Understanding Agile and lean

**Name Mekala Lokesh Kumar**

Table of contents

[Introductory part 3](#_Toc67900427)

Introduction …………………….…………………………………………………………………………………………………..3

Agile software development……………….…………………………………………………………………………….….3

Lean software development………………….………………………………………………………………………….....7

Conclusion………………………………………….………………………………………………………………………………..9

References………………………………………...……………………………………………….10

# Introductory part

## We understand the concept of agile, values in agile, and the principles present in agile.

## We discuss the relation between the values of agile and its principles.

## We explore the concept of lean software development, the values present in lean development, and the principles in lean.

## We discuss the differences and similarities between agile and lean.

## Introduction

Traditional software development started with simple methods like the waterfall model. These traditional methods seemed to be inadequate as the software started to fail. Despite many changes made in the software development phase, the result did not change. There were some problems such as people taking leave due to health issues, the time needed to get the required technology, etc which were unavoidable. Taking more time than promised, Exceeding the budget planned, overestimating the effort are some of the reasons for the failure of the software. Only if the software’s were developed within the promised deadlines, the software would have been a huge success [3]. Due to these continuous failures in the software industry, experts started to look out for new development methodologies. Eventually, the experts were able to come up with a new development method called Agile software development. Unlike traditional software development, with the implementation of Agile software methods, the software turned out to be successful. Soon everyone in the software industry started to use the Agile methodology.

# Agile Software development

All the drawbacks in the traditional software development methods were solved with the help of the Agile method. One of the major problems in the traditional methods is that requirements, resources, technologies kept on changing. Agile development accepts changes in requirements at any stage in the development cycle. The agile method helped to develop software’s within the agreed time and also without getting over budget [4]. A group of 17 individual industry experts sat down and developed an "Agile manifesto" by sharing all their experiences. From all these experiences they took out the common points and made them as agile manifesto{empirical studies of agile software development}. The agile manifesto's main motto is to define the importance of agile software development and find a group of people who have a similar mindset.

Agile values:

1. **Individuals and interactions** over processes and tools.
2. **Working software** over comprehensive documentation.
3. **Customer collaboration** over contract negotiation.
4. **Responding to change** over following a plan.

The first value suggests that we need to give more importance to selecting the people rather than tools and processes. Software development is a human-dependant process, so the group of people who work in a team in agile is to be selected carefully. Every day small meetings are conducted to keep track of progress and to discuss issues that are present. Every individual should join the meeting and share their thoughts and interact with the team. Individuals give respect to each other as fellow agile team members. In addition to receiving respect, individuals acquire responsibilities to complete given work in time.

A customer satisfies when working software is delivered. Coding starts as soon as possible in agile, the main objective of agile is to develop the working software as soon as possible. In the early stages of development, only the most important requirements are developed. Later in the development cycle, new functionalities are added to the software.

Due to miscommunications between the client requirements are wrongly interpreted. In agile after completion of development at every sprint, customers are subjected to use the software. Customer interaction helps the agile team to develop the software that makes the clients happy. Customer satisfaction is the most important factor in software development.

Requirements in real-time projects are never stable, they keep on changing even at the end of development stages. Agile accepts changes at every stage of development, agile is very much flexible to changes in requirements. When requirements are volatile there is no point in drafting a perfect plan. Even though the project plan was created perfectly without any flaws, due to changes in requirements plan needs to be recreated.

Agile consists of 12 principles they are as follows[1]:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
5. Businesspeople and developers must work​ together daily throughout the project.
6. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity—the art of maximizing the amount of work not done—is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

Explanation of principles:

1. 1. Customer is treated as a god in the software industry. Agiles' goal is to develop a working software as soon as possible with minimum or base requirements. The customer is satisfied when a working model is presented to him. Software is said to be successful if the customer is satisfied with the developed software.
2. Changes are very common in software development. Unlike other traditional methods agile always allows changes even late in the development phases. After using the software it is high that the customer suggests changes and agile always allows changes.
3. Time is one of the main factors in software development. Agile follows a short development period and submits the work done to the customer so that the customer is satisfied with the work of the team. The customer needs the software to be complete as soon as possible.
4. Everyday agile meetings are conducted to track the progress of the development and any other possible issues present. These meetings occur face-to-face so that everyone can share their ideologies. These interactions lead to innovative ideas.
5. Business people, developers must work side by side so that every requirement asked by the customer is developed and there will not be any other business-related issues for the developed project.
6. Agile teams are a group of highly motivated people. They do not need any motivation to do the work done, they are self-motivated to complete the work they are meant to be done. This type of environment creates the responsibility to complete the work in the given amount of time.
7. In every sprint, some parts of the requirements are developed and presented to the customer. The working project defines the amount of work done as one can see clearly how much work is done to complete the project. Working project is the primary source of determining the amount of work done in agile.
8. With the small sprint time, the agile team is able to assess the amount of work that will be completed within the sprint. It is easier to develop small parts of software in sprints rather than develop entire software at once.
9. Always keeping track of the technologies used and maintain good code. When bugs are solved as soon as they are discovered, less cost is used. But when they are not solved in earlier stages of development, the cost to solve the bugs increases. Similarly maintaining a neat design is easier compared to complicated design, which is difficult to maintain.
10. As Agile develops software in sprints new functionalities and new code is added to the already existing software. It is a good practice to develop code with as much low dependence between the modules.
11. Agile works in a self-organized team, decisions are taken individually by the team. Given a time limit and freedom to complete the project the team operates and takes decisions such that it can complete the development in the given time. With the group of self-motivated people working together every sprint develops the best architecture, plan, and working product.
12. When a certain issue occurs the agile team digs deep and finds the root cause of the issue and improves itself so that the issue is not created in future developments.

**Relation between agile values and principles.**

The first value “Individuals and interactions” describe that people play a crucial role in software development and, they must have trust and communicate with each other. Agile principle number 4 is talking about face-to-face interactions to obtain the most efficient method for communicating information. Face-to-face communication is possible from the interaction between team members. If the team members are highly motivated, everyone feels they are responsible for doing high-quality work thus developing the best quality software. Everyone trusts the team members that they are producing quality work, which is stated by principle 6. Principle 11 states that in the case of the self-organizing team, there is no need for someone to monitor the team's work and guide them. Given a project and deadline, the self-motivated team produces the best quality software by taking the best decisions and tools.

Working software is a means of determining the amount of work done in software development. Principle 1 describes the first priority is to develop the working software and complete it within less time with the help of continuous development. Principle 3 describes delivering the working software in less time frame and complete the development in less time to make the customer happy. As said earlier working software acts as a means of measuring the amount of software developed. With the working software at our hands' developers, managers, customers can find the amount of software that is yet to be developed, this is stated by principle 7.

Principle 5 describes that both the developers and business people must work closely and communicate frequently, which is nothing but "Customer Collaboration". Principle 8 states that sponsors, develops, stakeholders must work together to create plans and make the development process move forward. This is obtained by "customer collaboration".

The requirements are never stable in the development of software, principle 2 states that agile always welcome change even late in the development phase. Principle 9 defines that developers must always keep an eye on the technologies using and adapt to them, less complicated code can be maintained easily, as new changes are added constantly it is best to develop good design software. New code is added to the existing software in every sprint, hence developing code with less dependency, simplicity is essential for maintaining software. The development team measures its performance after every project and keeps adapting itself to avoid mistakes, failures, and poor quality of software.

|  |  |
| --- | --- |
| Agile Values | Principles |
| 1. **Individuals and interactions** | 4,6,11 |
| 1. **Working software** | 1,3,7 |
| 1. **Customer collaboration** | 8,5 |
| 1. **Responding to change** | 2,9,10,12 |

# Lean Software development

Lean is a method developed from the manufacturing domain with the main intent to develop a product that gives reasonable value to the customer for the money spent. The lean teams were created so that they outshine the traditional teams and produce the best quality products. The software industry experts came up with the idea to incorporate the ideology of lean development in the development domain. By taking the principles in the lean development manufacturing domain and making them as a base for developing the software.

There are four values in the lean development[1]:

1. Commitment: All the team members must show commitment towards completing the project.
2. Focus: Focus on the work they are doing so that they can excel in creating a high-quality product.
3. Openness: Team members must share their work with others so that everyone in the team has a good understanding of the progress of the project.
4. Last responsible moment: Decide the most important decisions as late as possible.

The lean principles are as follows[1]:

1. Optimizing the whole
2. Eliminating the waste
3. Build quality in
4. Learn Constantly
5. Deliver fastly
6. Engage Everyone
7. Keep getting better

Optimizing the whole represents, understand the project to be built and then try to understand the process better. Clearly understand the user requirements and needs then try to develop the product so that the customer is satisfied with the product.

Elimination of waste: By following this practice, it is possible to eliminate features that are no longer needed by the customer to improve the quality of the product. Avoid the development of unnecessary feature development called as “gold plating”

Build Quality in: This practice includes using tools for automated testing, unit testing, integration testing, and so on to improve the quality of the software. The underlying motto is to develop products with high quality.

Learn constantly: Find out all possible options present in the development of software, After assessing all the available options then take the best option present. This practice involves delaying the decision-making process till the very end to ensure that all possible decisions have been evaluated.

Deliver fastly: This practice involves techniques for developing a high-quality product in rapid cycles to cope with the rapidly growing software industry. Developing the working product is the main motive of lean development.

As a team project, everyone has equal importance in the development of the project. All the team players must be involved to get a good quality product. When all the members involve in the development of the project the best quality product is developed with minimum time.

After the completion of every project perform a self-analysis from the feedbacks obtained from the customer or own mistakes, so that these mistakes can be rectified and avoid repeating the same mistakes in future.

**Conclusion**[2]

Both agile and lean are profoundly used methodologies in software development, as both showed promising results of sucess. Agile and lean are developed from different environments, lean being developed from operational environment where as agile from development environment. Agile is a mouldable software development method. Lean does not specify any certain development practices, it gives the freedom to choose accordingly, Its scope is extended past software development. There are many different roles in lean compared to agile, even though they sound similar they are very much different. Both of the methods are used to develop high quality products. Some of the differences between Agile and lean are mentioned in the following table:

|  |  |
| --- | --- |
| **Agile** | **Lean** |
| A software development method for developing software in less time frame. | Lean developed by Toyota for car development, a method adopted by software Industry to develop software. |
| Small cycles of development with small goals. | Small lot, Just In Time production. |
| Every member has his own skills and importance | Team members are multi-skilled people |
| They follow work or product standards to develop best quality products. | Follow code-standards, design standards, testing standards for developing best quality software. |
| Constant development of product. | Evolution of product achieved by small cycles. |

**References:**

[1]Learning Agile: understanding Scrum, XP, Lean, Kanban

* Stellman, Andrew and Jennifer Greene. 2014.*Learning Agile: Understanding Scrum, XP, Lean, and Kanban*. Sebastopol: O'Reilly Media, Incorporated.

<https://learning.oreilly.com/library/view/learning-agile/9781449363819/ch03.html#principle_number_symble_10_colon_simplic>

[2]Lean Software development:

* Poppendieck, M. and M. A. Cusumano. 2012. "Lean Software Development: A Tutorial." *IEEE Software* 29 (5): 26-32.

<https://ieeexplore-ieee-org.miman.bib.bth.se/stamp/stamp.jsp?tp=&arnumber=6226341>

[3]Does Agile Work?

* Serrador, Pedro and Jeffrey K. Pinto. 2015. "Does Agile Work? — A Quantitative Analysis of Agile Project Success." *International Journal of Project Management* 33 (5): 1040-1051.

<https://www-sciencedirect-com.miman.bib.bth.se/science/article/pii/S0263786315000071?via%3Dihub>

[4] Empirical studies of agile software development: A systematic review

* Dyba, Tore and Torgeir Dingsoyr. 2008. "Empirical Studies of Agile Software Development: A Systematic Review." *Information and Software Technology* 50 (9): 833-859.

https://reader.elsevier.com/reader/sd/pii/S0950584908000256?token=F5F46C8DAA5D6670CA5E1D61456C16DE682AC723B092DD5FDFBD3269975B79C55BB2BB8C83D9F099D376D84733874325&originRegion=eu-west-1&originCreation=20210926014810