

Agile Software Development – A Beginners Tutorial and Detailed Guide

Before I start explaining about Agile Software Development, in a detailed, easy, step-by-step manner from scratch, let’s first understand the meaning of the term “Agile” first.

### **Agile Meaning**

If we search the meaning of Agile in a dictionary or google, you will get its meaning as ‘moving **quick**‘. Agile simply means **quick**.

As you have understood the general meaning of Agile, now let’s get started with the understanding of **Agile Software Development**.

*Agile Means Quick*

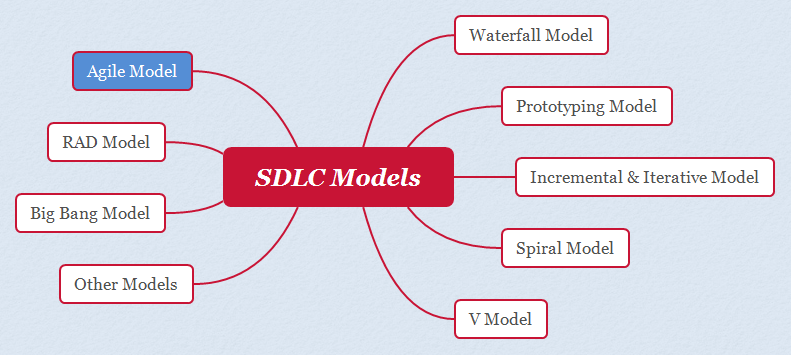
### **Agile Definition**

The agile Model is one of the different Software Development Life Cycle models (i.e. SDLC Models) available in the market.

The below are the different Software Development Life Cycle Models available in the market:

1. Waterfall Model
2. Prototyping Model
3. Incremental Iterative Model
4. Spiral Model
5. V Model
6. **Agile Model**
7. RAD Model (Rapid Application Development)
8. Big Bang Model
9. And many more

The below image depicts all the above-listed SDLC Models:

*Different SDLC Models*

So far, we have understood that Agile is one of the SDLC Models. Now, let’s understand why most of the Development Projects these days choose Agile as their SDLC Model.

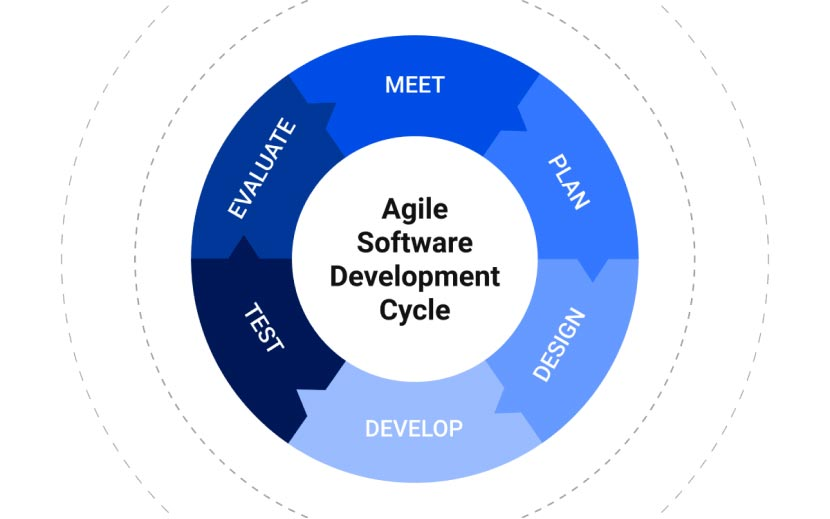
### **Agile SDLC Model**

Traditional Software Development doesn’t adapt to change and hence is not suitable for the projects these days which adapt/change quickly. Medium and Large Size Projects evolve over a period of time by implementing the changes coming in for surviving in the market.

In Traditional Software Development says Waterfall SDLC Model, all the requirements or changes required cannot be guessed at the beginning of the project and these models won’t adapt to the changes to requirements, code, the architecture of the Software to be developed. And the cost and time of incorporating and fixing these required changes in these Traditional Software Development projects are very high.

The solution for the above problem is Agile SDLC Model. The below are the different points, which explain how Agile solves to the problem of quickly adapting to changes:

1. Agile is an Iterative Development Model, where the development happens over multiple iterations. Instead of implementing all the features required for developing the software at a go, only a few sets of features will be identified based on priority and implemented in every iteration. Hence the development in Agile SDLC Model happens over multiple iterations.
2. In every iteration of the Agile SDLC Model, the requirements evolve, the changes will be quickly adapted and the problems will be fixed/resolved.
3. All the teams like Business, Development, and Testing, etc. will be collaborating effectively on the agreed functionalities to be implemented in an iteration and solving the problems in an efficient, quick, and focused manner.
4. Because of all the above three points, Agile enables the teams to deliver the Software faster with good quality and quick adaptability to incorporate changes.
5. The agile SDLC Model develops the software over multiple iterations in an incremental manner, where the Software to be developed will evolve with more features over a period of time.



*Agile SDLC Model*

The agile SDLC Model has several frameworks (i.e. methodologies) for its implementation in our software development projects. As we have understood the purpose of using the Agile SDLC Model for Software Development Projects, now let’s go a bit further and understand the different Agile frameworks (i.e. Agile Methodologies) that are available for us to implement the Agile SDLC Model in our Software Development Projects.

### 

### 

### 

### 

### 

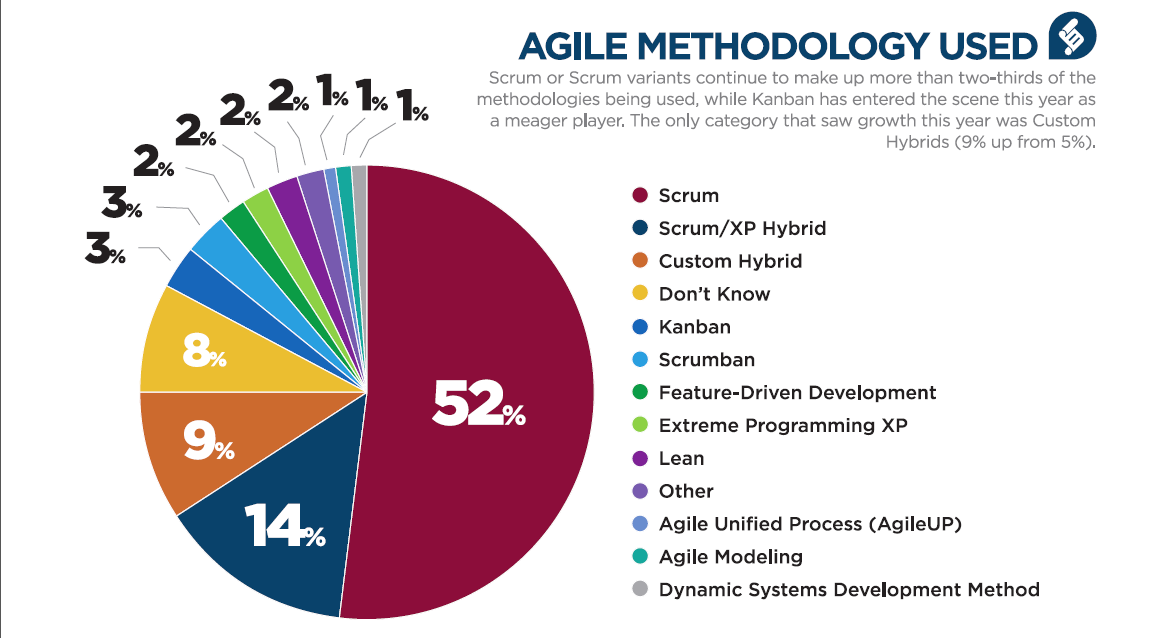
### **Agile Methodologies**

There are several frameworks/methodologies in Agile SDLC Model, which we can use for developing the projects. **Scrum** and **Kanban** are the most used frameworks/methodologies in Projects when the Agile SDLC Model is chosen for developing the Software.

The below are the different Agile Methodologies/Frameworks which can be used in the Software Project Development:

1. Scrum
2. Kanban
3. Lean
4. XP (Extreme Programming)
5. Crystal
6. FDD (Feature Driven Development)
7. DSDM (Dynamic Systems Development Methods)
8. And others methodologies

The below image depicts all the above listed Agile Methodologies:

**

*Different Agile Methodologies – Frameworks*

### 

### **Agile Scrum Methodology is Number One**

The Number one Agile Methodology which is most popular than any other Agile Methodologies in the market for developing Software Projects is **Agile Scrum Methodology.** In the coming sections, I will explain this Agile Scrum Methodology/Framework in detail.

### 

### **Agile is not Equal to Scrum**

As Scrum is the famous Agile Methodology/Framework, everyone tends to refer to Scrum as Agile, but they are not the same. Scrum is one of the different Agile Methodologies/Frameworks, whereas Agile is one of the SDLC Models, where Software is developed over multiple iterations.

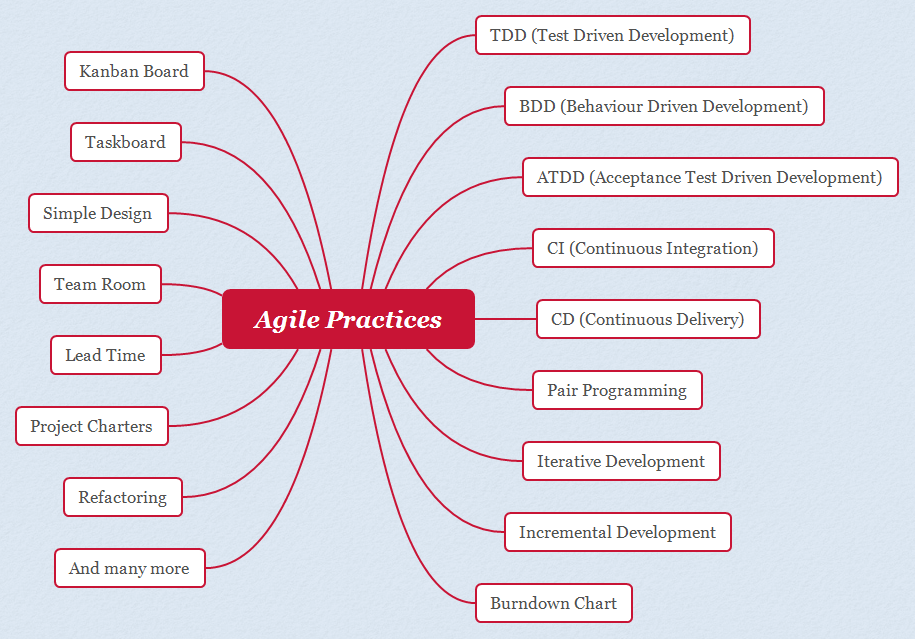
### **Agile Practices**

Agile Practices are nothing but the best practices that we can implement in any of the Agile Software Development projects. Multiple Agile Practices can be used/implemented in the Agile Development Projects. F0r example, there is a Software Project which is using the Agile SDLC Model for its development and Agile Scrum as the implementation Methodology/Framework, it can also implement several Agile Practices like Behaviour Driven Development (BDD) and Pair Programming to get better results.

The below are the different Agile Practices that can be implemented in the Agile Development Projects:

1. TDD (Test Driven Development)
2. BDD (Behaviour Driven Development)
3. ATDD (Acceptance Test Driven Development)
4. CI (Continuous Integration)
5. CD (Continuous Delivery)
6. Pair Programming
7. Iterative Development
8. Incremental Development
9. Burndown Chart
10. Kanban Board
11. Taskboard
12. Simple Design
13. Team Room
14. Lead Time
15. Project Charters
16. Refactoring
17. And many more

The below image depicts all the above listed Agile Practices:

*Agile Practices*

If you want to refer to the complete list of Agile Practices, you can refer to them here.

### 

### **Agile Manifesto**

Let’s forget about ‘Agile Manifesto’ for a while, and now let’s focus on understanding the term ‘Manifesto’ separately first.

For example, every Political party before elections and during the political campaign produces a manifesto (an open public declaration of their policies, how they are going to serve or make difference from other political parties, etc.), which thereby helps the general public in deciding whom to vote.

Agile SDLC Model also has a Manifesto, which was prepared by 17 developers during an outing. This Manifesto is nothing but a document, containing **4 Agile Values** and **12 Agile Principles**.

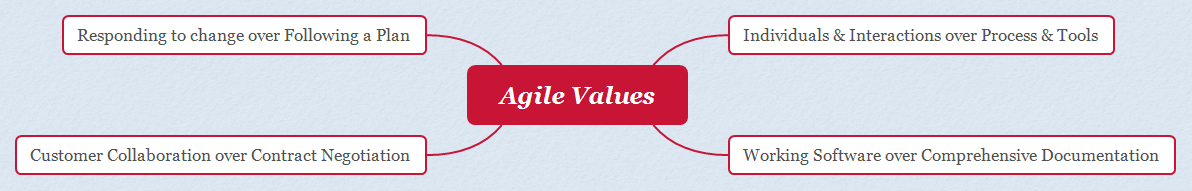
In the coming sections, I will explain more about these 4 Agile Values and 12 Agile principles in detail.

### **Agile Values**

As per Agile Manifesto, the below are the four values that Agile Methodologies and Agile Practices should follow:

1. Individuals and Interactions **over** Process and tools
2. Working Software **over** Comprehensive Documentation
3. Customer Collaboration over Contract Negotiation
4. Responding to change over Following a Plan

The below image depicts all the above listed Agile Values:

*Agile Values*

**Agile Value #1: Individuals and Interactions over Process and Tools**

In Traditional Development Projects, the Software Teams focus on having the best possible tools and processes for developing the software, but they don’t focus on the Individuals who are part of the Project and how their communication/interaction is helping the Project while building the software.

But Agile Manifesto, suggest that the people behind these process and tools are even more important in helping the Project move towards its success. Keeping the Project in the right hands, and making sure they are communicating well which each other, and solving the problems that arise in the projects, will actually help the project move towards its success, than just implementing the great processes and using the best tools. Hence Individuals & Interactions over Process & Tools is followed in the Agile Development Projects.

**Agile Value #2: Working Software over Comprehensive Documentation**

In Traditional Development Projects, more priority is given to the documentation and thereby showcasing these documents to the Customers/Clients on a regular basis. But what actually the Customer/Client wants is working software, overall these detailed and glittered documents and reports.

Agile Manifesto, recommend the Agile Development Projects to give priority to something that the Customer/Client really wants i.e. Workable/Deliverable Software over documentation. Hence in Agile Software Development Projects, more importance is given to the working of the Software and less importance is given to the documentation.

**Agile Value #3 : Customer Collaboration over Contract Negotiation**

In traditional Development Projects, when the Project Team completely develops the Software and delivers it to the Customers/Clients, the Customer may identify the functionalities in the Software that are not meeting their expectations or may want some additional functionalities to be incorporated into the Software. In this case, the Project Manager negotiates with the Customers on adding these additional changes to the originally agreed contract, instead of listening to them, understanding what is that they really want, and incorporating those additional changes into the Software.

Agile Manifesto, recommends that the focus should be on continuous development accompanied by intaking as much as feedback from the Customers and incorporating the needed changes into the Software, instead of negotiating with them over contracts.

**Agile Value #4: Responding to Change over Following a Plan**

In Traditional Development Projects, more emphasis/focus is given to Planning and road map creation in advance. But, Project Team cannot plan, know or design everything that is needed in advance (i.e. way before the Project starts). Customers/Clients may not know all the required equipment of the Software in Advance. Hence the Traditional Development Projects, tend to fail and go way behind the allocated budget.

Agile Manifesto, suggest the Software Teams have the ability to adapt to the required changes when needed in the Software Projects, rather than sticking to the initial or non-dynamic/non-changeable plans.

*4 Agile Values*

### 

### 

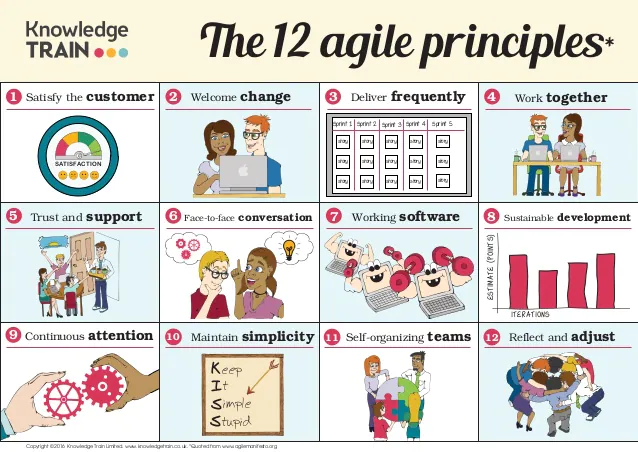
### 

### 

### 

### **Agile Principles**

As per Agile Manifesto, below are the Twelve Principles that Agile Methodologies and Agile Practices should follow:



1. Our highest priority is to satisfy the customer through the early and continuous delivery of valuable software.
2. Welcoming 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 for the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
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 behavior accordingly.

*12 Agile Principles*

So far, we have learned a lot about Agile SDLC Model, Agile Methodologies, Agile Practices, Agile Values, and Agile Practices. Now, let’s learn even more about Agile from the to be followed sections.

### **Agile Team**

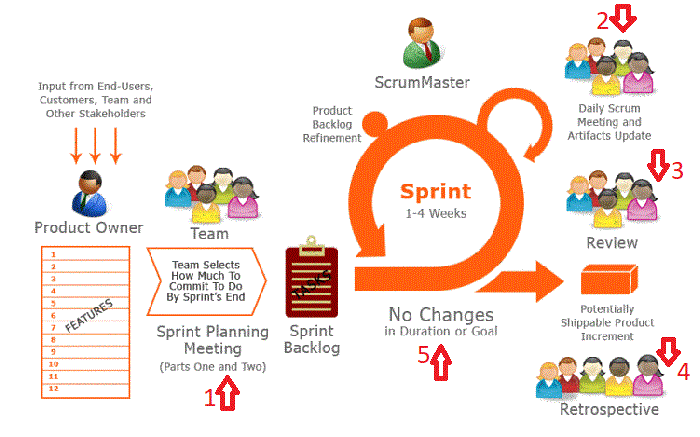
Agile refers to the team as Cross-Functional Team, instead of calling it as Development Team, Testing Team, etc.

Agile believes that each and every member of the Team, irrespective of whether a Developer or Tester, etc.

should contribute to the successful delivery of workable Software to the Customers.

In the coming sections, I am going to explain one of the methodologies of Agile i.e. Agile Scrum in detail in the next section, where we are going to use the term ‘Agile Team’ instead of terms or roles like Developer, Tester, etc.

*There are Five events in the Agile Scrum Framework.*

1. *Sprint Planning*
2. *Daily Scrum*
3. *Sprint Review*
4. *Sprint Retrospective*
5. *The Sprint  
     
   Figure: Life cycle of Scrum from*

***Sprint Planning*** *is the event in which the Product Owner presents the ordered product backlog to the development team.*

*As the word suggests, 'Sprint Planning' means we are going to plan the work to be done in the Sprint.*

*There are two main parts - 'What' and 'How'.*

* *'What' can be done in this Sprint?*
* *'How' will the selected work get done?*

***What can be done in the Sprint -***

* *In this part, the Product Owner presents the product backlog items with high business value tasks as a first priority to the development team.*
* *All team members collaborate to understand the work.*
* *The development team has all rights for taking the number of items in the sprint; the Product Owner or Scrum Master cannot force the team for more tasks.*
* *Sprint has a Sprint Goal that is decided in the Sprint Planning phase and helps the dev team to focus on more details of the tasks and what really needs to be accomplished.*

***How the selected works get done -***

* *In the second part of Sprint planning, the development team plans on how to produce the next product increment.*
* *The development team does planning and design to make sure that they will achieve Sprint Goal.*
* *Product Owner may remain available during the planning if the team has any questions.*
* *Scrum Master also helps the team in case any arrangements need to be done for achieving the Sprint Goal.*

***Summary***

* *Who will participate - Development Team, Product Owner, and Scrum Master.*
* *When - At the beginning of the sprint*
* *Time-box - maximum of eight hours for the four weeks Sprint.*
* *Input - Product backlog, latest product increment, the definition of done, team capacity, past performance.*
* *Output - Sprint goal, sprint backlog, a clear understanding of work to be done during the sprint, transparency.*

***Daily Scrum*** *is a daily standup meeting between Scrum Team. It's a 15-minute time-boxed daily event generally held at the same place and the same time.*

*In this meeting, the development team explains -*

* *What did I do yesterday?*
* *What will I do today?*
* *Are there any issues or Impediments?*

*The development team inspects the progress towards the Sprint goal in the daily standup. The everyday 15-minute meeting with the team helps each member to work together in a self-organized way, and they learn collaboration.*

*Scrum Master makes sure that each member participates in the daily standup. This helps the team in terms of better communication, decision-making, and improved level of knowledge.*

*The team learns about inspection and adoption.*

***Summary***

* *Who will participate - Development Team, Scrum Master (Product Owner – Optional)*
* *When - Everyday same place, same time*
* *Time-box - 15 mins Max.*
* *Input - 3 questions - "What did I do yesterday?", "What will I do today?" and "Are there any issues or Impediments?"*
* *Output - Clear idea about the progress towards Sprint goal, any issues or Impediment idea in the earlier stage.*

***Sprint Review*** *is held at the end of each Sprint. The main purpose of the Sprint Review is to inspect the incremented product created in the Sprint and adapt the product backlog if needed.*

*The Development Team demonstrates the work that it was done. During the review meeting, Scrum Team and Stakeholders collaborate to discuss what was done in the Sprint and what to do next.*

*It will give valuable feedback to the subsequent Sprint planning. The sprint review meeting is an informal meeting, not a status update meeting.*

***Summary***

* *Who will participate - Development Team, Scrum Master, Product Owner, and Stakeholders*
* *When - End of the Sprint and before sprint retrospective.*
* *Time-box - Four hours for one month's Sprint.*
* *Input - Product Increment, Changes to the product backlog during the Sprint.*
* *Output - Updated product backlog, New Idea, clear picture of tasks and product.*

***Sprint Retrospective*** *is the chance for the Scrum team to inspect itself and create a plan for improvements to be taken care in the next Sprint.*

*The main purpose of the Sprint retrospective is,*

* *Inspect how the sprint went with regard to process, tool, and people.*
* *Identify items that went well and potential improvements.*
* *Create an action plan to implement improvements in the scrum team.*

*Scrum Master encourages the team to improve. In the Scrum process framework, practices of processes make tasks more effective and enjoyable for the next Sprint.*

*Implementing these improvements in the next Sprint is the adaptation to the inspection of the Scrum Team itself.*

***Summary***

* *Who will participate - Development Team, Scrum Master, Product Owner*
* *When - at the end of the sprint after sprint review.*
* *Time-box - Three hours for one month's Sprint.*
* *Input - Results from the sprint, Sprint Events*
* *Output - Lesson learned Improvements and action list for the next sprint.*

***Sprint*** *is the heart of Scrum. It's a time-box of one month or 2 weeks and a container of all other events (Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective).*

*A new Sprint starts immediately after the end of the existing Sprint. During Sprint, the team works together to create a potentially shippable product increment.*

* *Sprint enables predictability.*
* *Sprint limits risk.*
* *Sprint duration should be consistent during the development effort.*
* *The sprint ends when the timebox expires.*

*During the sprint no changes were made in the sprint goal, quality goals do not decrease and scope may be renegotiated between the Product owner and Development Team as more is learned.*

***Canceling a Sprint***

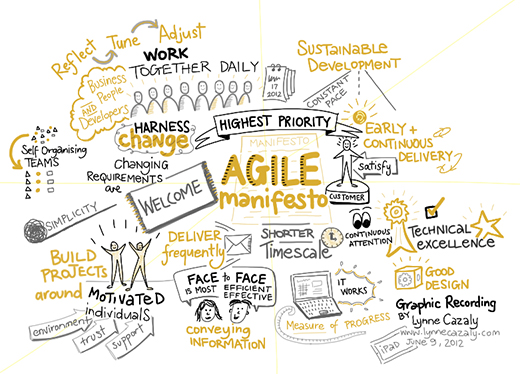
*A Sprint can be canceled before the Sprint duration is over. only the Product Owner can cancel the Sprint he has the authority to cancel*

*The product owner may do so under the influence of the stakeholders, the Development Team, or the Scrum Master. A Sprint would be canceled if the Sprint Goal becomes obsolete.*

*Hope you like this article. Please let me know your feedback in the comment section.*

***Reference***

* *https://www.scrumalliance.org/*

**