author search is done by searching the papers related to the authors name. A total number of 36 papers were retrieved from these searches but only 10 articles were relevant for this study.

The forward and backward literature search resulted in 33 numbers of articles but among these, the articles were only selected based on the inclusion criteria mentioned below with the relevance of the study [19].

Table 3.2: Articles retrieved

Research Articles	**Database**	**No. of articles retrieved**
[45], [22], [12], [31], [48], [10], [11], [39], [32], [47], [6], [30]	Google Scholar	12
[36], [37], [46], [57], [3], [25]	Engineering Village	6

The table 3.2 shows the articles selected for conducting this particular study. The inclusion and exclusion criteria for this particular study is:

Inclusion criteria:

- Papers discussing agile methodologies.
- Paper discussing agile practices.
- Papers related to the research problem domain (agile maturity model and agile maturity frameworks).
- Papers are selected only if it the text is in English.
- Articles available in full text.

Exclusion criteria:

- Papers older than 10 years were excluded for primary studies.
- Articles related to agile adoption in the non-software sector were excluded.

- Articles that are not peer reviewed were excluded for primary studies.
- Duplicate studies are filtered and were eliminated.

Based on the above-mentioned inclusion and exclusion criteria each article is read thoroughly and only the relevant papers are selected. The articles used for this study are show in the table 3.2. Nine different papers describe different agile maturity models. These are considered as primary set for the study. Both the authors were precisely involved selecting the articles for the study. These 9 articles which describe different agile maturity models are synthesized and the process is described in detail in the further sections.

3.1.2 Processing

This processing stage involves in identifying and extracting the essential data which is presented in the article. Processing stage involved in analyzing each document through studying the full text of the document and making notes of each document. For conducting an effective literature study the guidelines provided by J Rowley and F Slack in the article "Conducting literature reviews" were followed [42]. This is carried out in five sequential steps: Scanning documents, making notes, structuring the literature review, writing the literature review and building the bibliography [42].

- *Scanning documents:* The selected documents are carefully reviewed by both the authors and then they managed in grouping the documents with similar themes. This resulted in providing insights about the key themes related to agile maturity models which are essential for this study.
- *Making notes:* Each article is studied and all the essential information and data is noted through annotating and marking up the document. Here the aim and focus of the agile maturity models is identified. The structure and the components of the models are analyzed through identifying the agile practices from different maturity models. This information acts as backbone while answering the research question and so it was carefully reviewed.
- *Structuring the literature review:* This involved in identifying the key themes with the organization of concepts and documents together according to the actual research area. This helped in underlying the structure of different maturity models. An understanding of the maturity models with its set of agile practices is achieved through structuring the literature. Moreover, a defined purpose of the maturity models is gathered here.
- *Writing the literature:* An overview of each extracted maturity models is reported with a complete reference to that particular document. In this process of analysis, each maturity model with different maturity levels is

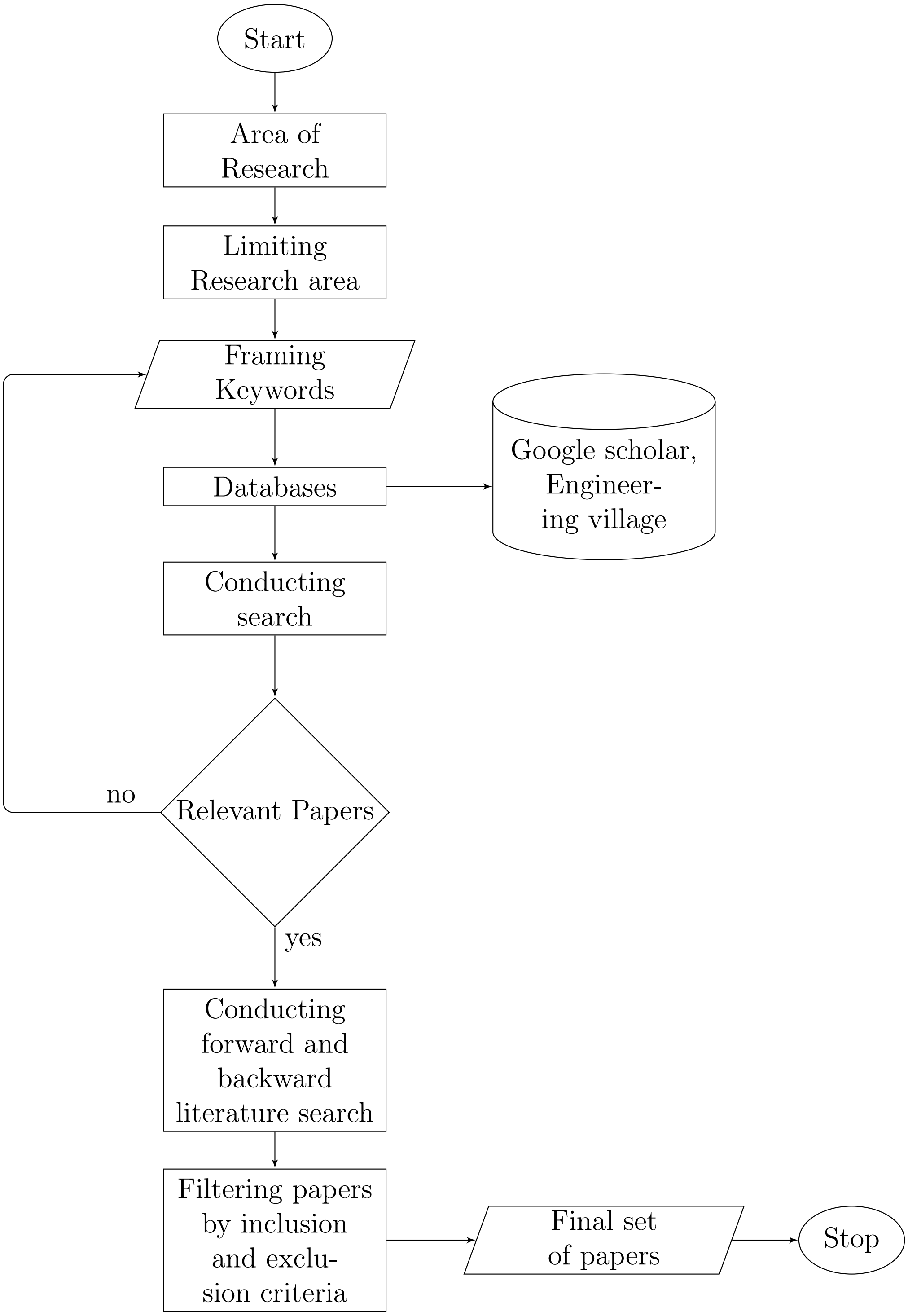

Start
Area of Research
Limiting Research area
Framing Keywords
Databases
Google scholar, Engineering village
Conducting search
Relevant Papers
no
yes
Conducting forward and backward literature search
Filtering papers by inclusion and exclusion criteria
Final set of papers
Stop

Figure 3.1: Search strategy

defined with its set of different agile practices. This provided a summary of required literature for the research field.

- *Building the bibliography:* All the credits to the selected articles are given by building the bibliography as they contributed to a new research. A bibliography is a list of all the sources that refer to in the literature review [42].

Through achieving all these above mentioned five steps it is clear that the process involved with literature review is almost complete. Triangulation was done in the best possible way. Both the authors were involved in performing the literature review. The material used for the study was kept in common between the authors and the work was done in parallel to identify potential results. Finally, the results achieved through the literature were reviewed by both the authors.

3.1.3 Output

The output of the literature review is presented with a clear academic style of writing with the logical structure of all extracted agile maturity models. All the nine agile maturity models included in this study are compared and presented. This document of the literature review provides the set of agile practices defined in each maturity model with respect to maturity levels of that particular model. A comprehensive overview of each maturity model is documented after the complete analysis of the model to uncover the inter-dependencies between the maturity levels. By performing these steps the overview of agile maturity models presented in the results chapter 4.

3.2 Survey

A survey is a strategy or design for an empirical study “to provide a quantitative description of some fraction of the population through collecting the data” [38]. Surveys are generally conducted as a representation of current or past situations.

Survey is opted as a part of research method for this particular research as this study aims at knowing the implementation of agile maturity models with respect to agile practices in the current IT industries. A quantitative research approach is chosen for this research as the quantitative data promotes comparison and statistical analysis [55].

Experiments are not chosen for this research as they are concerned with limited scope and most often they run in a laboratory setting [55]. So this type of approach for this research is not suitable. Moreover, experimentation objectives are to manipulate one or more variable and control all another variable at fixed levels [55]. This kind of approach will not help in gathering the required data i.e. retrieving of agile methods and practices from the current industry.

Whereas even case studies are also not suitable for this research because case studies are used for monitoring project, activities or assignment with an aim at tracking a specific attribute or establishing relationships between attributes [55]. The aim of this research is not confined to a specific agile maturity model rather it involves several models. Hence through performing a single case study it does not help in assessing all the nine maturity models considered for the study. Moreover this study does not focus on creating a new framework rather it involves comparing the agile practices implemented in the industries with literature findings. Hence, case study is not preferred in this case Post-mortem analysis can be performed for retrieving the current agile practices adopted in the industry. But post mortem is conducted by looking at project documentation [55]. It is not possible to gather the project documentation from industry as it is confidential post mortem analysis is ignored and is out of authors minds.

Implementation of agile methods in the current industry is vast and the survey has the ability to provide a large number of variables to evaluate. Moreover, the survey helps in collecting the data from a larger population from different geographic locations. Hence, survey questionnaire is used to achieve the objective of the RQ2.

Explorative surveys are used as a pre-study to investigate agile practices implemented in the current industry [40]. A professional questionnaire is created and the data is collected through a sample of the population from all over the globe. The main purpose of the questionnaire is to identify the order of practices of agile maturity models implemented in the current industry and facilitate the authors in identifying and understand the differences and commonalities with both the theoretical study and the exploratory study.

3.2.1 Rationale for survey

The rationale behind conducting the survey through questionnaire is to answer the second research question (RQ2). Several agile practices were identified while performing the literature review. These set of agile practices were incorporated into the questionnaire. This questionnaire is designed in an inclusive way to gather the data required even for the third objective by following the guidelines provided in [9]. This way of design helped in identifying and understanding the benefits and limitations of the order of practices currently implemented in the industry. Furthermore, the questionnaire helped in understanding the differences and commonalities between the literature study and current industrial experience in relation to agile maturity models. The main aim of this questionnaire is to extract all possible information from the respondents related to the agile practices that are implemented in a particular order.

Survey is opted as an empirical research method for this particular study because the survey can be administrated quickly and easily. Moreover, to identify the practices and activities from the current industry from different geographical locations within a short period of time the authors felt that survey serves as the best option for them. Also it helps in understanding the opinions of the software practitioners directly, involving wider population. Even from the respondent's side, the questionnaires gives them an option to share their own personal experience regarding the agile software development process.

3.2.2 Form of Data Collection

After the completion of literature, a self-administrated online web-based survey is conducted using Sosci survey (www.soscisurvey.de). This served as instrument conducting survey. For collecting the data, the sosci survey is used as it is convenient for use and also for distribution of the questionnaire to the respondents. Sosci survey is a free professional software package embedded different essential features like programmable filters, programmable layout, implementation of HTML and several others. These several vital features contributed towards creating a professional survey. The feature of automation in collecting the data provided by Sosci benefited both the respondents and authors in collecting the data without encountering any problems. So this form of data collection also provided flexibility and convenience for analysis. Hence, it is chosen as a means for collecting the data.

3.2.3 Population of the survey

The population involved in this particular study are current software engineers involved in agile development projects. The survey is conducted with the involvement of all functional groups from the developed companies. Convenience sampling was used other than probability sampling technique as the population chosen for this study involves the nearest and most convenient persons who act as respondents [40]. Most of the contacts involved with the survey were supervisor and authors' business contacts and so convenience sampling is adopted. For this survey, the experienced software practitioners are involved with the experience of agile development such as project managers, designers, developers, testers, analysts, etc. Respondents with agile experience and knowledge were only selected for answering the questionnaire to maintain the consistency in the quality of the responses, as this piece of valuable information is essential and crucial for further execution of the entire research. The respondents were contacted through email. With a superior request of the authors' contacts, some of the respondents have forwarded the questionnaire to their colleagues who are experienced with the agile development to answer the questionnaire.

Authors also managed in publishing the survey in the social groups like LinkedIn groups, Yahoo groups and Google groups related to agile software development. The questionnaire is published only in the groups related to the agile software development which involved groups from India, Sweden, Scotland, Spain, Belgium, Finland and the United States. The respondents were contacted through a defined email containing the survey link with a brief description and objectives of the questionnaire, providing them with the necessary contact details for further inquiries.

3.2.4 Survey Design

The questionnaire is focused on the topics related to adoption of agile practices, its benefits, and limitations. Each question is mapped to the respective research question and analyzed how well the developed question is able to answer the research goals. The questionnaire consists of 11 close ended and 5 open ended questions to answer the RQ2 and RQ3. Not all questions involved in the questionnaire are similar to each other, the format of the questions varies accordingly. The close-ended questions contain multiple choice questions whereas open-ended questions are provided with the text fields in which the respondents are requested to share his/her own experience in their own words expecting that this could provide a clear response to that particular question.

The survey questionnaire consists of three web pages. For the reasons of conciseness in the document the whole questionnaire is not presented here but the complete questionnaires can be found in the appendix A It can also be accessed online for a better professional experience when taking the survey. An overview of all the three webpages of the questionnaire is described below in separate sections.

- **Welcome Page:** In the first web page, a brief introduction about the survey is provided mentioning the non-disclosure statements of the respondent details. The contact details of the persons responsible for this particular questionnaire is included for any further inquiries.

- **First Page:** In this page, the questionnaire starts with a question related to the adoption of agile practices. A list of 17 core agile practices are listed in the first question, correspondingly to each agile practice, a timeline (refer appendix A) is provided which is created using javascript. The timeline has a pointer to mark the adoption of practices with respect to the time frame. A clear description of each agile practice is provided adjacent to the agile practice in the information tag for the convenience of the respondents. In the next part of the questionnaire, the respondents are requested to answer with reference to the agile practices adoption that they described in the previous question. Five open-ended questions are presented here which answers

  - The measures of success with agile adoption,
  - Limitations/ challenges faced during the implementing the agile practices,

  - Reasons for adopting the agile practice in that particular way of order and
  - Inquiries if any practice was terminated during the development process or not.

  Within this page, the data related to the adoption of agile practices is gathered. At the end of this page, a comment section is provided with an open text field for the respondent to add any additional information if they want to.

- **Second Page:** This second page is oriented with the research of this study including nine demographic questions. While entering to this page the respondent is requested to provide his/ her own background details regarding years of experience, roles, responsibilities and project characteristics. This page consists of 9 close-ended questions for the respondent to answer. These questions are related to the characteristics of their development teams, development type, industry domain and type of systems the respondent is experiencing or experienced previously. Also, the details of their distribution of team members are asked through a close ended question. This page completely focuses on retrieving the respondent's own experience with the adoption of agile practices.

- **Third Page:** And finally on the last page the respondent is optionally asked to add another experience. If the respondent does not want to add another experience, then he/she is asked to provide their contact details if they are interested in the survey results and for any further inquiries. Then the respondent is greeted with a vote of thanks for participating in the survey finally asking if they want to share anything else with us.

3.2.5 Piloting Survey

The questionnaire must be pre-tested before conducting the main survey to ensure whether the mentioned questions and inserted functionalities are functional, understandable and user-friendly or not. This helps the authors to find the difficulties faced by the respondents before conducting the main survey [18]. Initially a questionnaire is framed and discussed with the supervisor. Later from the discussions it was clear that the questionnaire design of the supervisor was completely addressing the research questions of this thesis. Hence for conducting this survey, the authors supervisor questionnaire was used since the questionnaire questions were completely answering the research questions of this thesis. Adding an advantage, the supervisor's questionnaire was already pre-tested. Hence, the authors of this thesis decided and further discussed with the supervisor regarding an additional question to add in her survey questionnaire. Finally the supervisor had provided the test link for the questionnaire. There after as suggested by the supervisor and the guidelines provided bt Kate Kelly [18] the pilot survey was conducted.

Three respondents were selected and the link was forwarded to them with an invitation email to participate in the survey. All the three respondents are well experienced with the agile software development methodologies. After a detailed analysis of the feedback given by the three respondents authors came to notice that the slider (refer appendix A) functionality which is included to answer the 1st question in the questionnaire was not functioning smoothly, overall every respondent commented that the questionnaire content was professional. After conducting the pilot survey, the questionnaire is validated by adding the information about the use of slider functionality in detail. Moreover, several enhancements were made to the code for the smooth running of functionality. Thus, the pilot survey helped the authors to finalize the questionnaire and publish the survey under the guidance of the supervisor.

3.2.6 Survey Execution

The survey is sent to the respondents through emails. With welcome note, a brief description of the survey and the estimated time to complete the survey is mentioned in the contents of the mail (refer appendix A). A log of all the respondents to whom the mail is distributed and the details of that particular respondent is maintained and updated frequently to avoid the reoccurrence of sending invitation mails to the same respondent again. The authors managed in publishing the survey link in the widely used social network websites namely LinkedIn, Yahoo groups and Google groups. The survey link is only posted in the frequently used groups related to agile methodologies. Above all, the respective supervisor also contributed in gathering the responses. The execution of the survey is described in the figure 3.2.

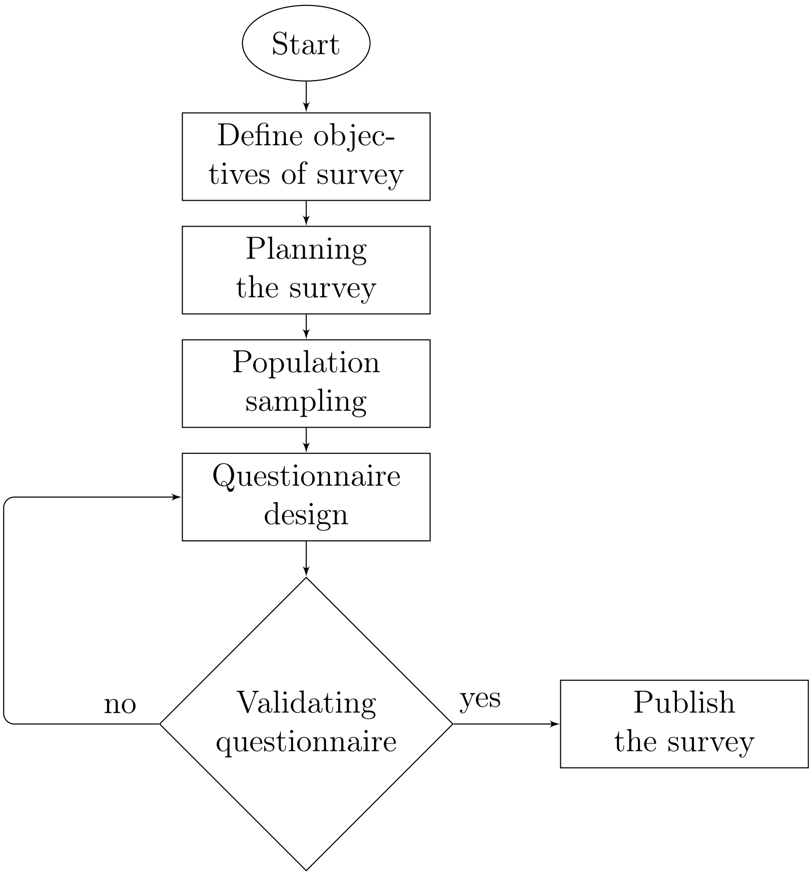


Figure 3.2: Survey execution

3.3 Mapping of Research Questions to Research Methodology

For answering each research question several sequential steps are followed as described in the following table with the mapping of the research methodology to the respective research questions.

Table 3.3: Mapping Research Questions to Research Methodology.

Research Questions	**Research Steps**	**Research Methodology**

1. RQ1	1.1. Identifying the agile maturity models present in the literature. 1.2. Identifying the order of practices from the each maturity model.	Literature review
2. RQ2	2.1. Identifying the order of practices implemented in the industry.	Survey
3. RQ3	3.1. Identifying the benefits and limitations in a certain order. 3.2. Differences and commonalities were gathered by comparing and analyzing both the results of literature review and survey.	Literature Review , Qualitative Data Analysis

3.4 Data Analysis

3.4.1 Narrative Analysis

Narrative analysis is a comprehensive narrative synthesis of previously published information [14]. So the extracted data from the articles found from the database search and through forward and backward search is subjectively analyzed through performing the narrative analysis. Both the qualitative and quantitative research can be analyzed through narrative analysis [35]. Narrative overviews, also known as unsystematic narrative reviews provide findings in a condensed format that typically summarize the whole content of each article. This utilization of narrative overviews provided a clear insight about each maturity model depicting the maturity levels with the set of agile practices within each level. The data extracted from the literature was analyzed using narrative analysis for achieving the O1 and also for answering the RQ1.

3.4.2 Statistical Analysis

For analyzing the extracted data from the questionnaire authors performed statistical analysis, as the obtained data is based on quantitative variables [44]. With the use of statistical methods, the quantitative data is analyzed to retrieve the order of agile practices implemented in the IT sector.

3.4.3 Comparative Analysis

Without comparisons, there is no complete fulfillment to any research. Qualitative data is useful in supplementing and illustrating the data obtained from the survey [40]. The qualitative comparative analysis supports with logical conclusions to the data set. There are many ways to conduct a comparative analysis through application of different logical techniques [40]. Comparative analysis for this study is conducted to discover the commonalities and differences between the results of survey and literature review to uncover the benefits and limitations of implementing the agile practices. This type of analysis is also carried out for further discussions and helped authors in framing new findings to the research.

3.4.4 Thematic Analysis

Thematic analysis is a method used to analyze the qualitative data and to report the patterns (themes) in the collected data [20], [54]. Thematic analysis is chosen to analyze the qualitative part of the study and it is one of the commonly used methods for analysis in empirical research. Authors performed thematic analysis in sequential steps as suggested in the article [20] and the steps are as follows

- **Transcribing:** The data collected is transcribed into a document in written format.
- **Organizing data:** The transcribed data is organized accordingly to analyze it for easy retrieval of the data.
- **Familiarization with data:** The data which organized are read carefully for a clear understanding of the data collected.
- **Coding:** Identifying the code is done by manually and carefully examining the data collected. A tag is assigned to each code for easy identification.
- **Generating themes:** After identifying the codes then the codes are categorized into themes. Later themes are labeled with a name for reporting the results.

3.4.5 Alternate analysis method

For both the quantitative and qualitative data there are several options to analyze the data. One of the suitable options to analyze the data would be grounded theory. According to Polit and Beck [16] “a generalization is an act of reasoning that involves drawing broad conclusions from particular instances” and they also espoused that knowledge is not gathered by testing a new theory but knowledge grows through confirmation [16]. There are several limitations with the grounded theory and in [16] reports that generalization is limited through grounded theory for the interpretation and analysis of data. Also, while performing grounded theory any prior consideration regarding the data shouldn't be made whereas for this particular research the extracted data is certain for its analysis and purpose. Hence grounded theory is not appropriate for this research.

Chapter 4

Results

4.1 Results from Literature Review

This document provides a detail comprehensive knowledge on nine agile maturity models published in recent years. Owing to the limited available resources on agile maturity models authors performed in-depth analysis and synthesis for presenting these nine agile maturity models. With a keen eye on the data extraction process, the overview of each maturity model is constructed based on the published structure of each maturity model with its defined set of agile practices. For each maturity model a respective ID is assigned for convenience in addressing the maturity model and is represented in the table 4.1.

Table 4.1: Assigned ID's for identified agile maturity models.

Model ID	**Paper Title**	**Model name**	**Author Name**	**Reference**
M1	Agile Maturity Model (AMM): A software process improvement framework for Agile Software Development practices.	Agile Maturity Model (AMM)	Chetankumar Patel and Muthu Ramachandran	[32]
M2	A Framework to support the evaluation, adoption and improvement of agile methods in practices	Agile Adoption and Improvement Model (AAIM)	A. Qumber and B. Henderson	[39]

M3	A Reference Model for Agile Quality Assurance: Combining Agile methodologies and Maturity Models	Agile Quality Assurance-Reference Model (Agile QA-RM)	Fernando Selleri Silva and Et.al	[48]
M4	A Structured approach to Adopting Agile Practices: The Agile Adoption framework	Sidky Agile Measurement Index (SAMI)	Ahmed Sidky	[47]
M5	Seven Dimensions of Agile Maturity in the Global Enterprise: A Case Study	Benfield's Model	Robert Benfield	[6]
M6	Scrum Maturity Model	Scrum Maturity Model	Alexandre Paulo Guo Yin	[57]
M7	The Agile Scaling Model (ASM): Adapting Agile methods for complex environments	Agile Scaling Model (ASM)	Scott W. Ambler	[3]
M8	A Road Map for Implementing XP	XP Model	Kim Man Lui and Keith C.C. Chan	[25]
M9	Towards Maturity Model for extreme Programming (XP)	The eXtreme Programming Maturity Model (XPMM)	Jerzy Nawrocki, Bartosz Walter, Adam Wojciechowski.	[30]

4.1.1 M1: Agile Maturity Model (AMM)

This model was developed by Patel, et.al. focusing on adaptability, sustainability and software maturity model [32]. This AMM is designed with the intention to improve, enhance and boost up the agile software development methodology through increasing the customer satisfaction, software quality, etc. Based on the agile principles and practices this model is designed with five maturity levels from initial to sustained level with predefined practices for each level.

1. **Initial:** The first level is named as "Initial" where there are no predefined goals or activities due to instability in the process of development.

2. **Explored:** The second level is defined as "Explored". The focal point of this level is on planning, requirements engineering, and customer satisfaction. At this level to give a good kick-start to the project an effective project planning, story card is driven development and involvement of the customer during the project for changes in requirements is followed in order to improve efficiency from the problems related to planning and requirements engineering.

3. **Defined:** At "Defined" level 3 the main focus is on the practices related to customer relationship management (CRM), frequent deliveries, pair programming, communication, coding, testing and quality of software[32]. The practices involved in this level are collective ownership, refactoring, frequent releases, coding standards, pair programming, and TDD.

4. **Improved:** At level 4 the maturity of the organizations is improved when comparing to previous levels hence it is named as Improved. At this point of time, the organizations are in a position to measure and control the software development process or practices and product quality with a main focus on the project management, working hours, risk assessment, self-organizing teams and problems related to the development team [32].

5. **Sustained:** Level 5 the highest level of this particular maturity model named as sustained mature level. At this level companies focus on performance management and defect prevention practices through quantitative feedback from the process and from testing innovative ideas and technologies [32].

After a detailed synthesis in this particular maturity model it is identified that the order of practices in this model helps the team in identifying and improving problems related to CRM, frequent releases, coding standards and testing [32]. But in especially at level 3 implementation of few essential practices like structured risk management, code optimization, and solving problems occurred due to the team is missing. While gaining maturity it aims at self-organizing teams and

Table 4.2: Agile Maturity Model

Model ID	**Maturity levelsMaturity levels**				
	Level-1: Initial	**Level-2: Explored**	**Level-3: Defined**	**Level-4: Improved**	**Level-5: Sustained**
M1		1. Planning game 2. On-site customer 3. Stories (Story card driven development)	1. Collective ownership 2. Refactoring 3. Short Iterations & Frequent releases 4. Coding standards 5. Pair Programming 6. Test Driven Development (TDD).	1. Tracking progress 2. Time boxing 3. Self-organizing teams 4. Continuous Integration	1. Time boxing 2. Continuous Integration 3. Story card with TDD

improving the efficiency of project management, maintaining the sustainable pace to accomplish the project. The structure of this maturity model is relevant to the CMMI levels with the implementation of various agile practices. Since implementing the CMMI model for software process improvement is still a challenging issue for organizations following agile methods. Thus, authors have developed this framework on agile development process with a focus on adaptability, suitability and process improvement.

4.1.2 M2: Agile Adoption and Improvement Model (AAIM)

An Agile Adoption and Improvement Model (AAIM) was developed by A. Qumber and B. Henderson and published as an article. A framework to support the evaluation, adoption and improvement of agile methods in practice, they summarized agile method as [39].

Agile method = f (agility, abstraction, people, process, product, tools, knowledge, governance)

In this agile adoption and improvement model authors have categorized the model into three agile blocks namely agile block: prompt, agile block: crux and agile block: apex. Where each level implements several agile practices throughout the software development process.

1. **Prompt:** It involves only one level called agile infancy the agile practices implemented in this level are iteration planning, TDD, and on-site customer [39]. Thereby this level is mentioned as Agile infancy.

2. **Crux:** While entering into the second block referred as crux it consists of three sub-levels. The first level is called as agile initial, it enables in establishing good communication and collaboration with both the customers and relevant stakeholders [39]. Whereas the next level termed as agile realization emphasizes on minimal documentation with the encouragement of verbal or face-to-face communication and tools. The final level of the crux is labeled as agile value aims at establishing the agile practices with a focus on development tools and practices [39].

3. **Apex:** The final block Apex contains two levels agile smart and agile process it aims at reducing cost while improving the production quality [39]. Agile smart involves a smart learning curve of software development, software process, software quality and new tools. The last level of this maturity model focuses on the establishment of the lean production environment to keep the process agile thus it is named as agile process [39].

Table 4.3: Agile Adoption and Improvement Model (AAIM)

Model ID	**Maturity Levels**				
	Level-1: Agile Infancy	**Level-2: Agile Intial**	**Level-3: Agile Realization**	**Level-4: Agile Value**	**Level-5: Agile smart** **Level-6: Agile progress**
M2	1. Iteration planning 2. Test-Driven Development (TDD) 3. On-site customer	1. face-to-face communication 2. Daily stand up meeting	1. minimal documentation	1. Self-organizing teams	

This AAIM model helps in measuring and assessing quantitatively with the degree of software agility through providing a systematic road map for implementing the agile practices to the software development environment [39]. This model provides a communication cooperation protocol in order to increase the efficiency of communication reducing the documentation. It comprises of mixed agile and traditional incremental development practices like product backlog, project planning, pair programming, product architecture, design, TDD, Coding, testing, and retrospective. The agile adoption and improvement model (AAIM) is validated empirically with one medium and one large sized organization in the area of embedded systems, network solutions, e-commerce systems, print solutions and multimedia production [39].

4.1.3 M3: Agile Quality Assurance-Reference Model (Agile QA-RM)

With several authors F. Selleri Silva, et.al combined agile methodologies and maturity models framed an Agile Quality Assurance- Reference Model (Agile QA-RM) [48]. This model consists of five maturity levels with 18 process areas similar to CMMI and MPS.BR structure. An overview of each level with defined agile practices is reported below with reference to the article [48].

- **Informal QA:** Initially, practices related to quality assurance (QA) like audits, monitoring and review are implemented in the level 1 i.e. Informal QA.

- **Managed QA:** At this level activity related to quality such as requirement analysis, testing and development are planned. This level is aligned with specific practices related to Quality Assurance Planning (QAP), Team Assistance (TEA), Processes Assessment (PCA), Non-Compliance Management (NCM), Product Assessment (PDA) and customer Satisfaction (CSA) are carried out to increase the satisfaction along the sprints [48]. The practices implemented at this level are planning game, on-site customer, daily meetings, coding standards, pair programming, iterations and sprint review meetings.

- **Defined QA:** The next level is Defined QA which includes the process areas and generic practices of PPQA namely Organizational Quality Assurance (OQA), Knowledge Management (KMW), Lesson Learned Management (LLM), Integration Management (ITM), Quality Assurance Quality (QAQ), Cost Analysis (CTA) and Risk Analysis (RKA). The practices implemented at this level are self-organizing teams, retrospectives, face-to-face communication, continuous integration, metaphor, sprint planning meeting and daily standup meetings.

- **Measure QA:** It comprises of application metrics responsible for enhancing the quality and development process or the product. It aims at Quality Assurance Measurement (QAM), Self-organization and Sustainability (SDS), overloads, schedule delays and costs increase. Practices like TDD, pair programming, self-organizing teams, and collective ownership are implemented at this level.

- **Optimized QA:** The level 5 aims at optimizing the process. In this level, the process areas deal with defect prevention (DFP) and Decision-Making Support (DMS) in a proactive way to minimize noncompliance and support in decision-making [48].

Table 4.4: Agile Quality Assurance-Reference Model

Model ID	**Maturity Levels**				
	Level-1: Informal QA	**Level-2: Managed QA**	**Level-3: Defined QA**	**Level-4: Measure QA**	**Level-5: Optimized QA**
M3		1. Planning game 2. On-site customer 3. Daily meetings 4. Coding standards 5. Pair programming 6.Iterations 7. Sprint review meeting	1. Self-organizing teams 2. Retrospectives 3. Face-to-face communications 4. Continuous integration 5. Metaphor 6. Sprint planning meeting 7. Daily meeting 8. Retrospective	1.TDD 2. Pair programming 3. Self-organizing teams 4. Collective ownership	1. On-site customer 2. Daily meetings

This maturity model showed similarities with Process and Product Quality Assurance (PPQ) with the CMMI. This model complies with CMMI and MPS.BR [48]. This model is not yet evaluated but is expected to evaluate with a set of

Brazilian companies.

4.1.4 M4: Sidky Agile Measurement Index (SAMI)

Sidky published an Agile Adoption Framework in his thesis "Sidky Agile Measurement Index" [47]. This Sidky Agile Measurement Index is described in 5 stage process which acts as a guide to the organization for software development process. Sidky has made an excellent attempt in relating the agile levels with the agile practices based on the agile principles. The set of practices listed with respect to each level is described below with complete reference of the article [47].

1. **Collaborative:** This level aims at intensifying the communication and collaboration in the software development process. The implementation of a set of agile practices in this level are retrospectives, planning game, self-organizing teams, coding standards, knowledge sharing, and on-site customer [47].

2. **Evolutionary:** The objective of this level is to enhance in collaborative work and delivering the software frequently. The set of practices involved at this level are short iteration, continuous delivery or frequent releases, sprint planning, tracking progress, simple design, customer contract reflective of evolutionary development [47].

3. **Effectiveness:** In this level, since the organization have already achieved effective communication and collaboration with frequent deliveries in the development process. The next objective at this level is to increase the efficiency and effectiveness of the development process [47]. Hence, this level is named as Effective with the adoption of several agile practices. The set of practices involved are risk driven iterations, product backlogs, metaphors, self-organizing teams, frequent face-to-face communication, continuous integration, refactoring, and unit test.

4. **Adaptive:** This level helps in adopting practices that help in stabilizing and automate the software development process [47]. This level the organization gather feedback from the customer to measure the correctness of the software product. Essential practices like on-site customer, iterative development, continuous customer satisfaction, frequent releases, adaptive planning, daily standup meeting, agile documentation, user stories and on-site customer [47].

5. **Encompassing:** In this level organizations accept the changes in the development process and maintain the agile nature. Seven essentials practices are implemented which ensure the highest level of maturity for the organizations through project estimation, low process ceremony, planning game,

implementation of TDD, pair programming in small teams, frequent face-to-face [47].

Table 4.5: Sidky Agile Measurement Index

Model ID	**Maturity Levels**				
	Level-1: Collaborative	**Level-2: Evolutionary**	**Level-3: Effective**	**Level-4: Adaptive**	**Level-5: Encompassing**
M4	1. Planning game 2. Self-organizing and cross functional teams 3. coding standards 4. Knowledge sharing tools 5. On-site customer 6. Retrospectives	1. Sprint planning 2. Tracking iteration progress 3. Short iteration 4. Continuous delivery or frequent releases 5. Simple Design 6. Customer contract reflective of evolutionary development	1. Risk driven iterations 2. Product backlogs 3. Metaphors 4. Self-organizing teams 5. Frequent face-to-face communications 6. Continuous integration 7. Continuous improvement (refactoring) 8. Unit tests	1. On-site customer 2. Iterations 3. Continuous customer satisfaction feedback 4. Frequent releases 5. Adaptive planning 6. Daily standup meetings 7. Agile documentation 8. User stories 9. On-site customer	1. Low process ceremony 2. Planning game 3. Test driven development 4. Pair programming 5. Frequent face-to-face interaction between developers and users (collocated)

Sidky proposed an Agile Adoption Framework consists of two components: Agile Measurement Index and 4-stage process. The Sidky Agile Measurement Index helps the organizations in measuring the agile potential. The 4-stage process helps the organizations in avoiding the conduct of unnecessary activities. This process is structured into four stages namely [47].

1. Identifying Discontinuing Factors

2. Project Level Assessment

3. Organizational Readiness Assessment

4. Reconciliation

Through framing a questionnaire with a total of 28 participants the goodness and effectiveness of the framework are validated.

4.1.5 M5: Benfield's Model

Benfield identified seven dimensions that are inconsistent to many of the teams. He published a new framework in the article "Seven Dimensions of Agile Maturity in the Global Enterprise: A Case Study" and named it as Benfield's model [6]. In this article, the author has considered seven dimensions to define the practices for each maturity level. The seven dimension are automated regression testing, code quality metrics, automated deployment and back out, automated builds and configuration management best practices, interlocked and interface integration testing, TDD, performance, and scalability testing. The five maturity levels which were defined based on the seven dimensions and also influenced by the CMMI, they are named as.

1. **Emergent Engineering Best practices:** This is the first level of the maturity model and the practices involved in this are unit testing, code reviews, repeatable builds, configuration management, code quality, and TDD. These are the practices implemented, valued and practiced by the team [6].

2. **Continuous practices at component level:** The level 2 is a repeatable rhythm of level 1. A deep understanding of previously introduced practices is made with the implementation of the practices. Reusable automation, interface testing, scalable testing is typically introduced. The unit harness is introduced with the implementation of the TDD in the implementation phase. The other practices implement at this level are user stories, metaphors, and tracking progress. The operation of previously introduced practices named this level as continuous practices at component level [6].

3. **Cross Component Continuous Integration:** This level focus on the robustness between a component with a flow through regular synchronized build/ test cycles across interface boundaries [6]. Here the teams are aiming for a better synchronize work in order to improve the level of maturity and automation. The practices involved at this level are continuous integration, collaborative teams.

4. **Cross Journey Continuous Integration:** It composes of more sophisticated injection test harness and instrumentation This level exhibits within XP teams and ever shrinking end to end test cycles are implemented to increase the user experience and quality of the product [6]. Powerful tools

are used to reduce the operational cost at the same time enhancing the service levels. Some of the practices implemented at this level are refactoring, frequent releases, iterations User Acceptance Testing.

5. **On Demand Just in Time Releases:** This level focuses on code refactoring and it is implemented to improve quality, supportability and reuse [6]. As the teams are highly productive an SOA based model is introduced for an effective risk assessment with reinforcement in planning for frequent delivers of the working software.

Table 4.6: Benfield Model

Model ID	**Maturity Levels**				
	Emergent engineering practices	**Continuous practices at component level**	**Cross component continuous integration**	**Cross journey continuous Integration**	**On demand just in time release**
M5	1.Coding standards 2.Automated regression testing 3. TDD 4. Unit Testing 5. Continuous Integration	1. User stories, 2. Metaphor 3.Tracking progress	1. Continuous integration, 2. Collaborative teams. 3. Frequent releases	1. Refactoring 2. Frequent releases, 3. Iterations 4. UAT testing	1. Refactoring 2. Self-organizing teams

Benfield dimensions were defined as Automated testing, code quality metrics, automated deployment and backout, automated builds and configuration management, interlocked delivery and interface integration testing, TDD and performance and scalability testing. These seven dimensions' drive towards significant quality and velocity improvement across the development process [6]. Most of the practices involved in this model are implemented with effective team collaboration and continuous integration. Benfield's model targets the quality if a product with vigorous different types of testing. An analysis showed that this model is not suitable for Commercial Off the Shelf (COTS) products and for the business side as this model evolved with the focus on engineering. Since the case study is only conducted at British Telecom company it lacks enough validation.

4.1.6 M6: Scrum Maturity Model

Scrum Maturity Model is proposed with five levels for the scrum development methodology by Alexandro Paulo Guo Yin. This model is evolved after several iterations in the research. A brief description of the model is presented in the below mentioned five levels.

1. **Initial:** This is the first level of the model which is named as initial. In this level the organization figure out the goals for the development process addressing the issues related to overtime, over budget, poor communication among stakeholders and unsatisfactory quality of the final product [57].

2. **Managed:** The practices performed at this level are more structured and complete when compared to level 1. Practices related to basic scrum management and software engineering management are implemented like sprint planning meetings, product backlog, project tracking, and daily standup meetings. A clear definition of the product owner is provided with defined roles, responsibilities and products vision [57].

3. **Defined:** This level mainly focuses on the customer relationship management to maximize the communication and collaboration with the customer and on iteration management to deliver the product on time. Several practices were adopted to achieve the goals defined at this level are an on-site customer, team estimate, daily standup meetings, metaphors/ sprint backlogs, TDD, tracking progress, continuous integration, and stakeholder feedback [57].

4. **Quantitatively Managed:** This level focuses on the standardized project management and the process performance management. Practices related to the project management falls under this level. The set of practices involved in this level are metaphors, project tracking, daily standup meetings, sprint planning meeting, on-site customer, team estimate, continuous integration [57].

5. **Optimizing:** In order to gain the level of maturity the organizations need to optimize their scrum maturity model. This level focuses on the performance management activities. At this level organization manage to improve the performance of the teams and customer satisfaction. This level aim is to measure and analyze their own set of actions to enhance their development process with benefits. The practices related to performance management falls under this level like time boxing, short iterations and retrospectives [57].

Table 4.7: Scrum Maturity Model

Model ID	**Maturity Levels**				
	Level-1: Intial	**Level-2: Managed**	**Level-3: Defined**	**Level-4: Quantitatively Managed**	**Level-5: Optimizing**
		1. Metaphors, 2. Project tracking, 3. Daily standup meetings, 4. Sprint planning meeting/ retrospectives	1. On-site customer, 2. Team estimate, 3. Daily stand up meetings, 4. Metaphor/ sprint backlogs, 5. Continuous integration, 6. Tracking progress	1. Metaphors 2. Project tracking, 3. Daily standup meetings, 4. Sprint planning, 5. On-site customer, 6. Team estimate, 7. Continuous integration	1. Time boxing 2. Retrospectives 3. Short Iterations.

This model was inspired by CMMI process area with the mapping of scrum practices. To monitor the assigned scrum practices at each level a set of metrics are suggested. This model is evaluated by Oscan-Top and Demirors through performing a case study in an organization. They both mentioned that the organization had reached the maturity level 2: Managed during their research [31].

4.1.7 M7: Agile Scaling Model (ASM)

Agile Scaling Model is a framework to reduce the hindrances faced by the team at an organizational level, which was developed by Scott W. Ambler [3]. ASM has three maturity levels consisting of the tailored agile practices. A brief description of the maturity levels is discussed below [3]:

1. **Core Agile Development:** The practices related to core agile development methodologies like Scrum, XP, and Agile modeling like daily standup meetings, requirements envisioning and self-organizing teams are the practices implemented at this level [3].

2. **Disciplined Agile Delivery:** In this level the practices which come under disciplined agile delivery process, DSDM and some of the core agile methodologies like self-organizing teams, on-site customer, continuous integration, daily standup meetings, metaphors, refactoring and TDD are implemented at this level [3].

3. **Agility at Scale:** This level also focuses on the disciplined agile delivery in addition to the application of one or more scaling factors [3]. The practices involved in the level 2 comes under this level.

Table 4.8: Agile Scaling Model

Model ID	**Maturity Levels**		
	Level-1: Core Agile Development	**Level-2: Disciplined Agile Delivery**	**Level-3:Agility at scale**
M7	1. Daily standup meetings. 2. Requirements envisioning 3. Self-organizing teams	1. Self-organizing teams 2. On-site customer 3. Continuous integration 4. Daily standup meetings 5. Metaphors 6. TDD. 7. Refactoring	1. Self-organizing teams 2. Daily standup meetings 3. Requirements envisioning 4. Face-to-face communication 5. On-site customer. 6. Pair programming 7. Refactoring. 8. Continuous Integration. 9. Collective ownership

This Agile Scaling Model focuses on a disciplined agile delivery life cycle addressing the full delivery process from project initiation to deployment into production [3]. Authors state that eight scaling factors that the team faces complexities are dependent on [3].

1. Team size

2. Geographical Distribution

3. Regulatory Compliance

4. Organizational Distribution

5. Technical Complexity
6. Domain Complexity
7. Organizational Complexity
8. Enterprise Discipline

This model shows some controversies as this model focuses on the integration of agile methodologies rather than process improvement. This proposed model lacks the enhancements on the quality of development process.

4.1.8 M8: eXtreme programming (XP) model

Lui and Chan designed a road map for implementing XP from Hong Kong [25]. Their purpose of the road map is to facilitate the learning of inexperienced teams by providing a clear picture of the relationship among XP practices [25]. Their primary vision is to provide a clear idea of twelve XP practices sequentially in a four staged process. Basing on their mathematical mapping studies and according to TDD their framework is described a complete phase road map, they are [25].

1. **Level-1:** In the initial stage of the maturity model, the practices involved are testing, simple design, refactoring, iterations and coding standards are implemented.
2. **Level-2:** Coming to the second level of the maturity model they added continuous integration.
3. **Level-3:** In the next stage they introduced pair programming and collective ownership and mentioned that pair programming involves partner rotation and collective ownership are closely connected [25].
4. **Level-4:** In the final stage they introduced the metaphors, 40-hour week, small releases, on-site customer, and planning game [25].

Table 4.9: eXtreme programming Model

Model ID	**Maturity Levels**			
	Level-1	**Level-2**	**Level-3**	**Level-4**
M8	1. Testing 2. Simple design 3. Refactoring 4. Coding standard 5. Iteration	1. Continuous integration	1. Pair Programming 2. Collective Ownership	1. Metaphor 2. Iteration. 3. Small release 4. On-site customer 5. Planning game

In this way, the authors suggested the inexperienced teams implement the XP for their development process. They achieved in facilitating the inexperienced teams through providing a clear description of relationships among the XP practices. The limitations identified with this method were this model does not define the timeline required for each phase neither it does not suggest the software team in how many phases the team needs to adopt all the set of practices in XP practices. But they achieved in providing a clear insight in describing the relationships among the twelve XP practices.

4.1.9 M9: The eXtreme Programming Model

Nawrocki, Walter and Wojciechowski from Poznan University, Poland proposed an eXtreme Programming Maturity Model (XPMM). This model is based on four levels in which the initial level is not compliant at all while the next level implements specific XP practices related to CMMI namely acceptance tests and the planning game. This level is oriented towards the project teams with a focus on two key process areas. Customer relationship management and product quality. The practices involved in level two are planning a game, user stories, release planning, velocity measurement, iteration planning, system metaphor, unit test and acceptance test [30]. Level-3 Advanced focus on pair programming where one-person act as a coding leader and other as a testing leader. Practices involved in level 3 are completely related to pair programming like collective code ownership, checking coding standards, automated testing, and frequent code integration [30]. While entering into the next level 4-Mature, it addresses issues related to customer's and developer's satisfaction. With the focus on project performance [30]. It involves the on-site customer, no overtime, achieving coding standards before release and customer complete satisfaction regarding the developed product [30].

This XPMM model simulates the CMMI and help in comparing the “real” and “pseudo” XP project. Their proposed model does not encourage the need for written documentation. During their research five project teams have started validating their proposed model believing in achieving highest possible competence in XP.

Table 4.10: The eXtreme programming model

Model ID	**Maturity Levels**			
	Level-1: Not compliant at all	**Level-2: Initial**	**Level-3: Advanced**	**Level-4: Mature**

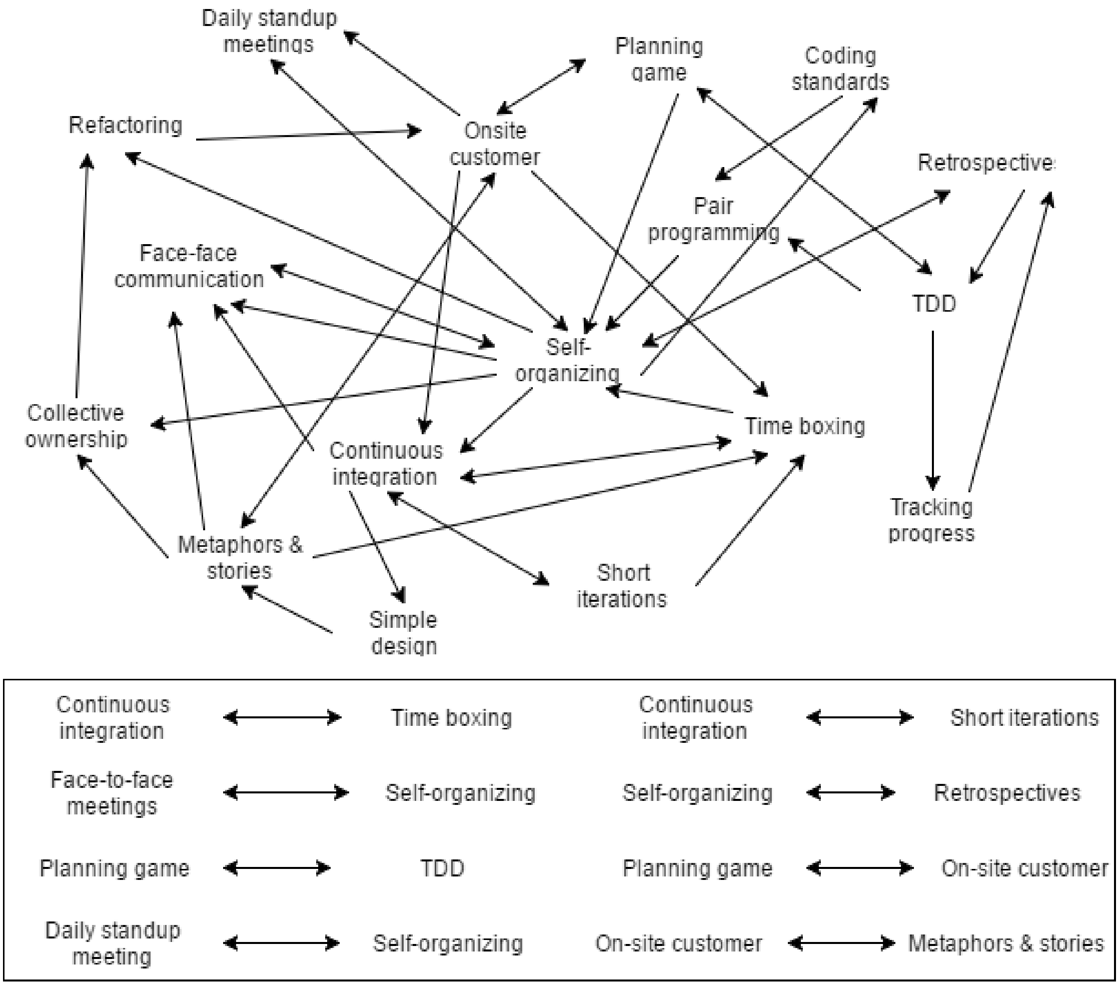
M9	1. Planning game 2. User stories 3. Release Plan 4. Frequent small releases 5. Iterations planning 6. Metaphor 7. on-site customer 8. Unit testing 9. Integration 10. Optimization 11. Acceptance test	1. Pair programming 2. Coding standards 3. Collective code ownership 4. Coding integration 5. Automated test for Integration tests	1. On-site customer 2. Customer satisfaction	

4.1.10 Analysis of Literature Review

The extracted data from each model is analyzed. If observed in each maturity model they are structured with different number of levels. M7 is with three levels, two of them M8 and M9 are with four levels, M2 with six levels and others with five levels. Hence, each maturity model consists of different levels and are subjected to different aims and objectives with respect to the agile practices aligned into those respective levels. The agile practices embedded in these maturity models were repetitive with one another irrespective of levels.

Few commonalities could be observed regarding the implementation of agile practices across different agile maturity models. Practices like planning game, on-site customer, metaphor, daily stand ups are suggested at the initial levels in the models M1, M2, M3, M5, M6, M7 and M9. Practices like collective ownership, re factoring, continuous integration are suggested in level-3 in the models M1, M4, M5, M7, M9.

The authors made an attempt in comparing each individual practices to observe their pre implemented and post implemented practices. For instance the practice on-site customer is suggested prior to daily stand-up meetings in M3, M6 and M7. Similarly the practices which reinforce each other were linked together with arrows as shown in the figure 4.1. The results indicate that the practice at the arrow head is suggested prior to the practice that is at the initial point. For example, if observed the figure 4.1 indicates that self organizing is suggested before collective ownership from the scientific literature.



Continuous integration	⟷	Time boxing	Continuous integration	⟷	Short iterations
Face-to-face meetings	⟷	Self-organizing	Self-organizing	⟷	Retrospectives
Planning game	⟷	TDD	Planning game	⟷	On-site customer
Daily standup meeting	⟷	Self-organizing	On-site customer	⟷	Metaphors & stories

Figure 4.1: Agile practices

From the nine extracted maturity models few of the practices like on-site customer and metaphor were suggested one after another and vice versa. So these practices are represented with a double headed arrow. Eight such combinations were retrieved from the nine maturity models as shown in the figure 4.1. The combinations are Continuous integration and Time boxing, Face to Face communication and self-organizing. Planning game and TDD, Self-organizing and Retrospectives,Daily standup meetings and self-organizing teams,Planning game and on-site customer, On-site customer and metaphors, Continuous integration and short iterations.

4.1.11 Summary

From the above discussions it implies that an ample research was made in knowing the agile maturity models. This research provided nine maturity models with several findings regarding the implementation of agile practices. From the literature it is clear that different models suggest differently regarding implementation of agile practices at different levels of maturity. Each model is structured with different aim and focus on different aspects.

However, literature does not answer on how industries actually adopt the agile practices. This can be known through identifying the implementation of agile practices in the industries through contacting software practitioners. And this can also result in finding the benefits and limitations of implementing the agile practices. The understanding of the results could bring efficient solutions and help the software practitioners in saving time and in improving efficiency of the development process.

4.2 Results of the Survey

4.2.1 Analysis of Demographic questions

This current section aims at findings from the questionnaire to answer the demographic information of the respondents and all the essential information needed for conducting this research i.e. regarding the implementation of agile practices. All together about 75 invitations emails had been sent to the current software practitioners to answer the questionnaire in which 16 invitations were posted in different social networking groups namely LinkedIn, Yahoo groups and google groups that are related to agile methodologies to maintain consistency with the responses. By the end of the survey period, data has been collected from 52 individuals, out of which 6 individuals for some particular reason couldn't complete the questionnaire. Hence these responses were not considered for further analysis of the data as the information provided by the respondents is unreliable. Hence out of 52 responses, only 46 individual responses were considered for analyzing the results. This indicated 61.33% completion rate of the questionnaire which means the overall responses to the questionnaire was very positive and is also sufficient enough as indicated in [8]. This overview of the responses showed that the quality of the questionnaire was satisfied and understandable to the respondents. Finally, these 46 responses were further carried out to perform the analysis of the data.

4.2.2 Roles of the Respondents

While performing a survey, the target population consists of different individuals each performing different roles and responsibilities. Therefore, the respondents were asked to mention their roles and responsibilities in their development process. The collected data of roles and responsibilities are analyzed and almost quarter of the respondents (24.56%) were scrum masters in different projects. About

22.8% of the respondents belong to programmers which included the developers and testers. Both the product owners and the program managers constitute of 10.53% individually from the overall respondents. 8.77% of the respondents are trainers whereas 7.02% of them were department unit heads. Each respondent possessing roles of system architect, consultant, and quality assurance had contributed 1.75% individually to the overall responses. 10.53% of the respondents are system analyst, c-level manager and others. A detail statistics of the roles of the respondents are presented in figure 4.2.

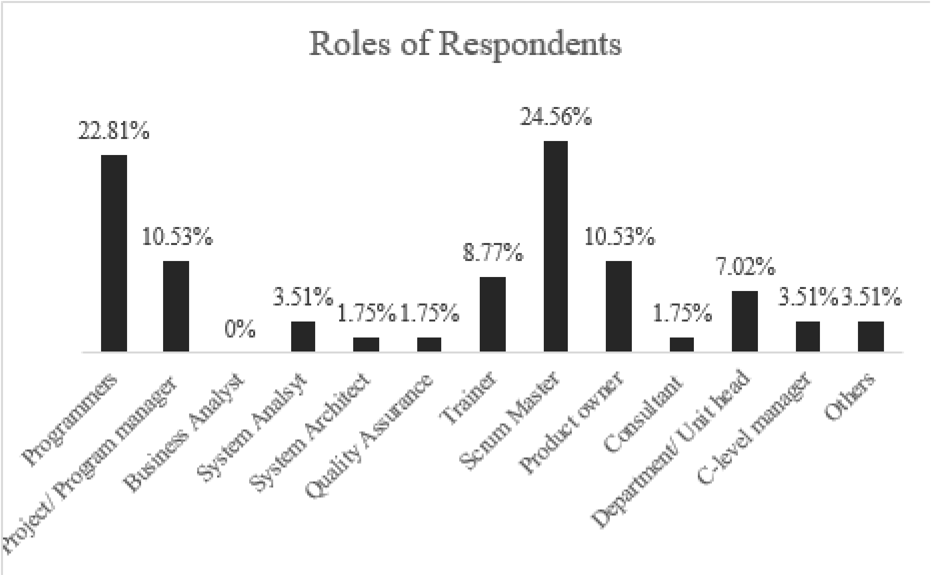


Figure 4.2: Roles of the Respondents.

4.2.3 Work Experience of the Respondents

The work experience of each respondent with the agile development is asked to report in the survey. As this is one of the key factor to examine the quality of the responses. Moreover, this research is focused on the agile practices and agile development so experience with the agile development is essential hence the respondents were asked to indicate their work experience with agile development methodologies. The analysis of the work experience of the respondents showed that 33.33% of the respondents are well experienced with 6-10 years of agile development. 22.8% of the respondents have 1-3 years of experience. About 17.54% of the respondents are experienced between 3-6 years with the agile development. 15.79% of the respondents are professionals with the experience of agile development for more than 10 years. Whereas only 10.53% of the respondent's population have experienced less than 1 year. This implies that almost two-thirds of the participant (66.66%) have experience for more than 3 years with the agile software development. Hence, based on work experience of the respondents the overall response to this questionnaire was satisfactory. For a clear overview, the work experience of the respondents is presented in figure 4.3.

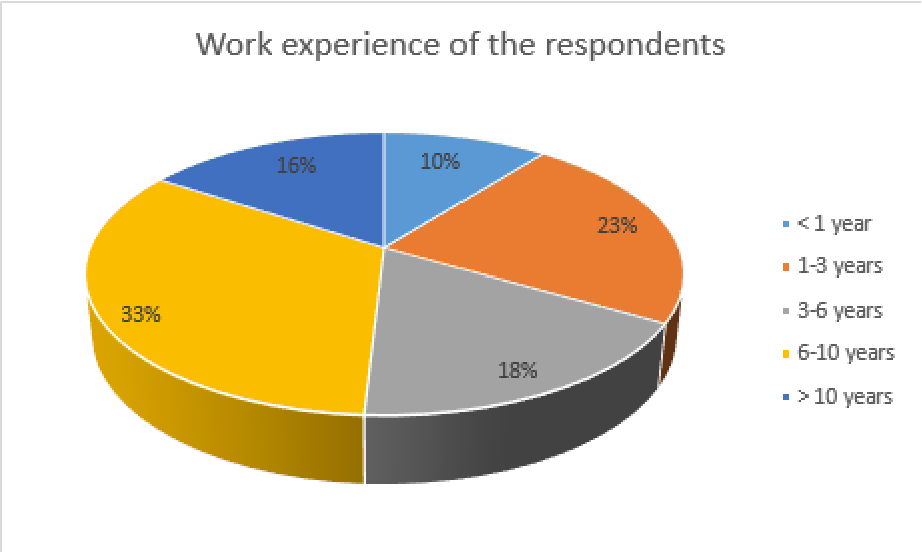


Figure 4.3: Work Experience of the Respondents.

4.2.4 Industry domain of the respondents

Respondents were asked to provide their industry domain in the questionnaire. Figure 4.4 represents the industry domains of the responses about 33.33% of the respondents are from independents software vendors. The second largest group is telecom with 22.81%. The third largest group of the respondents are from research and development with 10.53%. Both the media entertainment and financial services contributed 8.77% individually. These were followed by the health care 7.02%, government 3.51%, military 1.75% and other 3.51%. For this particular research, it is mainly focused on the agile development process and most of the data is collected from the industry domains related to independent software vendors, telecommunication, and research and development so this analysis provides that most of the collected data are reliable for this research.

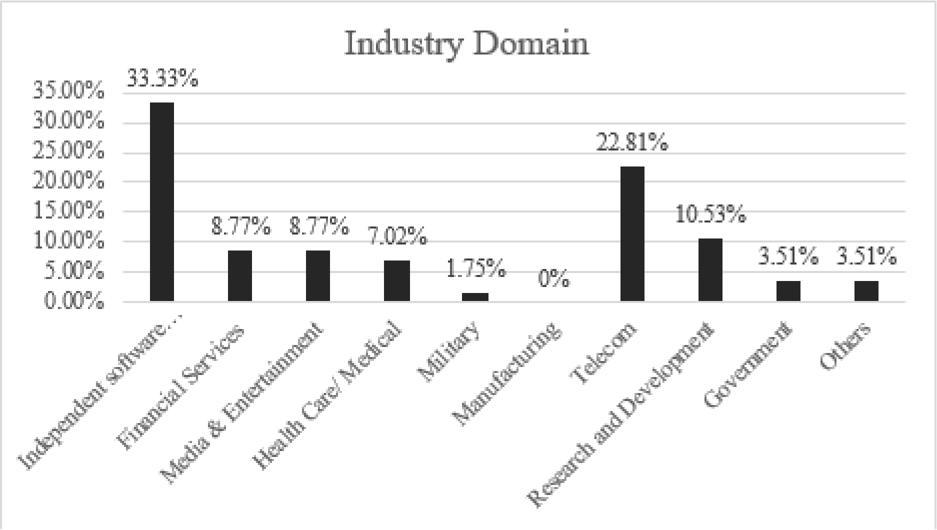


Figure 4.4: Domain of Industries.

4.2.5 Distribution of the team members

Respondents were asked to report their experience with the distribution of their team members. The below presented figure 4.5 represents the responses of the respondents regarding their team distribution.

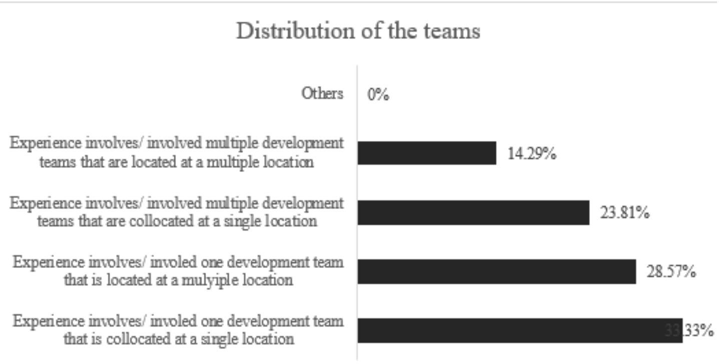


Figure 4.5: Distributed teams.

4.2.6 Development types of the industry

Respondents were asked to report the development type of the industry. The type of development during their experience. Market driven for the large open market has received 35.71% of all the responses being the highest. A detail description of all the development types of the industries of the respondents is represented in figure 4.6.

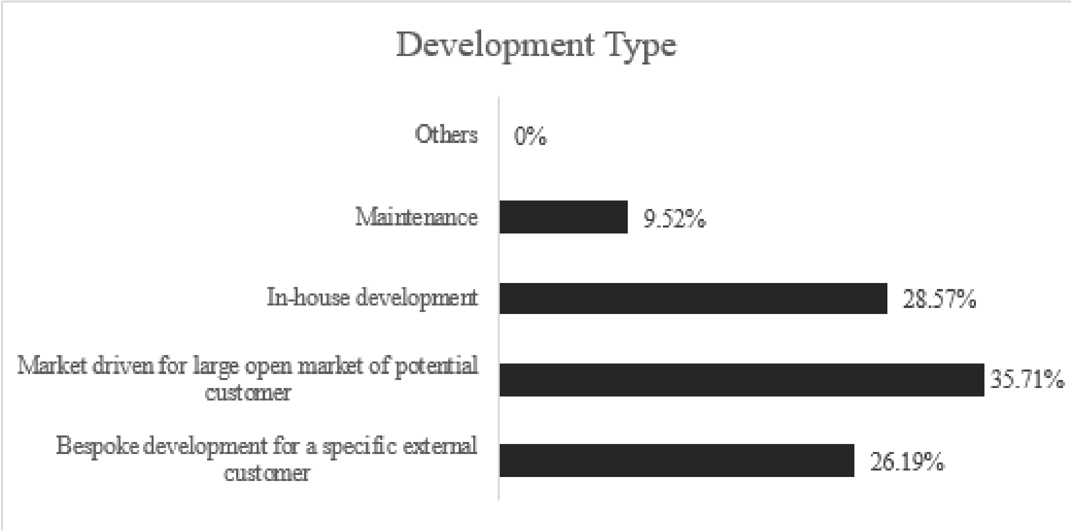
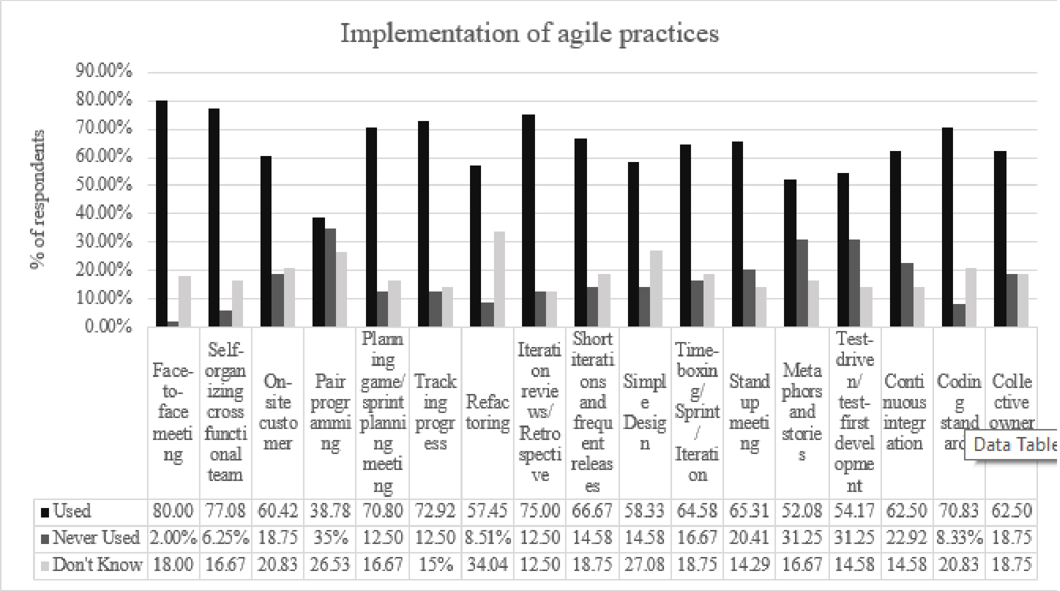


Figure 4.6: Development type of the industry responded by the respondents.

4.2.7 Usage of Agile Practices in the Industry

The responses of the respondents were analyzed to determine the frequency of the use of agile practices. Based on the collected data the response rate of use of each individual practice is calculated and presented in figure 4.7.



	Face-to-face meeting	Self-organizing cross functional team	On-site customer	Pair programming	Planning game/ sprint planning meeting	Tracking progress	Refactoring	Iteration reviews/ Retrospective	Short iterations and frequent releases	Simple Design	Time-boxing/ Sprint / Iteration	Stand up meeting	Metaphors and stories	Test-driven/ test-first development	Continuous integration	Coding standard	Collective owner
■ Used	80.00	77.08	60.42	38.78	70.80	72.92	57.45	75.00	66.67	58.33	64.58	65.31	52.08	54.17	62.50	70.83	62.50
■ Never Used	2.00%	6.25%	18.75	35%	12.50	12.50	8.51%	12.50	14.58	14.58	16.67	20.41	31.25	31.25	22.92	8.33%	18.75
■ Don't Know	18.00	16.67	20.83	26.53	16.67	15%	34.04	12.50	18.75	27.08	18.75	14.29	16.67	14.58	14.58	20.83	18.75

Figure 4.7: Implementation of Agile practices mentioned by the respondents.

The data is extracted from the timeline provided in questionnaire. This is analyzed to understand the implementation of agile practices in industries. The extracted data from each respondent shows that the agile practices are implemented in distinct order in their respective timeline. Hence both the authors were cautiously involved while performing the analysis. The analysis was performed through inspecting each individual agile practice. With this the practices, that are implemented before and after with a particular practice can be determined. The analysis and the results of each individual practice is tabulated in table 4.11. For instance if observed for practice P1, 78% of the respondents implemented P2 after P1 i.e self-organizing and Face-to-face communication.Similarly 74.18% of the respondents implemented P3 after P2 i.e. on-site customer and self-organizing.

Table 4.11: Analysis of practices.

	Agile Practices	Total Used (%)	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17
1	P1	80	-	**78.03**	4.62	5.57	0	0	0	0	0	0	0	3.7	0	3.5	0	1.23	3.35
2	P2	77.08	0	-	**74.18**	6.6	0	0	0	0	0	0	0	5.49	0	6.23	0	4.2	3.3
3	P3	60.42	2.58	4.64	-	0	0	0	0	2.54	4.23	0	**53.22**	4.7	5.4	15	0	7.69	0
4	P4	38.78	0	0	0	-	0	0	0	12.29	0	0	9.25	0	18.63	12.7	23.48	23.65	0
5	P5	70.83	0	0	**71.05**	0	-	0	7.8	0	0	5	2.6	0	0	0	8.23	5.32	0
6	P6	72.92	0	3.8	0	0	0	-	0	0	22.3	0	0	13.9	0	0	0	**60**	0
7	P7	57.45	0	0	0	9.78	0	0	-	0	0	0	0	0	0	0	0	0	0
8	P8	75	0	1.3	8.2	0	0	3.42	0	-	4.79	0	0	**74.63**	0	0	7.66	0	0
9	P9	66.67	0	0	0	0	11.21	2.4	0	1.9	-	0	0	1.49	**83**	0	0	0	0
10	P10	58.33	6.4	0	0	8.2	0	**68.16**	0	0	0	-	0	0	0	11	0	6.24	0
11	P11	64.58	0	0	0	9.18	**74.03**	0	0	0	14.64	0	-	0	2.6	0	0	0	0
12	P12	65.31	0	0	0	0	0	4.33	0	0	0	0	13.04	-	0	0	**82.63**	0	0
13	P13	52	0	0	0	15	0	0	0	0	0	62.3	6.12	0	-	16.58	0	0	0
14	P14	45.83	0	0	0	0	0	0	33.6	21.96	32.37	0	0	0	8.9	-	0	3.17	0
15	P15	62.5	0	0	0	5.3	6.23	4.33	0	2.12	0	4.46	**73.56**	0	0	0	-	4	0
16	P16	70.83	0	0	0	0	5.32	0	0	**63**	0	0	0	17.3	0	12.02	2.36	-	0
17	P17	62.5	0	2.33	0	5.39	1.2	11.69	0	0	1.17	0	5.62	0	2	0	**70.6**	0	-

Looking at the results of the survey, the following deductions have been made. The practices which indicated highest percent of implementation are combined together as a pair since the results indicate that these two practices reinforce each other in agile development process as per the software practitioners. Then the combined pairs of agile practices are then compared with the results of scientific literature to observe the commonalities and differences. With these findings an assessment on agile maturity models is performed on how effectively these agile maturity models are helping the software practitioners. The analysis of survey results from table 4.11 resulted in nine pairs of agile practices namely

1. P1 and P2 (Face-to-face communication and self-organizing teams)
2. P2 and P3 (Self-organizing and On-site customer)
3. P3 and P5 (On-site customer and Planning game)
4. P8 and P12 (Iteration Review and Standup meetings)
5. P9 and P13 (Short Iterations and Metaphors & stories)
6. P11 and P5 (Time boxing and Planning game)
7. P12 and P15 (Standup meeting and Continuous Integration)
8. P15 and P11 (Continuous integration and Time boxing)
9. P17 and P15 (Collective ownership and continuous Integration)

4.2.8 Results from open-ended questions

The first open ended question provided in the questionnaire to the respondents was regarding the measure of success with their experience in the agile development process. This particular question was intended to gather the benefits and success of implementing agile practices in their organization. Most of the respondents responded the measure of success in a positive manner. 80.44% of the respondents reported that their experience with the implementation of the agile practices was successful (refer figure 11). A variety of perspectives on the measure of success were expressed by the respondents few of them were:

"Measure of success=5(1-5) with good team collaboration quality product can be delivered. Collective ownership contributed to project success".

"Better quality and happier staff with increased productivity, reduced defects thereby reducing time spent on defects. Agility was my main attraction to the methodology".

"Delivering working software more quickly with the happier team which reduces the stress/ pressure on the team. Fewer defects to fix with reduced emphasis on

deadlines so the team enjoys build software that clients love. On a scale of 1 to 10, I can give 8/10".

"Value delivered; stakeholder satisfaction; the great rate of learning with minimal product issues; financial success with on-time delivery; a measure of success maximum extent".

Similarly, most of the respondents have reported their measure of success with respect to their agile development process. All the responses of the respondents were analyzed and the benefits regarding the use of agile practices are reported below:

- Enhanced coordination and communication between the team members.
- Decrease in production issues.
- Able to adopt changes frequently with on-time delivery.
- Transparency in the development process for all the stakeholders.
- Focus on both the customer and business value.
- Improved quality with reduce in cost.
- Flexibility and adaptability in the development process.
- Frequent releases of the developed product after each iteration.
- Implementation of code refactoring.
- Knowledge sharing with the implementation of daily status meetings.
- Early access to test the software.
- Minimal documentation.
- Constant feedback on products and process for better enhancements.
- Better risk management.
- Better control over the development process and project.
- Velocity measurement and control charts help team to develop.

Correspondingly, the respondents were asked to report the limitation/ challenges faced while adopting the agile practices. This question was intended to gather the limitations/ challenges faced while adopting the agile practices in their software organization. From all over the responses the most common challenges faced by the practitioners are identified through performing a detail investigation on the responses and these challenges faced by the practitioners. A variety of perspectives on limitations/ challenges were expressed by the respondents some of them are described below:

*"Agile methods shift focus towards a true collaborative "team" as opposed to a group of individuals working on independently on parts of the same project. This creates a need for a broader "soft skills" set around key areas such as communication and teamwork. Some software development companies have invested heavily as a part of their team's professional development. Pure technical skill is not the *only* measure for a high performing agile team member, and these skills takes time to develop".*

"Getting the agile mindset into the organization is a bigger challenge. Sometimes documentation is needed when you are working on support project whereas in agile documentation scope is limited. Team members who are not familiar with the agile practices take longer learning time".

"Most of the time is spent on "agile administration", such as debating on whether to have an electronic board or not. Describing stories / tasks often fail, because the short descriptions are misunderstood by team members. Continuously improving the agile process is complex, especially if there is no dedicated scrum master in the team. The official scrum master is also a developer and spends most time developing the code. Occupying testers / business users within the team at the initial stages of the sprint (scrum) is often harder".

"Not all agile practices can be adopted. A lot depends on the type of project you are working on and the phase of the project you are in. We don't refactor that often because the priority was on developing new features, not existing ones. The culture of the organization plays a major role in Agile adoption. We can't go for Agile practices when users want to do production parallel testing for the whole project".

Similarly, most of the respondents have reported their challenges with respect to their agile development process. All the responses of the respondents were analyzed and the limitations/ challenges regarding the use of agile practices are reported below:

- Depending on project domain, project type and phase of the project, agile practices should be adopted. Not all agile practices can be implemented at once. This requires a superior decision maker.

- Extra effort is needed to maintain good interaction between the customer and development team to develop the product as per customer needs.

- Due to minimal documentation, the scope is limited and sometimes documentation is needed when working on support project.

- Additional heavy investment and extra effort is needed to train the agile team and also the training takes longer time.

- A broader sort skills are needed for effective communication and team work.

- Adopting agile mindset and maintaining agile culture is challenging in an outsourced environment.
- Dealing with legacy code possess additional challenges to teams.
- Parallel testing is not flexible with agile practices.
- More elitism caused more trouble.
- Developing and structuring user stories is challenging sometimes. Describing stories often fail because short descriptions are misunderstood by team members.
- Without dedicated agile administration continuously improving the agile process is a bigger challenge and threat to the entire team.
- Frequent build sometimes leads to code breakage.
- Communication challenges between globally distributed teams.

In the next open ended question, the respondents were asked to report their reasons for adopting the practices in that particular way/ order with respect to their reported order of practices. This question was intended to gather the reasons behind adopting the agile practices in that particular way or order. From all over the responses the most common reasons for implementing agile practices by the practitioners are identified through performing a detail evaluation on the responses reported by the respondents. A variety of perspectives on implementing agile practices in a particular way/ order were expressed by the respondents are described as follows:

"Agile practices need to be aligned in a particular order for successful implementation of agile practices and to eliminate integration issues".

"They need to be aligned in particular order to complete all the assigned user stories in stipulated time".

"To increase efficiency by having more control over planning process and also to eliminate unnecessary actions. Thereby to save time, budget and to improve the quality".

Most of the respondents have reported their reasons with respect to their agile development process. All the responses of the respondents were studied and the reasons regarding the implementation of agile practices in a particular way/ order are reported below:

- To increase efficiency and have more control over planning process by eliminating unnecessary actions.
- Agile administration containing executive supervisors decides the order of the agile practices which needs to be implemented.

- Alignment of agile practices is important to minimize the integration issues.
- Implementing of agile practice in a certain order help in faster execution, faster feedback and time to market.
- To shorten lead times and avoid delivering functionality which is no longer relevant.
- To save time, budget and improve quality in the development process.

The fourth question was asked to report whether any practices were terminated during the development process. This question was intended to gather reasons behind if any of the practices were terminated during the development process. All of the respondents except one reported similar statements that none of the practices were terminated during the development process. One responded who reported the termination of the practice reported that pair programming was terminated in their organization in the middle of the development process as this practice was not efficient moreover creating overheads with in the team. Two of the respondents in spite of non-termination of practices comment in provided text field:

- One of the respondent reported that *"In his experience most team members (especially developer) are often positive using agile practices and told that no practices were terminated".*
- Other respondent reported that *"I don't see any practices which were using in our project have been terminated. But on my personal account, I would terminate a practice if I find it redundant, doesn't resolve any issue or uncertainties in the project".*

The last open ended question was provided optionally to add additional remarks or comments regarding their adoption of agile practices with respect to their agile experience. Few of the respondents responded to this field thereby all the additional comments were gathered, analyzed and reported below. One of the impressive comment was

"If you are aiming at a strategic adoption of 'agile' methods, it's worth considering what Peter Drucker said: 'culture eats strategy for breakfast'. Adoption of the process will not produce an agile culture; however, they will reinforce it. This is emphasized in the 'agile manifesto' where it states that 'Individual and iterations' need to VALUED MORE than 'processes and tools'; the latter is important, but the former is more important. If you are implementing a cultural change, you need to be able to understand how to analyze the current organization culture. The motion of the 'Cultural Web' as a tool for analysis is useful here. Common wisdom is that this will take about three years and will cost 20-30% of your team who will not be able to adjust. That matches our experience".

Additional comments provided by the respondents are

- Agile needs to be modified or combined with other practices to fit the business needs.
- Extreme care should be taken in the sprint planning meetings regarding the prioritization of user stories with a clear description of the requirements for each user story included in the product backlog.

4.3 Summary of the results

A survey was conducted with the current software practitioners to identify and understand the implementation of agile practices of agile maturity model implemented in the industry. Also, the benefits and limitations of implementing these agile practices in a certain order are derived. The collected data is analyzed by using statistical and comparative analysis methods to attain reliable results. Analysis of the demographic results showed that 67% of the respondents were having experience for more than three years. If observed clearly, 33% of the respondents have experience of the respondents for more than 10 years in the field of agile software development. Therefore, with respect to the experience of the respondents, the summary of the results showed a positive sign indicating that the respondents have high enough experience to answer the questionnaire which means the collected answers from the respondents are reliable to perform the research. Moreover, the respondents are located from different geographical locations adding an advantage of analyzing the data from different continents thereby understanding the agile development process followed by the organization from all over the world. Each of the respondents possesses different roles and are from different levels this helped in understanding the perspectives of each respondent with respect to their designation.

The next part of the questionnaire consists of five open ended question to gather the practitioner's perspective regarding the benefits and limitations of implementing the agile practices in the software development process. The first part of the open ended question aims at knowing the benefits and measure of the success of implementing agile practices. 54.35% of the respondents felt that the adoption of agile practices was successful moreover 26.09% of the respondents reported that according to them the adoption of the agile practices was very successful. In the bellow pie chart 4.8, a detail perception of all the respondents regarding the success of adoption of agile practice is described in detail.

The results of the analysis showed that 80.44% of the respondents felt that implementation of agile practice was successful. In which 48% of the respondents are scrum masters, 34% of them are programmers and developers, 6% of them are trainers and other 12% constitute managers and others. Most of the responded that the agile is flexible and adaptable to change in requirements. The essential benefits of the agile development are frequent releases of the shippable product.

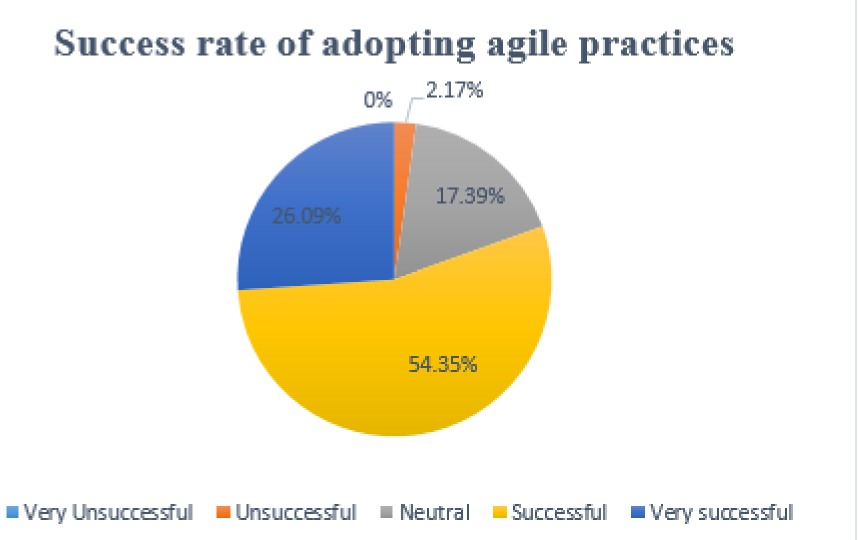


Figure 4.8: Success rate of adopting agile practices.

Several other identified benefits of implementing agile practices are documented in the previous section (4.2.8). Next part of the open ended question focus on understanding the challenges faced by the software practitioners due to the adoption of agile practices and also the respondents were investigated if any of the adopted agile practices are terminated during the development process due to some particular reasons. The respondents were asked to report the difficulties faced during the process if agile development. Several challenges were identified after performing the analysis. The results are documented in the section 4.2.8 out of which most commonly faced challenges by the practitioners due to agile development is adoption of agile practices in a particular order based on their development type of project. And issues with the training of the agile to the practitioners who are new to agile development, time constraints, and issues with the agile administration. In detail, all the challenges reported by the respondents were reported in the previous section (4.2.8).

Chapter 5

Analysis and Discussion

5.1 Validity Threats

The survey can be affected by four types of validity threats they are external validity, internal validity, construct validity and conclusion validity [56].

5.1.1 Internal Validity

Initial validity of the survey is related to research design. Petersen [33] states that "This validity deals with factors that might affect cause and effect relationships, but is unknown to the researcher". In this particular study this threat might occur with the type of questions in the questionnaire i.e. respondents might feel uncomfortable in answering the question/they might not have knowledge regarding that question/they might feel uneasy to answer the question/they might not have implemented that practice before. This kind of threats might decrease the response rate of the questionnaire or there is a high possibility of incomplete responses. This will definitely have an impact on the overall output of the results. Therefore, to reduce these kind of threats the questionnaire is designed to be in the simplest possible way with the options "Don't know" and "Never used" for all the close ended questions. In addition, the personal information of the respondents was not requested but an open ended question was provided to mention their email address if they were interested to the results of the survey. Pilot survey was done with the software practitioners experienced with agile development who are not involved in the research to identify any flaws or issues with the questionnaire. These practitioners did not find any flaws and additionally commented that the questionnaire was professional.

Furthermore, the questionnaire 1 was validated by the supervisor prior to the execution of the survey. The sample of the population was selected based on the convenience sampling with this the internal threat caused by the respondents' selection was mitigated.

5.1.2 External Validity

As indicated by Wohlin [56], external validity covers the generalizability of the results. Since the survey is anonymous, there is no guarantee that the respondents belong to required sample population. This kind of anonymous responses can affect the results of analysis and also the final output. Hence to mitigate this threat the survey invitation is only sent to the practitioners who are involved with the software development process and also the survey link is only posted in the social networking groups related to the agile software development or agile methodologies. So that only software practitioners with the agile background can only participate in the survey from all over the globe.

Invalid selection of the practitioners can affect the results of the survey. Hence the participants selected for answer the survey questionnaire was based on the selection of convenience sampling and an extreme care is taken while selecting the respondents to make sure that every software practitioner is valid for this study. Thereby, with this the authors were able to mitigate the threat to external validity.

5.1.3 Construct Validity

This type of threat might occur when the survey is unable to provide the essential data or information for conducting the research. This threat might occur when the questionnaire is designed with irrelevant or unnecessary questions which might not provide potential answers to the actual research area. In order to mitigate this threat, the results of the literature review are analyzed and then the necessary questions were designed and incorporated into the questionnaire. The questions were framed basing on the research idea. Also the most important factor 'time' is considered while framing the questions. The increase in the number of questions in questionnaire increases the time enormously, this might also affect the response rate. Therefore, before publishing the questionnaire a pilot survey is conducted to check the time required, usability and understandability of the questionnaire.

5.1.4 Conclusion Validity

Petersen [33] mentions that "this validity threat deals with the degree to which conclusions/inferences we draw are reasonable i.e. the threat of quality and trustworthiness of the results". In order to alleviate this threat, the authors conducted a literature review prior to the execution of survey within the same topic. The obtained results from the survey was then compared with the results of literature review to observe if there exists any relation between the scientific literature and industry field.

5.2 Answering the Research Questions

5.2.1 RQ1

This research question is intended to identify the order of agile practices recommended in agile maturity models from the literature. To answer this question a literature review was conducted and the articles that describe various maturity models are extracted from the scientific databases. These several models were analyzed in detail and synthesized in order to identify the different agile maturity models with its set of agile practices. In total nine maturity models were extracted from the current literature. Each maturity model is carefully examined to retrieve its structure and set of agile practices. A comprehensive overview of each maturity model is presented in the results section 4.1. According to authors knowledge, there are no similar studies presenting nine maturity models in a single study.

One study by Schweigert et.al [45] describes the currents status of agile maturity models and provide an approach to extract and analyze the agile maturity models. There are few studies which had compared utmost of five maturity models in their studies like Ozcan Top [31] had made an attempt in assessing five agile maturity models. Whereas another study by Sidky [47] had compared three agile maturity models and proposed a new maturity framework. Whereas this particular study is designed to extract and analyze nine maturity models identified from the literature. All the nine maturity models included in the study are compared and presented in section 4.1 which represents a logical structure of each maturity model with their set of agile practices. Thereby, adding an advantage of comparing the different agile maturity models with one another. After the analysis on nine maturity models one interesting finding was the practices like planning game, on-site customer and daily standups are suggested at the initial levels in most of the models. The analysis and comparison of nine maturity models resulted in mapping of agile practices. The results are presented in figure 2.1. Kent Beck performed a similar study in mapping the XP practices. Thereby the findings from the literature are compared with the Kent Beck study to observe if there exists any common relation among the agile practices. Interestingly three pairs of agile practices i.e onsite customer and planning game, continuous integration and short iterations, TDD and planning game are found. Out of six combinations retrieved from nine maturity models three pairs of agile practices showed similar results with the Kent Beck study. Since his study was confined to XP practices it was unable to address all the agile practices.With this, the authors conclude that their first objective to identify the order of agile practices from the maturity models has been achieved successfully.

5.2.2 RQ2

To obtain this, a questionnaire was framed and the survey was conducted with the software practitioners who are experienced with agile development. Survey was performed in a structured professional manner under the supervisor guidance. The data is collected from the different practitioners located geographically from all over the world. The responses were collected from the targeted population each performing different roles and responsibilities. The collected data is analyzed by performing both the statistical analysis and comparative analysis.

Several studies conducted by Santos et.al [43], R.Vijayasastry et.al [50] and Begel et.al [5] had focused on usage of agile practices and their benefits. Whereas this particular study focuses on retrieving the order of implementation of agile practices in the IT industries. The findings of the survey results are compared with the results of literature to observe the commonalities and differences. From the literature eight combination of agile practices were retrieved from the nine maturity models as mentioned earlier. Now the results of the survey showed nine combination of agile practices which were implemented in the industries. When both the literature and survey results are compared surprisingly, only three combinations of agile practices were in common namely face to face communication and self organizing, on-site customer and planning game, continuous integration and time-boxing. Whereas five other findings retrieved from the survey were not in common with the literature. The results indicate that the software practitioners are not completely following the agile maturity models for their development process. This implies that the agile maturity models available are not completely efficient in addressing the needs of software practitioners. And also it seems that implementation of agile practices depends on the organization, project type and their current practices.

5.2.3 RQ3

This respective question is intended to identify the benefits and limitations of implementing the agile practices. To answer this, respondents were provided with open ended questions to provide clear cut benefits and limitations which they have experienced during the implementation of agile practices. The collected qualitative data was analyzed for identifying the benefits and limitations. The most common benefit reported by the respondents was regarding the flexibility and adaptability of the agile development process with the acceptance of change in requirements even in the middle of the development process. Whereas one of the most common challenge reported was maintaining the agile culture throughout the development process. A detail results regarding the benefits and limitations are reported in the section 4.2.7. By this, the third research question of the study is achieved.

Chapter 6

Conclusions and Future Work

6.1 Conclusion

In the recent years, agile has gained lot of importance due to its adaptability and flexibility in nature. Agile development have been implemented in both the maintenance and development projects due to its several principle of responsiveness, transparency, collaboration and iteration in the development process [17]. Agile approaches have contributed several benefits and enhancements to the software development process in terms of better enhanced communication, improved responsiveness to change in project requirements, minimal documentation with in person meetings, satisfied working environment among team with better relationships, efficient frequent delivery, self-organizing teams and most importantly customer satisfaction [17], [5], [49], [29]. Hence, this research had focused to contribute additional strengthen knowledge on implementing the agile practices in the software development process for a better understanding and enhanced process of development.

The purpose of the current study is to examine the current literature of agile maturity models and validate the findings from the literature through an empirical study. Three research questions have been framed to conduct this research. Initially, a background study was conducted to understand several distinct agile practices used in the process of software development. For answering the RQ1 to extract different agile maturity models, a literature review has been conducted. This helped the authors in identifying different agile maturity models which links the agile software development practices to maturity levels and helps in guiding the organizations in a structured manner for a smooth development process [48]. After careful examination authors were able to identify nine agile maturity models from the literature. Each agile maturity model is analyzed and synthesized to extract the different maturity levels with its set of agile practices embedded in each level. Following this all the nine maturity models are extracted. An innovative approach of combining them at one place has been done to compare the commonalities and differences of maturity model with one another. When compared no maturity model showed resemblance with one another. If observed from each maturity is structured with different number of levels one with three levels (M7), two of them four levels (M8, M9), M2 with six levels and others in five levels (M1, M3, M4, M5, M6). Therefore, it is clear that each maturity model is incorporated with different levels and are subjected to different aims and objectives with respect to the agile practices aligned in different levels. Hence, there were only few particular similarities observed among the extracted agile maturity models. But the agile practice embedded in these maturity models were repetitive with one another irrespective to the levels. Therefore, this study further aimed at identifying the agile practices that are implemented in the industry to understand how effective these agile maturity models are benefiting the organizations and in which particular way/ order the agile practice are implemented to successfully accomplish the project objectives.

In order to achieve the objective of the RQ2, 17 agile practices which were identified in the study were incorporated into a questionnaire. A survey was conducted with the software practitioners who are experienced with the agile methodologies to understand the order of agile practices that several organizations are following. After the execution of the survey, authors could summarize the results of the survey. The results of the survey had showed the variance in perception of implementing the agile practices from the literature in terms of order. Results from the survey helped in performing a critical analysis in comparing the agile maturity models.

Finally, the questionnaire is designed in a way to identify the benefits and limitations of implementing the agile practices in a particular way/ order to answer the RQ3. Interestingly results from the overall respondents shows success rate of 80.44% for the implementation of agile practices in the software companies. This success rate itself just implies that there are several benefits in implementing agile methodologies for developing the project. In addition, several respondents have reported the benefits in provided open-ended questions. The responses reported by the respondents indicated that implementation of agile practices increases with the flexibility and adaptability of development process with the acceptance of change in requirements any stage of the development process.

6.2 Future Work

There are several studies which had focused on agile methodologies. It has been observed that implementing the agile practices is challenging. Also, there is no particular proper study that has focused on identifying the importance of implementing the agile practices in a certain order. Within this particular research a total of 13 challenges have been identified regarding the implementation of agile practices. Therefore, there is need for finding the mitigation strategies for each of the challenge that is identified in the study. In addition to overcome these challenges there is a need to provide recommendations, techniques and tools for the software practitioners to improvise their development process. Framing a basis with this view point a future research can be performed by providing an alleviated way of implementing the agile practices which would benefit the software practitioners in a much more advanced way.

References

[1] N. Abbas, A. M. Gravell, and G. B. Wills. Using factor analysis to generate clusters of agile practices (a guide for agile process improvement). In *Agile Conference (AGILE), 2010*, pages 11–20. IEEE, 2010.

[2] A. Al Tamimi. Empirical Investigation on Success Factors in Adapting Agile Methodology in Software Development at Public Organizations. 2014.

[3] S. W. Ambler. The agile scaling model (ASM): adapting agile methods for complex environments. *Environments*, 2009.

[4] K. Beck. *Extreme Programming explained: embrace change.* XP series. Addison-Wesley, Reading, Mass, 2000.

[5] A. Begel and N. Nagappan. Usage and perceptions of agile software development in an industrial context: An exploratory study. In *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on*, pages 255–264. IEEE, 2007.

[6] R. Benefield. Seven dimensions of agile maturity in the global enterprise: a case study. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1–7. IEEE, 2010.

[7] T. Dyba and T. Dingsoyr. Empirical studies of agile software development: A systematic review. *Information and software technology*, 50(9):833–859, 2008.

[8] A. Fink. *How to analyze survey data*, volume 8. Sage, 1995.

[9] A. Fink. *The survey handbook*, volume 1. Sage, 2003.

[10] R. M. Fontana, I. M. Fontana, P. A. da Rosa Garbuio, S. Reinehr, and A. Malucelli. Processes versus people: How should agile software development maturity be defined? *Journal of Systems and Software*, Vol. 97:pp. 140–155, November 2014.

[11] R. M. Fontana, V. Meyer, S. Reinehr, and A. Malucelli. Progressive outcomes: A framework for maturing in agile software development. *Journal of Systems and Software*, vol. 102:pp. 88–108, 2015.

[12] R. M. Fontana, S. Reinehr, and A. Malucelli. Maturing in agile: what is it about? In *Agile Processes in Software Engineering and Extreme Programming*, pages 94–109. Springer, 2014.

[13] M. Fowler and J. Highsmith. The agile manifesto. *Software Development*, 9(8):28–35, 2001.

[14] B. N. Green, C. D. Johnson, and Alan Adams. Writing narrative literature reviews for peer-reviewed journals: secrets of the trade. *Journal of chiropractic medicine*, 5(3):101–117, 2006.

[15] G. W. Hislop, M. J. Lutz, J. F. Naveda, W. M. McCracken, N. R. Mead, and L. A. Williams. Integrating agile practices into software engineering courses. *Computer science education*, 12(3):169–185, 2002.

[16] M. E. Hussein, S. Hirst, V. Salyers, and J. Osuji. Using grounded theory as a method of inquiry: Advantages and disadvantages. *The Qualitative Report*, 19(27):1–15, 2014.

[17] N. Jigeesh. An Empirical Study of Agile Testing Attributes for Higher Customer Satisfaction in IT Projects in India. *International Journal of Business and Information*, 10(3), 2015.

[18] K. Kelley, B. Clark, V. Brown, and J. Sitzia. Good practice in the conduct and reporting of survey research. *International Journal for Quality in Health Care*, 15(3):261–266, 2003.

[19] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman. Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology*, 51(1):7–15, 2009.

[20] A. Lacey and D. Luff. *Qualitative data analysis.* Trent Focus Sheffield, 2001.

[21] C. Larman. *Agile and iterative development: a manager's guide.* Addison-Wesley Professional, 2004.

[22] M. Leppänen. A comparative analysis of agile maturity models. In *Information Systems Development*, pages 329–343. Springer, 2013.

[23] Y. Levy and T. J. Ellis. A systems approach to conduct an effective literature review in support of information systems research. *Informing Science: International Journal of an Emerging Transdiscipline*, 9(1):181–212, 2006.

[24] Y. Levy and T. J. Ellis. Towards a framework of literature review process in support of information systems research. In *Proceedings of the 2006 Informing Science and IT Education Joint Conference*, volume 26, 2006.

[25] K. M. Lui and K. C. Chan. A road map for implementing extreme programming. In *Unifying the Software Process Spectrum*, pages 474–481. Springer, 2005.

[26] R. C. Martin. *Agile software development: principles, patterns, and practices.* Prentice Hall PTR, 2003.

[27] G. Melnik and F. Maurer. Perceptions of agile practices: a student survey. In *Extreme Programming and Agile Methods—XP/Agile Universe 2002*, pages 241–250. Springer, 2002.

[28] N. B. Moe, T. Dingsoyr, and T. Dyba. Understanding self-organizing teams in agile software development. In *Software Engineering, 2008. ASWEC 2008. 19th Australian Conference on*, pages 76–85. IEEE, 2008.

[29] A.B.M. Moniruzzaman and D. S. A. Hossain. Comparative Study on Agile software development methodologies. *Global Journal of Computer Science and Technology*, 13(7), 2013.

[30] J. Nawrocki, B. Walter, and A. Wojciechowski. Toward maturity model for extreme programming. In *Euromicro Conference, 2001. Proceedings. 27th*, pages 233–239. IEEE, 2001.

[31] O. Ozcan-Top and O. Demirörs. Assessment of agile maturity models: a multiple case study. In *Software Process Improvement and Capability Determination*, pages 130–141. Springer Berlin, Heidelberg, 2013.

[32] C. Patel and M. Ramachandran. Agile maturity model (AMM): A Software Process Improvement framework for agile software development practices. *International Journal of Software Engineering, IJSE*, 2(1):3–28, 2009.

[33] K. Petersen and C. Gencel. Worldviews, research methods, and their relationship to validity in empirical software engineering research. In *Software Measurement and the 2013 Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2013 Joint Conference of the 23rd International Workshop on*, pages 81–89. IEEE, 2013.

[34] K. Petersen and C. Wohlin. The effect of moving from a plan-driven to an incremental software development approach with agile practices. *Empirical Software Engineering*, 15(6):654–693, 2010.

[35] M. Petticrew, L. Arai, H. Roberts, N. Britten, and J. Popay. Testing Methodological Guidance on the Conduct of Narrative Synthesis in Systematic Reviews. *Evaluation*, 15:1, 2009.

[36] M. Pikkarainen, J. Haikara, O. Salo, P. Abrahamsson, and J. Still. The impact of agile practices on communication in software development. *Empirical Software Engineering*, 13(3):303–337, 2008.

[37] M. Pikkarainen, O. Salo, and J. Still. Deploying agile practices in organizations: a case study. In *European Conference on Software Process Improvement*, pages 16–27. Springer Berlin Heidelberg, 2005.

[38] T. Punter, M. Ciolkowski, B. Freimut, and I. John. Conducting on-line surveys in software engineering. In *Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on*, pages 80–88. IEEE, 2003.

[39] A. Qumer and B. Henderson-Sellers. A framework to support the evaluation, adoption and improvement of agile methods in practice. *Journal of Systems and Software*, 81(11):1899–1919, 2008.

[40] C. Robson and K. McCartan. *Real world research*. Wiley, 2002.

[41] P. Rodríguez, J. Markkula, M. Oivo, and K. Turula. Survey on agile and lean usage in finnish software industry. In *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement*, pages 139–148. ACM, 2012.

[42] J. Rowley and F. Slack. Conducting a literature review. *Management Research News*, 27(6):31–39, 2004.

[43] M. D. A Santos, P. H. D. S. Bermejo, M. S. de Oliveira, A. O. Tonelli, and E. J. Seidel. Improving the management of cost and scope in software projects using agile practices. *International Journal of Computer Science and Information Technology (IJCSIT)*, 5(1), 2013.

[44] R. Sapsford and V. Jupp. *Data collection and analysis*. Sage, 2006.

[45] T. Schweigert, D. Vohwinkel, M. Korsaa, R. Nevalainen, and M. Biro. Agile maturity model: a synopsis as a first step to synthesis. In *Systems, Software and Services Process Improvement*, pages 214–227. Springer, 2013.

[46] P. Sfetsos and I. Stamelos. Empirical studies on quality in agile practices: A systematic literature review. In *Quality of Information and Communications Technology (QUATIC), 2010 Seventh International Conference on the*, pages 44–53. IEEE, 2010.

[47] Ahmed Sidky. *A structured approach to adopting agile practices: The agile adoption framework*. PhD thesis, Virginia Polytechnic Institute and State University, 2007.

[48] F. S. Silva, Santana F. S. F., Lima P. A., Monteiro D. A. I., Peeira P. P., and Romero D. L. M. S. A Reference Model for Agile Quality Assurance: Combining Agile Methodologies and Maturity Models. In *Quality of Information and Communications Technology (QUATIC), 2014 9th International Conference on the*, pages 139–144. IEEE, 2014.

[49] A. Solinski and K. Petersen. Prioritizing agile benefits and limitations in relation to practice usage. *Software Quality Journal*, pages 1–36, 2014.

[50] L.R. Vijayasarathy and D. Turk. Agile Software Development: A survey of early adopters. *Journal of Information Technology Management*, 19(2):1–8, 2008.

[51] X. Wang, K. Conboy, and M. Pikkarainen. Assimilation of agile practices in use. *Information Systems Journal*, 22(6):435–455, 2012.

[52] R. Wendler. The maturity of maturity model research: A systematic mapping study. *Information and software technology*, 54(12):1317–1339, 2012.

[53] L. Williams. Agile software development methodologies and practices. *Advances in Computers*, 80:1–44, 2010.

[54] C. Wohlin and A. Aurum. Towards a decision-making structure for selecting a research design in empirical software engineering. *Empirical Software Engineering*, 20(6):1427–1455, 2015.

[55] C. Wohlin, M. Höst, and K. Henningsson. Empirical research methods in software engineering. In *Empirical methods and studies in software engineering*, pages 7–23. Springer, 2003.

[56] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in software engineering.* Springer Science & Business Media, 2012.

[57] A. Yin, S. Figueiredo, and M. M. da Silva. Scrum maturity model. *Proceedings of the ICSEA*, pages 20–29, 2011.

Appendices

Appendix A
Invitation letter for participating in survey

Dear Practitioner,

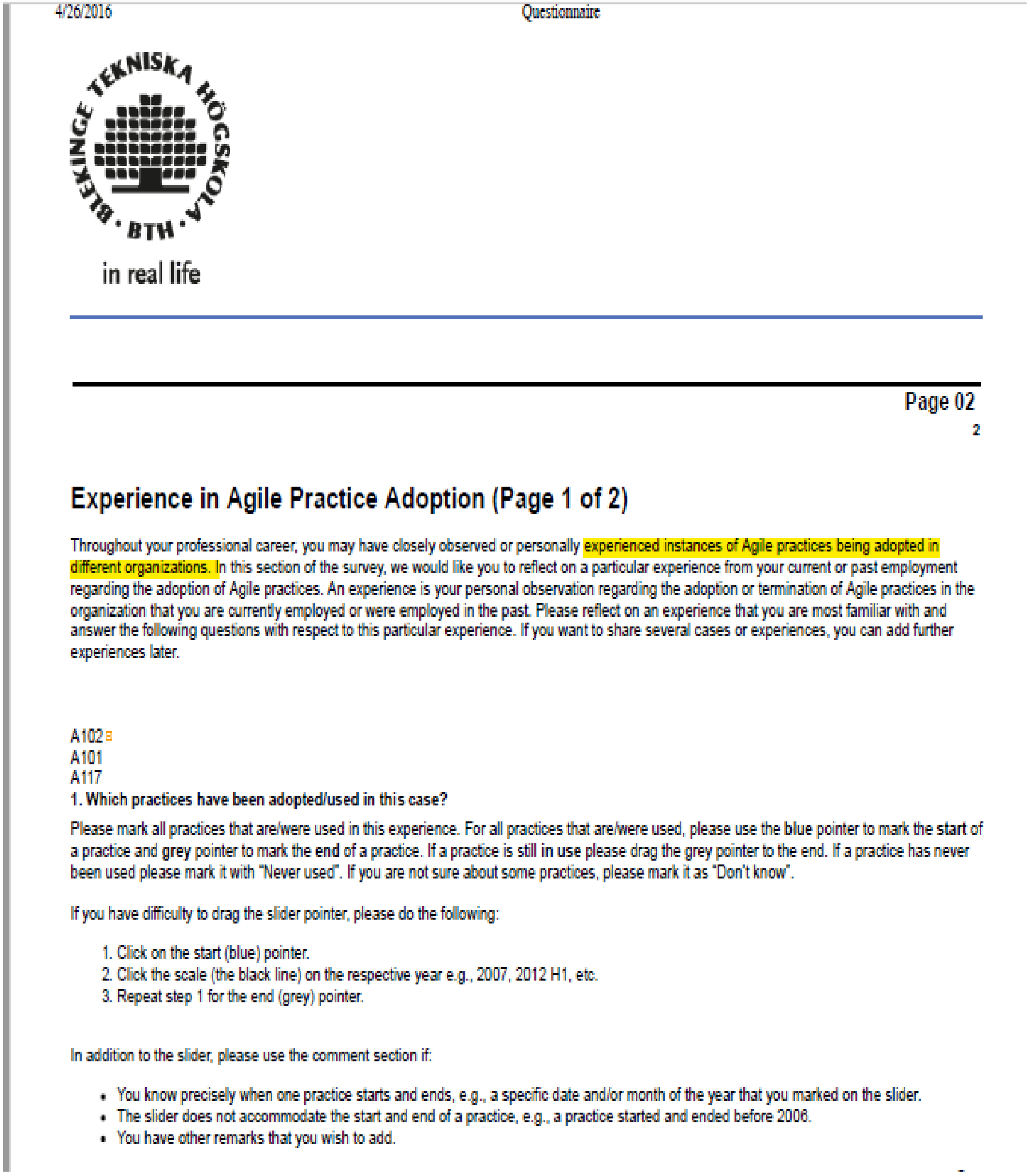
We are conducting a survey on how Agile practices, e.g., retrospective meetings, pair programming, user stories, and metaphors, etc., are adopted and/or abandoned. This survey is a part of a research project at Blekinge Institute of Technology (BTH). Please kindly find a brief introduction to the survey below.

When adopting an Agile method, e.g., Scrum, eXtreme Programming (XP), etc., usually a number of practices are adopted together. Sometimes you start adopting two or more practices together. Sometimes you start adopting one practice after you master another practice. Which practices should be adopted together? Which practices should be adopted after another? Should a practice be terminated before another can be adopted? Whichever scenario is selected, it might bring different benefits and limitations. The goal of this survey is to identify experiences of adoption and/or termination of Agile practices. We would like to learn from you as practitioners, which Agile practices are being adopted, when they started, if they are still in use or already abandoned, and what are the associated benefits and limitations of this experience. We would then use your feedback as input for research into how to maximize benefits and minimize limitations of Agile practices adoption. We are interested in understanding how things are done in practice and not a "perfect" experience or scenario. The link to the survey is: https://www.soscisurvey.com/AgilePracticeSurvey2016/

In case of any queries regarding the survey, you can contact:

Indira Nurdiani
Indira.nurdiani@bth.se
bit.ly/indiran
Prithvi Raj Sirigudi
prsi15@student.bth.se
Contact No: +46 76 78 52287
Rahul Deekonda
rade15@student.bth.se
Contact No: +46 767 996 377

Appendix B
Questionnaire for survey



4/26/2016 Questionnaire

in real life

Page 02

2

Experience in Agile Practice Adoption (Page 1 of 2)

Throughout your professional career, you may have closely observed or personally experienced instances of Agile practices being adopted in different organizations. In this section of the survey, we would like you to reflect on a particular experience from your current or past employment regarding the adoption of Agile practices. An experience is your personal observation regarding the adoption or termination of Agile practices in the organization that you are currently employed or were employed in the past. Please reflect on an experience that you are most familiar with and answer the following questions with respect to this particular experience. If you want to share several cases or experiences, you can add further experiences later.

A102
A101
A117

1. Which practices have been adopted/used in this case?

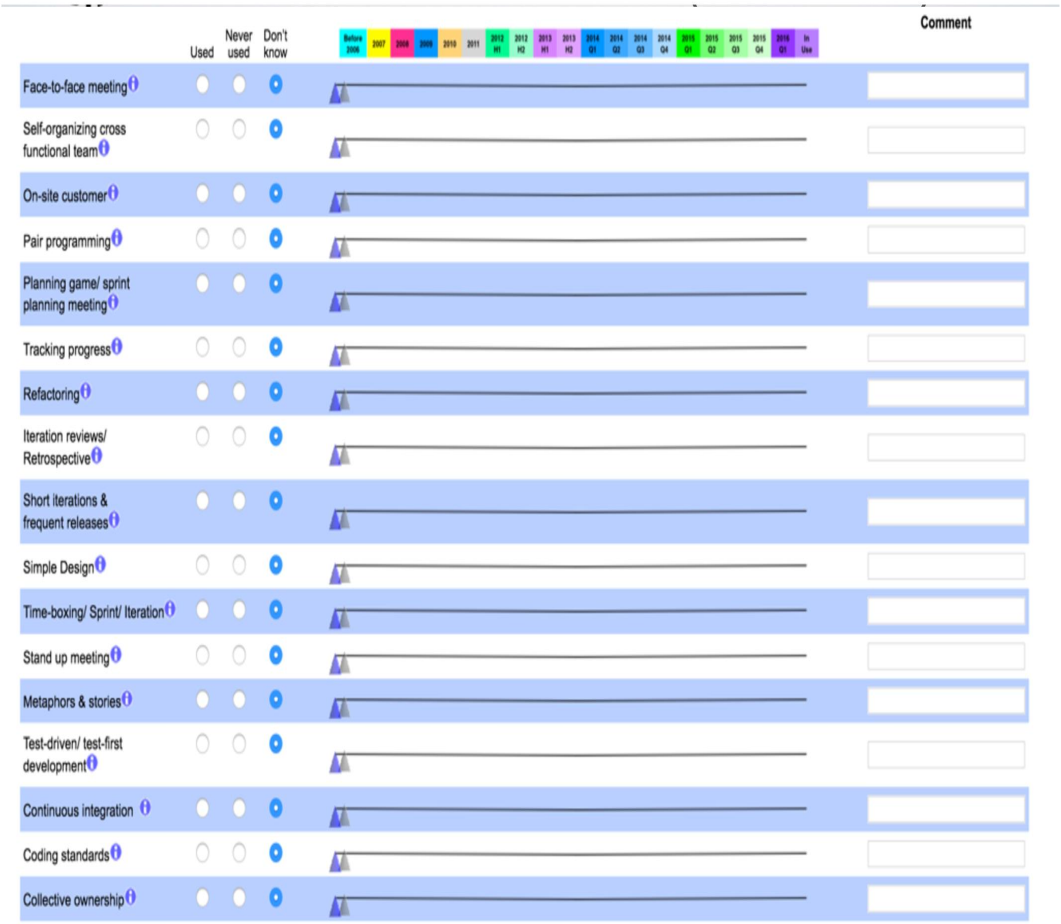
Please mark all practices that are/were used in this experience. For all practices that are/were used, please use the **blue** pointer to mark the **start** of a practice and **grey** pointer to mark the **end** of a practice. If a practice is still **in use** please drag the grey pointer to the end. If a practice has never been used please mark it with "Never used". If you are not sure about some practices, please mark it as "Don't know".

If you have difficulty to drag the slider pointer, please do the following:

1. Click on the start (blue) pointer.
2. Click the scale (the black line) on the respective year e.g., 2007, 2012 H1, etc.
3. Repeat step 1 for the end (grey) pointer.

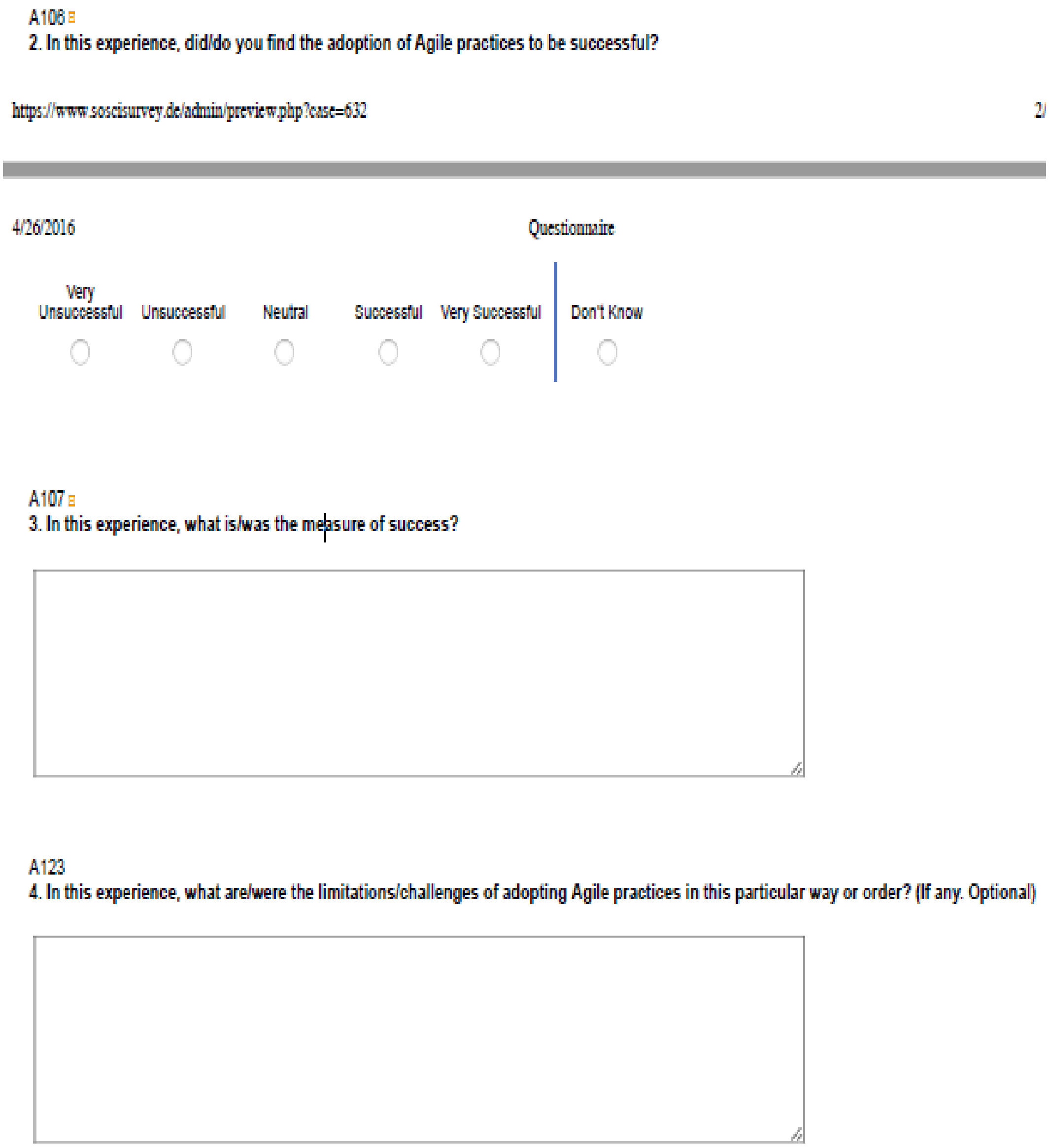
In addition to the slider, please use the comment section if:

- You know precisely when one practice starts and ends, e.g., a specific date and/or month of the year that you marked on the slider.
- The slider does not accommodate the start and end of a practice, e.g., a practice started and ended before 2006.
- You have other remarks that you wish to add.



Before 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 H1 | 2012 H2 | 2013 H1 | 2013 H2 | 2014 Q1 | 2014 Q2 | 2014 Q3 | 2014 Q4 | 2015 Q1 | 2015 Q2 | 2015 Q3 | 2015 Q4 | 2016 Q1 | In Use

	Used	Never used	Don't know	Comment
Face-to-face meeting	○	○	●	
Self-organizing cross functional team	○	○	●	
On-site customer	○	○	●	
Pair programming	○	○	●	
Planning game/ sprint planning meeting	○	○	●	
Tracking progress	○	○	●	
Refactoring	○	○	●	
Iteration reviews/ Retrospective	○	○	●	
Short iterations & frequent releases	○	○	●	
Simple Design	○	○	●	
Time-boxing/ Sprint/ Iteration	○	○	●	
Stand up meeting	○	○	●	
Metaphors & stories	○	○	●	
Test-driven/ test-first development	○	○	●	
Continuous integration	○	○	●	
Coding standards	○	○	●	
Collective ownership	○	○	●	



A106

2. In this experience, did/do you find the adoption of Agile practices to be successful?

https://www.soscisurvey.de/admin/preview.php?case=632 2/

4/26/2016 Questionnaire

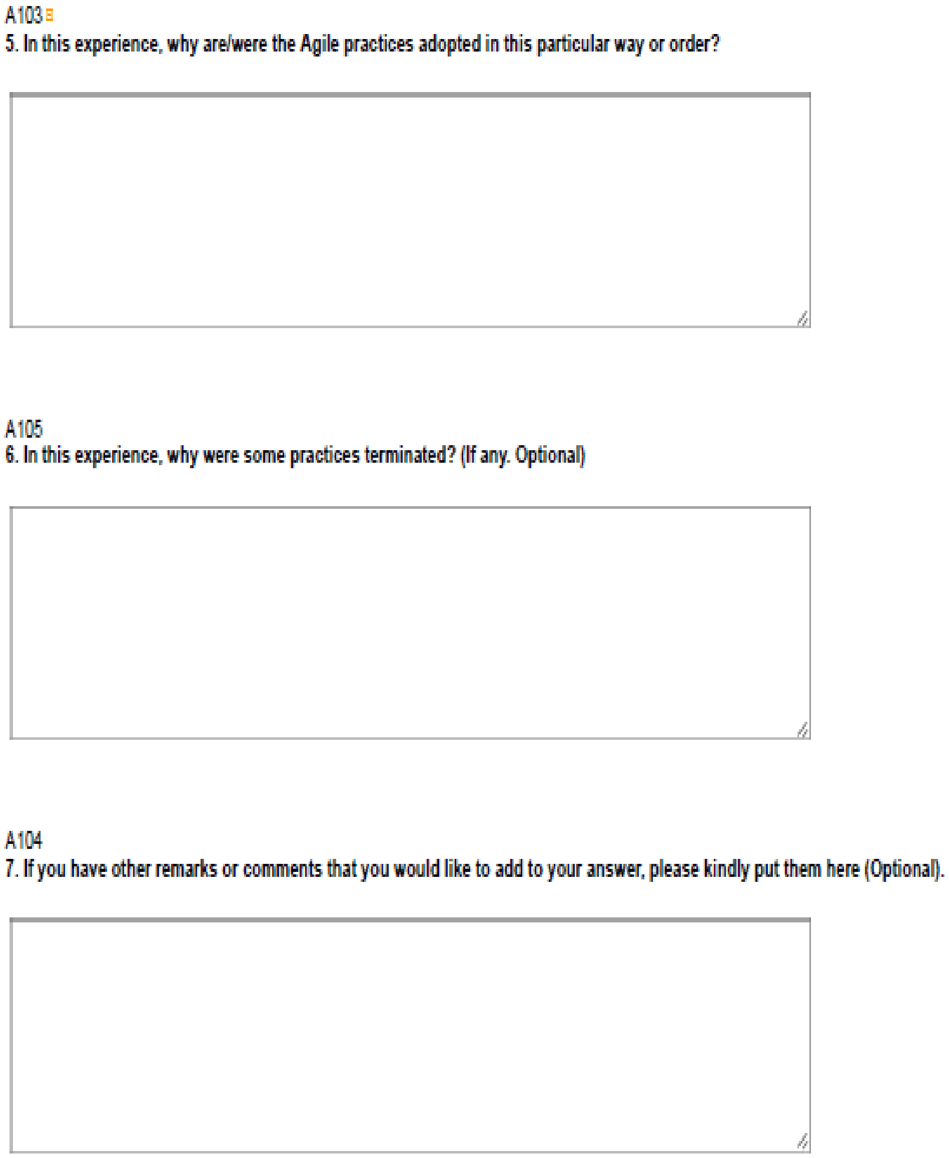
Very Unsuccessful	Unsuccessful	Neutral	Successful	Very Successful	Don't Know
○	○	○	○	○	○

A107

3. In this experience, what is/was the measure of success?

A123

4. In this experience, what are/were the limitations/challenges of adopting Agile practices in this particular way or order? (If any. Optional)



A103

5. In this experience, why are/were the Agile practices adopted in this particular way or order?

A105

6. In this experience, why were some practices terminated? (If any. Optional)

A104

7. If you have other remarks or comments that you would like to add to your answer, please kindly put them here (Optional).