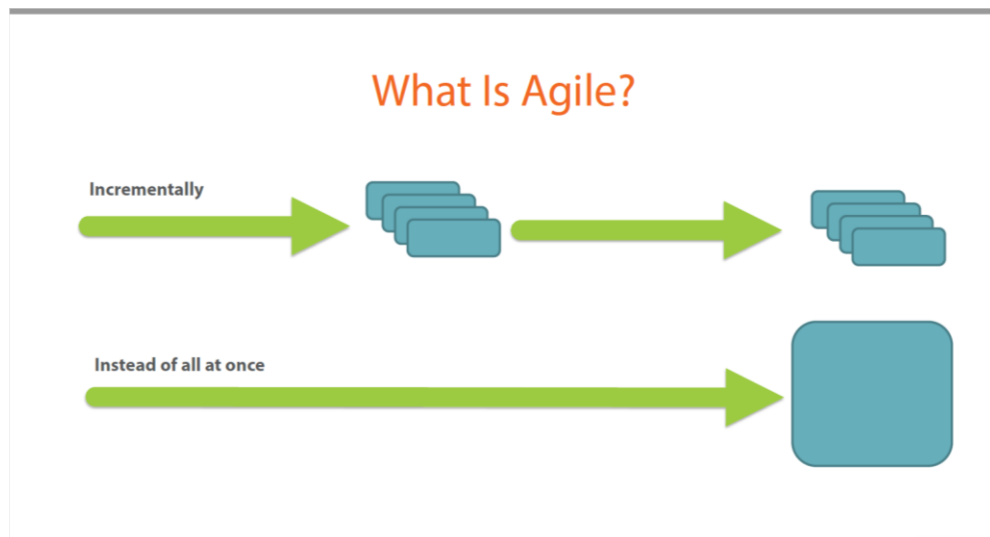
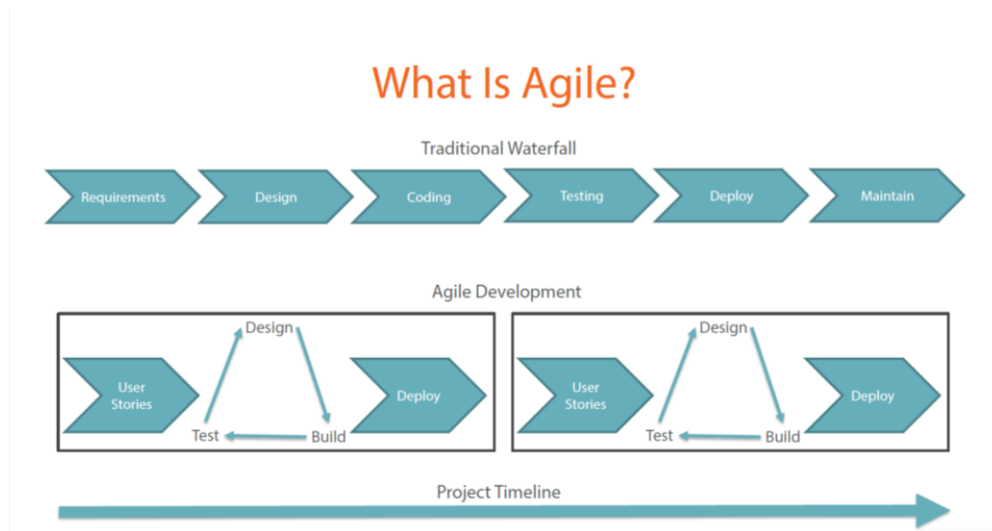


Agile

Agile is a software development methodology to build a software incrementally using short iterations of 1 to 4 weeks so that the development is aligned with the changing business needs.





Working software is the primary measure of progress.

<http://agilemanifesto.org/>



Agile Methodologies Overview



- Scrum
- Extreme programming (XP)

Agile Methodologies Overview



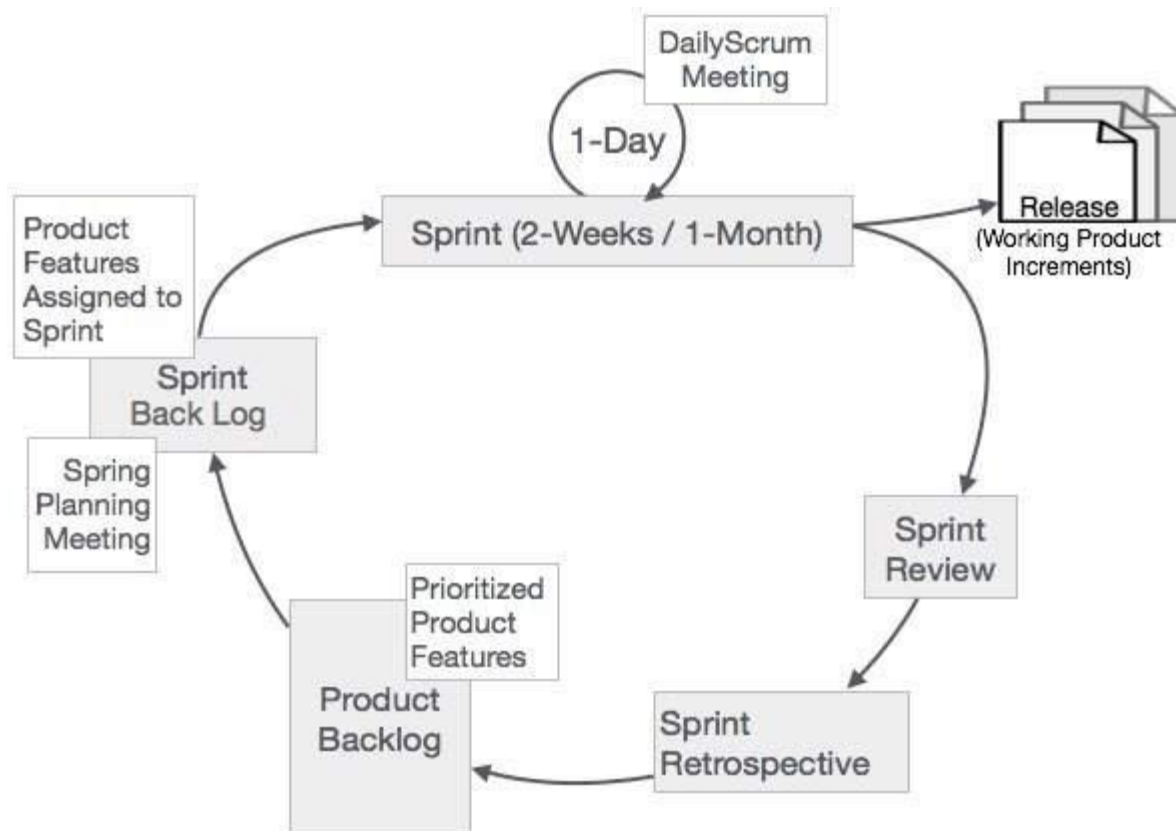
- Scrum
- Extreme programming (XP)
- Crystal
- Dynamic systems development method (DSDM)
- Feature driven design (FDD)

Key Principles of Agile

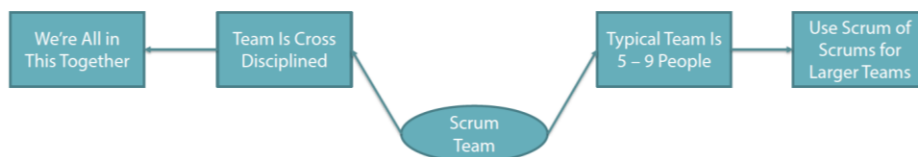
The Agile Manifesto is based on the following principles:

Principle	Description
Satisfaction and Delivery	Customer satisfaction through early and continuous working software.
Welcoming Change	Welcome changing requirements, even at later stages of development.
Deliver Frequently	Deliver working software frequently (weekly rather than monthly).
Communication is the Key	Ensure close association of developers with business people on daily basis.
Environment and Trust	Build projects around motivated individuals. Give them necessary support and trust them.
Face-to-face Communication	Encourage face-to-face conversation to ensure efficient and effective communication.
Software as Measure of Progress	Working software is the primary measure of progress.
Sustainable Development	Promote sustainable development with the ability to maintain a constant pace throughout the development.
Attention to Details	Continuous attention to technical excellence and good design.
The Power of Less	Simplicity is essential.
Self-organizing Teams	Regular attention of the team on becoming effective in changing circumstances.

Scrum Process Framework



Scrum Team



Scrum - Roles

ScrumMaster

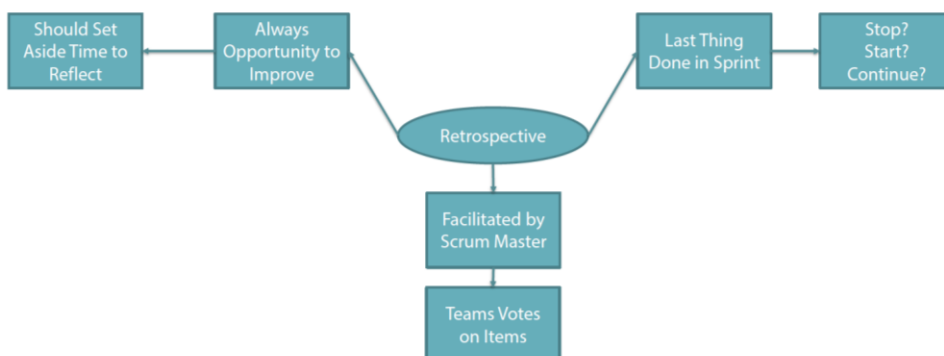
The ScrumMaster (sometimes written as the Scrum Master, although the official term has no space after "Scrum") is the keeper of the scrum process. He/she is responsible for-

- making the process run smoothly
- removing obstacles that impact productivity
- organizing and facilitating the critical meetings

Daily Scrum



Sprint Retrospective

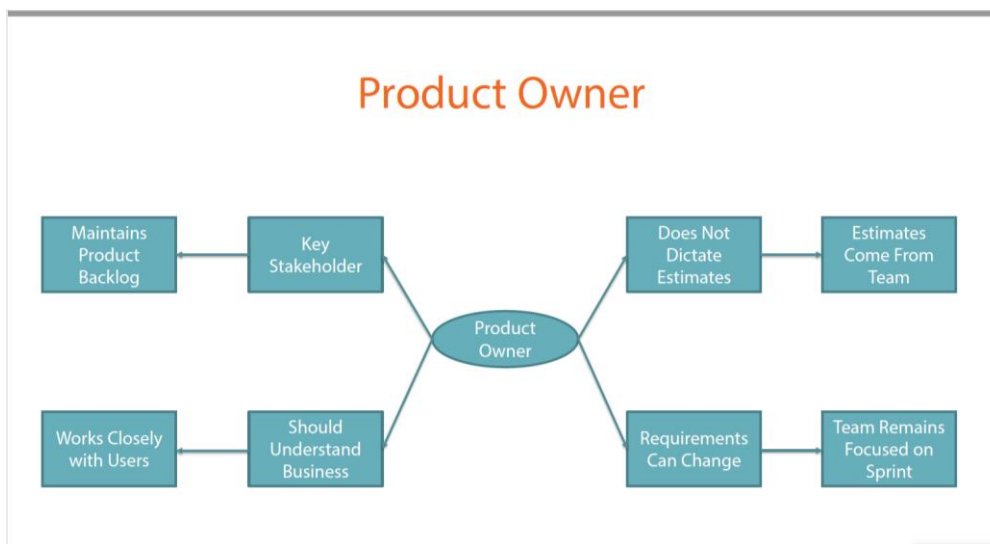


Product Owner

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.



Scrum - Events

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.

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

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?

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.

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.

Scrum - Artifacts

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.

Product Backlog refinement means adding detail, estimates, and priority order to the Product Backlog items. This is an ongoing process performed by the Product Owner and the Team. The Scrum Team decides how and when refinement is to be done.

Product Backlog items can be updated at any time by the Product Owner or at the Product Owner's discretion.

Higher-ordered Product Backlog items are usually clearer and more detailed than lower-ordered ones. More precise estimates are made based on the greater clarity and increased detail. The lower the order, the lesser is the detail.

Product Backlog items that may likely be the candidate requirements for the upcoming Sprint are refined so that these items can be developed during the Sprint. Product Backlog items that can be developed by the Team within one Sprint are deemed to be ready for selection in a Sprint planning meeting.

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. It must be in working condition regardless of whether the Product Owner decides to actually release it.

The Scrum Team needs to have consensus on what is considered to be an Increment. This varies significantly per Scrum Team, but, team members must have a shared understanding of what it means for work to be complete. This is used to assess when work is complete on the product Increment.

The same understanding guides the Team in knowing how many Product Backlog items it can select during a Sprint Planning. The purpose of each Sprint is to deliver Increments of potentially releasable functionality.

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.

User Stories

In software development, the product features play a crucial role. It is the features that the user ultimately likes to use in the final product. They are known as Requirements in the general terminology. The software development project success lies in understanding the user requirements accurately and appropriately, and then implementing them in the final product. Thus, requirements or product features need to be thoroughly known to the development project team.

In 1999, Kent Beck came up with a term User Stories for the product features. He described that a User Story is narrated from user perspective regarding what he or she wants to have rather than what system can do for him. Thus, the view changed from product to user completely and User Stories became de facto standard for Requirements in all Agile frameworks.

In Scrum projects, the Product Backlog is a list of user stories. These User Stories are prioritized and taken into the Sprint Backlog in the Sprint Planning Meeting.

Estimation is also based on user stories and the size of the product is estimated in User Story Points.

The User Story Structure

The User Story structure is as follows -

As a *<Type of User>*,

I want *<To Perform Some Task>*,

So that *<I can achieve some goal/benefit/value>*.

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.

Managing User Stories

User Stories are managed in the Product Backlog. The User Stories are ordered according to priority. The most prioritized user stories are refined to granular level, while the least priority user stories are kept at a lesser detail level. For every sprint, the most prioritized and hence more granulated user stories are taken into the sprint backlog. If a user story is to be added to the product backlog, its priority is first determined, and it is placed according to its place as per the priority. The user stories can be reprioritized at any time. It is also possible to remove any of the user stories if required.

Benefits with User Stories

- The major benefit of User Story lies in the user centric definition itself. This is because, ultimately, it is the user who will be using the product in the relevant user scenarios. It connects the end users to the team members.
- The syntax of the User Story itself ensures to capture the goal or benefit or value that the user wants to achieve.
- Since the acceptance criteria forms part of user story itself, it will be an added advantage to the Scrum Team.
- It is possible to make changes to a user story in course of the execution of the project. If the scope of the user story becomes large, it needs to be split into smaller user stories. The conditions in the acceptance criterion can also be changed.
- As working product increments are delivered to the users at the end of each sprint, the scrum team can get feedback from the users in sprint review meeting. This enables incorporation of feedback into the product continuously.

Scrum - Burn-Down Charts

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

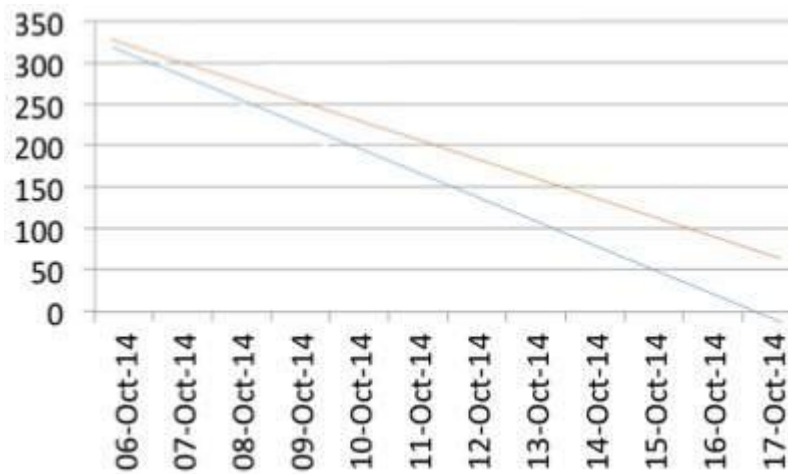
No. of Days per Week: 5

No. of Hrs. per Day: 6

No. of Resources: 6

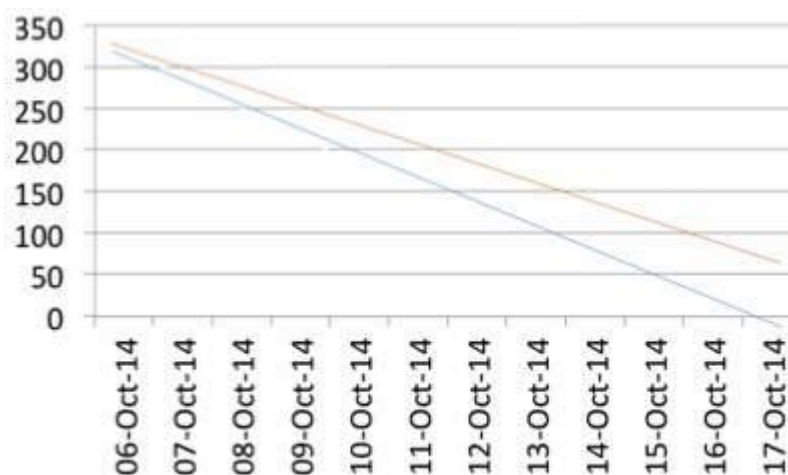
Hence, total remaining effort at the beginning of sprint is $2*5*6*6 = 360$ hrs.

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

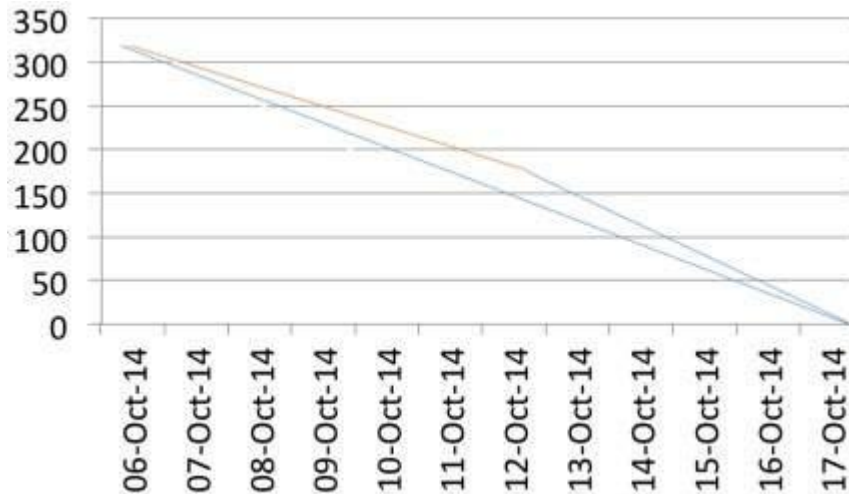


If the sprint work is done as planned daily, the scrum progress is almost aligned to the ideal bar.

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.

