Table of Contents

[Introduction to Agile: 2](#_Toc23859952)

[Agile Vs Waterfall: 3](#_Toc23859953)

[Team Psychology: 5](#_Toc23859954)

[Team communication: 7](#_Toc23859955)

[Agile Processes: 8](#_Toc23859956)

[Version Control 11](#_Toc23859957)

[Implementation of Version Control 12](#_Toc23859958)

[GIT and GitHub 12](#_Toc23859959)

[GIT commands 13](#_Toc23859960)

[Benefits of using Version Control 16](#_Toc23859961)

[Test Driven Development (TDD): 17](#_Toc23859962)

[Behavior Driven Development (BDD): 19](#_Toc23859963)

[BDD Workflow: 19](#_Toc23859964)

[Tool used for BDD: 20](#_Toc23859965)

[Figure 1: Research Result on success rate of Agile and Waterfall Methodology 3](#_Toc23843866)

[Figure 2: Agile Mindset (Agile Infographics, 2019) 5](#_Toc23843867)

# Introduction to Agile:

So what really is Agile? Agile Alliance define agile as, “The ability to create and respond to change in order to succeed in an uncertain and turbulent environment”. It is particularly vital in case of software development since it is intrinsically erratic. Also, quick and steady feedback is crucial in software development. Therefore, Agile portrays a lot of techniques and practices to convey continuous incentives to clients. At its core, teams that are self-organizing and cross-functional implement methods that are suitable to their setting in order to generate solutions through joint effort. If we look at the beginning of Agile, we need to go back to the late 1990’s, when numerous software development processes were in rise with their own set of principles. Nevertheless, those methodologies had common overlaps emphasizing:

* Collaboration between the development team and business stakeholders
* Frequent delivery of business value
* Small, self-organizing teams
* Innovative ways to create, test and deploy code

And in 2001, that collection of methodologies was termed as “Agile” with a gathering of 17 software development specialists in Utah to discuss and offer their different ways to deal with software development and hence, Agile Manifesto was created.

The Agile Manifesto values the following points:

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

Although both the items have certain value, the Agile Manifesto concludes that the items on the left are of greater value and be respected accordingly.

## Agile Vs Waterfall:

At first glance, waterfall methodology is a more traditional method and as the name suggests, the work flow is steady and downwards. Here, single one of the stage of software development life cycle occurs in a progression. On the other hand, Agile is one of the new and widely popular methodology that offers a linear, iterative and sequential approach. Recently, there has been a rise in the number of organizations adopting Agile. It is the result of a research in 2015 by the Standish Group that concluded Agile processes produce higher success rate when compared to Waterfall method.

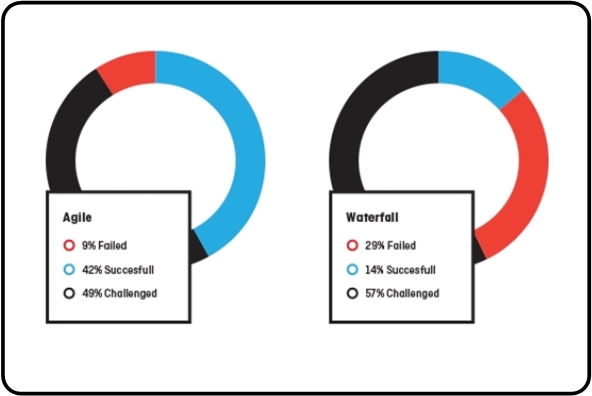


Figure 1: Research Result on success rate of Agile and Waterfall Methodology

Below is a side by side comparison of Agile and Waterfall methodologies:

1. While Waterfall model partitions the procedure into various stages, Agile proposes to isolate the development lifecycle into iterations/sprints.
2. The development procedure in Waterfall ought to be actualized as one single undertaking and at this undertaking is separated into stages. However, Agile consists of numerous ventures that are the emphases of the various stages. They are centered on enhancing the quality and criticisms from the clients.
3. Next up, Waterfall is strictly structured and often rigorous. On the other hand, Agile is more flexible and so is preferred by most of the Project Managers.
4. Also, Agile pursues an iterative development approach and hence some stages may occur more than once. But in Waterfall, all of the venture stages are finished at once.
5. One of the major difference is that Waterfall does not allow to change the prerequisites once the development begins. However, due to the flexibility offered by Agile, changes in requirements are always welcomed even after the planning is finished.
6. There is also a major difference in testing and quality brackets. In Waterfall, testing cannot be initiated unless development phase is completed. Whereas, Agile suggests to perform testing simultaneously with the coding phase.
7. In case of Waterfall, the customer’s participation in development is not required at all but in Agile, customers are included throughout the development stage to focus on customer satisfaction and feedback.
8. If the project we are working on has a clearly characterized necessities and does not anticipate any changes, Waterfall method is useful. Nevertheless, if the project demands varying and evolving requirements then Agile suits better.

# Team Psychology:

Generally, psychology is referred as the study of the mind and its capacities, especially those influencing behavior in some random setting. Nevertheless, in context of team circumstances, team psychology is referred as the psychological attributes, attitudes and behavior of the group/team including its individuals. With regards to the same, agile processes aim to shape behavior with the goal of creating productive, effective and successful teams. Hence, any agile team respects psychology as one of the fundamental aspect.



Figure 2: Agile Mindset (Agile Infographics, 2019)

The following points illustrate how Agile processes impacted on our team psychology:

* As per Agile, rather than taking guidance and direction from the directors/managers, members of the team are given the power to settle on their own choices and this practice has been proven effective too. However, this technique demanded our team to take up an altered psychology, one that guaranteed that procedures were possessed by the team, forming the project from the base up, and conflicts with our characteristic manner for adhering to the directions went down the levels of leadership.
* Likewise, our team had to figure out how to work and self-arrange in a manner that it is most likely unfamiliar to us, which took a tremendous measure of self-restraint, and it additionally entrusted individuals by providing us obligation and making us responsible for our tasks.
* Furthermore, there is a notable psychological marvel referred as “***The Pygmalion Effect***”, which relates to the hypothesis of inevitable outcome, or in business terms, that more prominent desires drive noteworthy performance. Embracing agile strategies and taking responsibility for ventures, our team had better standards of ourselves and in that capacity be increasingly persuaded to convey an epic outcome.
* Similarly, Agile’s one of the core tenant is to *“fail fast and often”*- which considers quick development, minor changes and blunders for cost efficiency while permitting individuals to learn and build up their aptitudes. It helped us to overcome dread of disappointment in case of failures.
* Agile’s another key component is distribution of work into reasonable segments, referred as sprints or iterations. This feature enabled our team to praise our accomplishments often, which have been demonstrated to be fundamental for improved team psychology, improving confidence and inspiration.

To conclude, each agile procedure, may it be open plan workplaces, daily stand-ups, reviews, etc. are intended to encourage social collaborations which thus advance a positive working psychology.

# Team communication:

In terms of project management, communication is defined as the formal and casual ways the individuals on the project team transfer information or messages to one another. Although good communication is a need for agile projects, communication in Agile differs by a certain margin than in conventional projects. Nevertheless, agile principles set an alternate pace for agile projects, stressing effortlessness, straightforwardness, and face to face discussions.

In fact, the real key to agile success is also communication. It is because of the common practices of an Agile team : daily stand-up meetings, retrospectives after every sprint, pair programming and buddy reviews, collaboration with customers, and more face-to-face time instead of mountains of documentation. If we analyze the agenda behind all these tasks, the answer is obviously frequent and open communication.

There were certain communication issues within our team during the project and they are as follows:

* Absence of written as well as verbal communication skills among the individuals of a team.
* Lack of communication between developers and testers about the user stories.
* Exclusion of testers from vital design-related or technical discussions for applicable user stories.
* Outweighing other’s thoughts and ideas by ruling the discussion.
* Absence of team discussions such as daily stand-ups and retrospective meetings for iterations or sprints.
* Hesitation of team members while recommending enhancements or changes at the time of designing.

The above mentioned issues were addressed by following the listed measures:

* Verbal/written communication skills training to the naïve members.
* Scheduled meeting to discuss user stories with both developers and testers in one place.
* Encouraging shy members to make mistakes and openly discuss them.
* Utilizing informal communications such as chats and emails to keep everyone updated about the status of projects and discuss problems.

# Agile Processes:

Typically speaking, agile processes/methodologies are just subsets of agile that has much of the same underlying philosophy including practices and characteristics. The only differentiating factor is an execution outlook as single one of them has its own one of a kind blend of practices, strategies and name. Below is a list of popular agile methodologies:

* **Scrum:** It is one of the lightweight agile processes and is widely utilized to oversee iterative and steady projects of numerous types. Because of its proven efficiency, simplicity, and capacity to join different general practices advanced by other agile models, it has been widely popular. Its main features are:
* Product backlog to prioritize system functionality along with features, bug fixes, and non-functional requirements, etc.
* Define the sprint duration.
* After prioritization, estimate and sign-up to deliver “potentially shippable increments” of product during each sprint/iteration.
* If necessary, analyze and reprioritize the Product backlog after delivering a sprint.
* **Lean:** An iterative and highly flexible process without strict rules, guidelines, or methods and solely focuses on delivering value to the client via effective value stream mapping. Its main principles are:
  + Eliminating Waste
  + Amplifying Learning
  + Deciding as Late as Possible
  + Delivering as Fast as Possible
  + Empowering the Team
  + Building Integrity In
  + Seeing the Whole

Its main aim is to reduce waste by ensuring that the customer only selects the truly valuable features for a system. Next, it focuses on efficient use of team resources and also suggests to write automated unit tests along with the coding.

* **Kanban:** It is a child methodology incorporated in Lean and is highly visual workflow management technique. Almost 83% of Lean users use Kanban as it visualizes and administers the development of product with significance to steady delivery without piling up load on development life cycle. Its vital principles are;
  + Visualize what you’ll do today (workflow automation)
  + Limit the amount of work in progress (WIP)
  + Enhance flow
* **Extreme Programming (XP):** Also regarded as one of the popular and controversial agile method, it is a diligent process for high grade software development, focused on quick and steady delivery. Moreover, it aims to enhance software quality as well as responsiveness in the face of changing customer requirements. As the name suggests, it takes traditional practices to “extreme” level. For example, code review is done continuously via ***pair programming***. The following are its supporting practices:
  + Planning Game
  + Small Releases
  + Customer Acceptance Tests
  + Simple Design
  + Pair Programming
  + Test-Driven Development
  + Refactoring
  + Continuous Integration
  + Collective Code Ownership
  + Coding Standards
  + Metaphor
  + Sustainable Pace
* **Crystal:** One of the lightweight method comprised of other models including Crystal Clear, Crystal Yellow, Crystal Orange and etc. These depend upon several factors including team size, system criticality, and project preferences. The primary focus is on individuals and synergy between them during the development process. Additionally, it also addresses business priority and criticality of the system.
* **Dynamic Systems Development Method (DSDM):** Based on the eight key principles, which guides the team and builds a mentality to hand-over on time and budget, DSDM explicitly calls out “fitness for business purpose”. Likewise, it aims to deploy 80% of system within 20% of time. Its eight key principles are:
* Focus on the business need
* Deliver on time
* Collaborate
* Never compromise quality
* Build incrementally from firm foundations
* Develop iteratively
* Communicate continuously and clearly
* Demonstrate control
* **Feature Driven Development (FDD):** The fundamental purpose of FDD is to convey substantial, working software, over and again. The upside of implementing FDD is that it is scalable even to enormous groups because of the idea of “just enough design initially**” (JEDI).** As a result of its component driven procedure, FDD is an incredible answer to uphold control for gradual and innately complex Agile projects. To conclude, its five basic activities are:
* Development of an overall model
* Building a feature list
* Planning by feature
* Designing by feature
* Building by feature

Out of all these Agile processes, for our project we implemented the **Scrum** framework as shown in the figure below:

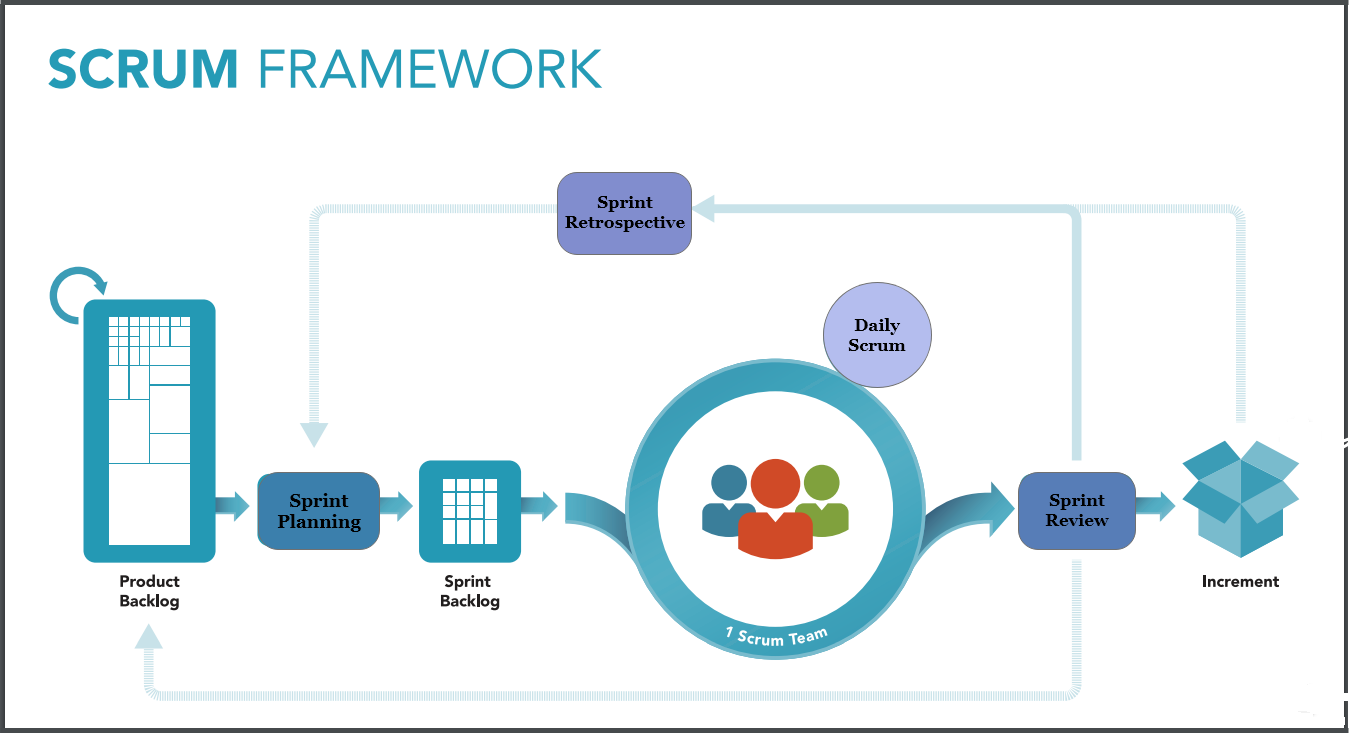


Figure 3: Scrum Framework (The Scrum Framework Poster, 2019)