Improving Project Management with Agile Thinking

Bhaskar Rauniyar

Masters in Project Management & Operational Development (1yr), Stockholm, Sweden 2010
Master Thesis

Masters of Science in Project Management & Operational Development
Department of Mechanical Engineering (IEI)
KTH, Royal Institute of Technology

Author       Bhaskar Rauniyar
Supervisor   Mr. Roland Langhé
Department   Department of Mechanical Engineering
             KTH, Royal Institute of Technology
Acknowledgements

First of all, I would like to thank Dr. Roland Langhé, my supervisor at KTH, who has given me many helpful suggestions during the thesis. I would also like to thank my colleague Lisa Chapuis, R&D Project Manager, Ericsson AB. She has made me interested in this topic and helped a lot with regard to theoretical issues and problems which encountered during the thesis.

I would also like to extend my thanks to Bosse Wesshagen, Project Manager at Ericsson Sweden AB who helped me in practical issues.

Special thanks to my parents, my brother Vikash who always took care of me during the thesis and supported me in all situations. Last but not the least; I would like to thank God for giving me the courage and opportunity to pursue my dreams.
Abstract

The rapid change in today’s dynamic environments has increased the uncertainty in project management, forcing the organizations to adapt new methods and approaches to deal with such situations. Software firms have adapted modern and contemporary approaches to deal with uncertainty and to adapt changes by bringing changes in the product during the time of its development. During each of its development phase, customers are involved and teams divide the work in short tasks. In this research study, Agile software development project management is discussed where companies have adapted iterative, adaptive and extreme strategies to be more Agile and flexible in order to meet the deadlines and customer’s requirements. The most commonly used approach is Scrum in which tasks are divided into short sprints to get the full functionalities. However extreme programming, Kanban, adaptive project framework, adaptive software development and a blend of other Agile methods are being used by many organizations to carry out projects. Organizations adapt these methodologies to increase productivity and to add values to customer and themselves but at the same time many problems emerge during the development and evaluation phases.

Key Words: Agile, Software development project, Project management, uncertainty.
Contents
Chapter 1  Introduction .............................................................................................................. - 1 - 
  1.1  Introduction .................................................................................................................. - 1 - 
  1.2  Research Concept ......................................................................................................... - 1 - 
  1.3  Research Context .......................................................................................................... - 1 - 
  1.4  Research Questions....................................................................................................... - 1 - 
  1.5  Delimitations ................................................................................................................ - 2 - 
  1.6  Disposition of the thesis .............................................................................................. - 2 - 
Chapter 2  Research Methodology ...................................................................................... - 3 - 
  2.1  Overall Research Design ............................................................................................. - 3 - 
  2.2  Research Strategy ......................................................................................................... - 4 - 
  2.3  Research Approach ...................................................................................................... - 4 - 
  2.4  Literature Search and Empirical Data ......................................................................... - 5 - 
     2.4.1  Literature Search and Selection ............................................................................. - 5 - 
     2.4.2  Selection of Interviewee and Method .................................................................... - 5 - 
  2.5  Research Quality ......................................................................................................... - 6 - 
     2.5.1  Internal Validity: ..................................................................................................... - 6 - 
     2.5.2  External Validity ..................................................................................................... - 6 - 
     2.5.3  Reliability ............................................................................................................... - 7 - 
Chapter 3  Literature review.................................................................................................. - 8 - 
  3.1  History of Agile Project Management ......................................................................... - 8 - 
  3.2  Agile Development ....................................................................................................... - 8 - 
     3.2.1  Agile Software Development Manifesto & Principle ........................................... - 9 - 
  3.3  Software Development Projects ................................................................................ - 11 - 
  3.4  Role of Agility in Software Development .................................................................... - 11 - 
     3.4.1  Product Oriented Process ....................................................................................... - 13 - 
     3.4.2  Project Management Process ................................................................................ - 13 - 
     3.4.3  Traditional Project Management Approaches ...................................................... - 13 - 
     3.4.3.1  Linear Strategy ................................................................................................... - 13 - 
     3.4.3.2  Incremental Strategy .......................................................................................... - 14 - 
     3.4.4  Iterative Strategy ................................................................................................... - 15 -
Chapter 1  Introduction

1.1  Introduction

This chapter introduces research overview by elaborating research concept and purpose of research. It also discusses questions raised in this thesis as well as presents a brief overview of the chapters covered in this thesis. Research concept answers questions on what the study is about and the main theme of the research. Furthermore, the research context discusses why the study is conducted, and there is a brief contextual background of the study.

1.2  Research Concept

Software is important for the all facets of the modern world; software development itself is not a perfect process. Despite all efforts to employ software engineering methodologies, software development has not been consistently successful resulting in delayed, failed, abandoned, expensive and rejected software projects. Sometimes many ongoing projects require expensive maintenance and corrective releases or service packs. These shortcomings have affected the bottom line of information technology and software development organizations in a big way, challenging them on how to develop software management to improve and avoid waste and inefficiency. There has been a recent emergence of a new class of software development process called Agile Methods, which operates differently than the traditional method [9].

1.3  Research Context

The purpose of this research is to explore the role of Agile thinking in software development projects and how it is helping project development teams to come up with a better suitable solution by improving project productivity.

1.4  Research Questions

Question 1: Which concepts of Agile thinking are relevant for project management?

Question 2: How can Agile thinking bring improvement in project productivity?
Question 3: What are the problems that are encountered during the Agile software development project management?

1.5 Delimitations

Agile methodology is a broad topic and due to limited time and resources to conduct the study; it is not possible to analyze all the concepts of Agile methodology. The focus of the study remained limited to Agile software development projects and Scrum methodology. The study is mainly conceptual and is based on limited empirical study consisting of six interviews which restricts the ability to generalize the results on a broader scale. As a matter of fact, the interviews provide information from the prospective of different companies. The results can be generalized to some extent.

1.6 Disposition of the thesis

Chapter one, is based on introduction to the thesis, the problem description, explaining purpose of the research and raising research questions which needs to be answered during this research. Chapter two is based on methodology on how this research will be conducted and which strategy and approaches are used to find a suitable solution. Chapter three is based on literature review; it provides an insight to Agile development, role of agility in project management, strategies of Agile software development and approaches used to implement these strategies. Chapter four is presenting the empirical data in the form of interviews. Chapter five is based on the analysis of literature review and empirical data. Chapter six discusses the results and findings and chapter seven will provide a summary of the research conducted and its implications for the future research including suggestions.
Chapter 2  Research Methodology

This chapter details out the approaches, techniques and systematic steps taken to conduct the research study. With the help of research methodology, it is intended to provide an idea on gathering required information and using it to answer research questions.

2.1  Overall Research Design

Based on limited empirical basis, consisting two interviews, this study is mainly conceptual and exploratory. The idea behind this research work is to review the literature on project management, agility and Agile software project management and thereafter investigate the state of the art practices and strategies in software development project. It is an exploratory research through which suitable answers to research questions will be explored. The overview of the research design is shown in the figure 1.

![Figure 1: Research Design](image)
As mentioned earlier in delimitations (section 1.5), it was difficult to find a company for an in-depth case study and it is difficult to use an alternate method of case study to conduct research. The thesis aims to answer both ‘What’ and ‘How’ related to Agile software project management. To accomplish this aim, it was decided that a literature review coupled with interviews would be an appropriate research design.

2.2 Research Strategy
Exploratory and descriptive are two aspects of this research. It is exploratory as there has been little research on Agile project management and Agile software project management. The idea behind this research is to explain the strategies and approaches of Agile project management that are used by project management teams.

2.3 Research Approach
This thesis is based on qualitative approach. For analytical reasoning in order to arrive at results, qualitative measures are considered to be the best. The qualitative data is explained in [37] as “information is considered to be qualitative in nature if it cannot be analyzed by means of mathematical techniques”. Qualitative approach was chosen because research is based on evaluating, comparing and linking of Agile project management and Agile software project management strategies and approaches. Two distinct logical systems are considered to be important for this study. These two logical systems are inductive reasoning and deductive reasoning.

In deductive research approach, theories will be applied in real world for the purpose of testing and assessing their validity [24]. It works from more general to more specific, whereby conclusions follow logically available facts. Therefore it is informally called top-down approach. In Inductive approach, researchers develop hypothesis and theories with a view towards explaining empirical observations found in real world [24]. Therefore inductive reasoning works the other way, moving from specific observations to broader generalizations and theories. Conclusion is likely based on premises. It is in formally called bottom-up approach. Beveridge has explained the two logics as, “Logicians distinguish between inductive reasoning: from particular instances to general
principles, from facts to theories and deductive reasoning; from the general to the particular, applying a theory to a particular case” [4].

2.4 Literature Search and Empirical Data
The literature has been searched from within the areas of project management and Agile software development project management. Secondary data has been collected using online resources, journal articles, conference papers, books and e-books. Secondary data is defined as data which is collected by some other person and is available in the form of published resources e.g. secondary data are quarterly profits published in the Wall Street Journal, data collected by Government departments, business information sources, encyclopedia of Business information sources, journals and library data bases [38].

The secondary data collection helped in finding and suggesting suitable methods to answer questions raised in this research as well as interpreting and understanding primary data in a cost efficient and time saving way which will be collected through interviews.

To analyze current practices in project based organizations, empirical data is collected through interviews. It is also called primary data gathering as interviews will be conducted from persons who will share their own personal experiences.

2.4.1 Literature Search and Selection
In order to find enough research material for literature review, a detailed search has been conducted. Initially various databases e.g. web of science, business source premier, academic source premier were used to find articles and other research work using key words, Agile project management, history of Agile methods, project management, Agile software development project management. Later on using the university catalog, e-books were searched and a very relevant work done by Wysocki, Robert K in [34] was found and this work provided a base for conducting the research study.

2.4.2 Selection of Interviewee and Method
To examine the state of the art practices in Agile project management carried out in the industry. Six Semi structured interviews were arranged by the author who personally asked the questions. A complete interview guide was carefully planned and questions were arranged and developed in
advance. Before conducting the actual interview, a pilot interview was conducted informally to test the questions and how much time will be required for the interview. After the pilot interview some of the questions were reviewed for the actual interview. The reason for choosing a semi-structured interview was to provide liberty in explaining things to the respondent. The interview questions are available in the Appendix.

Transcript notes were produced directly from the interview. These transcript notes were used to make sure that all necessary information is available in written form and nothing has been missed out. These questions were arranged in such manner that they are creating a sequence with our literature review and are also systematically answering the research questions.

2.5 Research Quality

Valid and reliable information is required for every research; the quality can be checked using number of tests in case of quantitative research but the quality of qualitative research is typically assessed by considering the validity and reliability of the study. There are two types of validity of a study:

a. Internal Validity
b. External Validity

2.5.1 Internal Validity:

The internal validity is defined as the extent to which result of the study corresponds to reality. It is relevant in establishing a causal relationship where certain conditions lead to other conditions [35] although Golafshani in [16] states that the degree of internal validity is usually considered even though qualitative research is not of casual nature.

The internal validity for this research can be checked by going through information collected in the form of interview, several times in order to correctly present the findings.

2.5.2 External Validity

The degree to which results of the study can be generalized. In other words, degrees to which results are applicable to situations other than the studied one. Some studies aim to generalize the results to broader theories; generalization can be done more easily by studying several situations [35]. As the study was based on interviews from people who had worked on different projects,
external validity can still be considered since results can be generalized to some extent. However it is not possible to get high external validity without including greater number of interviews.

2.5.3 Reliability

The extent to which the research study can be repeated and arrive at the same result. According to Golafshani [16] it is impossible to conduct a social research in exactly the same way more than twice, hence the reliability of the qualitative research cannot be established. However to ensure reliability establishing trustworthiness is crucial, which lies in internal validity so for qualitative research reliability internal validity is sufficient.
Chapter 3 L iterature review

This section aims at identifying the theoretical background of Agile thinking in software development project management that replaces the traditional project management techniques. Three main strategies have been discussed in this phase.

3.1 History of Agile Project Management

Agile project management’s history stretches back a little more than 25 years. As recently as 2001, Agile software development was first explained through “Agile manifesto” by Marting Fowler and Jim Highsmith [20]. Agile manifesto is the guiding principle in all Agile project management models. Most of the Agile project management models originated with software development and are based on very specific software development practices.

3.2 Agile Development

Many new product development companies operate nowadays in uncertain and dynamic competitive environments [22]. The principles of Agile manufacturing were discovered to address the needs of new post-mass production competitive environments. It was no longer possible to achieve significant improvements by developing the old mode of operation, but a paradigm shift was necessary [13]. Agile development was an alternative solution to the inability of traditional methods to address problems such as time, cost, lack of responsiveness [25], intensified global competition, reduction in lead-time and life expectancy of products, diversification of demand, and new technologies [21]. The vision behind to adapt Agile practice is to create the organization ability to respond the changes by being able to deliver or change direction at any time with no any extra cost[41].

The core values, principles and practices of Agile development have been derived mainly from past experiences and many organizations adopt Agile development approach without clearly understanding how agility is defined. The origins of the Agile software development models date back to late 1980s and early 1990’s [22] and has been evolved since the mid 1990’s [25].
The term Agile is often described as a system engineering method but going further deeper Agile is a collection of practices influenced by many disciplines; from software engineering it inherits extreme programming, from both systems and software engineering it borrows Rational Unified Process (RUP) to deliver useful output every few weeks, from manufacturing it gains the influence of lean for elimination of waste and from product development Agile inherits short daily “Scrum” update meeting that facilitates the collaboration and keeps the team focused[17]. Agile has been defined by many authors as;

“Capability to survive and prosper in a competitive environment of continuous and unpredictable change by reacting quickly and effectively to changing markets, driven by customer-designed products and services [18]”

“Ability to cope with unexpected changes, to survive unprecedented threats of business environment, and to take advantage of changes as opportunities [30].”

“Ability of an organization to respond rapidly to changes in demand, both in terms of volume and variety; a business-wide capability that embraces organizational structures, information systems, logistics processes, and mindsets [10].”

“Ability to respond to, and create new windows of opportunity in a turbulent market environment driven by individual (bespoke) customer requirements cost effectively and rapidly [21].”

The above definitions reflect the point that organizations must have the capacity to respond to change which gives them the capability of enabling flexibility, responsiveness and pro-activity. Flexibility results in enabling the organizations to respond to unpredictable changes in demand in a cost effective and timely fashion whereas pro action creates future capabilities [22]. Thus, flexibility could be highly advantageous when faced with certain types of projects and projects scenarios [14]. Agile project management focuses on people development, self-management, participatory decision making, customer focus and less bureaucracy [6].

3.2.1 Agile Software Development Manifesto & Principle

The Agile Software development has developed into an alternative to document driven thorough software development process. Software developers have realized that processes which require many documents, artifacts and actions to follow, is too slow to fulfill customer needs. There is a
need to follow a methodology where one can be up to date to the customer requirements in the current competitive market. The common ground of Agile software development was defined in 2001. Seventeen expert professionals agreed and signed the Agile software development manifesto, which is as follows [36]:

“We are uncovering better ways of developing software by doing it and helping others does it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.”

Manifesto of Agile software development is followed by twelve basic principles. The principles behind Agile software development manifesto are as follows [36]:

a. “Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

b. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

c. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

d. Business people and developers must work together daily throughout the project.

e. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

f. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

g. Working software is the primary measure of progress.

h. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

i. Continuous attention to technical excellence and good design enhances agility.
j. Simplicity—the art of maximizing the amount of work not done—is essential.

k. The best architectures, requirements, and designs emerge from self-organizing teams.

l. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.”

3.3 Software Development Projects

“A software development project (SDP) is a complex undertaking by two or more persons within the boundaries of time, budget, and staff resources that produces new or enhanced computer code that adds significant business value to a new or existing business process” [34].

The definition is bit restrictive as it does not define all types of software development projects. The criteria for SDP are to add significant business values, be highly visible and of moderate complexity. It is important to understand the difference between SDP and software development project management (SDPM). SPDM is defined as:

“Software development project management is the discipline of assessing the characteristics of the software to be developed, choosing the best fit software development life cycle, and then choosing the appropriate project management approach to ensure meeting the customer needs for delivering business value as effectively and efficiently as possible” [34].

3.4 Role of Agility in Software Development

For today’s managers dealing with an increasingly volatile organizational environment is a serious challenge especially in software development projects. The traditional formal methodologies or strategies can be characterized as linear, Sequential process [2] or incremental process [14]. In most real world development efforts are conducted in much more volatile environments as organizations adapt to changes in technology, market and social conditions due to which requirements for systems must be able to change at the same rate. As the system becomes more complex and components become more independent, minor changes produce unanticipated effects, the approach based on the traditional linear and sequential development become a mismatch in such dynamic systems [2]. To deal with such dynamic systems, environment strategies like iterative, adaptive and extreme can be used. The core Agile Project Management (APM) needs to build Agile and adaptable products and there is need to build Agile and adaptable development teams
Agile principles are based on flexibility and simplicity [6] and add value to customers by means of short term deliveries [8] [20].

Koskela and Howell argue in [23] that underlying theories of project management are obsolete. As PMBOK theories guided by the PMI (2004) is determined to be based upon the transformation view of production and three theories of management which are

a. Management as planning

b. The dispatching model of execution

c. Thermostat model of control

Koskela and Howell have questioned these theories especially when it comes to the projects managing the uncertainty and change. They proposed new ingredients to the theories as

a. Focus on flow and value generation in addition to transformation

b. Inclusion of management as organization for planning, the language/action prospective for execution.

Koskela and Howell have demonstrated that Scrum: an Agile project management approach is comprehensive project management method with an underlying theoretical foundation that includes flow and value generation emphases, management as organization, the language/action prospective.

In [32], William also presented that traditional/conventional methods of project management have become inappropriate and disadvantageous for projects that are structurally complex, uncertain and heavily time limited. The new project management approaches such as Agile or lean are more promising for the projects which have such characteristics.

Project management has three kinds of goals [23]:

a. The goal of getting intended products produced.

b. Internal goals such as cost minimization and level of utilization.

c. External goals related to the need of customers like quality dependability and flexibility.
According to the PMBOK (cited in [23]) projects are composed of two types of process project management process and product oriented process, which specify and create the project product. Project management is further divided into initiate, planning, execution, controlling and closing process.

3.4.1 Product Oriented Process

Turner in [40] explained that scope of management serves following functions:

a. An adequate or sufficient amount of work is done

b. Unnecessary work is not done

c. The work that is done delivers the stated business purpose.

d. The scope is defined through the work breakdown structure

Koskella and Howell in [23] explained Turner’s [40] viewpoint as Turner claims that project management is about managing work, this is conceptualization, Turner claims that work can be managed by decomposing the total work efforts into small chunks of work, which are called activities and tasks as in PMBOK guide. Thirdly he claims that this conceptualization and the principal of decomposition serve the essential purpose of project management.

3.4.2 Project Management Process

PMBOK Guide reveals that activities and tasks are the units of analysis in the core process of project management like scope management, time management and cost management.

The PMBOK guide divides project management process into initiating, planning, execution, controlling and closing process. The planning process provides a plan that is realized by execution process and variance from the base line or request for change lead to corrections in executions or change in further plans.

3.4.3 Traditional Project Management Approaches

3.4.3.1 Linear Strategy

Linear strategy is a traditional strategy consists of dependent, sequential phase that are executed with no feedback loops. The solution is only provided after the final stage. The entire project is
scheduled and resource requirements are known. It does not need much skilled resources and teams can be disrupted [14]. It is the simplest of all constitute nearly perfect information about goal and solutions which can be expected. Usually it is based on assumptions and does not accommodate any deviations or uncertainty.

Scope phase is usually of few hours with customers and the project manager and it could be weeklong planned agenda meeting of core project team and representative of customers. This week long planned agenda covers project scope, business case, requirement gathering and high level project schedule with perhaps milestone identification [34].

During planning and launch session work breakdown structures and initial project schedules are decided. Customers are involved during this phase [34].

The designs build and test phase are executed linearly against a project schedule and are monitored and controlled for conformance to the project schedule. Project schedule are revisited due to external factors and unforeseen events.

Close phase is started when the acceptance test criteria has been demonstrated. The Deliverables are put into production status and formal closing activities are done. The best example of linear strategy is waterfall method [34].

3.4.3.2 Incremental Strategy

It is usually identical to a linear strategy except that each phase of the project is released as a partial solution. In this strategy the solution is delivered before the final stage. Incremental strategy produces the business value at earlier stage of project. Change is accommodated through incremental solutions and it has a strong focus on customer value [14].

The goals are clearly defined and solutions & results are delivered in stages over the life of project. The total solution is decomposed into chunks of deliverables to cater the need where results are needed early. Each of the deliverable provides its functionality. These chunks are released sequentially until all functionalities reach to production status.

The scope phase of incremental strategy is almost same as of linear strategy but the deliverables in increments adds complexities. During the plan and launch phase the work is not done contiguously
and it comes into chunks. If some increment slips it affect the resources scheduled for later increments. The development resources usually make scheduled commitments with other projects as well and lacks flexibility due to which it gets difficult to revise project schedule and complexities involves extends to multiple projects. During the monitoring and control phase the scheduling and resource management are fixed and any slippage have to be recovered in given schedule. As all features are defined up front at the beginning of the project but when the customers use the deliverables new features come into their mind which they would like to include in original scope.

### 3.4.4 Iterative Strategy

Iterative approaches are used when an initial version of the solution is known but they are known to fall short in terms of features and functions. It is designed to uncover missing pieces of solution [34]. It consists of a number of repeated phases that include a feedback loop once a group of phase is completed see Figure 2. This strategy is more like learning by doing strategy that uses intermediate solutions as a pathway to discover the details of complete solution. Iteration can be on requirements, functionality, features, design, development, solution. An example of this strategy is Scrum project management [34] [14] there are various phases in iterative strategy [34].

- The scope phase is a bit more complex which involves moving into waters where the complete solution is not known
- Plan and launch phases involves planning at two levels. Development of high level plan without much detail is done in the initial phase as much information is not available. The functionality is known and its design and development can be planned across any number of iterations but in some cases functionality is designed and developed in the first iteration. In the later phase iterations then further drill down to possible areas for further identification and development of features which might be the most efficient of all the designs develop alternatives we may consider.
- Monitor and control phase changes in the iterative strategy, much of the heavy documentation and status reporting gives way to more informal reporting due to speculative nature of the project and formulism become non value added working and begins to burden the teams which no long help them in reaching to the final solutions.
• In the closing phase customer specified criteria must be met and project deliverables to be considered complete. The criteria were specified during the planed phase and update as scope changes request were approved and integrated into the solution.

Fernandez, D & Fernandez, J. (2009) in [14] has summarized the Wysocki [34] iterative strategy’s strength as:

• Customer can review the current solution for suggested improvement
• Scope change can be accommodated between iterations
• Adapts to changing environment.

They have also summarized the weaknesses of the strategy due to with some of the problems faced, these weaknesses are

• Requirement for a more active customer in projects
• The final solution cannot be specified to the customer at the outset of the project.

![Figure 2: Iterative Strategy](image)

3.4.5 Adaptive Strategy

The adaptive strategy is the first approach encountered where the solution is not known or partially unknown see Figure 3. Unlike the iterative approach where some depth of the solution is not known, the adaptive approach is missing both depth and breadth of the solution.
Figure 3: Adaptive Strategy

Figure 3 shows the adoptive SPDM strategy. It is also the model that lies between the iterative and extreme models. In the above model the goal is clearly defined but the solution to reach the goal is not. It is the process that thrives on learning, discovery and change. It is the best suited approach to projects whose solutions are partially known. The degree to which the solution is known might vary over a wide range from knowing a lot but not all to knowing very little. The less that is known about the solution the more risk uncertainty and complexity will be present. To remove the uncertainty associated with these projects, the solution has to be discovered which will only happen through continuous change process which will create a convergence on a complete solution and if not adaptive projects will be frequently cancelled and restarted in other promising directions.

The success of adaptive SDPM strategies is leveraged by accommodating frequent change which is the result of learning and discovery by the team and, most important, by the customer. As change will have a dramatic impact on the project, only a minimalist approach to planning is employed. Planning is actually done just in time. No effort is wasted on planning the future. The future is unknown, and any effort at planning that future will be viewed as non-value-added work [33]. The example of this strategy of Agile project management methods are Adaptive Project Framework [33] and Adaptive Software Development [19].

1. The scope phase will lay out the approach to clear the clouds covering the solution. The objective is to move the clouds and discover what lies behind it. The adaptive SDPM strategy must accommodate development of the known and unknown parts of the solution. Discovering the unknown parts of the solution will require tight collaboration of the
customers with the project teams. Projects adapting Adaptive approach are high risk projects [34].

2. The planning phase of the adaptive SDPM strategy occurs at two different phases.

   a. It occurs at the beginning of the project at a high level. In this planning phase the deliverables are the parameters to define the overall approach of the project, cycle length and the number of cycles that are executed during the project.

   b. In the 2nd phase the deliverables are specific to the next cycle. Detail planning takes place, the tools and process used in earlier projects are used here. The deliverables are the list of functions and features to be delivered, the task level schedule and team assignments and the daily status reports takes place during this cycle [34].

3. The adaptive SDPM strategy differs in principle from iterative SDPM in one important way. In Adaptive SDPM strategy, each iteration presents the customer with complete solution, as well as the results of probes into the missing functions of the solution. The complete solution can be deployed if it is deemed by the customer to have sufficient business value. Most likely that will require several cycles to accomplish. The results of the probes will be to gain further knowledge of the missing pieces and probe again if needed to dig even deeper. Four tasks must be done in the launch phase [34].

   a. Processing scope change requests
   b. Handling solution handoffs
   c. Handling solution rollout
   d. Scheduling iterations

4. There is a marked shift from formality to informality during this phase as we move from iterative to adaptive strategy. In this phase the client is more proactively involved with the development team. The learning and discovery is now used to identify next steps which include not only integrating new functions and features but also probing with new initiatives aimed at further learning and discovery. In the most extreme of the adaptive cases very little will be known about the solution from the standpoint of functionality which has to be discovered but it is not easy to discover without the input and hard work of
development team and client. During the monitoring and controlling phase four elements must be considered [34].

a. Project progress reporting  
b. Discovery of new/revised functions  
c. Discovery of new/revised features  
d. Processing scope change requests

5. The closing phase begins once the customer has signed off that the requirements are satisfactorily met. There are similarities between the closing phases of iterative and adaptive strategies but there are some key differences as well. The similarities centers on satisfaction of requirements at the time of completion. In Iterative strategy the focus was on discovering the details of features that were identified by not accepted by the customers whereas in adaptive strategy the closing was focused on the discovery of features and the functions for later integration into the solution. This is the major difference between the two strategies. The iterative strategy deals in known features that need change and are acceptable to the customer whereas adaptive strategy is seeking new features and functionalities that can be built into the solution. The closing phase not only assesses the results of changes to function and features that were integrated into the solution but also access the discovery and learning from the probative initiatives [34].

3.4.6 Extreme Strategy

When a project is at a stage where there is no solution and the goal is clearly unknown and project is at the edge of the uncertainty and complexity it needs an extreme approach. No one knows for sure if the goal is attainable or not and if there is something to be achievable related to the goal it needs to be discovered and it can be discovered using the extreme approach [34]. It is the most creatively managed approach, as the uncertainty and complexity are at the high levels the failure rates of extreme projects are at highest among the three strategies [33] see Figure 4. Extreme SDPM strategy means entering into the world of research and development where the solution might be quite elusive. Extreme SDPM strategies are designed to handle projects whose goal can be only fuzzily defined or really not defined at all [34]. Example Agile project management methods are extreme Project Management [11] [3].
Figure 4: Extreme Strategy

1. The scope phase of extreme strategy resembles Adaptive strategy. The scope phase includes brainstorming sessions to identify all avenues of possible solutions which usually become concurrent swim lanes in early cycles. The best strategy here is to leave all the options open and not prematurely discard any idea which might lead to a possible result.

2. Planning and launch phases are also the same as adaptive strategy, but as these projects are more speculative there are more expectations of having more concurrent swim lanes at earlier stages.

3. The swim lanes that do not lead to any productive results are cancelled during this monitor and control phases, it is like fishing expedition. If one thing is not working then move on to something else that holds a promise.

4. The closing phase occurs in two phases, as it is very speculative project it might not give any productive directions so it can be cancelled with at least a partial solution or the project might be continued with new funding if any different direction is available [34].

Agile project management emphasizes on the concepts of minimizing focus on short iteration, clearly defined deliverables and directing communication with partners in the development process which helps the product team to adapt quickly to the unpredictable and rapid changing requirements [5]. As we have discussed above, the modern strategies there are several takes on how to best apply the Agile methods as illustrated by the examples, the most important procedure that can be adopted include Scrum and it is the most often used [5]. Due to limited time and resources we will only discuss Scrum method in this research.
3.5 Traditional Vs. Contemporary Agile Methods

The improvement in project productivity can be analyzed by comparing the traditional and contemporary methods. In the traditional methods (linear and incremental), the goal and solutions are known, the final solutions are only provided at the final stage and the projects are scheduled and resources are assigned at the beginning of the project. During the projects the customer's feedback is minimal in linear strategy and changes are not accommodated however in incremental method some of the changes are accommodated after the release of partial solution. In both of these strategies the goals are clearly defined and it does not have any effect of changing environment. At the same time skilled resources and teams can be changed and it has not much effect on the projects. Scope, planning & launch, test and closing phases are all scheduled and any change in the schedule are usually done because of external factors and unforeseen events which also results in passing the deadlines.

Figure 5: Agile Project Management Strategies
The projects where the solutions are partially known, partially unknown or completely unknown are best served by the Agile methods like iterative, adaptive and extreme. Scope change requests are accommodated according to the customers. As the solutions are not known there are more risks and complexities involved in achieving the goals but due to higher flexibility in working methods of these approaches, the projects are dealt with more effectively in uncertain conditions. The roles of the team members are very important and they cannot be replaced that easily. The involvement of the customer at every stage of development results in achieving the goals and satisfying customer needs. These kinds of projects involve minimum planning time and most of the planning is done in time after learning by doing, similarly the formalism has been reduced to minimum in order to take decisions effectively. Both traditional and contemporary approach are used to manage the projects but traditional approaches are unable to deal with complexities in uncertain environments due to which there has been a paradigm shift in project management techniques. The contemporary approaches have enabled project managers to deal with uncertainties and complexities due to their flexibilities hence increasing the productivity of the projects by helping to complete the sufficient amount of work, reducing the unnecessary work and delivering the business value to customers by dividing the scope of the project through small work breakdown structures. The comparison between traditional and agile approaches is shown in Figure 5 and Table 1.
### Summary Table of Contemporary Agile vs. Traditional Approaches

<table>
<thead>
<tr>
<th>Description</th>
<th>Traditional Methods</th>
<th>Contemporary Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Linear</td>
<td>Incremental</td>
</tr>
<tr>
<td><strong>Environment</strong></td>
<td>Unchanged</td>
<td>Unchanged</td>
</tr>
<tr>
<td><strong>Solution</strong></td>
<td>Known</td>
<td>Known</td>
</tr>
<tr>
<td><strong>Schedule</strong></td>
<td>Scheduled at the beginning</td>
<td>Incremental Planning</td>
</tr>
<tr>
<td><strong>Resources</strong></td>
<td>Fixed at beginning</td>
<td>Fixed at beginning</td>
</tr>
<tr>
<td><strong>Customer Involvement</strong></td>
<td>Planning and Launch phase</td>
<td>Planning and Launch phase</td>
</tr>
<tr>
<td><strong>Teams</strong></td>
<td>Replaceable</td>
<td>Replaceable</td>
</tr>
<tr>
<td><strong>Delivery</strong></td>
<td>End of Project</td>
<td>End of Project</td>
</tr>
</tbody>
</table>

Table 1: Summary of Traditional vs. Agile Approach
3.6 Scrum

The Scrum development method was proposed in 1995 by Ken Schwaber. This method is based on work of Pittman and Booch and adheres to the principles of Agile software development [31]. It is simply an Agile, light weight process for managing and controlling software development projects in dynamic environments, as Scrum is intentionally iterative, incremental process that is predicted on a team based approach and one of the major reason for using this process is to control the chaos that can result from conflict of interest and needs within the project development teams [5]. Many processes during the development cannot be predicted and Scrum addresses software development in a flexible way and the only two parts of the process is completely defining the planning and the closure phase. Development of the final product involves several teams in a series of flexible black boxes called “sprints” and no new requirements can be introduced to these teams to ensure the development of product with success in changing environment [31].

3.6.1 Components of Scrum

Scrum Model consists of three main components

a. Role
b. Process
c. Artifacts

Scrum master is the role traditionally assumed by the team leader or facilitator who is responsible for enacting the Scrum vales and practices and removing impediments. Scum master is not a project manager.

Scrum teams are cross functional team of five to ten full time working members. The leadership role within the self-managing teams is not fixed and changes depending on the needs of the iteration (known as sprint) in process of the time.
### Table 2: Components of Scrum

<table>
<thead>
<tr>
<th>3 Roles</th>
<th>5 Events</th>
<th>3 Artifacts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Owner</td>
<td>Sprint</td>
<td>Product Backlog</td>
</tr>
<tr>
<td>Scrum Master</td>
<td>Sprint Planning Meeting</td>
<td>Sprint Backlog</td>
</tr>
<tr>
<td>Development Team</td>
<td>Daily Scrum (Stand Up)</td>
<td>Increment</td>
</tr>
<tr>
<td></td>
<td>Sprint Review (demo)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sprint Retrospective</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Release Planning (Optional)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Backlog Grooming(Optional)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Scrum of Scrums (Optional)</td>
<td></td>
</tr>
</tbody>
</table>

The product owner is typically a functional unit manager who knows how, when and what needs to be built. The product owner has vision of the product and is responsible for profitability and delivery of the product see Table 1.

Scrum process has five basic activities:

#### Figure 6: Scrum Process

In the kick off meeting the group defines the high level backlog for the project and the major project goals. It is slightly different from the sprint planning meeting. Vlaanderen in [31] explained that two types of backlog instruments which are important in Scrum process i.e. product backlog and development sprint backlog. Product backlog is central to Scrum method as it contains prioritized list of items relevant to product and also consists of bugs, customer requests, enhancements, competitive product functionality, competitive edge functionality and technology upgrades [29]. Development Sprint backlog (DSB) is an instrument where each team that is participating in the development process maintains its own DSB and contains all the requirements that are assigned at the beginning. The tasks are then broken down and assigned to specific members. It is fed by the product backlog with items that have been fully specified.
The sprint planning meeting is a meeting of the Scrum team, the Scrum master, and the product owner at the beginning of each sprint (iteration). These meetings, which may take up to a day, consist of two parts. In the first part of the meeting, two major activities occur. First, the group defines the product backlog, which is basically a list of project requirements. After this, the group determines the sprint goal, which is the formal outcome from this particular sprint. In the second part of the meeting, the focus of work is on creating the sprint backlog [5].

The Sprint can begin after sprint planning meeting. Sprint is limited to a month long iteration cycle in which functionality of the product is further developed and no outside interference is allowed with the work of the Scrum team which means product requirement cannot be changed during the sprint. The Scrum meeting takes place every day between the Scrum master and the Scrum team and usually lasts for 15 minutes. Every team member answers briefly about their last Scrum and what they will be doing next and what are the problems. The daily Scrum is not a problem solving session is and only held to collect information about the work schedule and progress.

The sprint review meeting is held after each sprint where the functionality of the sprint is demonstrated to the product owners. This meeting is usually information and is not a distraction for the team members.

Scrum Artifacts is the last major component that includes the product backlog, the sprint backlog, and burn down chart. Product backlog is the prioritized list of requirements for the project and is managed and owned by the product owner. The product backlog is a major deliverable of the kickoff or sprint planning meetings. The Sprint backlog is the subset of product backlog. Unlike the project backlog the sprint backlog is created by the Scrum team members. It is updated every day and contains not more than 300 tasks and the team can breakdown the tasks if it is more than 16 hours of work. Scrum intentionally focuses on working through the use of burn-down charts. Burn down charts provides information easily and in comprehensive manner. Three types of burn down charts are commonly used; sprint burn down chart documents the progress of the sprint, the release burn down chart documenting the progress of the releases and the product burn down chart documenting the overall project progress [5].
3.6.2 Agile Principle vs. Scrum

Scrum is an Agile software development method and follows the Agile principle presented in subchapter 3.21. These methods are used to accomplish the Agile goals. We prefer Scrum because of its strong focus on self-managing teams, daily Scrum measurements. How Scrum is aligned to basic principle of Agile methodology can been visualized by the Table 2.

<table>
<thead>
<tr>
<th>Agile Principle</th>
<th>Scrum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Satisfy the customer</td>
<td>Scrum Pre game planning</td>
</tr>
<tr>
<td>Welcome changing requirements</td>
<td>Scrum sprint planning</td>
</tr>
<tr>
<td>Deliver working software frequently</td>
<td>Scrum sprint</td>
</tr>
<tr>
<td>Motivate Individuals</td>
<td>Scrum master firewall</td>
</tr>
<tr>
<td></td>
<td>Blocks gone in one day</td>
</tr>
<tr>
<td></td>
<td>Common room</td>
</tr>
<tr>
<td>Communicate face to face</td>
<td>Scrum meetings</td>
</tr>
<tr>
<td>Maintain Constant Pace</td>
<td>Look priorities in Scrum meetings</td>
</tr>
<tr>
<td>Sustain technical excellence and good design</td>
<td>High level design and daily build of software</td>
</tr>
<tr>
<td>Keep it simple</td>
<td>Fast decision</td>
</tr>
<tr>
<td>Empower self-organizing teams</td>
<td>Self-managed teams of 7 -8 people</td>
</tr>
<tr>
<td>Reflect and adjust continuously</td>
<td>Sprint review</td>
</tr>
<tr>
<td>Interact frequently with stakeholders</td>
<td>Planning and Feedback Meetings</td>
</tr>
<tr>
<td>Measure by working software</td>
<td>Quality Meetings</td>
</tr>
</tbody>
</table>

Table 3: Agile Principle versus Scrum
Chapter 4 Empirical data: State of the art in project management

This section presents the empirical data which has been collected through interviews to find out the best practices of Agile thinking in software development project management.

To connect the theories with the practicalities, interviews have been conducted to find the best practices within the Ericsson. The interview has been conducted in Ericsson product development unit (PDU) in Sweden and China with Release Project Manager, Feature Development Program/Project Manager, Release Line Manager, and Feature Development Line Managers and with an Agile Coach.

First interview was with Bosse Wesshagen, project manager for feature teams of Session Border Gateway (SBG) at Ericsson. During his projects he has gained the experience of working in Agile teams and how to manage the Agile teams in Scrum way.

The second interview was conducted with Lucia Suarez, Research and Development (R&D) release project manager. Her current assignment is to assist MTAS (Multimedia Telephony Application Server) release project manager, MTAS product introduction manager and Telephony Evolution project manager. She has gained the experience of working with 3rd party products, working externally with other actors and experience of working internally with own designs in projects with MTAS.

Third interview was conducted with Mattias Åhlander, Project Manager (R&D). Matthias’s current assignment is as an Agile coach. He is coaching several Agile teams. His current assignment is to coach the software development teams, test teams and leadership team of product development unit in Ericsson.

Fourth interview was conducted with Doris Yang. She is the Line Manager of feature development unit at Ericsson China. Her current assignment is line manager responsibilities along with Agile coach. She has worked as a software developer, team leader and Scrum masters role.
Fifth interview was conducted with Janos Harris. He is Line Manager of feature development unit at Ericsson Sweden. His current assignment is line manager activities he also act as product proxy owner backup as per needed. He has also worked as Scrum master and team leader.

Sixth interview was conducted with Justin Lawless. He is Release Line Manager at Ericsson Sweden. His current assignment is line manager activities he is also a single point of contact (SPOC) for Session Border Gateway (SBG) node.

4.1 Interview with Project Manager

Boose Wesshagen described the working methods in project management as doing the right things in the right way with the right people. Project management contains many areas like planning, estimation, execution, monitoring, risk management, etc. It is quite a complex and big system of methodology to ensure project success. However Ms. Suarez does not see project management as a work method. Organizing work in a project could be considered a work method and if the project is big enough then a project manager can be needed in order to help the project achieving its goals. The project management would rather be a tool than a work method.

4.1.1 Development and Evaluation Phase

From Agile way of working, setup a feature backlog at the beginning and then divide the whole project into several sprints. For each sprint, select several story points from the feature backlog, called sprint plan. At the same time, the whole team will define a definition of done. For instance, to one user story, if completed the implementation, the code should be reviewed, and after it is tested by both block testing and function testing, it is agreed that this task is done. So after each sprint, team will go through all these tasks to see the progress and quality. There is a retrospective meeting after each sprint to make an evaluation and reflection of this sprint and perform better in next sprint. Besides this sprint-based mechanism, there is a Status Point (SP) process, which divides the whole work package into several stages, for each stage many criterion should be fulfilled. This is another kind of evaluation during the development phase. There are check points where the project’s ability to achieve the goals are evaluated by an external (to the project) quality person, also called tollgates (although not any more in the IMS organization). The evaluation of the project is usually done by the design owner however in some cases project manager or the release manager will take care the delivery time and over-all quality. Total Project Manager (TPM) and
Technical-Coordinator of Session Border Gateway (TC-SBG) will be in charge of technical detailed solution within the projects. All participants take their own responsibility and collaborate together to manage and control the whole project. In some projects, Project Steering Group (PSG) evaluates and controls the project. It usually includes people representing the product owner, the design owner, the line. Basically, those either involved in the project or affected by the result.

It is mentioned before, team will create this feature backlog at the beginning, all the team members contributes their opinion on this and during the beginning and end of each iteration, and team will review the iteration plan and do the retrospective meeting together. All these scenarios will give opportunity to check if team members have common vision or not. In practice, it depends on the size of the project and the level where the common vision is established.

Activities in projects are performed through daily meetings every morning. Action for each person in the team is then assigned. The status of each task is checked in next daily meeting. During the daily work, if another problem is found, it is shared within the team immediately. Action will be taken as soon as possible to meet the time frame.

According to Boose Wesshagen, they use a combination of Scrum and XP methods from the Agile. It depends on the real situation of the team and also the project status. It is quite flexible and adjustable. The most common rules used are the quick response to change and short-time iteration delivery. However Ms. Suarez works with the release project, where she converts Agile development from the Feature Development Center’s (FDC) to a waterfall release of a product that Strategic Product Manager (SPM) can sell as a whole. What we would like to see from Agile is the smaller deliveries of full working functionality and fully empowered teams that make own decisions (not quite there yet on any of them). Scrum is the most common used way of working in Ericsson IMS, some team also uses the XP and Kanban methods.

The major problem is that the problems are that the Product Management does not work in a very Agile manner. They still tend to define a whole release. The scope of a feature is to a large extent set from the beginning and to some extent the time as well. The commitment process that does not fit very well with Agile working. This process may change though. Some of the problems are as follows:
a. *Delivery deadline is not well described in the Agile process* i.e. how to meet the deadline efficiently. The process itself is just providing methods to do the team daily business. But when the deadline pressure is coming, prioritization should be used to make decision.

b. *New team members’ competence building.* This is always a problem when the project’s deadline is near. How to balance the investment between involving new members to bring him/she up to enhance the competence force or use this effort to concentrate more on the project itself. Sometimes companies face the problem of not enough mature teams.

### 4.1.2 Value

Everything that contributes to the project goal in an effective way adds value to the project. The value of the project is stated by product management in the project assignment. Project Portfolio department stands for outside customers and they will specify and assign work package to the feature development center. They are also taking responsibility of make an evaluation on the quality of the outcome from FDC.

The waste of different kinds: studies that are not implemented, parts of feature implementation that is not finalized, problems with environment, lack of efficiency (for example, badly prepared meetings, “parked” work packages that must be re-started later) all adds up to non-value adding things in project management.

Work outside the feature is measured by each team. Lab Environment problems are measured by most teams. The waste mentioned above is unfortunately not measured. No measurements that managers are aware of today. But they are following up the final costs of a work package compared with the expected ones at the end of the project and in cases of big discrepancies; a root cause analysis is done.

The capacity measurement is taking place on the implementation time for a feature and compared with quick study estimate, and pre study estimate. Also at both man hour consuming and the iteration spent. For the man hour Ericsson has a system to report all the consumed man hour for the whole project; it also impacts the budget for the future. For the iteration number, there is a SP process to control and measure the whole project progress.
To some extent Ericsson share information if a customer asks. Information is shared with external parties on a need-basis on how the project is being carried and its status. Work flow for the projects in Ericsson is created using SP checklist which divides the whole work package into different stages, SP start SP Commit, SP iterations, SP done, SP close. Based on established ways of working, the work flow for a specific project is based on the project anatomy, developed as work packages are scoped in. Once the work flow is created teams do the sprint planning. Feature backlogs are present to follow up task status. Teams try to stay on time and waiting times are eliminated from the project through planned studies so that there are always assignments for the teams. Sometimes a few may solve TR for some time.

a. Prioritize resource to solve the most important tasks in the project.

b. Run the tasks in parallel to utilize the gap between

c. Make a plan in advance and do a risk management as soon as possible.

d. Work packages are scoped in before the team is ready with the previous one. Each team has a percentage of time dedicated to improvements and corrections.

Ericsson has a quite well-established system for different roles and responsibilities for project management. However the border lines between some roles are fuzzy but all the team members come to know about the project through project meeting and information via Scrum master.

4.1.3 Quality

Ericsson Project Management has proper ways of working for quality measurement. There are several checkpoints at each stage of project. At each checkpoint there are a number of quality related questions e.g. whether the release is performing same as the previous one or better than it. Another quality measurement is number of faults detected. It is presented at checkpoints in the form of software maturity curve that depicts tendencies of amount of faults and predicts the remaining amounts to be found. At Ericsson the quality is checked through various procedures and at different stages e.g. number of trouble reports (TR’s) is measured. Test suites run every night. This test includes Unit testing, Block testing and functional testing. LSV team test results: daily testing, legacy testing, KPI (robustness and characteristics) testing and early system test. System
test results: Security tests, Upgrade tests, performance and integration testing. Jointed testing activities and its results: Network feature testing, network system testing, Tier 1, Tier 2 testing.

Project team’s work can be improved by retrospectives and case by case.
   a. Learn and improve every day.
   b. Retrospective meeting after each sprint
   c. Add small percentage of improvement work in each iteration

The delivery of the product is usually decided together by the customer and the project manager depending on customer’s requirements and also the project team’s competence, delivery plan. In Ericsson, feature teams get these requirements from the release manager from product release department. However in some cases it is decided by the person who orders the project or the project sponsor.

4.2 Interview with Agile Coach

According to Mattias Åhlander, Ericsson Sweden AB is not mature in Agile yet. There are group of people are acting as an Agile coach and trying to implement the Agile methodology throughout the organization. He said that most of the leadership team has Kanban as an Agile methodology. But they do not have proper visibility of backlogs as they are doing several things in parallel and that creates the problem while prioritizing the backlog also most of the leadership team do not have product owner role.

According to him the strategy behind the Agile, mission and vision is not very clear is each and every one. In order to follow Agile what to change and how to change have been big problems as Agile transformation solves the process oriented but all the problems are not always process oriented.

He said that Adaptive strategy is mainly followed in product development units (PDU) in Ericsson. Ericsson also uses Extreme strategy in its R&D units, when there is no solution and the goal is clearly not known.

According to him most of the feature development teams are following Scrum and Kanban, but they are not following it completely. When it comes to Scrum roles they do not have Product
Owner role However, test teams are following Kanban. The implementation way of Agile at most of the software development units at Ericsson differs to theory by means of Agile role, artifacts and events but follows core Agile principle.

When it comes to customer collaboration the development teams are bit far away from customers as it is very long feedback loop which is really a big problem, mature products has feature development and release units are at different sites of Ericsson.

Ericsson still has project managers and project management unit because they have not been much involved in the Agile transformation yet, however Ericsson in Finland and Italy doesn’t have this role, but in order to implement complete Agile transformation there is need to have more reorganization which is not easy. However they are trying different changes in organization and checking if works or not. They believe that according to Agile, changes are welcome so it is also acceptable to have reorganization. When it comes to measuring the velocity or capacity of the team development teams are not using user stories instead they still use man hours. They do not have proper estimation of it.

He said that by adopting Agile Ericsson has become more been more cost to revenue oriented. The new features are prioritized based on its worth e.g. increased revenue or market share. At Ericsson it is Strategic project manager role is to make sure that the Agile project team is not devoting too much into the development of the new solution. If they do so then product development will cost more than gain. However the project manager at Ericsson focuses on project costs, the strategic project manager focuses on the total cost of ownership that includes development costs as well as the cost of operating the system after it is deployed.

He said that by using Agile resources are much optimized. The visibility of backlog, contribution of individual has become visible and velocity of development teams has been increased. He said that Agile transformation in bottom up in the company and line managers should promote Agile more.
4.3 Interview with Line Manager

4.3.1 Interview with Feature Development Unit Line Manager

According to Dorris Yang, most of the development teams follow Scrum and each Scrum teams has Scrum masters and functional testers. The Scrum teams normally follow all the events and artifacts of the Scrum and follow the principle of Agile. However, when it comes to role and responsibilities they are not fully Scrum. They do not have product owner role in Ericsson china but they still do have the team leader role, because job roles in Chinese culture are a bit hierarchical and people are not so empowered as they are used to have team leaders or someone to have command and control and team has feeling that Scrum master doesn’t have authority to lead the team. Most of the team is far away from self-managing teams in Ericsson china. The project/program manager is in charge of project delivery and execution. Program manager is responsible for quality during entire project life cycle. She said according to Scrum we should not have this role but Ericsson’s project management unit has been doing very well and known for their excellence, so they still have this trend. Also it is not easy to transform this role to someone else.

In Ericsson Agile teams are long living; the development teams are responsible for development as well as maintenance. In traditional waterfall method the after development phase project was closed and maintenance was another project and was not done by the feature development team but in Agile feature development teams are still responsible for it and the project is long living, because in practice for maintenance team it is not easy to understand the complete feature and code which is written by some other team.

The development teams are far away from the customer collaboration because most of the product line sits at Sweden and it is very long feedback loop. When it comes to measure the capacity or velocity of the team they are not in fully Scrum yet because most of the team doesn’t follow it on the basis of user sprint story points they still follow the man hours, the disadvantage is that some people could pretend that they are busy but actually not.

The performance indicators in the development projects are seen weekly program alignment meetings over there they discuss about the trouble reports, integration speed, customer trouble reports backlog, automation rate, unit test coverage.
When it comes to problems to implement Agile she said that it *is not easy to change mind set* but they have been bit successful in it. Some people thinks that *rules not easy to implement*. People *break the rule* very often and excuse that it will not work in Scrum without practicing it. However, the rules are important to follow in order to have better end result. She also told that other problems could be arise during agile implementation *e.g.* resistance to collaboration, low-trust environment, reluctance to change, inflexible management hierarchy, less development experience and lack of automated tool support.

The strategies behind to follow Agile at Ericsson is to *respect people and continuous improvement, individual interaction, proper documentation, reduce waste and deliver valuable things, see top prioritized backlogs, speed and quality.*

She said that in very few time spans they have found that productivity and capacity has increased in Ericsson China. The development team *has become empowered and people have become visible.*

According to Janos Harris, feature development unit at Ericsson Sweden follows mostly Kanban and Scrum and follow the Scrum principle, but they are in *transition phase* and they have not been completely Agile yet.

When it comes to the Scrum roles they *do not have product owner role* in the development teams but they do have proxy product owner role, who is beside the team and communicates to external interfaces and helps teams as per needed. Sometimes line managers also supports the team as per needed. Apart from proxy product owner role the development teams do have Scrum master and team itself. Teams are *self-managing and doing pretty well* at Ericsson Sweden.

When it comes to customer collaboration they are far away from the Scrum rules, because the requirement are not coming directly to the feature team and it has a *huge feedback loop.*

The flow of the project starts from customer unit and then it goes to the project portfolio office where quick studies of the features are done and features are put into feature pool and then there is a meeting where proxy product owner, feature team line managers, release project manager, Strategic Product Manager decided which feature should be prioritized as per market and customer needs and then line manager of feature development unit creates the team and assigns the feature to the feature development team and then pre-study is done and divided into user stories and
sprints are created and they do continuous integration and tested on daily and weekly basis by testers by legacy (LSV) team and follows until last function delivery of the project.

When it comes to measuring the capacity and velocity of the team they do not use user stories point as they are not completely Scrum, they use man hours instead. Agile at feature team focuses on long living team and promotes the team performances. He said that Agile doesn’t help much in project management as its organization is undefined in Agile. It doesn’t gives more information about check point and check list.

He said that Agile has helped them to increase in productivity and prioritize the backlog, since it has become much visible by Agile. Also Agile has helped a lot when it comes to measuring the capacity and velocity.

4.3.2 Interview with Release Management Line Manager

According to Justin Lawless release management unit at Ericsson Sweden follows mostly Kanban and partially Scrum. The teams in release management are quite experts and specialized in their area (System Test and Network System Test). They can be seen as group of individuals in practice rather than as a real team. But they have common group that test the each release of the project and they share a common Kanban board. They do have Scrum master and team role and proxy product owner roles. However they get supported by the release project manager.

He said that the Agile team still needs project manager as team by themselves cannot manage everything. It is impossible to deal with different releases ongoing at same time without project manager, as there is a very much project administrative work and too much management with a lot of external things. He said that some of the product has tried to remove the project management role at beginning of Agile transition but they didn’t succeed to manage and went back to have project managers role.

The release team does almost tests same tests in each and every release also takes care of external trouble reports. The task of sharing the customer trouble reports are normally shared by the team. The team doesn’t follow much retrospective meetings.

He said that they are trying being more Agile and changing their roles in the team but due to specialized in one area it is bit difficult for them but they have started competence development in
each other areas and created backup for each other. They are quite closely to their customers both internally and externally because they are in release organization and responsible for quality. The team measures their capacity in man hours because they are not feature development teams. When it comes to the productivity they haven’t seen much difference in Agile adoption except the transparency of each other’s work and prioritization of tasks. This is because it the team of experts and they all team members are self-going individuals.

According to Justin the productivity in Agile in release organization has not seen as an advantage of Agile yet but backlogs has been more visible.

He said that in Agile there is no downtime during the project which is not really good for team. In the traditional setup it was quite feasible that when the project get completed the resources (team) gets back to line and when the project starts it takes a while and that time people has a but relaxed time which was good for resources and they were always active and motivated. However in Agile there is no downtime and people gets tired by continuous pressure and stress. Agile is much on resource optimization.
Chapter 5 Analysis and Discussion

This chapter will provide analysis of empirical data and theories in order to answer the research questions.

Kettunen in [42] explained that companies are operating in uncertain and dynamic environments and many organizations have adopted the Agile methods without having proper knowledge. However, empirical data in shows that Ericsson is working in a very dynamic and competitive environment and is following the Agile approach with full knowledge as the interviewees from Ericsson believe that project management is a way of doing the right work with right people at right time. It proves the fact that companies have adopted modern methods to address the need for new post mass production competitive environments [13].

Dealing with a relative unstable organizational environment is a serious challenge for the managers of any software development project, most real world development efforts are much more likely to be conducted in more unstable environments as organizations adapt to moving technologies and markets [2]. However observing the trends for software requirements to change the underlining processes, it can be characterized as multi-level, multi loop, multi agent feedback systems. Software developers have responded to this complexity with iterative, often ad-hoc approaches. Recently, Agile development methods including extreme programming (XP), Scrum, adaptive software development methods and feature driven development has sought to focus on rapid iterative deliveries and flexibility [1]. Scrum is an Agile project management is a comprehensive project management approach which includes flow and value generation, management as organizing, and language/action prospective and scientific experiment model and also referred this approach as paradigmatic transformation of the discipline of project management [39]. IT companies are successfully using Scrum to improve process and Agile approaches are delivering key business values faster and cheaper in response to not just a rapidly changing environment but also saving costs in mature environments.
5.1 Concepts of Agile Thinking Relevant for Project Management

Theoretical and empirical evidences showed that iterative, adaptive and extreme strategies are the most relevant strategies for Agile project management and under these concepts many approaches are used.

Iterative strategy is used when the requirements are uncertain, unfinished, or matter of change. During the development phase, high level plans without much detailed information is made at initial level and its functionality is designed and developed during the first iteration.

Adaptive strategy is used when the business requirements and solution are uncertain and the schedule is tight. Adaptive approach is more suitable when the solutions are partially known. More risk, uncertainty and complexities are present when solution is little known and to deal with this situation continuous change process is needed. The adaptive approach is also learning by doing approach but in this approach customers are more involved during the development phases. Planning is usually done just in time as per change in environment because it is considered to be a non-value adding work due to the unknown future.

It is mostly used Agile approach which is not only suitable for software development projects but for all kinds of projects. Ericsson is using adaptive strategy in most of its product development units (described in section 4.2).

Extreme strategy is used to provide solution when there is no solution and the goal is clearly not known. Extreme strategy is mainly used when big changes in project are expected, high speed is required, and uncertainty on requirements exists. It is considered as most creatively managed approach in uncertain and complex environments and has the highest failure rates. Ericsson is using this approach when there is plan of move to new platforms, as there is a lot of research work is required.

The Agile methodology is successful because nowadays requirements are very unstable. It uses iterative and incremental process where project stakeholders dynamically work together to understand the business domain, figure out the business requirements and prioritize functionality. Agile methods are used when worth of the project is significant even though when business requirements are not clear, customer can contribute in the project development phases, technical
experts are co-located, incremental feature based development is feasible, and visual documentation is appropriate.

Theoretical and empirical evidences have shown that Agile thinking is mainly used to resolve key challenges that negatively impact the development projects. The Agile methodology provides the techniques to resolve the major business challenges for the development projects. Agile provides solution to challenges to project management is shown is Table 4.

**Agile Solution to Project Management**

<table>
<thead>
<tr>
<th>Challenges to Project Management</th>
<th>Agile Solution</th>
</tr>
</thead>
</table>
| Requirements are volatile and difficult to fasten | Active Customer/Stakeholder involvement as a project team member  
Collective requirements, design and prototyping sessions  
Usage of envisioning and prototyping tools  
Team explanation of requirements with flexible scope change  
Iterative approach allowing key components to be developed to address |
| Deliver Project/Business value more speedily | Prioritization of needs based on business goal  
Timeboxing to fixed dates and fixed cost  
Incremental delivery of highest priority project components first  
Decomposition of large targets into multiple releases  
Small, dedicated, co-located teams in teaming environment |
| Decrease risk of take up new technologies | Designers participate and directly to the developers  
High risk proof of concepts are analysed early in the project  
Right skills are devoted to the project team |

**Table 4: Agile Solution to Project Management**
5.3 Agile in Practice

The interview result has shown in *Chapter 4* that how an Agile practice at industry differs than the Agile in theory however the Agile in practices follows the core concept is same as it is in Agile manifesto and Agile principle (*described in section 3.2.1*). The case of Agile in practice from the project management perspective can be seen as follows:

<table>
<thead>
<tr>
<th>Role of Project Manager</th>
<th>Agile</th>
<th>Agile in Practice (Ericsson)</th>
<th>Traditional Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do not need Project Manager</td>
<td>Project Manager is able to empower and shepherd the development team, and keeps team focused and fruitful</td>
<td>Project Manager often becomes an un-empowered project administrator</td>
<td></td>
</tr>
</tbody>
</table>

| Planning | On-time, next iteration focused, design and usability often fall through the cracks | End-to-end planning for Agile technique usage with involvement of key professionals/experts | Commit too early to levels of detail that are unknown throughout waterfall phases |

| Risk | Many missed due to lack of planning | Risks are identified and reduced by proactive out of the box opinions | Frequently accepted and reacted to and allowed to unfavorably impact delivery |

| Problems/Impediments | High focus on impediments removal on a daily basis | Impediments are discussed on daily standup meetings and actions to resolve it | Impediments identification lags the need for the team to have it fixed |

| Status | Burn down chart of effort within an iteration is often the only stakeholder level status provided | PM is always aware of status, and provides summary to Stakeholders on a weekly basis including a burn down of story completion against the iteration baseline. | Something the Project Manager records during their one weekly meeting with the team |

| Scope Management | Goals for one month iterations and sprint selected from backlog of requirements | Baseline three iterations, then allow variation of prioritized scope within fixed deadline and budget | Change to scope changes deadline and budget |

Table 5: Agile in Practice
The empirical evidence (*described in section 4.3.2*) has shown that the project management is still required. The project must be aligned to the business management system and for that there is always need of project management and project manager. Agile Projects needs some kind of Project Management System tailored the existing project management methodology. Agile projects typically form a component of a larger project which needs a light Project Management system at the Scrum level and majority of the Project Management parts resides at the overall project level.

### 5.4 Improvement in Project Productivity

The strategy behind adopting Agile is to have continuous improvement, individual interaction, reduce waste and deliver valuable things, see top prioritized backlogs, speed and quality. The empirical and theoretical evidences have shown that how Agile implementation helps in the productivity. It mainly helps in resources optimization; visibility of team backlog and visibility of contribution of individuals in the team. The teams those have implemented Agile correctly and in mature stage they have gained in capacity and velocity. By implementation of self-managing team concepts the team has become more empowered, team members has become motivated, accountable and visible. People who do not speak and give the opinion they started talking and became able to think and give their opinion, which is very good for the team as team get more views (*described in section 4.3.1*).

The projects are evaluated continuously through attending of daily meetings, through weekly reporting or sometimes via escalations. Scrum projects are usually evaluated after each sprint. At beginning feature logs are set and project is divided into several sprint plans, the quality is checked at the end of each sprint. However, in the projects there are checkpoints to check the quality but usually quality is evaluated externally. When the projects are being checked at every stage of development the non-value adding activities are removed like in iterative approach documentation is reduced, planning is reduced in adaptive strategy and swim lines which do not promise any solution are cancelled. Removing the non-value adding activities, project teams increase the project productivity (*described in section 4.1, 4.3.1*).

In Agile (Scrum), the impediments are visualized on day to day basis and action taken to remove the impediments is also taken on day to day basis which helps a lot to the planning and velocity of the team. The delivery capacity of organizations is increased using Agile methods as well as
smaller organizations can easily manage their projects. Regular meeting on daily basis helps to identify daily task priorities which results in helping organizations to identify the needs and save valuable time of the teams. Besides delivering, for customer value Agile methods helps technical experts by encouraging exploration and simplification of practices through adaptive teams. The Agile approaches are developing cooperative organization culture by replacing the hierarchical culture as verbal communication is bringing people face to face, facilitating more collaboration. As the work is scheduled on regular basis there are no as such overtime costs and projects usually have small teams and have upfront cost estimations. The comparison of project productivity of traditional and agile approach can be seen in the Table 6.

Comparison of Project Productivity of Traditional & Agile Approaches

<table>
<thead>
<tr>
<th>Descriptions</th>
<th>Traditional Approach</th>
<th>Agile Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation</td>
<td>At the End of Project</td>
<td>Continuous Evaluation</td>
</tr>
<tr>
<td>Quality</td>
<td>At the End of Project</td>
<td>At the end of each Sprint</td>
</tr>
<tr>
<td>Task Priority</td>
<td>Scheduled at the beginning of project</td>
<td>Set during the daily meetings</td>
</tr>
<tr>
<td>Documentation</td>
<td>High Documentation</td>
<td>Minimum Documentation</td>
</tr>
<tr>
<td>Activities</td>
<td>High Non Value adding activities</td>
<td>Reduction in Non-value adding activities.</td>
</tr>
<tr>
<td>Teams</td>
<td>Large teams</td>
<td>Small Teams</td>
</tr>
<tr>
<td>Project Delivery</td>
<td>Long delivery times</td>
<td>Short Delivery time</td>
</tr>
</tbody>
</table>

Table 6: Comparison of Project Productivity of Traditional and Agile Approach

5.5 Problems Faced During the Agile Software Development

The empirical evidences have shown that Agile practices in the industries are not completely Agile or matured (described in Chapter 4). The challenges in implementing agile are as follows:

a. Resistance to collaboration: People are not willing to collaborate as there is much interaction and collaboration required in agile. This might be due the organizational culture which they are
used to or people are too busy or when there is low trust between people and team and organization.

b. **Reluctance to change:** People are not willing to change their ways of working as they feel that they are doing good or used to of working in the older way.

c. **Inflexible management hierarchy:** Some organization is too much hierarchical and there is people are less empowered and they do not want to take much responsibility due to their hierarchical culture in organization.

d. **Less development experience:** Organization where developers don’t have much experience it is difficult to implement agile there as people don’t feel to be empowered due to lack of experience and don’t wish to take decision.

e. **Lack of hardware, automated tool support:** There is need of new hardware’s, tools and support for continuous integration and organization does not have money for that as it is expensive to have it.

f. **Mind Set:** People think that they are doing well with non-Agile approach. To implement Agile requires more than a set of practices; it needs changes in thinking (*mind set*).

g. **Misinterpretation of Agile:** The major risks for Agile implementation are not processes or tools but it is fear of transform and misinterpretation Agile values. There is problem when teams are not clear with what they would like to change and how they should adopt the transformation to Agile.

h. **Customer Relations:** In Agile software project management the customers are very much involved and bad customer relation can be a big problem. Sometimes it is difficult to have proper customer collaboration when it comes to internal customer, as normally there is a long feedback loop even for internal customer. People are sitting at different sites in large multinational companies. This problem can be solved by having more end to end cross functional team and the team will have more responsibilities and it will remove some unnecessary deliveries and over there they will improve in internal customer collaboration side.

i. **Scrum-Buts:** There are a lot of Scrum Buts are in Agile practice (*Scrum*). People don’t follow Agile method correctly and regularly. This normally happens in the feature development and test teams that they break the rules of Agile methods (*Scrum*) and as a result they have a lot of problems in continuous integration. The reason behind it is that normally people are over loaded and do not wish to put effort on the change, rather than fixing their current work and
also many people don’t understand the agile concept properly. People think that Agile works well in small companies; it is not for industries where have very huge product line. It is very important to follow the Agile method correctly throughout the organization in order to have gain in productivity and better advantage of Agile.

The empirical evidences have also shown some limitations of Agile methods are as follows:

a. **No Downtime for Employees:** Agile has a negative point that there is no down time for employee and people gets burned out soon. However in the traditional way of working when the project finishes the employee gets a down time for a while in the beginning phase of new project. It is very important for the leaders in organizations to create such a working environment in the organization where employees can also get relaxed in order to promote such transformation.

b. **Undefined Organization:** Agile doesn’t help in core project management as organization is not well defined in agile. It doesn't gives much information about check point and check list.

c. **Deadline is Undefined:** Delivery deadline is not well described in the Agile processes i.e. how to meet the deadline efficiently. The process itself is just providing methods to do the team daily business.

d. **Competence build-up for new employees:** There is problem with new team members' competence building when the project's deadline is near and experts are always involved in the project work and don’t have time to help new employee and in agile there is no downtime for such activity. How to balance the investment between involving new members to bring him/she up to enhance the competence force or use this effort to concentrate more on the project itself is not easy in agile.

Necessary skill sets and project management competencies play a huge role in the success of a project. However sometimes organizations lack correct Agile practices and inappropriate technology and tools due to which they face problems. Sometimes organizations face issues right from the beginning of the projects. If the scope of the project is not properly defined and the requirements are not properly identified, teams will not be able to do the project planning which will ultimately result in lack of Agile process tracking and ultimately resulting in the failure of the project.
Chapter 6 Conclusion

This chapter will provide the final conclusion on the principals of Agile software development project using the Agile approaches and conclude the findings of this research work.

Traditional formal methodologies of software development have become a mismatch with the dynamic environments and have been replaced by the modern contemporary Agile approaches of software development as organizations adapt to work in more volatile environments. During current times in fast pace market environments Agile approaches have given organizations a competitive edge as well as survival in the competitive markets. To develop the products more quickly and on time according to the needs of the customer, Scrum has enabled development teams and the product owners to monitor and evaluate the product functionality at every stage of the development, which results in arriving to the solution. The purpose behind following Agile approach is to meet requirements of the changing environments along with creating value for the customers by evaluating quality through monitoring every stage of development.

This research work was conducted to identify reason to adopt Agile methodology and which Agile methods are currently used by software development teams in the industry, how Agile methodology increases the productivity and problem faced during Agile implementation. The aim behind adopting Agile is to have continuous improvement, individual interaction, reduce waste, deliver valuable things, see top prioritized backlogs, transparency of work, low cost, less down time for employees (resource optimization) and achieve high customer value, high speed and high quality (described in section 4.2). There are several Agile methods are followed in industry e.g. Scrum, Kanban, Extreme Programming. The most commonly used Agile approach is Scrum, however other approaches like Kanban is also very much in use.

Industries are currently using Scrum approaches and due to maturity of the organization and teams they are also applying some lean techniques like fountain methods in order to meet delivery dates, even in the release projects Agile approaches are being used not just for the product development
but also for its evaluation. Industries have gained in productivity and quality by resource optimization, continuous evaluation of the project, transparency of work, monitoring quality on regular basis and short term deliveries. By adopting Industries has become more cost to revenue oriented. The new features are prioritized based on its worth e.g. increased revenue or market share. This is really good for profitability. The Agile adoption at Industries is not matured yet but they are finding ways to do it. Industries are trying to follow Agile throughout the organization. The Agile implementation approach is more likely bottom-up approach in Industries. They are continuously trying different ways of working, trying to grow cross functional team and reorganize the team in order to implement the Agile, increase productivity and quality (described in section 4.2, 4.3.1).

The empirical evidences have shown that Agile practices in industry are not very mature yet (described in section 4.2). The Agile practices in the industry differ from the theory to some extent but the core concepts are same. The implementation way of Agile practices differs to theory by means of Agile role, artifacts and events but follows Agile principle. Also there are several team and organization where Agile methods don’t align as per the theory. This is due to the organization culture, maturity of current ways of working and lack of resources. However, it can be followed by aligning business requirements and Agile principle in correct manner which requires time to figure out the gaps which needs a lot of time, resources and correct understanding of the Agile theories. The organizations must fully understand the Agile process before starting the Agile practices to enjoy higher success rates and quality deliveries.

6.1 Research Limitations

Qualitative research has been conducted by interviews of professionals of the IT/Telecom industry. However there were some limitations which may cause some differences in results of our research.

- This study emphasizes on the identification of key practices of organizations of Agile software development methods. In depth study of each and every practice has not been done.
- The information has been collected from only one Company.
- Information from people using different project management techniques has not been gathered.
Due to the limited resources and time only six interviews were conducted.

6.2 Recommendation for future research

This research was conducted in the context of identification of practices of software development companies and how they are operating in current unpredictable and uncertain dynamic environments.

Qualitative research can be extended to more professionals, working in related markets through interviews, so further depth of discussed factors can be obtained and understood.

- Future research can explore more of these factors and new techniques that may be used.
- More questions can be raised like solutions to problems, pressure faced by the team members, team coordination etc.
- A detailed study can be done on each of the Agile strategy that has been discussed here.
- Future researcher can discuss the benefits of Agile software development projects.
- How many organizations have benefited from this Agile development?
7 References


Appendix

Interview Questions

- What is your designation and what are your assignments?
- Years of experience in this field?
- What experiences you have gained from projects? General

Project management
- How do you describe project management as a work method?
- Who decides the project delivery date?
- Who is responsible to evaluate and control the project?
- How do you evaluate a project during development phases?
- Do you see project team members having any common vision?
- How activities are performed in projects?
- What strategies of Agile are important, which one is the most commonly used?
  - What methods are most commonly used, Scrum, xp, adaptive etc.
  - What problems are usually faced during Agile process?
- Who specifies the value of a project?
- What is value adding in a project?
- What is non-value adding in a project?
  o How do you measure?
- What all, you measure about projects? (Hours consumed? Number of iterations, Etc?)
  o Do you use any special tools or methods for measuring?
- What is your opinion about activates, of Scrum.
- How activities are sequenced in projects?
- Who determines when a specific task should end?
- How decisions are made?
- Do companies share information of in house process with partners and suppliers?
- How work flow is created in projects?
- How project team members come to know what activity is performed before and after their task?
- How waiting times are eliminated in project management?
- Do you see clarity of roles in project teams?
- How the project team members come to know about the project
- How quality is measured in projects?
- Who measure quality?
- How do project teams work to improve?
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDC</td>
<td>Feature Development Center</td>
</tr>
<tr>
<td>SPOC</td>
<td>Single Point of Contact</td>
</tr>
<tr>
<td>LSV</td>
<td>Latest Software Version</td>
</tr>
<tr>
<td>PSG</td>
<td>Project Steering Group</td>
</tr>
<tr>
<td>PMI</td>
<td>Project Management Institute</td>
</tr>
<tr>
<td>PDU</td>
<td>Product Development Unit</td>
</tr>
<tr>
<td>PMBOK</td>
<td>Project Management Body of Knowledge</td>
</tr>
<tr>
<td>APM</td>
<td>Agile Project Management</td>
</tr>
<tr>
<td>RUP</td>
<td>Rational Unified Process</td>
</tr>
<tr>
<td>DSB</td>
<td>Development Sprint Backlog</td>
</tr>
<tr>
<td>TC</td>
<td>Test Coordinator</td>
</tr>
<tr>
<td>SPM</td>
<td>Strategic Product Manager</td>
</tr>
<tr>
<td>SP</td>
<td>Status Point</td>
</tr>
<tr>
<td>SPOC</td>
<td>Single Point of Contact</td>
</tr>
<tr>
<td>SPM</td>
<td>Strategic Product Manager</td>
</tr>
<tr>
<td>TR</td>
<td>Trouble Reports</td>
</tr>
<tr>
<td>TPM</td>
<td>Total Project Manager</td>
</tr>
</tbody>
</table>