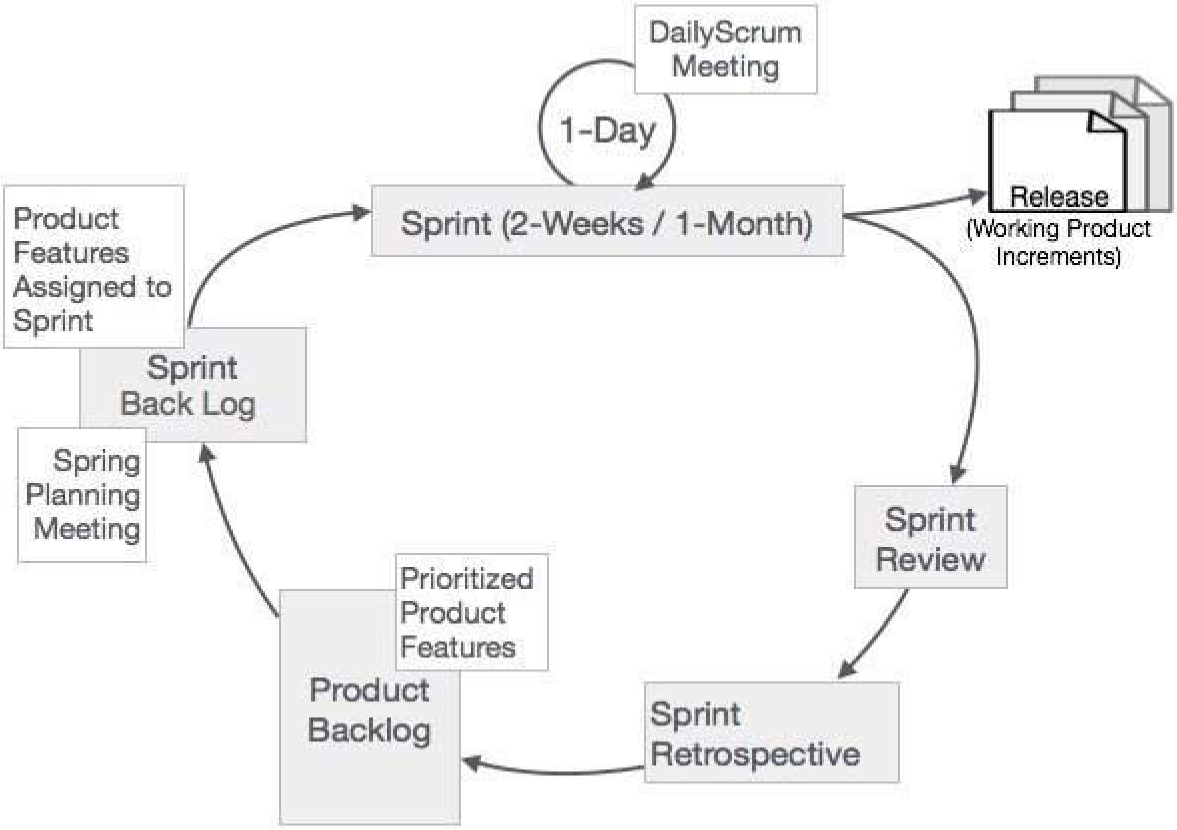
1)Scrum Process Framework


DailyScrum Meeting
1-Day
Sprint (2-Weeks / 1-Month)
Release
(Working Product Increments)
Product Features Assigned to Sprint
Sprint Back Log
Spring Planning Meeting
Sprint Review
Prioritized Product Features
Product Backlog
Sprint Retrospective

Scrum Process Framework can be viewed by means of a sequence of events and the corresponding artifacts. The Scrum events are time-boxed events. That means, in a project, every scrum event has a predefined maximum duration. These events enable transparency on the project progress to all who are involved in the project. The vital events of scrum are-

The Sprint
Sprint Planning
Daily Scrum Meetings
The Sprint Review
The Sprint Retrospective

The Sprint

During a Sprint, a working product Increment is developed. It is usually of duration two weeks or one month, and this duration remains constant for all the sprints in the project. We cannot have varying durations for the different sprints in a project. A new Sprint starts immediately after the conclusion of the previous Sprint.

The Sprint Goal is an objective set for the Sprint. It provides guidance to the Team on why it is building the Increment. It is created during the Sprint Planning meeting. The scope of the sprint is clarified and re-negotiated between the Product Owner and the Team as more about the requirements is learned. Thus, each Sprint is associated with it, a definition of what is to be built, a design, and the flexible plan that will guide building it, the development work, and the resultant product increment.

A Sprint should be cancelled if the Sprint Goal becomes obsolete. This might occur if the organization changes direction or if market or technology conditions change. A sprint can be cancelled only by product owner, though others have an influence on the same.

Due to the short duration nature of Sprints, cancellation during a sprint rarely makes sense. As the sprint cancellations consume resources, for getting re-organized into another Sprint, they are very uncommon.

If a Sprint is cancelled, and part of the work produced during the sprint is potentially releasable, the Product Owner typically accepts it. All the incomplete Sprint Backlog Items are put back into the Product Backlog.

Sprint Planning

The work to be performed in the Sprint is planned in the Sprint Planning Meeting. Sprint Planning Meeting is of duration of maximum of four hours for two weeks sprints and eight hours for one month Sprints. It is the responsibility of the Scrum Master to ensure that the meeting takes place and that all the required attendees are present and understand the purpose of the scheduled meeting. The Scrum Master moderates the meeting to monitor the sustenance of discussion and closure on time.

Sprint Planning focuses on the following two questions -

- What needs to be and can be delivered in the Sprint Increment?
- How will the work needed for the execution of Sprint be achieved?

The inputs to this meeting are -

- The Product Backlog
- The latest product Increment
- Projected capacity of the Team during the Sprint
- Past performance of the Team

The Scrum Team discusses the functionality that can be developed during the Sprint. Product Owner provides clarifications on the Product Backlog items. The team selects the items from the Product Backlog for the Sprint, as they are the best to assess what they can accomplish in the Sprint. The Team comprises of analysts, designers, developers, and testers. The work is carried out in a collaborative fashion, thus minimizing re-work.

The Scrum Team then comes up with Sprint Goal. The Sprint Goal is an objective that provides guidance to the Team on why it is building the Product Increment. The Team then decides how it will build the selected functionality into a working product

Increment during the Sprint. The Product Backlog items selected for this Sprint plus the plan for delivering them is called the Sprint Backlog.

Work during a sprint is estimated during sprint planning and may be of varying size and/or effort. By the end of the Sprint Planning meeting, the work is divided into tasks of duration of one day or less. This is to enable the ease of work allocation, and tracking the completion. If the Team realizes that it has too much or too little work, it can renegotiate the selected Product Backlog items with the Product Owner.

The Team may also invite others (not part of Scrum Team) to attend the Sprint Planning meeting to obtain technical or domain advice or help in estimation.

Daily Scrum Meetings

The Daily Scrum Meeting is a 15-minute meeting for the Team, conducted daily to quickly understand the work since the last Daily Scrum Meeting and create a plan for the next 24 hours. This meeting is also referred to as Daily Stand up Meeting.

The Daily Scrum Meeting is held at the same time and same place every day to reduce complexity.

During the meeting, each Team member explains -

- What did he do yesterday that helped the Team meet the Sprint Goal?
- What will he do today to help the Team meet the Sprint Goal?
- Does he see any impediments that prevent him or the Team from meeting the Sprint Goal?

Daily Scrum is mistaken to be a status tracking event, though, in fact, it is a planning event.

The input to the meeting should be how the team is doing toward meeting the Sprint Goal, and the output should be a new or revised plan that optimizes the team's efforts in meeting the Sprint Goal.

Though the Scrum Master coordinates the Daily Scrum Meeting and ensures that the objectives of the meeting are met, the Meeting is the responsibility of the Team.

If necessary, the Team may meet immediately after the Daily Scrum Meeting, for any detailed discussions, or to re-plan the rest of the Sprint's work.

Following are the benefits of Daily Scrum Meetings -

- Improve communication within the Team.
- Identify impediments, if any, in order to facilitate an early removal of the same, so as to minimize impact on the Sprint.
- Highlight and promote quick decision-making.
- Improve the Team's level of knowledge.

Sprint Review

A Sprint Review is held at the end of every Sprint. During the Sprint Review, a presentation of the increment that is getting released is reviewed. In this meeting, the Scrum Team and the stakeholders collaborate to understand what was done in the Sprint. Based on that, and any changes to the Product Backlog during the Sprint, the attendees arrive at the next steps required that could optimize value. Thus, the objective of Sprint Review is to obtain feedback and progress unitedly. If the product still have some non-achievable features then it will be checked in this stage and then the product is passed to the Sprint Retrospective stage

The Sprint Review is normally held for two hours for two week sprints and for four hours for one month sprints.

The Scrum Master ensures that -

- The meeting takes place.
- The participants understand the purpose.
- The meeting is focused on the required agenda and is completed within the required duration.

The Sprint Review includes the following aspects -

- Attendees include the Scrum Team and key stakeholders, as invited by the Product Owner.
- The Product Owner explains what Product Backlog items have been completed during the sprint and what has not been completed.
- The Team discusses what went well during the Sprint, what problems it ran into, and how those problems were solved.
- The Team demonstrates the work that it has completed and answers questions, if any, about the Increment.
- The entire group then discusses on what to do next. Thus, the Sprint Review provides valuable input to Sprint Planning of the subsequent Sprint.
- The Scrum Team then reviews the timeline, budget, potential capabilities, and marketplace for the next anticipated release of the product increment.
- The outcome of the Sprint Review is an updated Product Backlog, which defines the probable Product Backlog items for the next Sprint.

Sprint Retrospective

The Sprint Retrospective occurs after the Sprint Review and prior to the next Sprint Planning. This is usually a one hour meeting for two-week duration sprints and a three hour meeting for one month duration Sprints.

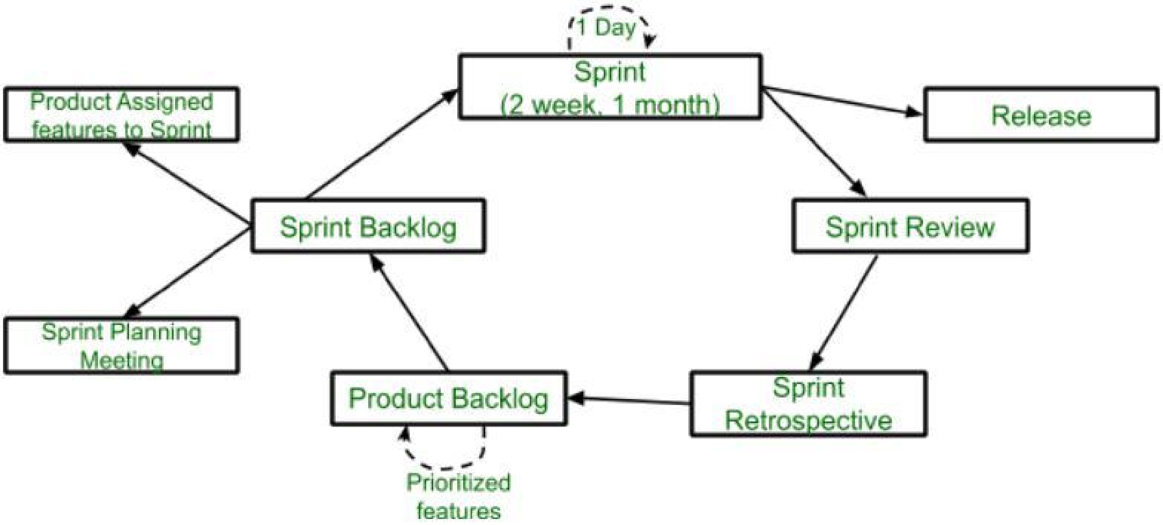
The purpose of the Sprint Retrospective is to -

- Combine the learnings from the last Sprint, with regards to people, relationships, process, and tools.
- Identify the major items that went well and potential improvements.
- Creation of a plan for implementing improvements to increase product quality.

The Sprint Retrospective is an opportunity for the Scrum Team to introspect and improve within the Scrum process framework so as to make the next Sprint outcome more effective.

2)

SCRUM LIFECYCLE



In Scrum, the prescribed events are used to create regularity. All events are time-boxed events, such that every event has a maximum duration.

Sprint

The heart of Scrum is a Sprint, a time-box of two weeks or one month during which a potentially releasable product increment is created. A new Sprint starts immediately after the conclusion of the previous Sprint. Sprints consist of the Sprint planning, daily scrums, the development work, the Sprint review, and the Sprint retrospective.

In Sprint planning, the work to be performed in the Sprint is planned collaboratively by the Scrum Team.

The Daily Scrum Meeting is a 15-minute time-boxed event for the Scrum Team to synchronize the activities and create a plan for that day.

A Sprint Review is held at the end of the Sprint to inspect the Increment and make changes to the Product Backlog, if needed.

The Sprint Retrospective occurs after the Sprint Review and prior to the next Sprint Planning. In this meeting, the Scrum Team is to inspect itself and create a plan for improvements to be enacted during the subsequent Sprint.

It is a process framework that defines certain rules, events, and roles to bring in regularity. However, it can be adapted to any organization, based on needs, provided the basic scrum rules are not violated.

3) CHARACTERISTICS OF A USER STORY

Good user stories are vital for a Scrum project as user requirements are primarily captured in this form. Good user stories alone are not enough to ensure that the product is of high quality. However, good user stories are important to communicate user requirements to the Scrum team to develop products of a high quality.

The INVEST acronym, given by Bill Wake, suggests characteristics of good user stories. The acronym stands for **I**ndependent**, N**egotiable**, V**aluable**, E**stimative**, S**mall, and **T**estable. Let us examine each characteristic in detail.

Independent

User Stories are often inherently dependent on each other. In such situations, it is not clear which story should be given the higher estimate. One of the solutions is to combine these stories into one, big independent story. If some of the needed functionality has already been created or implemented, then the estimate should be revised to reflect this change. Also, it is important to define a non-functional User Story that defines the pre-conditions for each dependent one.

Negotiable

User Stories should not be set in stone and should have enough scope to allow negotiation between the Scrum Team and the business team. Sometimes, the business team will have to sacrifice increased functionality due to budget constraints, or the Scrum Team might have to convince the business team to increase the budget if a critical user feature is required despite budgetary concerns.

Valuable

It is important that each feature in the backlog possess value. Saying that a feature should have value does not necessarily mean value in terms of prerogative for the end user. Some features may work in the background or may indirectly support another functionality that the user requires, so these features still possess value. This value should be clearly perceived by the Product Owner to be able to prioritize it and include it in the backlog.

Estimable

One of the characteristics of a good User Story is that it is easily estimated. The estimates do not have to be accurate, but they should be good enough to use the estimate for prioritization. Large User Stories are difficult to estimate, and small stories are generally easily estimated (we will discuss size later in the chapter). Stories that are difficult to estimate can point to underlying issues in the story—it may be that the story is too large or there is ambiguity in terms of what the User Story is.

Small

A small User Story is relatively easy to estimate. They are easier to track and can usually be completed within one iteration. Big stories take longer, and any delays take longer to report. When the Product Owner is trying to create stories that are the right size, he or she should consider the Scrum Team's experience and capabilities.

Testable

If a User Story cannot be tested and its functionality verified, then it becomes difficult to assess whether it can be considered "done." To verify a User Story's testability, the acceptance criteria must be clearly defined.

Example of good user story

here is an example of a good user story

Example 1

As an online visitor I want to add products in my shopping cart so that I can purchase multiple products at one go.

Acceptance Criteria [Abstract]

Products can be added to the cart

Products can be removed from the cart.

Shopping cart will be empty initially

Shopping cart will be empty after purchase

Products can be added with multiple quantities in the cart

Shopping cart will show the total product breakdown quantity and cost with grand total

Example 2

As an online visitor I want to add products in my shopping cart so that I can purchase multiple products at one go.

Acceptance Criteria [Elaborated]

1.Given my Shopping cart is empty
When I add a product to my cart
Then my shopping cart should contain 1 quantity of the added product

2. Given my Shopping cart contain 1 product
   When I add the same product to my cart
   Then my shopping cart should contain 2 quantity of the product, and I can see a warning the product has more than one quantity.
3. Given my Shopping cart contain 1 or more product
   When I click on delete button inside cart page for any specific product
   Then my shopping cart should remove the deleted product and show the remaining product in the cart
4. Given Shopping cart contain one or more product
   When click on confirm order on cart page
   Then My shopping cart should go empty after successful order
5. Given shopping cart contain one or more products
   When Open Cart page
   Then It Should show all the products break down with its selected quantity and cost and appropriate totals of cost and quantity
6. Given shopping cart contain one or more products in Cart page
   When change the quantity of any product Then the product's cost should be updated and subtotals of quantity and cost should change accordingly

4) SCRUM ROLES

The Scrum Team consists of three roles,

ScrumMaster, a Product Owner, and Team.

The ScrumMaster (sometimes written as the Scrum Master, although the official term has no space after “Scrum”) is the keeper of the scrum process. He/she is responsible for-

making the process run smoothly

removing obstacles that impact productivity

organizing and facilitating the critical meetings

The Product Owner is responsible for maximizing the value of the product and the work of the Team. How this is done may vary widely across organizations, Scrum Teams, and individuals.

The Product Owner is the sole person responsible for managing the Product Backlog. Product Backlog management includes-

Expressing Product Backlog items clearly.

Ordering the Product Backlog items to best achieve goals and missions.

Optimizing the value of the work the Team performs.

Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Team will work on further.

Ensuring that the Team understands items in the Product Backlog to the level needed.

The Product Owner may do the above work, or have the Team do it. However, the Product Owner remains accountable for these tasks.

The Product Owner is one person, not a committee. The Product Owner may represent the desires of a committee in the Product Backlog, but those wanting to change a Product Backlog item’s priority must address the Product Owner.

For the Product Owner to succeed, the entire organization must respect his or her decisions. The Product Owner’s decisions are visible in the content and ordering of the Product Backlog. No one is allowed to tell the Team to work from a different set of requirements, and the Team is not allowed to act on what anyone else says. This is ensured by ScrumMaster.

The Team

The Team is self-organizing and cross-functional. That means the team comprises of analysts, designers, developers, testers, etc. as appropriate and as relevant to the project.

Some people in the industry refer to this team as development team. However, such a reference is leading to controversy that the team can have only developers and no other roles. It is an obvious understanding that it is only a misconception. To develop a software product, we require all the roles and that is the essence of scrum – the team will function in collaboration. Cross-functional teams have all competencies needed to accomplish the work without depending on others not part of the team, and thus time and effort can be saved. The team model in Scrum is designed to optimize flexibility, creativity, and productivity.

Reference

Scrum Guide © 1991-2013 Ken Schwaber and Jeff Sutherland, All Rights Reserved.

5)Scrum Artifacts provide key information that the Scrum Team and the stakeholders need to be aware of for understanding the product under development, the activities done, and the activities being planned in the project. The following artifacts are defined in Scrum Process Framework -

Product Backlog
Sprint Backlog
Burn-Down Chart
Increment

These are the minimum required artifacts in a scrum project and project artifacts are not limited by these.

Product Backlog

The Product Backlog is an ordered list of features that are needed as part of the end product and it is the single source of requirements for any changes to be made to the product.

The Product Backlog lists all features, functions, requirements, enhancements, and fixes that constitute the changes to be made to the product in future releases. Product Backlog items have the attributes of a description, order, estimate, and value. These items are normally termed as User Stories. The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering.

A Product Backlog is an evolving artifact. The earliest version of it may contain only the initially known and best understood requirements. The Product Backlog gets developed as the product, and the environment in which it will be used, progress. The Product Backlog constantly changes to incorporate what is required to make it effective. As long as a product exists, its Product Backlog also exists.

As the product being built is used and gains value, the Product Backlog becomes a larger and more exhaustive list. Changes in business requirements, market conditions, or technology, cause changes in the Product Backlog, making it a live artifact.

Sprint Backlog

The Sprint Backlog is the set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and realizing the Sprint Goal.

The Sprint Backlog is a forecast by the Team about what functionality will be made available in the next Increment and the work needed to deliver that functionality as a working product Increment.

The Sprint Backlog is a plan with enough detail that can be understood but the Team to track in the Daily Scrum. The Team modifies the Sprint Backlog throughout the Sprint, and the Sprint Backlog emerges during the Sprint. This emergence occurs as the Team works through the plan and learns more about the work needed to achieve the Sprint Goal.

As new work is required, the Team adds it to the Sprint Backlog. As work is performed or completed, the estimated remaining work is updated. When elements of the plan are deemed unnecessary, they are removed. Only the Team can change its Sprint Backlog during a Sprint. The Sprint Backlog is a highly visible, real-time picture of the work that the Team plans to accomplish during the Sprint, and it belongs solely to the Team.

Increment

The Increment is the sum of all the Product Backlog items completed during a Sprint combined with the increments of all previous Sprints. At the end of a Sprint, the new Increment must be a working product, which means it must be in a useable condition.

Teams deliver an Increment of product functionality every Sprint. This Increment is useable, so a Product Owner may choose to release it immediately. If the understanding of an increment is part of the conventions, standards, or guidelines of the development organization, all Scrum Teams must follow it as a minimum. If it is not a convention of the development organization, the Scrum Team must define a definition of Increment appropriate for the product.

Each Increment is additive to all prior Increments and thoroughly tested, ensuring that all Increments work together.

As Scrum Teams mature, it is expected that their definitions of Increments expands to include more stringent criteria for higher quality. Any one product should have a definition of Increment that is a standard for any work done on it.

Sprint Burn-Down Chart

At any point in time in a Sprint, the total work remaining in the Sprint Backlog can be summed. The Team tracks this total work remaining for every Daily Scrum to project the likelihood of achieving the Sprint Goal. By tracking the remaining work throughout the Sprint, the Team can manage its progress.

Sprint Burn-Down Chart is a practice for trending the work expended by the Scrum Team. This has been proven to be a useful technique in monitoring the Sprint progress towards the Sprint Goal.

The Product Owner tracks this total work remaining at least every Sprint Review. The Product Owner compares this amount with work remaining at previous Sprint

Reviews to assess progress toward completing the projected work by the desired time for the goal. This information is shared with all stakeholders.

Case study:

Let us take a look at how a user story is framed for the scenario of a Bank Customer withdrawing cash from ATM.

User Story: Customer's Cash Withdrawal

As a **Customer**,

I want to **withdraw cash from an ATM**,

So that **I don't have to wait in line at the Bank**

User Story Acceptance Criteria

Each User Story also has Acceptance Criterion defined, so that correctness of implementation of the user story is confirmed by passing the Acceptance Test that is based on the Acceptance Criterion.

Following are the sample acceptance criterion for the example of User Story Customer's Withdrawal of Cash.

Acceptance Criterion 1:

Given that the account is creditworthy

And the card is valid
And the dispenser contains cash,

When the customer requests the cash

Then ensure the account is debited

And ensure cash is dispensed
And ensure the card is returned.

Acceptance Criterion 2:

Given that the account is overdrawn

And the card is valid **When** the customer requests the cash

Then ensure the rejection message is displayed

And ensure cash is not dispensed
And ensure the card is returned.

Writing User Stories

Product Owner is responsible for the Product Backlog and thus for the User Stories. However, it does not mean that only product owner writes the user stories. Anyone in the Scrum Team can write the user stories, and the activity can be spread across the project as requirements get refined and new functionalities get added.

Non-Functional Requirements in User Stories

It is possible to incorporate the non-functional requirements also in the user stories. In the given ATM example, the ATM to be available to the user 24X7, 365 days is a non-functional requirement, which can be described by a use case.

6)Burn-Down Chart

The sprint tracking is usually done using Burn-Down Chart. Burn-Down Chart shows the remaining effort in day-wise number of hours.

For example, let us consider a 2-week sprint -

Sprint Duration: 2 Weeks

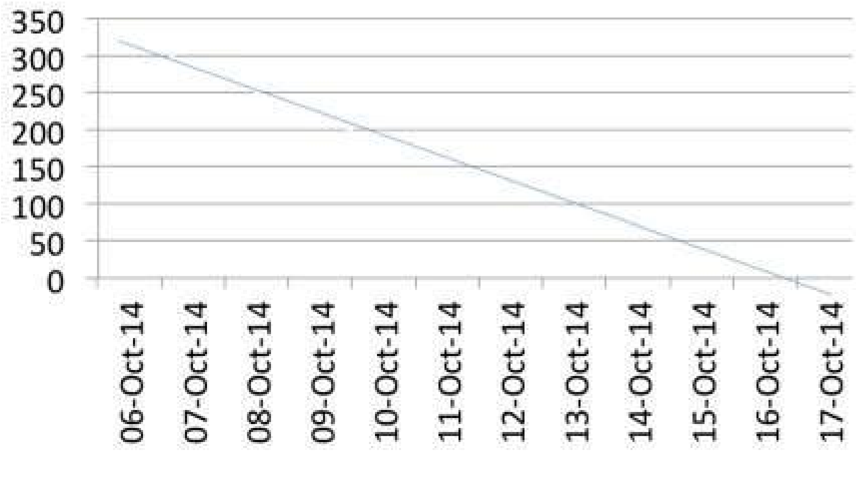
No. of Days per Week: 5

No. of Hrs. per Day: 6

No. of Resources: 6

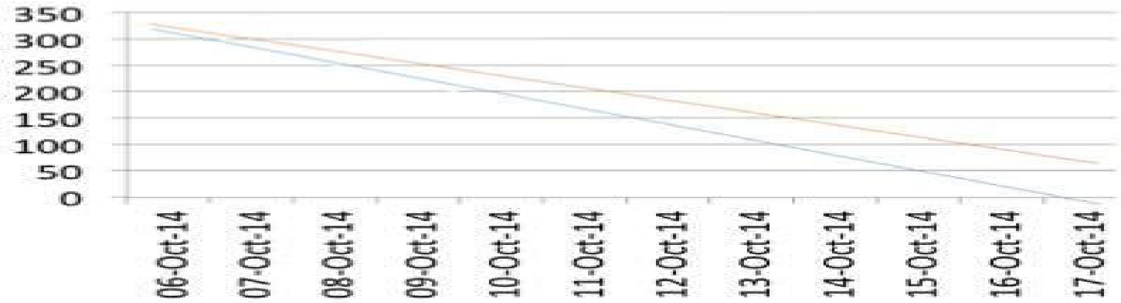
Hence, total remaining effort at the beginning of sprint is 2*5*6*6 = 360 hrs.

Therefore, in an ideal scenario, 36 hours of work gets reduced in the remaining work and the burn-down chart looks as follows -

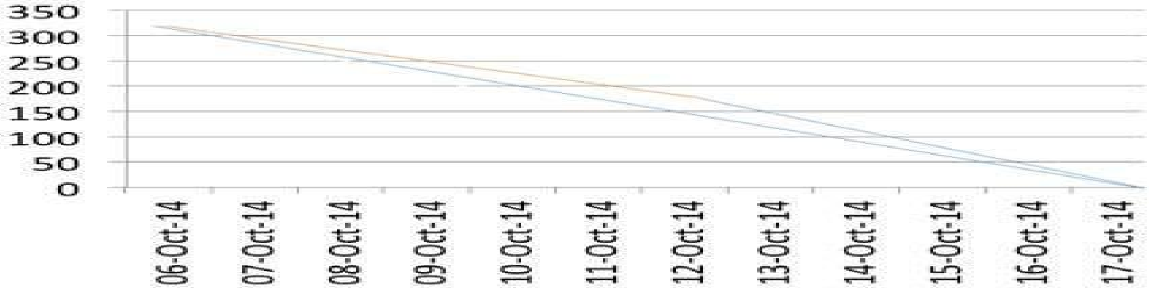


If the sprint work is done as planned daily, the scrum progress is almost aligned to the ideal bar.

If the sprint work gets delayed and time commitment is not met, the burn-down chart looks as follows -



But, as the burn-down chart is drawn daily, and the slippage is known early, corrective actions can be taken to meet the sprint time line. Suppose, the team stretches to meet the timeline, the burn-down chart looks as follows -



Thus, at any point in time in a Sprint, the total work remaining in the Sprint can be visualized and possibility of meeting sprint timeline can be improved.

Conclusion

Burn-down charts aid the Scrum team to keep track of their progress and what needs to be done to meet the sprint goal.

Agile Model

"**Agile process model**" refers to a software development approach based on iterative development.

Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements.

Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.

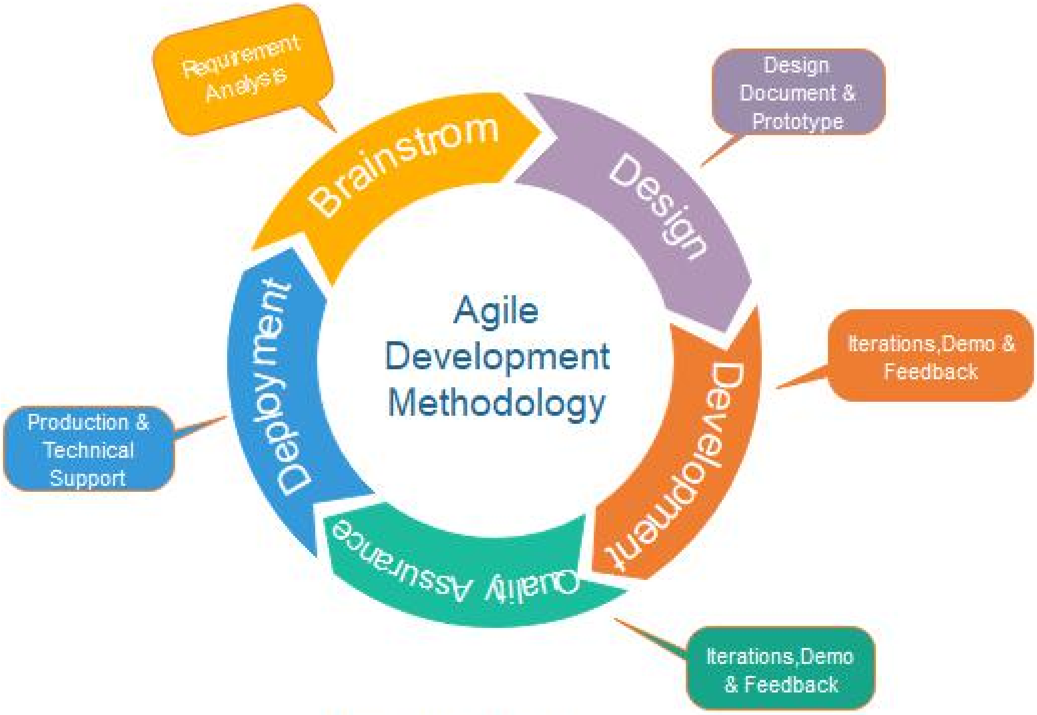

Requirement Analysis
Brainstrom
Design Document & Prototype
Design
Agile Development Methodology
Iterations,Demo & Feedback
Development
Production & Technical Support
Deployment
Quality Assurance
Iterations,Demo & Feedback

Fig. Agile Model

Phases of Agile Model:

Following are the phases in the Agile model are as follows:

1. Requirements gathering
2. Design the requirements
3. Construction/ iteration
4. Testing/ Quality assurance
5. Deployment
6. Feedback

1. Requirements gathering: In this phase, you must define the requirements. You should explain business opportunities and plan the time and effort needed to build the project. Based on this information, you can evaluate technical and economic feasibility.

2. Design the requirements: When you have identified the project, work with stakeholders to define requirements. You can use the user flow diagram or the high-level UML diagram to show the work of new features and show how it will apply to your existing system.

3. Construction/ iteration: When the team defines the requirements, the work begins. Designers and developers start working on their project, which aims to deploy a working product. The product will undergo various stages of improvement, so it includes simple, minimal functionality.

4. Testing: In this phase, the Quality Assurance team examines the product's performance and looks for the bug.

5. Deployment: In this phase, the team issues a product for the user's work environment.

6. Feedback: After releasing the product, the last step is feedback. In this, the team receives feedback about the product and works through the feedback.