

শ্রীমতী সুনন্দ প্রান্ত বিশ্ববিদ্যালয়



# CITY UNIVERSITY

## LAB REPORT

Assignment Name: Agile Methodology

Course Code: CSE-325

Course Title: System Analysis & Design

*Submitted To: Richard Philip*

*Senior Lecturer, Department of CSE*

*City University, Bangladesh*

*Submitted By: ...creating a culture of excellence*

ID : 171442624

Name : Mustafizur Rahman Khan

Program : CSE (Eve)

Batch : 44<sup>th</sup>

**DATE OF ASSIGNED : 20-04-2019**

# **AGILE DEVELOPMENT**

## **Introduction**

**Agile methodology** is a type of project management process, mainly used for software development, where demands and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customers.

## **Agile Methodology**

### **Where did Agile come from?**

In 2001, a small group of people, tired of the traditional approach to managing software development projects, designed the agile manifesto. It is a more improved method for managing the progress of software projects.

**The agile [manifesto](#) has four important values:**

1. Focus should be more on individuals and interactions instead of processes and tools
2. Working software is more important than comprehensive documentation
3. Customer collaboration is more vital than contract negotiation
4. The process should respond to change rather than follow a plan

**There are 12 principles of agile software development:**

- ❖ Deliver customer satisfaction by delivering valuable software continuously
- ❖ Always accept change of requirements matter how early or late in the project
- ❖ Deliver software that works within a shorter timescale
- ❖ Both developers and business professionals must work closely together daily throughout the duration of the project
- ❖ Information is best transferred between parties in face-to-face conversations
- ❖ Motivate people to build a project by creating an environment of appreciation, trust, and empowerment

- ❖ Working software is the key measure of progress
- ❖ The agile process promotes sustainable development
- ❖ Continuous attention to excellence and quality in technical development and design boosts the agility
- ❖ 10 Simplicity is a vital part of effective agile management
- ❖ Self-organized teams produce the best architecture, requirements, and design
- ❖ Teams should reflect through inspection and adaption to be more effective
- ❖ There are different methods of agile that promote the values and principles of the manifesto. Scrum and XP are two well-known examples.

## **The Benefits of Agile Software Development**

Top software developers developed agile meetings. After repeatedly experiencing challenges and limitations from traditional waterfall development in real life projects, they wanted to create a more efficient process for analyzing project development. The approach they used addresses the issues regarding the philosophies and processes of traditional methods directly.

**There are many benefits to the agile software development. They include:**

### **Stakeholder Engagement and Satisfaction**

The agile process creates many opportunities throughout each sprint meeting for genuine engagement between the team and the stakeholders. Because the client is actively involved in the entire project, there is a continuous level of collaboration between all parties. This gives the team a chance to fully understand the client's vision. By delivering high quality, working software frequently, the stakeholders quickly develop a trusting and authentic relationship with the team. This also further promotes engagement between the client and the team.

### **Transparency**

The agile approach actively involves the client throughout the entire project including the iteration planning, review sessions, and new feature builds in the software. Clients, however, must understand that during the transparency of the project, they are seeing a work in progress and not the final product.

### **Early and Predictable Delivery**

Sprints are held on a fixed schedule of 1 to 4 weeks duration. By using this time-boxed method, predictability is high as new features can be delivered to the stakeholders quickly and frequently. It also allows the team to beta test or release the software sooner if it has sufficient business value.

### **Predictable Costs and Schedule**

Because the Sprints are on a fixed schedule, the costs are limited and predictable, and based on the amount of work done. By combining the estimated costs before each Sprint, the client will better understand the approximate costs of each feature. This offers more improved decision-making opportunities when prioritizing the features or adding iterations.

### **Flexible Prioritization**

Scrum methodologies allow more flexibility by prioritizing the customer-driven features. The team has more control in managing the shippable units of work with each sprint boundary; making continued progress towards the final product milestone. To get a prompt RIO from the engineering, the work needs to be shipped early to the customers so they will realize the value from the features.

### **Allows for Change**

While the focus should be to deliver the agreed subset of the products features, Agile processes create an opportunity to continually reprioritize and refine the product backlog. These changes can be added to the next iteration so the new changes can be introduced within a few weeks.

### **Focuses on Business Value**

The team has a better understanding of what is most important for the client's business and can deliver features that give the most value to the business.

### **Focuses on Users**

The user's stories are commonly used to define the product features as they relate to business-focused acceptance criteria. By focusing on the user's needs, each feature delivers real value and not just an IT component. It provides a better opportunity to gain valuable feedback through beta testing the software after each Sprint. This provides vital feedback earlier in the project so that changes can be made as needed.

### **Improves Quality**

The projects are broken down into manageable units, making it easier for the team or focus on high-quality development, testing, and collaboration. By creating builds and conducts tests or

reviews throughout the iteration, defects and mismatches can be found and fixed early, improving over-all quality.

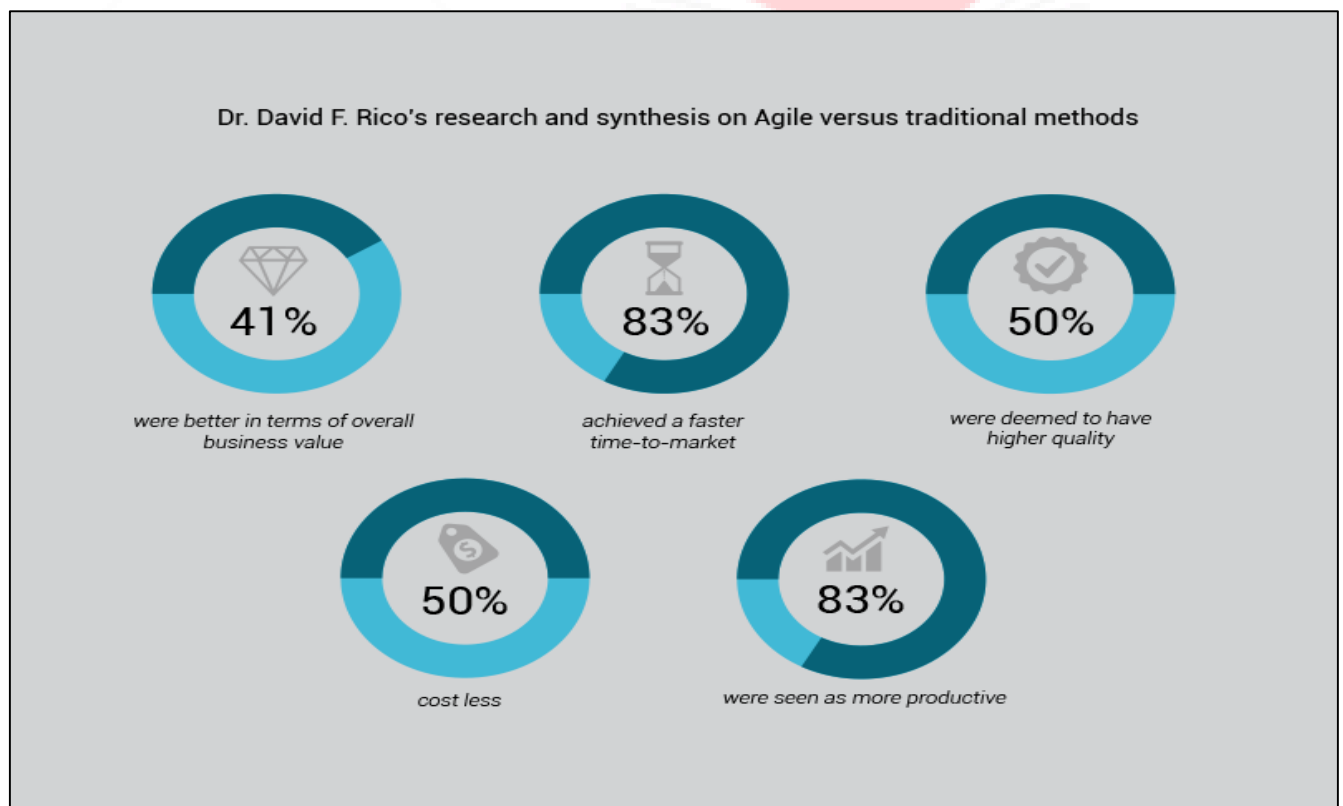
### It gives your team Purpose

Most agile processes focus on creating a shared sense of ownership and goals for all team members. This to gives your team purpose rather than creating a false sense of urgency. Purposeful teams are more productive and challenge themselves to be faster and more efficient.

## The Business Benefits of Agile

Agile management reduces the common risks that are associated with the delivery, scope, and budget of the project.

It encourages collaboration between the customer and the team; offering mutual benefits in the mitigation of the high risks during the development of the software.



In 2009, Dr. David F Rico compared Agile with traditional methods of software project management. During his research and synthesis, he analyzed 23 Agile processes, comparing them with 7,500 traditional projects.

**He found 20 benefits to Agile projects:**

- 41% were better in terms of overall business value
- 83% showed quicker time-to-market speeds
- 50% were higher in quality
- 50% were less costly
- 83% were more productive

## Agile Methodologies

There are several agile methodologies; all share similar philosophies, characteristics, and practices. However, from the point of implementation each agile has its own practices, terminology, and tactics. Some of the main agile software development methodology components include:

### Scrum

[Scrum](#) is a management framework with far reaching abilities to control and manage the iterations and increments in all project types. They are lightweight and can be used with other agile methodologies for various engineering practices. Scrums have grown in popularity within the agile software development community because they are simple and have a proven productivity rate.

### Lean and Kanban

#### 1. Lean Software Development

Lean software development is an iteration methodology originally developed by Mary and Tom Poppendieck. Many of the principles and practices in Lean Software Development came from the lean enterprise movement and were first used by big companies like Toyota. This value based method focuses on giving the customer an efficient “Value Stream” mechanism that delivers the value to the project.

The main principles of this methodology are:

- Eliminate waste
- Amplify learning
- Make decisions as late as possible
- Deliver results as quickly as possible
- Empower the team
- Build integrity

- Envision the whole project

By choosing only the features that have real value for the system, prioritizing and delivering them in small batches eliminates waste. Instead, the lean methodology emphasizes on speed and efficiency; relying on rapid, reliable feedback between the customers and programmers. It focuses on the idea that customer requests “pull” the product. The focus is more about the faster and more efficient decision-making abilities of individuals or small teams instead of a hierarchy controlled method. This methodology concentrates on the efficiencies of its team’s resources, ensuring everyone is as productive as possible always.

## **2.The Kanban Method**

Organizations use the Kanban method to manage the project’s creation while emphasizing on the continued delivery and not overburdening the development team. Like Scrum, Kanban processes are designed to help teams work more efficiently together.

### **There are three principles:**

Visualize what you do: see all the items within context of each other – more informative

Limit the amount of work in progress (WIP): balance the flow-based approach so teams are not committed to doing too much work at once

Enhance the flow: as soon as one task is finished, start on the next highest job from the backlog

The Kanban method promotes continued collaboration by the client and team. It encourages ongoing learning and improvements to provide the best possible workflow for the team.

### **Extreme Programming (XP)**

Extreme Programming (XP) was originally described by Kent Beck. It is one of the most popular and controversial agile methodologies. XP is a highly disciplined method of continually delivering high-quality software faster. The customer is actively involved with the close-knit team to perform continued planning, testing and rapid feedback to deliver working software frequently. The software should be delivered in intervals everyone to three weeks.

### **The original XP method is based on four simple values:**

- Simplicity
- Communication
- Feedback
- Courage

### **It has 12 supporting practices:**

1. Planning game
2. Small releases

3. Customer acceptance tests
4. Simple design
5. Pair programming
6. Test-driven development
7. Refactoring
8. Continuous integration
9. Collective code ownership
10. Coding standards
11. Metaphor
12. Sustainable pace

## Crystal

Crystal methodology is one of the most lightweight and adaptable approaches in developing software. It is made up of several agile processes including Clear, Crystal Yellow, Crystal Orange, and other uniquely characterized methods. There are several factors that drive these processes including: size of the team, criticality of the system, and the priorities of the project.

The Crystal family focuses on the realization that each project has unique characteristics, therefore, the policies and practices must be custom tailored to accommodate these features.

**The Crystal method has several essential tenets including:**

- Teamwork
- Communication
- Simplicity
- Reflection
- Frequent adjustments
- Improve processes

This agile process, like other methodologies, promotes early and frequent working software delivery. It encourages high user involvement, adaptability, and eliminations of distractions and bureaucracy.

## Dynamic Systems Development Method (DSDM)

The Dynamic Systems Development Method (DSDM) originated in 1994 to provide an industry standard framework for project delivery for what was then known as Rapid Application Development (RAD). Although it was very popular in the 1990's, the RAD approach developed in an unstructured fashion.



Since its beginning, the DSDM has evolved and matured; it provides a comprehensive foundation in planning, management, execution, and scaling of the agile process and iteration projects.

**DSDM has six key principles revolving around the business needs:**

1. Value
2. Active user involvement
3. Empowered teams
4. Frequent delivery
5. Integrated testing
6. Stakeholder collaboration

DSDM uses a “fitness for business purpose” approach for delivery and acceptance criteria. It focuses on the formula: 80% system deployment in 20% time.

## **Feature-Driven Development (FDD)**

Jeff De Luca, along with contributors A.m. Rajashima, Lim Bak Wee, Paul Szego, Jon Kern, and Stehen Palmer developed Feature-Driven Development (FDD). It is a model-driven, short iteration process that begins by first establishing the shape of the agile model. Iterations on “design by feature, build by feature” are held biweekly. The features appeal to clients because they are small and useful.

**The FDD design and development is delivered using these eight practices:**

1. Domain object modeling
2. Development of features
3. Component and class ownership
4. Feature teams
5. Inspections
6. Configuration management
7. Regular builds
8. Visibility of progress and results

## **Roles in Agile**

In the agile development process, the [Scrum](#) most clearly defines what agile is in project management.

There are three roles in the Scrum project: Product Owner, Scrum Master, and team members.

[The Product Owner](#) oversees all the business conditions of the project to ensure the right product is built and in the right order. A good product owner balances the competing priorities, is available for the team, and makes decisions about the project.

[The Scrum Master](#) is the team's coach; they help the team work together effectively. [Scrum Masters](#) service the team by removing barriers that impair the progress, facilitating meetings and discussion groups, tracking progress, problem solving, and performing other project management duties.

The team works together to determine the best approach to achieve the product goals that are outlined by the product owner. The team decides which members will manage specific tasks, and outline the technical practices required to achieve the desired goals.

### **Advantages of Agile Methodology**

- In Agile methodology the delivery of software is unrelenting
- The customers are satisfied because after every Sprint working feature of the software is delivered to them.
- Customers can have a look of the working feature which fulfilled their expectations.
- If the customers has any feedback or any change in the feature then it can be accommodated in the current release of the product.
- In Agile methodology the daily interactions are required between the business people and the developers.
- In this methodology attention is paid to the good design of the product.
- Changes in the requirements are accepted even in the later stages of the development.

### **Disadvantages of the Agile Methodology**

- In Agile methodology the documentation is less.
- Sometimes in Agile methodology the requirement is not very clear hence it's difficult to predict the expected result.
- In few of the projects at the starting of the software development life cycle it's difficult to estimate the actual effort required.

- ## Conclusion

When Agile programming is done properly, organizations can continually find ways to increase the value to their customers. It gives more meaning to those who are actively working on the project and creates a more positive experience for the customer, producing more generous end results for the company.

## Reference

**Agile Development Methods: Philosophy and Practice. (2010). CPSC 315 – Programming Studio. Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile Software Development Methods Review and Analysis. University of Oulu: VTT Publications 478.**

**Ambler, S. W. (2012). The Agile System Development Life Cycle (SDLC). Retrieved March 26, 2013, from ambysoft: <http://www.ambysoft.com/essays/agileLifecycle.html>**

[illegible]