

Survey and Comparative Study on Agile Methods in Software Engineering

¹Achrak El Mehdi, ²Chkouri Mohamed Yassin, ¹El Kadiri Kamal Eddine

¹LIROSA Lab, National School of Applied Sciences, TETOUAN, Morocco

²SIGL Research Team, National School of Applied Sciences, TETOUAN, Morocco

mehdi.achrak@gmail.com, yassin.chkouri@gmail.com, elkadiri@uae.ma

ABSTRACT

Today's business environment is very much dynamic, and organizations are constantly changing their software requirements to adjust with new environment. They also demand for fast delivery of software products as well as for accepting changing requirements. In this aspect, traditional plan-driven developments fail to meet up these requirements. Though traditional software development methodologies, such as life cycle-based structured and object oriented approaches, continue to dominate the systems development few decades and much research has done in traditional methodologies, Agile software development brings its own set of novel challenges that must be addressed to satisfy the customer through early and continuous delivery of the valuable software.

This article is located in the field of engineering Information Systems (IS) and in particular in the world of engineering process. In order, to produce information systems that the customer needs, many methods engineering were introduced. The first methods introduced are called sequential methods like Cascade method, V cycle or the Y cycle then there are iterative methods like RUP or Symphony and finally agile methods like Scrum and XP have emerged.

Keywords—Agile Methods, Development Process, Information Systems, Software Engineering.

1 Introduction

Software development is an organized process that thrives to deliver products in faster, better and cheaper ways. There have been many studies and suggestion in improving the development process. Recently, this interest has paved way to a new software development method called Agile Software Development. Agile methods strive to deliver small sets of software features to customers as quickly as possible in short iterations. As part of this paper scope most commonly used methods will be examined from the angle of their applicability, strengths and weaknesses and their adoption in industry. This will lead us to find benefits, limitations and difficulties in agile software development.

The term agile stands for 'moving quickly'. Agile methodology is a lightweight methodology for software development. [1] Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In agile development, rather than a single large process model that implemented in conventional SDLC, the development life cycle is divided into smaller parts, called "increments" or "iterations", in which each of these increments touches on each

of the conventional phases of development. Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer. [2]

The success of a project requires regular and frequent feedback from the client. A contract that specifies requirements, planning and project costs prior reports a vision of the IT project. The best way forward for the customer is to work closely with the development team to provide continuous feedback that ensures better control of the project. So, changes in specifications may occur very late in the project development cycle. It is a solution that truly meets customer expectations is performed, not a solution that meets the requirements of a contract established a priori. This of course requires great maturity of the customer and the service provider to establish a real relationship of trust and a good understanding of the operational reality of the project by the lawyers handling the case.

When planning, it is necessary to ensure that the schedule is flexible and adaptable to changes that may occur in the context, technologies and specifications. Indeed, it is difficult to think from the beginning that all features you would like to have and it is very probable that the customer modifies its requirements once he saw an early version of operating system.

This paper is organized as follows. Section 2 gives an overview of agile methods. In Section 3, we analyze and compare the agile methods. In Section 4, we give some recommendation to use agile methods in IS. Conclusions close the article in Section 5.

2 Agile Methodology

2.1 Agile Model

IT project management methods know as well as the technologies used a permanent challenge. The high proportion of failed projects is often fed more or less constructive feedback methods implemented. Developments in technology architectures and development tools also contribute for an important part.

The term "Agile" defines a project management approach that against the foot predictive and traditional approaches sequential cycle type V or waterfall. The very notion of "project management" is called into question in favor of "product management". In order to think more "product" that "project". After all the purpose, the project gives birth to a product.

Agile methods are mainly dedicated to IT project management methodologies. They are based on iterative and adaptive development cycles according to the customer's changing needs. They allow in particular involving all employees and customers in the development of the project.

For over a decade now, there has been an ever increasing variety of agile methods available includes a number of specific techniques and practices of software development. Agile methods are a subset of "iterative and evolutionary methods" [14,15] and are "based on iterative enhancement" [85] and "opportunistic development processes". Most of agile development methods promote development, teamwork, collaboration and process adaptability throughout the life-cycle of the project. The major methods include eXtreme Programming [14], Scrum, Dynamic Systems Development Method, Adaptive Software Development, Crystal, and Feature-Driven Development [17].

These methods generally allow the best responding to customer expectations in a limited time (thanks in part to the involvement of the latter) while driving up employee skills. These methods are therefore a productivity gain competitive advantage as well as a client side and the provider side.

2.2 The Process of Agile Software Development

The process contains the six basic phases

- Pre project preparation
- Start
- Construction
- Production
- Selection
- Departure

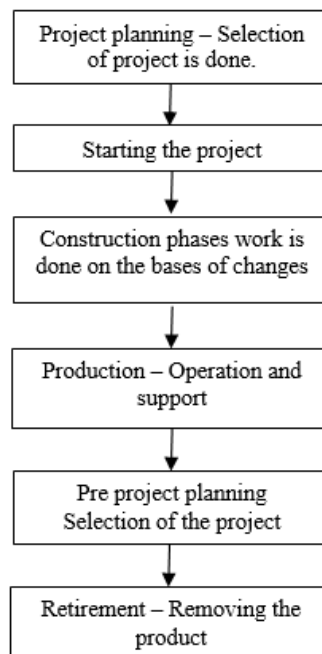


Figure. 1. Process of Agile Software Development

- (1) Pre project scheduling: The basic of the project and the market features are well-defined in this phase.
- (2) Start-up-phase: Requirement displaying done through active participation of stakeholders in order to identify the initial requirement basic or high-level requirements for the system. The main objective is to understand the problem.
- (3) Construction: In this phase excellent working software is delivered incrementally, which meets the variable needs of the customer. Adjustment in the requirements is allowed to meet the exact needs of the customers. At the end of each progress iteration there is a fractional, working system which is shown to the customer and testing can be done on that like system integration testing.
- (4) Production: It handles all operations and provides supports.
- (5) Selection: The goal of the construction phase is to keep systems useful and active even after the product has been deployed to the user.

(6) Departure: It is also known as system deactivating phase.

3 Study of Agile Methods

3.1 eXtrem Programming

The main purpose is to reduce the costs of change. In traditional methods, the needs are defined and often set at the beginning of the computer project which increases costs subsequent for changes. XP works to make the project more flexible and open to change by introducing basic values, principles and practices. [4]

The client (project owner) pilot project, the team around the project delivers early a first version of the software, and deliveries of new versions then linked together at a steady pace for maximum feedback on the progress of development. The team organizes itself to achieve its objectives by encouraging maximum cooperation between its members. The team sets up automatic tests for all the features it develops, its guarantee a very high level of robustness. Developers are constantly improving the internal structure of software that changes are still easy and quick. [6]

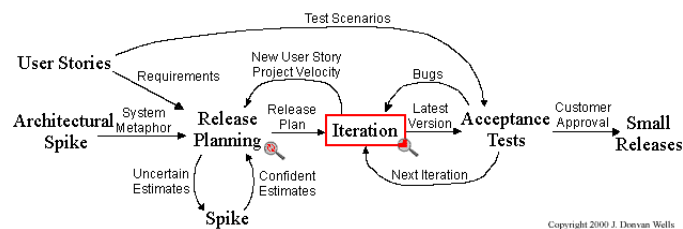


Figure 2 Overview of XP

The XP values are: communication, simplicity, feedback, respect

3.2 Scrum

Scrum is a framework for development of multifunctional teams perform iterative and incrementally products. Scrum structure development in work cycles called sprints. These iterations never last more than four weeks (two weeks most of time) and are linked one after the other without interruption. Sprints are time-limited or “Time box” they end on a specific date that the job is done or not, and are never extended. Scrum teams generally choose duration of sprint and maintained during the project, until they could further increase their productivity and then use a shorter cycle. At the beginning of each sprint, a cross-functional team (about seven people) selects items (customer requirements) in a prioritized list. The team collectively agrees on a target consisting of what expects to deliver at the end of sprint, tangibly and effectively “finished”. No new elements are added during the sprint, scrum accept change for next sprint, but the fixed term of a current sprint is made to focus on a relatively stable objective, clear and limited. [3]

Every day, the team met briefly to monitor his progress and adjust the next steps required to complete the remaining work. At the end of each sprint, a review is organized with parts during which the team shows what it has achieved. The feedback received may be taken into account on the following sprint. Scrum insists to deliver a functioning product at the end of each sprint, and really finished. In the case of software, it means an integrated, fully tested, documented for the users and potentially deployable.

The roles, artifacts and key events are show in the overview below:

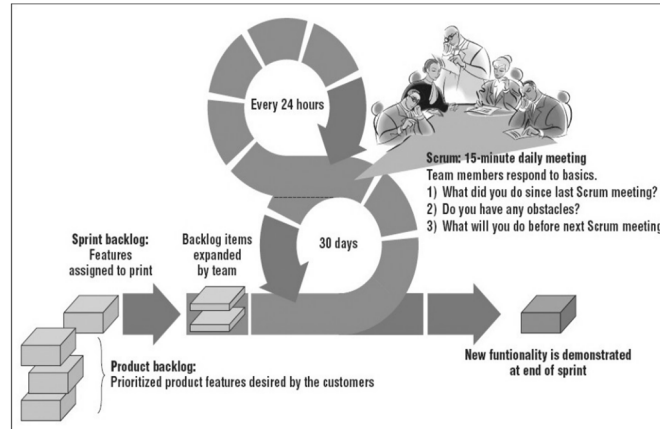


Figure 3 Scrum overview for Agile Software Development

In scrum, there are three roles: [5]

1) Product Owner: is responsible for maximizing the return on investment by identifying features of the product, transferring them in a prioritized list, selecting those to be top of the list for the next sprint, and re-prioritizing and refining continually this list.

2) Team: built product that is defined by the product owner: an application or website for example. In scrum, the team is “multi-function:” it includes all the expertise necessary to provide a version of product deliverable in each sprint.

3) Scrum Master: helps the groups learns and supply Scrum so that business value materializes. The Scrum Master is doing everything in its power to assist the team, the product owner and organization to succeed the project. The Scrum Master is not the manager of the team, or even project manager, a team leader or a representative of the team. Its role is to serve the team, its helps to remove barriers, protect the team from external interferences and facilitates the adoption by the team of modern development practices.

The values for Scrum are transparency, inspection, adaptation

3.3 Rational Unified Process (RUP)

This method can be considered the less agile methods presented here, is a mixing of practices from traditional and agile methods. The principle is browse a life cycle (inspection, elaboration, construction, transition) during iteration. Each stage in the life cycle is very precisely detailed.

It's heavy approach and the investment cost of this method the reserve for projects large and medium size detailed. [7]

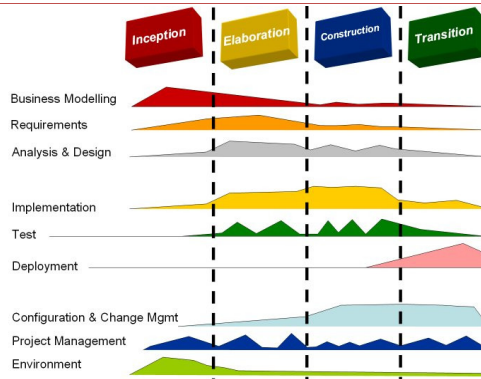


Figure 4 RUP phases

Rational Unified Process is composed of 4 phases:

Inception allows:

- Delineate the system (range), and identifying its interfaces with the external world,
- Offer or sketch a plausible system architecture,
- Identify risks regarding the feasibility of the system.

Elaboration allows:

- Develop an architecture / infrastructure to implement functionality to architectural impact,
- Identify the risks that could result in additional costs, delays, etc.
- Specify the quality requirements,
- Identify use cases,
- Make a project plan with estimated costs and resources.

Construction can:

- Complete the requirements capture,
- Complete the analysis, design, implementation and test use cases,
- Maintain architectural integrity,
- Monitor and manage risks.

Transition can:

- Prepare the deployment environment,
- Produce documentation for the 'Release' (user manuals, maintenance manuals)
- Correct errors discovered in beta.

3.4 Feature Driven Development (FDD)

Feature Driven Development (FDD), was developed by Peter Coad, Jeff Luca and Stephen Palmer. This method prioritizes risk management, based on short iterations to deliver tangible functionality to the user. This one is quickly informed of the progress, reducing the risks. According to its creators, the latter can be used in the development of critical systems.

Less known than the two previous methods, FDD focuses mainly on design and development. For this it relies on a formalization of object model using UML diagrams, cutting functions that will be developed by small teams responsible for one or two functions. It gives a very important aspect to the quality of the finished product, and support tools to monitor the progress of the project.

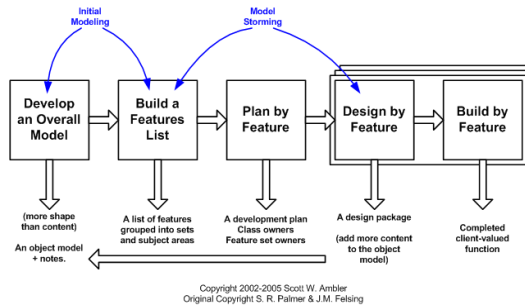


Figure 5 The FDD project lifecycle [5]

3.5 Rapid Application Development (RAD)

It is the oldest agile method and that was the first to be breaking with traditional methods. It introduced the concepts of iteration and increment. It aims to adopt the most strategic solution (in terms of time), the least risky, most reliable and least expensive. Its development cycle is simple: framing, design, construction and completion in full respect of a period of between 90 and 120 days.

The RAD cycle consists of five phases:

- Initialization determines whether the project can be followed with the RAD method,
- Expression of needs, determines the outline of the project: objectives, participants, main functions
- Design, precisely defines the processes, data,
- Construction, is the development and validation phase of the system,
- The implementation is the delivery of the newly created system.

The phases of design and construction are included in a cycle, and each is the subject of several iterations to reduce the risk of failure. [8]

This method includes 5 main steps; the last two are in the iterative cycle.

- Develop a general model.
- Build a list of features.
- Plan by features.
- Design by features.
- Build in features.

The RAD method's main objective is to produce a system in which the charge and the project period be reduced without sacrificing quality and meet the needs of users. Thus the objectives of RAD are to produce software: faster, better, cheaper.

The RAD objective is to produce a useful software when the business needs it. For this reason, the life cycle RAD is set at a maximum of 120 days. The project is contained in a short time to avoid the classic

escape ahead of the project delivery and user expectations that increase increasingly delaying project completion

3.6 Dynamic Software Development Method (DSDM)

DSDM is agile method developed in England in the mid-90s. It uses the principles already seen in other methods (user involvement, autonomy of the team, visibility and relevance of the result, iterative and incremental development, reversibility of changes, tests continuous, cooperation of actors).

The DSDM method was born of the same recognition that all agile methods, namely that development times were far too long, namely that development times were far too long, the delivered applications corresponded not always exactly the needs, the developers were very involved in the design and that no one had full view of the systems developed.

Fundamental Principles of DSDM:

- Active user involvement is imperative:
- The DSDM teams must be allowed to make decisions
- The product is made tangible as often as possible:
- An iterative and incremental development:
- Any changes during the realization is reversible
- The needs are defined at a summary level
- Testing is integrated throughout the lifecycle
- A spirit of cooperation between actors is essential.

The phases of DSDM:

- The Feasibility Study.
- The Research Business.
- The Functional Model iterative.
- Design and iterative development.
- The Implementation in the work environment.

4 Analysis of Agile methods

Agile is now widely used in web services or agencies companies. I have often heard that one of the advantages of this method was that we could have what we wanted in, but contrary to popular belief, this method will bear fruit only if it is followed to the letter. Indeed, each step is important. We will see the advantages and disadvantages and potential risks in using this method. [9]

4.1 The strenghts of Agile Methods

In this part we will present you the highlights of agile methods:

- They allow the minimization of the differences between the developed product and business needs,
- They cause a reduction in costs of study that is conducted jointly with the developments,
- They negate the possibility of tunneling to users and management, who are involved in the project,
- They are born great flexibility and responsiveness for short projects with high added value,
- The communication is better, quality is assessed continuously,

- The risks are detected early, costs are controlled.

4.2 The limits of Agile Methods

This part will allow us to know the limits which cannot exceed using agile methods among these limits:

- One of the first problems is availability. It is more difficult to obtain in small teams, including the drafting of project documentation, as everything is done progressively;
- A lack of forward planning as the project progresses custody;
- A need for consultation with other projects because the fact to all employees does not always create a team receptive to change. (Barry, January 2002)

As part of the evolution infrastructure, there may be cycles and iterations, but the user is involved or is pressed for the realization. Agility may lie in the choice of architecture (clouding, clustering, and virtualization (Boehm)) and financing options.

4.3 Gain and Downside

4.3.1 For the Customer:

Putting the customer at the center of the steps and people involved in it many advantages:

- In terms of satisfaction, because the customer can adapt, even in development, the software to its business needs by giving her feedback through regular access to different functional versions made. The goal of agile methods is indeed the real satisfaction of need rather
- In terms of competitiveness: direct adaptation needs of the software definitely provide great value to it. Management priorities in cost estimates also provide greater efficiency in economic terms for the client.

Agile methods fall short cycle as a project but resulting length that is important, so customers who have little time or interest in the proposed development of their software will provide a final product much lower in terms of quality. [10]

4.3.2 For the developer:

One of the principles of agile methods is self-organization of team development; developers enjoy greater creative freedom and greater responsibility in the project. In addition, the prominence of trade and the collective spirit promises better quality of working life. For their managers, the advantage is greater visibility on project progress.

Finally, management will appreciate the focus on return on investment and the sustainability of the program throughout its lifetime.

However, by focusing on direct dialogue, agile methods leave little room to documentation, which can become a big problem if development team when changes or software maintenance is handled by another team than which has been responsible for the development. [11]

4.4 Comparaison between Agile Methods

After listing the agile methods now we go to make compare between these methods. We will start by seeing the percentage of use, development phases, advantages, disadvantages, size of the project, cost of development, rapid of use and quality for each method.

Percentage of use:

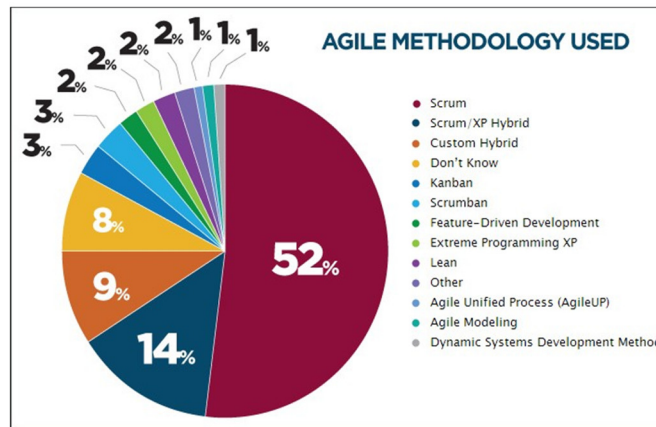


Figure 6 Li State of Agile Survey Results 2011 by Version One In [13]

Development phases

- Scrum : Vision / Product Backlog / Sprint Backlog / Product Increment
- XP : Planning / Writing tests / Development / Test / Integration
- RUP : Inception / Development / Construction / Transition
- FDD : Development / Build / Plan / Designate
- RAD / DSDM : Initialization / Expression of needs / Design / Construction / Implementation

Advantages :

- Scrum : fully developed and tested for short iterations, simplicity of processes, clearly defined rules, increased productivity, personal organization, each team has its share of responsibility and improved communication.
- XP : reduce costs of change, permanent code reviews, development driven by testing, design during the project and systematically privilege the simplest method.
- RUP : risks are mitigated from the start of the project, change is better managed, improved portability through the use of UML and verify the constant quality.
- FDD : risk management, frequent deliveries and code inspection.
- RAD : rapid development, low development cost and a high-quality application.
- DSDM : quick implementation of priority solutions, adequacy of the system to the need, respect of deadlines and costs and integrated tests

Disadvantages:

- Scrum : little written documentation and violation of liability.
- XP : small and medium-sized projects only, requires strong customer involvement and not enough documentation.
- RUP : heavy management to set up and not suitable for a small project.
- FDD : number of senior developers available and code inspection.
- RAD : applications are very long to develop.
- DSDM : difficult maintenance and bad understanding with users.

Size of the project:

- Scrum : many processes implicitly define.
- XP : 12 people in the team.
- RUP : importance of well-defined processes.
- FDD : 20 people in the team.
- RAD : presence of a coordination and reporting group.
- DSDM : extensible due to roles and well-defined responsibilities.

Cost of development :

- Scrum : \$633 043
- XP : \$630 860
- RUP : \$756 157

Rapid :

- The deadlines, the costs.

Quality :

- The potential for anomalies
- Efficiency in repairing anomalies (DE)
- Anomalies present during delivery
- Major anomalies

After the comparison between the agile methods, we can say that the two ideal methods that can be used : XP and Scrum. So we will do a detailed comparison between these two methods. [12]

XP Method	Scrum Method
Duration of the iteration (1 to 2 weeks)	Duration of the iteration (2 to 4 weeks)
The changes during the iteration features are not accepted.	Possibilities to changes opportunities scenarios or features to implement in the iteration.
Different roles are assigned to team members 'XP' (programmer, tester, coach, manager, ...)	The team members are multidisciplinary and only three roles are defined (scrum master, product owner and the team)
Management practices, engineering practices and tools for collaboration and the support of management	Management practices and tools for collaboration and the support of management

Scrum following an iterative process for obtaining a product of close customer needs by taking into account the evolution of the latter and thus maximize the value of the delivered product.

If during a Sprint, a problem occurs, the responsibility lies not with one person but is shared between the Product Owner, the Scrum Master and Scrum team.

4.5 Recommendation for use agile methods in SI

To have a well-structured information system and especially without any problem agile methods are used according to our needs and the project.

For this we made a vision and we follow three steps:

- First step: formalize the product vision: objectives, target users
- Step two: establish a list of functional and non-functional requirements
- Third step: estimating each requirement by the development team

Requirements will be converted into usable features as scheduling. [9]

So you have to back up the functional requirements of the highest value at the top of the list.

One can also use a planning poker: it is a technique to rapidly reach an optimal and objective estimation. Before or during the estimates, the product owner may be solicited to answer the questions of the development team. To conclude, we can say that Scrum offers several advantages with the possibility to apply on a large project panel. It is for everyone to have their own idea and decide if the Scrum methodology is adapted or not to this or that project because it's important to keep in mind that each team, each project and context are different.

5 Conclusion

Agile approaches are increases the flexibility, agility and to be more adjusted to the environment where software development projects are present and working today. The idea behind Agile methodology would likely to break large projects into smaller projects which would become more flexible. Also it's proven that agile provides means for software development with minimum risk where project requirements are well defined, development team is self-dependent. It also ensures that every team member understands, evaluates and actively participate in development which ensures no single point of dependency and failure. It also facilitates more flexibility to adopt changes in requirement and control development life cycle through micro breakdown of requirements and planning of same.

REFERENCES

- [1] K. Beck, Extreme Programming explained: Embrace change. Reading, Mass., Addison-Wesley, Nov16, 2004
- [2] Jim Highsmith "Agile Project Management:creating Innovative product" (2nd Edition)
- [3] Charlotte, H. . Méthodes, modèles et outils pour la méta-modélisation des processus d'Ingénierie de Systèmes d'information. France, 2009
- [4] Chausson, C. (2008). Méthodes agiles: le renouveau des relations client.
- [5] Deemer, P. (2013). The Scrum Primer: Guide Léger de la Théorie et de la Pratique de Scrum - Version 2.0.
- [6] Jean-Louis, B. (2011). Extreme Programming - Agile Methods . Tour d'horison / Business Interactif.

- [7] Koch, A. S. (2004). Agile Software Development Evaluating the Methods for Your Organization. ARTECH House.
- [8] Messenger, V., & Tabaka, J. (2007). Gestion de projet vers les méthodes agiles . Groupe Eyrolles,.
- [9] Paasivaara, M., & Lassenius, C. (s.d.). Could Global Software Development Benefit from Agile Methods? IEEE International Conference on Global Software Engineering , 5.
- [10] Palmer, S. R. (s.d.). A Practical Guide to Feature-Driven Development 1st Edition.
- [11] Rahmania, a. e. (2012). Project Lifecycle, Project Management, tagged agile methodology, extreme programming, rational unified process, RUP, SDLC.
- [12] STEPHANE, L. Transition vers l'agilité à l'échelle d'une organisation. Valtech, 2012
- [13] <http://www.versionone.com/Agile101/Agile-Software-Development-Benefits/>
- [14] C. Larman, Agile and Iterative Development: A Manager's Guide. Boston: Addison Wesley, 2004.
- [15] C. Larman and V. Basili, "A History of Iterative and Incremental Development," IEEE Computer, vol. 36, no. 6, pp. 47-56, June 2003.
- [16] V. R. Basili and A. J. Turner, "Iterative Enhancement: A Practical Technique for Software Development," IEEE Transactions on Software Engineering, vol. 1, no. 4, pp. 266
- [17] A Practical Guide to Feature Driven Development. 2002 S Palmer, M Felsing - Prentice Hall