



SCRUM

Agile framework for completing complex projects

tutorialspoint

S I M P L Y E A S Y L E A R N I N G

www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About the Tutorial

Scrum is an efficient framework within which you can develop software with teamwork. It is based on agile principles.

This tutorial will help you understand agile development in general and Scrum in specific. You will get familiar with its associated terminology along with appropriate examples.

Audience

This tutorial is prepared for the beginners to help them understand the basics of Scrum framework and its implementation.

After completing this tutorial, you will find yourself at a moderate level of expertise, from where you can take yourself to next levels.

Prerequisites

Before proceeding with this tutorial, you need a basic knowledge of software development concepts such as software requirements, coding, testing, etc.

Disclaimer & Copyright

© Copyright 2015 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher. We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com.

Contents

About the Tutorial	1
Audience	1
Prerequisites	1
Disclaimer & Copyright	1
Contents	ii
1. OVERVIEW	1
Waterfall Model	1
Iterative Incremental Model	1
Agile Development	2
Agile Manifesto	3
Definition of Agile Manifesto Items	3
Key Principles of Agile	4
Agile Methodologies	4
Dynamic System Development Methodology (DSDM)	4
Scrum	5
Extreme Programming (XP)	5
Test-driven Development (TDD)	5
Lean	5
Kanban	5
2. FRAMEWORK	7
Scrum Definition	7
Scrum Process Framework	8
Sprint	8
3. SCRUM ROLES	10
ScrumMaster	10
Product Owner	10
The Team	11
4. SCRUMMASTER	12

ScrumMaster Services to the Product Owner.....	12
ScrumMaster Services to the Scrum Team	12
ScrumMaster Services to the Organization	12
5. EVENTS	14
The Sprint	14
Sprint Planning	15
Daily Scrum Meetings	15
Sprint Review.....	16
Sprint Retrospective	17
6. ARTIFACTS.....	18
Product Backlog	18
Sprint Backlog	19
Increment	19
Sprint Burn-Down Chart.....	20
7. USER STORIES	21
User Stories.....	21
The User Story Structure	21
User Story: Customer’s Cash Withdrawal	21
User Story Acceptance Criteria	22
Writing User Stories	22
Non-Functional Requirements in User Stories.....	22
Managing User Stories	23
Benefits of User Stories.....	23
8. BURN-DOWN CHARTS.....	24
9. ESTIMATION.....	27
Estimation Techniques.....	27
Planning Poker Technique.....	27

Benefits of Planning Poker Estimation	28
10. SCRUM TOOLS	30
Available Scrum Tools	30
11. SCRUM – BENEFITS	32
Benefits to Customer	32
Benefits to Organization	32
Benefits to Product Managers	32
Benefits to Project Managers	32
Benefits to Development Team	33
12. SCRUM – CERTIFICATIONS	34
Certified ScrumMaster (CSM)	34
Certified Scrum Product Owner (CSPO)	34
Certified Scrum Practitioner (CSP)	34
Certified Scrum Coach (CSC)	34
Certified Scrum Trainer (CST)	35
13. SCRUM – FAQs	36

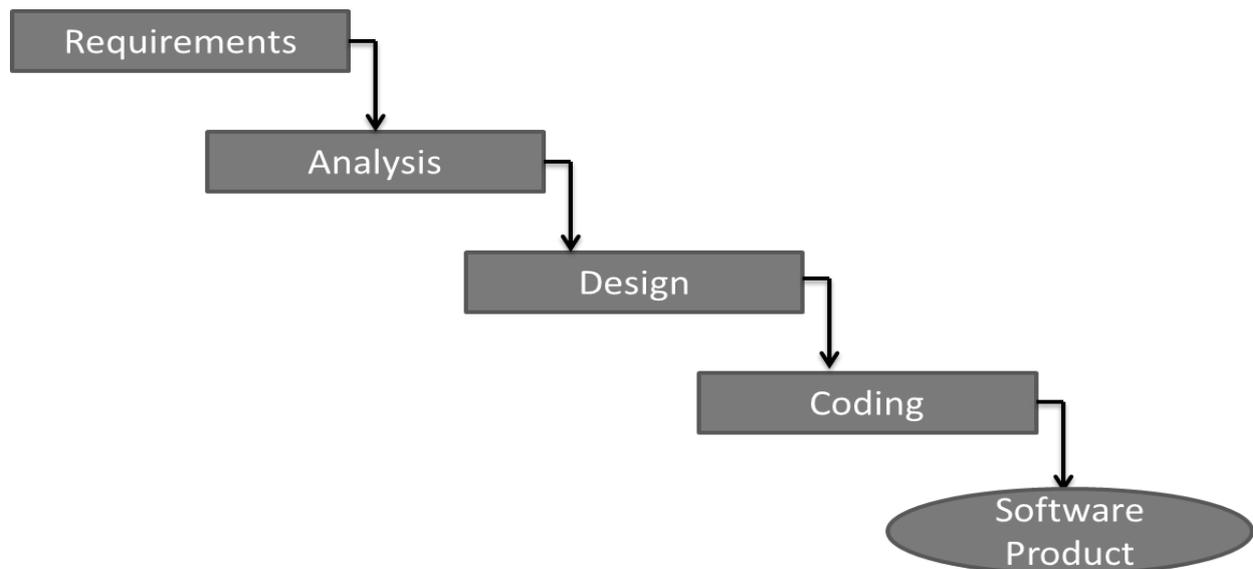
1. OVERVIEW

Agile has become one of the big buzzwords in the software development industry. But what exactly is agile development? Put simply, agile development is a different way of executing software development teams and projects.

To understand what is new, let us recap the traditional methods. In conventional software development, the product requirements are finalized before proceeding with the development.

Waterfall Model

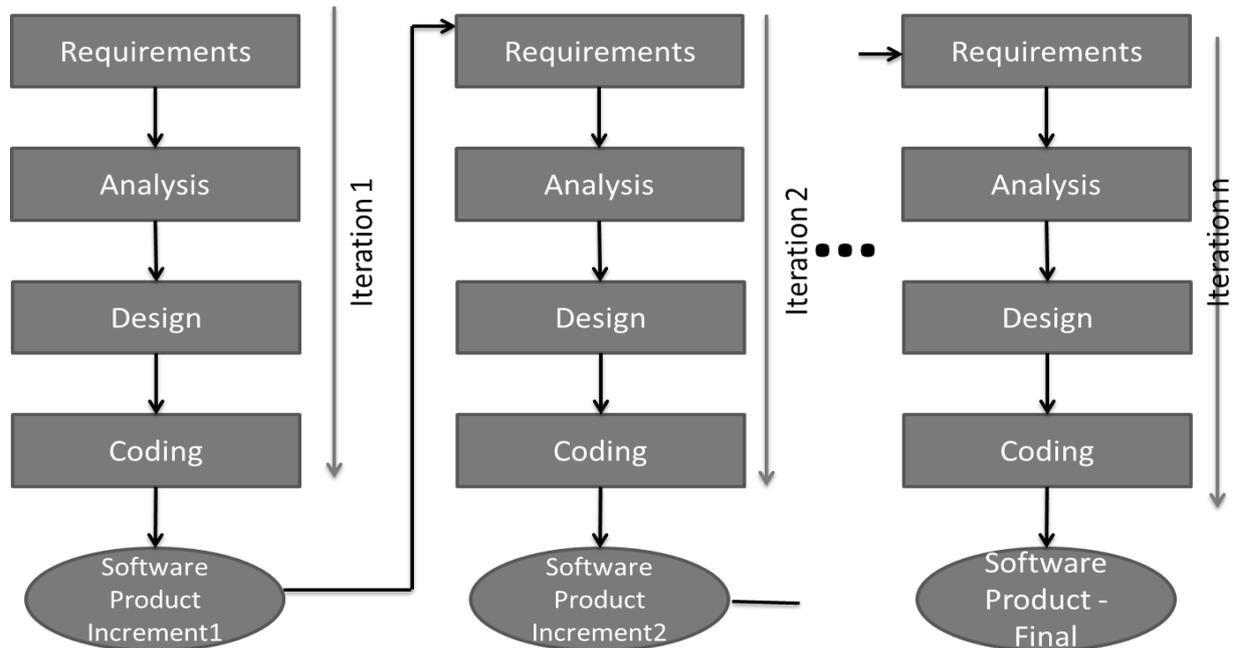
The most commonly used software development model with this characteristic is the Waterfall Model as depicted in the following diagram. However, in most of the cases, new functionalities get added, and also earlier requirements may change. The Waterfall model is not structured to accommodate such continuous changes in requirements. Further, the user will not have clarity on the functionality of the product till the product becomes available in its entirety.



Iterative Incremental Model

In the iterative incremental model, the development starts with a limited number of finalized and prioritized requirements. The deliverable is a working increment of the product. A set of activities ranging from requirements to code development is called an iteration. Based on the functionality of the increment and any or all of the new, modified, pending requirements, the next lot of requirements is given to the subsequent iteration. The outcome of the subsequent iteration is an enhanced

working increment of the product. This is repeated till the product accomplishes the required functionalities.



The user is usually not involved in the development work and it may cause communication gaps resulting in incorrect functionalities. The involvement is positive for the development team, but is demanding on the time of the team and can add delays. Further, any informal requirement changes during an iteration may lead to confusion and may also create scope creeps. With this premise, Agile development came into existence.

Agile Development

Agile development is based on iterative incremental development, in which requirements and solutions evolve through team collaboration. It recommends a time-boxed iterative approach, and encourages rapid and flexible response to change. It is a theoretical framework and does not specify any particular practice that a development team should follow. Scrum is a specific agile process framework that defines the practices required to be followed.

Early implementations of agile methods include *Rational Unified Process* (1994), *Scrum* (1995), *Crystal Clear*, *Extreme Programming* (1996), *Adaptive Software Development*, *Feature Driven Development* (1997), and *Dynamic Systems Development Method (DSDM)* (1995). These are now collectively referred to as **agile methodologies**, after the Agile Manifesto was published in 2001.

Agile Manifesto

The Agile Manifesto was published by a team of software developers in 2001, highlighting the importance that needs to be given to the development team, accommodating changing requirements, customer involvement.

The Agile Manifesto is as follows:

"We are uncovering better ways of developing software by doing it and helping others do it. Through this work, we have come to value:

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

That is, while there is value in the items on the right, we value the items on the left more."

...Manifesto for Agile Software Development, Authors: Beck, Kent, et al. (2001)

Definition of Agile Manifesto Items

The manifesto items on the left can be described as follows:

Manifesto Item	Description
Individuals and interactions	Importance needs to be given to: <ul style="list-style-type: none"> • self-organization and self-motivation of the team members • continuous interaction for work, clarifications, information among the team members
Working software	Delivery of working software at short duration intervals helps gain customer trust and assurance in the team.
Customer collaboration	Constant involvement of customer with the development team ensures communication of necessary modifications.
Responding to change	Focus on quick response to the proposed changes, which is made possible with short duration iterations.

The key element of Agile Manifesto is that we must trust people and their ability to collaborate. For this reason, the specific agile methodologies developed tap the abilities of team members by emphasizing teamwork and collaboration throughout the life-cycle of the project.

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.

Agile Methodologies

Agile methodologies include the following:

- Dynamic System Development Methodology
- Scrum
- Extreme Programming
- Test-driven Development
- Lean
- Kanban

Dynamic System Development Methodology (DSDM)

It is an agile framework for software projects. It was used to fine-tune the traditional approaches. The most recent version of DSDM is called DSDM Atern. The name Atern

is a short for Arctic Tern - a seabird that can travel vast distances that represents many features of the method which are natural ways of working such as prioritization and collaboration.

Scrum

It is the most popular agile framework, which concentrates particularly on how to manage tasks within a team-based development environment. Scrum uses iterative and incremental development model, with shorter duration of iterations. Scrum is relatively simple to implement and focuses on quick and frequent deliveries.

Extreme Programming (XP)

It is a type of agile software development. It advocates frequent releases in short development cycles, which is intended to improve productivity and introduce checkpoints where new customer requirements can be adopted. The methodology takes its name from the idea that the beneficial elements of traditional software engineering practices are taken to extreme levels. (Extreme Programming is a software-development discipline that organizes people to produce higher-quality software more productively.) XP addresses the analysis, development, and test phases with novel approaches that make a substantial difference to the quality of the end-product.

Test-driven Development (TDD)

It is a software development process that relies on the repetition of a very short development cycle: first the developer writes an automated test case that defines a desired improvement or a new function, then it produces the least amount of code to pass that test, and finally brings the new code to acceptable standards.

Lean

It is a production practice that considers the expenditure of resources for any goal other than the creation of value for the end-customer to be wasteful, and thus a target for elimination. Working from the perspective of the customer who consumes a product or service, the term value is defined as any action or process that a customer would be willing to pay for. Lean is centered on preserving value with less work.

Kanban

It is a system to improve and keep up a high level of production. Kanban is one method through which Just-In-Time (JIT), the strategy the organizations employ to control the inventory expenses, is achieved. Kanban became an effective tool in support of running a production system as a whole, and it proved to be an excellent way for promoting improvement.

Conclusion

Over the last 10 years, there is an ever-increasing volume of success stories, where companies have dramatically improved the success and performance of their IT development teams and projects with agile practices. This has caused agile to be widely adopted across a variety of industries, including media and technology, large corporates, and even government.

Agile Framework helps teams to benefit from:

- Faster Time to Deliver/ Market
- Reduce Uncertainty and Risk
- Increase Return on Investment (ROI) by focusing on Customer Value

Among these different agile methodologies, Scrum has proved to be extremely successful worldwide over the last 20 years.

2. FRAMEWORK

Scrum is a framework for developing and sustaining complex products. Ken Schwaber and Jeff Sutherland developed Scrum. Together, they stand behind the Scrum Rules.

Scrum Definition

Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.

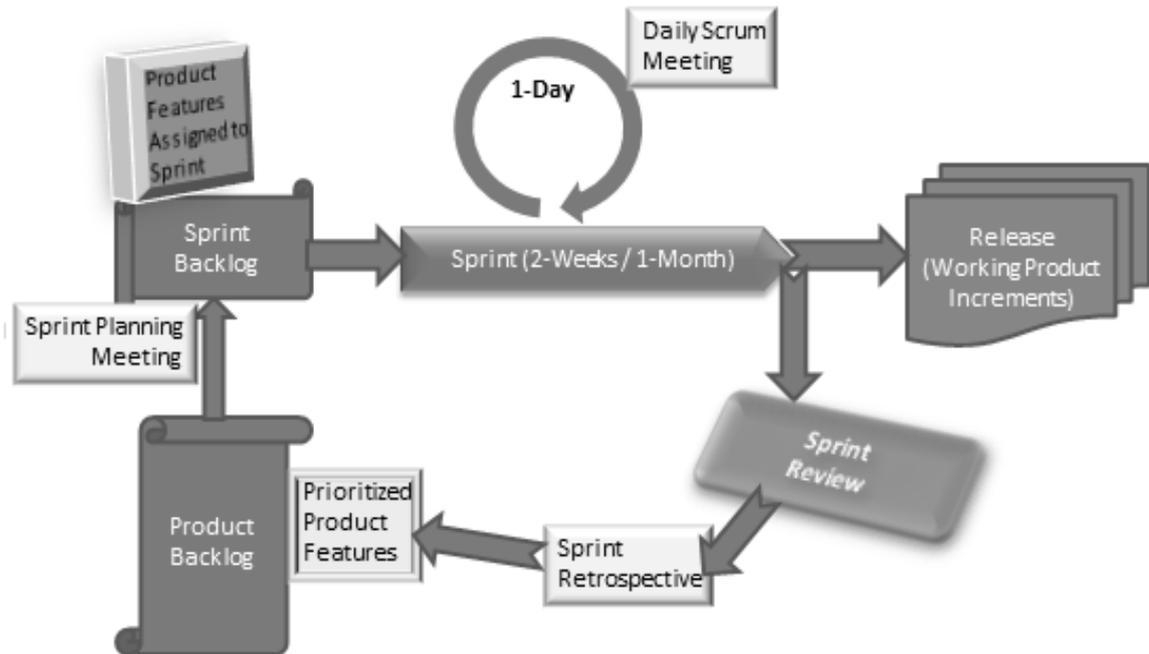
Scrum is a process framework that has been used to manage complex product development since the early 1990s. Scrum is not a process or a technique for building products; rather, it is a framework within which you can employ various processes and techniques. Scrum makes clear the relative efficacy of your product management and development practices so that you can improve.

The Scrum framework consists of Scrum Teams and their associated roles, events, artifacts, and rules. Each component within the framework serves a specific purpose and is essential to Scrum's success and usage.

The rules of Scrum bind together the events, roles, and artifacts, governing the relationships and interaction between them. The rules of Scrum are described throughout this tutorial.

Note: Across the industry, there are misconceptions that Scrum means no documentation, scrum team consists of only developers, and so on. It is not entirely so; we will give clarifications on these in later sections.

Scrum Process Framework



In Scrum, the prescribed events are used to create regularity. All events are time-boxed events, such that every event has a maximum duration. The events are described more elaborately in the subsequent chapters.

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.

Conclusion

Scrum 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. SCRUM ROLES

The Scrum Team consists of three roles, namely a ScrumMaster, a Product Owner, and the Team.

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

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.

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.

Optimal Team size is small enough to remain nimble and large enough to complete significant work within a Sprint. The Team size should be kept in the range from five to nine people, if possible. Fewer than five team members decrease interaction and results in smaller productivity gains. Having more than nine members requires too much coordination.

The scrum team works together closely, on a daily basis, to ensure the smooth flow of information and the quick resolution of issues. The scrum team delivers product iteratively and incrementally, maximizing opportunities for feedback. Incremental deliveries of a complete product ensure a potentially useful version of working product is always available.

4. SCRUMMASTER

ScrumMaster is a trained responsible person, who renders services as described below:

ScrumMaster Services to the Product Owner

The ScrumMaster serves the Product Owner in several ways, including:

- Finding techniques for effective Product Backlog management.
- Helping the Scrum Team understand the need for clear and concise Product Backlog items.
- Understanding product planning in an empirical environment.
- Ensuring that the Product Owner knows how to arrange the Product Backlog to maximize value.
- Understanding and practicing agility.
- Facilitating Scrum events as needed.

ScrumMaster Services to the Scrum Team

The ScrumMaster serves the Scrum Team in several ways, including:

- Coaching the Scrum Team in self-organization and cross-functionality.
- Helping the Scrum Team to create high-value products.
- Removing impediments to the Scrum Team's progress.
- Facilitating Scrum events as requested or needed.
- Coaching the Scrum Team in organizational environments in which Scrum is not yet fully adopted and understood.

ScrumMaster Services to the Organization

The ScrumMaster serves the organization in several ways, including:

- Leading and coaching the organization in its Scrum adoption.
- Planning Scrum implementations within the organization.

- Helping employees and stakeholders understand and enact Scrum and empirical product development.
- Causing change that increases the productivity of the Scrum Team.
- Working with other ScrumMasters to increase the effectiveness of the application of Scrum in the organization.

Conclusion

Scrum 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.

5. 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.

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.

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.

Reference

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

6. 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.

Conclusion

Scrum's roles, events, artifacts, and rules are inevitable. If only some parts of Scrum are implemented, the result is not Scrum. Scrum needs to be implemented in its entirety and functions well if aligned with other techniques, methodologies, and practices.

Reference

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

7. USER STORIES

As you have understood, the User Stories are commonly used to describe the product features and will form part of the Scrum Artifacts – **Product Backlog** and **Sprint Backlog**.

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 of 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.

Conclusion

Scrum's User Stories bring the users closer to the Scrum team and prevent last-minute surprises.

8. 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

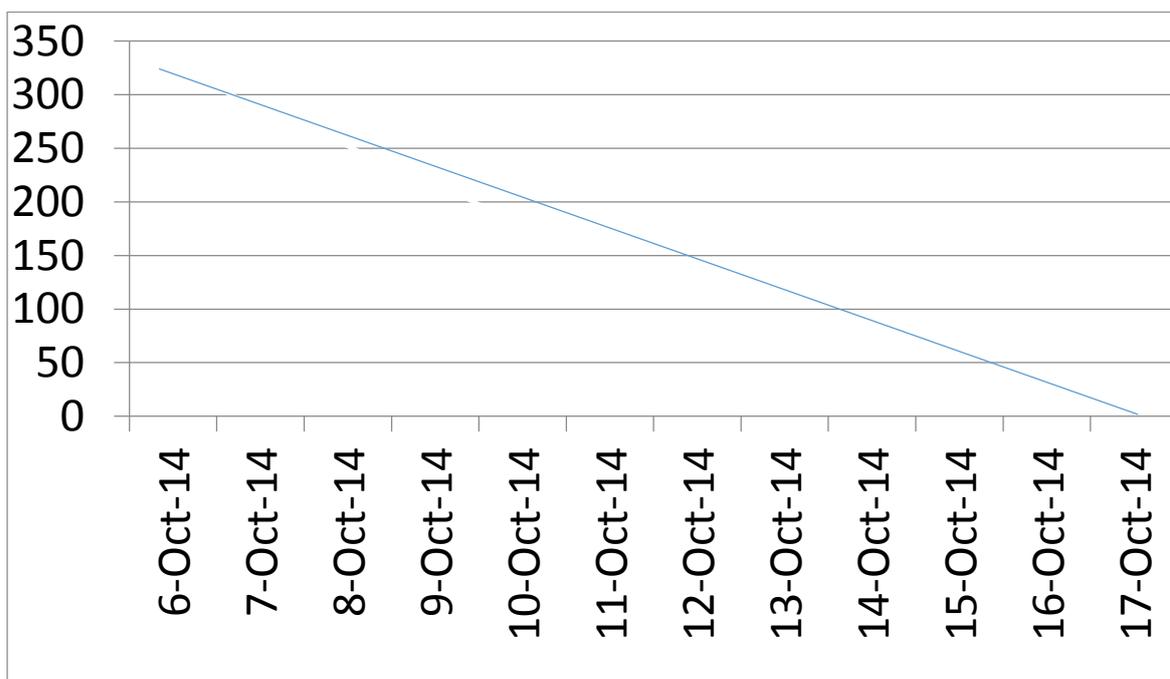
No. of Days per Week: 5

No. of Hrs. per Day: 6

No. of Resources: 6

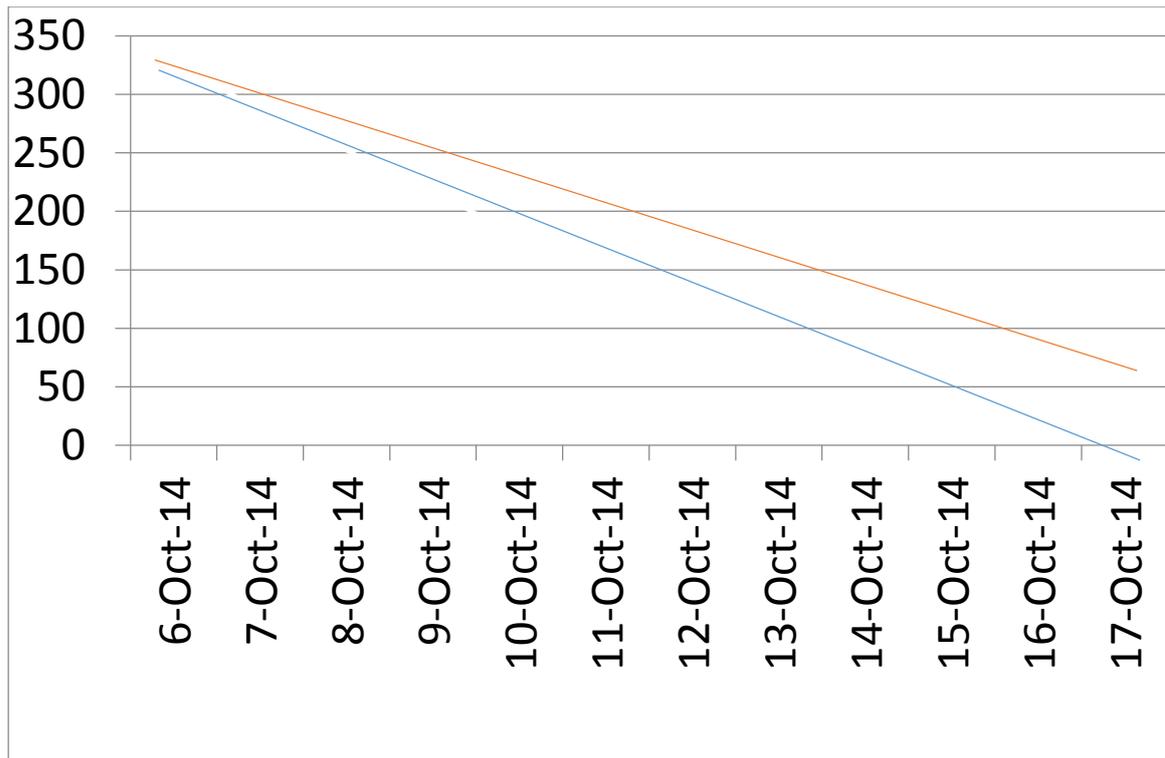
Hence, total remaining effort at the beginning of sprint is $2*5*6*6 = 360$ hrs.

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

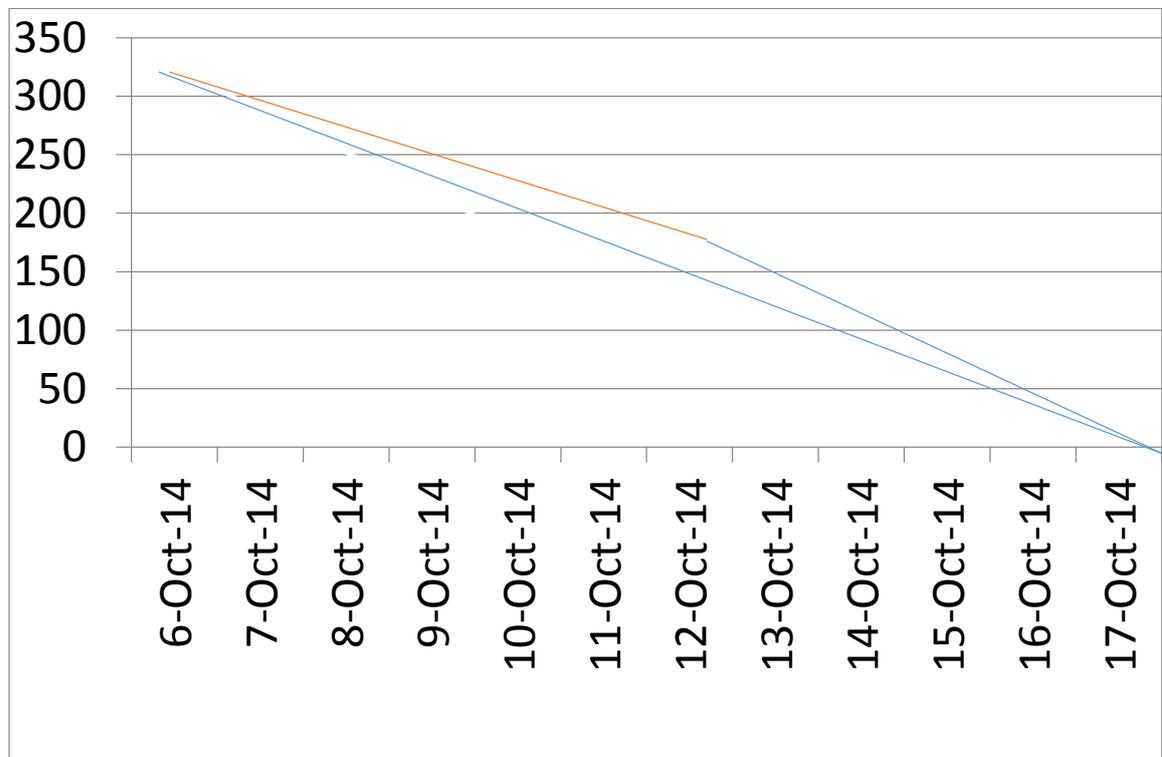


If the sprint work is done as planned daily, the scrum progress is almost aligned to the ideal bar.

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.

9. ESTIMATION

In Scrum Projects, Estimation is done by the entire team during Sprint Planning Meeting. The objective of the Estimation would be to consider the User Stories for the Sprint by Priority and by the Ability of the team to deliver during the Time Box of the Sprint.

Product Owner ensures that the prioritized User Stories are clear, can be subjected to estimation, and they are brought to the beginning of the Product Backlog.

As the Scrum Team in total is responsible for the delivery of the product increment, care would be taken to select the User Stories for the Sprint based on the size of the Product Increment and the effort required for the same.

The size of the Product Increment is estimated in terms of User Story Points. Once the size is determined, the effort is estimated by means of the past data, i.e., effort per User Story Point called Productivity.

Estimation Techniques

The Scrum Estimation of User Stories is in terms of the degree of difficulty for each of the User Stories. To assess the degree of difficulty, a particular scale is used.

There are several types of scales that are used in Scrum Estimation. Following are some examples:

- Numeric Sizing (1 through 10)
- T-shirt Sizes (XS, S, M, L, XL, XXL, XXXL)
- Fibonacci Sequence (1, 2, 3, 5, 8, 13, 21, 34, etc.)
- Dog Breeds (Chihuahua,.....,Great Dane)

The estimation technique is normally chosen in such a way that the entire scrum team is acquainted and comfortable with scale's values. The most commonly used and most popular technique is Planning Poker which is based on Fibonacci sequence.

Planning Poker Technique

In Planning Poker Estimation Technique, estimates for the User Stories are derived by playing planning poker. The entire Scrum Team is involved and it results in quick but reliable estimates.

Planning Poker is played with a deck of cards. As Fibonacci sequence is used, the cards have numbers - 1, 2, 3, 5, 8, 13, 21, 34, etc. These numbers represent the Story Points. Each estimator has a deck of cards. The numbers on the cards should

be large enough to be visible to all the team members, when one of the team members holds up a card.

One of the team members is selected as the Moderator. Moderator reads the description of the User Story for which estimation is being made. If the estimators have any questions, Product Owner answers them.

Each estimator privately selects a card representing his or her estimate. Cards are not shown until all the estimators have made a selection. At that time, all cards are simultaneously turned over and held up so that all team members can see each estimate.

In the first round, it is very likely that the estimations vary. The high and low estimators explain the reason for their estimates. Care should be taken that all the discussions are meant for understanding only and nothing is to be taken personally. The moderator has to ensure the same.

The team can discuss the story and their estimates for few more minutes.

The moderator can take notes on the discussion that will be helpful when the specific story is developed. After the discussion, each estimator re-estimates by again selecting a card. Cards are once again kept private until everyone has estimated, at which point they are turned over at the same time.

Repeat the process till the estimates converges to a single estimate that can be used for the story. The number of rounds of estimation may vary from one user story to another.

Benefits of Planning Poker Estimation

Planning poker combines three methods of estimation:

Expert Opinion: In an Expert Opinion based Estimation approach, an expert is asked how long something will take or how big it will be. The expert provides an estimate relying on his or her experience or intuition or gut feel.

Expert Opinion Estimation usually doesn't take much time and is more accurate compared to some of the analytical methods.

Analogy: Analogy Estimation uses comparison of User Stories. The User Story under Estimation is compared with similar User Stories implemented earlier. This results in accurate results as the estimation is based on proven data.

Disaggregation: Disaggregation Estimation is done by splitting a User Story into smaller, easier-to-estimate User Stories. The user stories to be included in a Sprint are normally in the range of two to five days to develop. Hence, the User Stories that possibly take longer duration need to be split into smaller Use Cases. This approach also ensures that there would be many stories that are comparable.

Conclusion

Planning Poker is an enjoyable, yet productive approach to estimating. As the session is open for discussions before the final estimate is arrived, it would be easy for the team to come to a consensus and also have a broad view of the implementation of the User Story at hand.

10. SCRUM TOOLS

Scrum Tools facilitate planning and tracking for Scrum projects. They provide a single place for managing the product backlog, sprint backlog, planning and tracking Sprints, displaying Burndown charts, conducting daily Scrum Meetings, and conducting Scrum Retrospectives.

There are many different types of Scrum Tools available. Some are free (open source), some are paid, and for some, you get a distilled version of the tool. However, to get all the features and scalability, you need to buy a full version.

Available Scrum Tools

Following is a list of some Scrum Tools available in market as of day. The Open Source Tools are marked with Asterisk.

Axosoft	Airgile	Agile Cockpit	Jira (GreenHopper)	Mingle
Scrumwise	Agilo For Scrum	Banana Scrum	Kunagi	OnTime Now
Version One	AgileWrap	Daily-Scrum	Intervals	Pango Scrum
Acunote	Agile Tracking Tool*	Digaboard*	iMeta Agility	Pivotal Tracker
Agile Agenda	Agile Task	EasyBacklog	Ice Scrum*	pmScrum
Agile Bench	Agile Soup	Explain PMT	Hansoft	Prj Planner
Agile Buddy	Agile Manager	Agile Express*	GravityDev	Project Cards
Agile Fant*	Agile Log	Fire Scrum*	Fulcrum*	Quantum Whisper
Quick Scrum	Retrospectiva*	Scrum'd	Scrum Factory*	Scrumpy
Rally Dev	Scrinch*	Scrum Dashboard*	Scrum Edge	Scrum Pad
Redmine Backlogs	Scrum 2 Go	Scrum Desk	Scrum Do	Tweet Scrum
Scrumrf	Scrum Time*	Scrumwise	Select Solution Factory	Tackle*
Urban Turtle	ScrumTool	Scrum Works	Timebox	Tangy Orange Scrum

Conclusion

Agile in general, Scrum in specific does not mean there is no documentation work. The Scrum Artifacts are defined, Scrum Planning and Tracking are well established.

Scrum Tools facilitate in capturing and tracking information regarding the Scrum Projects. The choice of the tool depends on the features required by the organization, in addition to the needs for any other tool.

11. SCRUM – BENEFITS

Scrum supports continuous collaboration among the customer, team members, and relevant stakeholders. Its time-boxed approach and continuous feedback from the product owner ensures working product with essential features all the times. Additionally, Scrum provides different benefits to the different roles in the project.

Benefits to Customer

The Sprints are of shorter duration and prioritized user stories are taken up at every sprint planning. It ensures that at every sprint delivery, the features as required by the customer immediately are included. Further, if a customer raises any change request, it will be absorbed in the current sprint, or included in the very next sprint. Thus, the development team quickly responds to the customer's requirements very fast.

Benefits to Organization

Organization can focus on the effort required for development of the prioritized user stories and thus reduce overhead and rework. Due to the specific benefits of scrum to customer, increased efficiency of the development team, customer satisfaction and hence customer retention and customer references will be possible. It increases the market potential of the organization.

Benefits to Product Managers

Product Manager plays the role of Product Owner in the project. The responsibility of the product owner is to ensure customer satisfaction. Since Scrum facilitates quick responses, work prioritization, absorbing changes, product manager can easily ensure that the work is aligned to customer needs, which in turn ensures customer satisfaction.

Benefits to Project Managers

Project Manager plays the role of Scrum Master in the project. The collaborative nature of Scrum facilitates easy and concrete planning and tracking. The use of Burndown Charts to understand the work left, and the Daily Scrum meetings give the Project Manager awareness about the state of the project at all times. This awareness is essential to monitoring the project, and for catching and addressing issues quickly.

Benefits to Development Team

Due to the time-boxed nature of sprints and working product increment delivery at the end of every sprint, the development team becomes enthusiastic to see that their work is used immediately. The built in team collaboration makes the team enjoy the work they do. As the user stories for every sprint are based on customer priorities, team also understands that their work is valued.

12. SCRUM – CERTIFICATIONS

Scrum certifications are offered by the Scrum Alliance. Following Certifications are being offered:

- Certified ScrumMaster (CSM)
- Certified Scrum Product Owner (CSPO)
- Certified Scrum Practitioner (CSP)
- Certified Scrum Coach (CSC)
- Certified Scrum Trainer (CST)

Certified ScrumMaster (CSM)

Certified Scrum Master is the basic certification to become a member of Scrum Alliance, play Scrum Master's Role, and be eligible for other certifications. The certification requires attendance of the CSM course. After that, the candidate gets an email specifying the details of the Scrum membership and the CSM online examination. After taking the examination, the candidate is given the Certified ScrumMaster (CSM) certification.

Certified Scrum Product Owner (CSPO)

Certified Scrum Product Owner is the basic certification to become a member of Scrum Alliance, play Product Owner's role, and be eligible for other certifications.

Certified Scrum Practitioner (CSP)

Certified Scrum Practitioner is the certification for experienced ScrumMasters and Product Owners. The candidate should be a ScrumMaster or a Product Owner for at least one year. The candidate must submit an application containing a detailed description of what he or she has done in the specified role.

It is possible for a candidate to acquire the CSP certification immediately after the CSM certification or CSPO certification, provided the candidate is actively practicing the ScrumMaster's role, or Product Owner's role for the required duration.

Certified Scrum Coach (CSC)

Certified Scrum Coach is the certification for those who focus on coaching. The certification requires that the candidate has coached Scrum Teams through their adoption and mastery of Scrum for at least 1500 hours in the past 5 years.

Certified Scrum Trainer (CST)

Certified Scrum Trainer is the certification for those who want to teach CSM or CSPO classes. Applicants must have either a CSM or CSPO, and should be a CSP for at least a year before applying.

13. SCRUM – FAQs

Following are some FAQs regarding Scrum:

Question: What is the difference between Scrum and Agile Development?

Answer: Agile Development is a software methodology, whereas Scrum is one of process frameworks that follows Agile.

Question: Are Sprints and Iterations the same?

Answer: Both Sprints of Scrum and Iterations of Iterative Incremental model deliver working product increments. However, these differ in that:

- Lifecycles of Sprint and Iteration are different.
- Sprints are time-boxed, while Iterations are not.
- Duration of Sprints is much less compared to durations of Iterations.

Question: Is Scrum Master a job title or a role that someone with an existing job title fills?

Answer: Scrum Master is a role that someone with a job title fills. Normal practice is that the person playing the role of project manager plays the ScrumMaster's role as well.

Question: Can Product Owner and ScrumMaster's roles be played by the same person?

Answer: No, since the ownership differs. Product Owner takes care of the Product Backlog, Prioritization of User Stories, and Validation of the working product increment with the user stories allocated to the Sprint.

Question: Is it that Scrum Projects need not have any Documentation?

Answer: No. Scrum Projects, like any other Projects require documentation such as user stories, design, test cases, etc.

Conclusion

Agile and Scrum are not the same. Scrum is one of the process frameworks adapting Agile. Scrum is advised to teams with experienced team members as the Framework requires great collaboration and self-organization as well. If the Scrum rules are not followed strictly, a project can lead to failure. Hence, it is necessary to have a proper understanding of Scrum concepts among the entire team. Since the Sprints are of short durations and are time-boxed, there is no time to learn the Scrum specifics on the job, even when a Scrum Master continuously monitors the project.