Sprint

Scrum organizes work in iterations or cycles of up to a calendar month called sprints.

key characteristics of sprints: They are time-boxed, have a short and consistent duration, have a goal that shouldn't be altered once started, and must reach the end state specified by the team's definition of done.

Sprints should be consistent in length, though exceptions are permitted under certain circumstances.

As a rule, no goal-altering changes in scope or personnel are permitted during a sprint.

Finally, during each sprint, a potentially shippable product increment is completed in conformance with the Scrum team's agreed-upon definition of done.

Time Boxed

Sprints are rooted in the concept of **time-boxing, a time-management technique that** helps organize the performance of work and manage scope.

Each sprint takes place in a time frame with specific start and end dates, called a time-box. Inside this time-box, the team is expected to work at a sustainable pace to complete a chosen set of work that aligns with a sprint goal.

Epics

An epic is a large user story that cannot be delivered as defined within a single iteration or is large enough that it can be split into smaller user stories.

There is no standard form to represent epics. Some teams use the familiar user story formats (As A, I want, So That or In Order To, As A, I want) while other teams represent the epics with a short phrase.

Product backlog items included the following epic-level user stories:

As a Certified Scrum Trainer I want to be able to post my public Scrum class on the Scrum Alliance website so that the community will know the details surrounding where and when I am offering the class.

As a prospective student I want to be able to see details of all publicly available Scrum classes so that I can find one that meets my criteria for attendance.

EPIC Example

Epic 01: Provide ordering and priority options to user to manage backlog easily

User Story 01: As a release manager, I want to have my releases mapped to different sprints. I also want to see the priority of each item on same board.

User Story 02: As a system admin, I should be able to provide rights who can prioritize product backlog

User Story 03: As a user I should be able to re-order my backlog using numbered items, star priority or simple drag n drop

Product Backlog

The product owner, with input from the rest of the Scrum team and stakeholders, is ultimately responsible for determining and managing the sequence of this work and communicating it in the form of a prioritized (or ordered) list known as the **product backlog**. On new-product development the product backlog items initially are features required to meet the product owner's vision. For ongoing product development, the product backlog might also contain new features, changes to existing features, defects needing repair, technical improvements, and so on.

The product owner collaborates with internal and external stakeholders to gather and define the product backlog items.

He then ensures that product backlog items Scrum Activities and Artifacts are placed in the correct sequence (using factors such as value, cost, knowledge, and risk) so that the high-value items appear at the top of the product backlog and the lower-value items appear toward the bottom.

Product Backlog

The product backlog is a constantly evolving artifact.
Items can be added, deleted, and revised by the product owner as business conditions change, or as the Scrum team's understanding of the product grows (through feedback on the software produced during each sprint).

Sprint Backlog

To acquire confidence in what it can get done, many development teams break down each targeted feature into a set of tasks. The collection of these tasks, along with their associated product backlog items, forms a second backlog called the **sprint backlog.**

Sprint backlog is a list of the tasks and requirements to be completed within the sprint.

The sprint backlog includes:

- The list of user stories within the sprint in order of priority.
- The relative effort estimate for each user story.
- The tasks necessary to develop each user story.
- The effort, in hours, to complete each task.
- A burn down chart that shows the status of the work the development team has completed.

Story Points

Story point is used for effort estimation for a user story, it’s a number without any unit, and sp value represses the comparative complexity from one story to another.
Classic example is reading a book or travelling from point A to C. Usually this could be value such as t-shirt size or number of cookies.

Velocity and Burn down Chart

Velocity is a measure in the Agile methodology. Simply put, velocity measures the rate at which the team gets work done in a single iteration. It deals with what actually was achieved as opposed to what was planned in that iteration.

A burn down chart is a graphical representation of work left to do versus time. The outstanding work is often on the vertical axis, with time along the horizontal.

It is useful for predicting when all of the work will be completed.

Burn up Chart

A burn-up chart tracks progress towards a project's completion. In the simplest form of burn up chart, there are two lines on the chart:

A total work line (the project scope line)

A work completed line

Sprint Activities

A sprint starts with sprint planning, encompasses the development work during the sprint (called sprint execution), and ends with the review and retrospective.

To acquire confidence that the development team has made a reasonable commitment, the team members create a second backlog during sprint planning, called the sprint backlog.

The sprint backlog describes, through a set of detailed **tasks, how** the team plans to design, build, integrate, and test the selected subset of features from the product backlog during that particular sprint.

Next is sprint execution, where the development team performs the tasks necessary to realize the selected features. Each day during sprint execution, the team members help manage the flow of work by conducting a synchronization, inspection, and adaptive planning activity known as the daily scrum. some, but not all, of the product owner’s vision.

Sprint Activities

At the end of sprint execution the team has produced a potentially shippable product increment that represents some, but not all, of the product owner's vision.

The Scrum team completes the sprint by performing two inspect-and-adapt activities. In the first, called the sprint review, the stakeholders and Scrum team inspect the product being built.

In the second, called the sprint retrospective, the Scrum team inspects the Scrum process being used to create the product. The outcome of these activities might be adaptations that will make their way into the product backlog or be included as part of the team's development process.

Sprint Planning

A product backlog may represent many weeks or months of work, which is much more than can be completed in a single, short sprint.

To determine the most important subset of product backlog items to build in the next sprint, the product owner, development team, and Scrum Master perform **sprint planning**

During sprint planning, the product owner and development team agree on a **sprint goal that defines what the upcoming sprint is supposed to achieve.**

Using this Start date End date Fixed length Time box of up to a calendar month Sprint 1 Sprint 2 Sprint 3 Sprint 4

Sprint planning is the first part of every sprint goal, the development team reviews the product backlog and determines the high priority items that the team can realistically accomplish in the upcoming sprint while working at a **sustainable pace—a pace at which the development team can comfortably**

work for an extended period of time.

Sprint Planning

To acquire confidence in what it can get done, many development teams break down each targeted feature into a set of tasks. The collection of these tasks, along with their associated product backlog items, forms a second backlog called the **sprint backlog.**

The development team then provides an estimate (typically in hours) of the effort required to complete each task. Breaking product backlog items into tasks is a form of design and **just-in-time planning for how to get the features done.**

Sprint Planning

Most Scrum teams performing sprints of two weeks to a month in duration try to complete sprint planning in about four to eight hours.

A one-week sprint should take no more than a couple of hours to plan (and probably less).

The approach I use most often follows a simple cycle: Select a product backlog item (whenever possible, the next-most-important item as defined by the product owner), break the item down into tasks, and determine if the selected item will reasonably fit within the sprint (in combination with other items targeted for the same sprint). If it does fit and there is more capacity to complete work, repeat the cycle until the team is out of capacity to do any more work.

Sprint Execution

Once the Scrum team finishes sprint planning and agrees on the content of the next sprint, the development team, guided by the Scrum Master's coaching, performs all of the task-level work necessary to get the features done, where "done" means there is a high degree of confidence that all of the work necessary for producing good-quality features has been completed.

Exactly what tasks the team performs depends of course on the nature of the work (for example, are we building software and what type of software, or are we building hardware, or is this marketing work?).

Nobody tells the development team in what order or how to do the task-level work in the sprint backlog. Instead, team members define their own task-level work and then self-organize in any manner they feel is best for achieving the sprint goal.

Daily Scrum

Each day of the sprint, ideally at the same time, the development team members hold a time boxed (15 minutes or less) **daily scrum. This inspect-and adapt** activity is sometimes referred to as the **daily stand-up because of the common** practice of everyone standing up during the meeting to help promote brevity.

A common approach to performing the daily scrum has the Scrum Master facilitating and each team member taking turns answering three questions for the benefit of the other team members:

What did I accomplish since the last daily scrum?

What do I plan to work on by the next daily scrum?

What are the obstacles or impediments that are preventing me from making progress?

Daily Scrum

By answering these questions, everyone understands the big picture of what is occurring, how they are progressing toward the sprint goal, any modifications they want to make to their plans for the upcoming day's work, and what issues need to be addressed. The daily scrum is essential for helping the development team manage the fast, flexible flow of work within a sprint.

The daily scrum is not a problem-solving activity. Rather, many teams decide to talk about problems after the daily scrum and do so with a small group of interested people.

The daily scrum also is not a traditional status meeting, especially the kind historically called by project managers so that they can get an update on the project's status.

Daily Scrum

A daily scrum, however, can be useful to communicate the status of sprint backlog items among the development team members. Mainly, the daily scrum is an inspection, synchronization, and adaptive daily planning activity that helps a self organizing team do its job better.

Definition of Done

In Scrum, we refer to the sprint results as a **potentially shippable product increment**, meaning that whatever the Scrum team agreed to do is really done according to its agreed-upon definition of done.

This definition specifies the degree of confidence that the work completed is of good quality and is potentially shippable.

For example, when developing software, a bare-minimum definition of done should yield a complete slice of product functionality that is designed, built, integrated, tested, and documented.

An aggressive definition of done enables the business to decide each sprint if it wants to ship (or deploy or release) what got built to internal or external customers.

To be clear, “potentially shippable” does not mean that what got built must actually be shipped. Shipping is a business decision, which is frequently influenced by things such as

Definition of Done

“Do we have enough features or enough of a customer workflow to justify a customer deployment?” or “Can our customers absorb another change given that we just gave them a release two weeks ago?”

Potentially shippable is better thought of as a state of confidence that what got built in the sprint is actually done, meaning that there isn’t materially important undone work (such as important testing or integration and so on) that needs to be completed before we can ship the results from the sprint, if shipping is our business desire.

Sprint Review

The goal of this activity is to inspect and adapt the product that is being built.
Critical to this activity is the conversation that takes place among its participants, which include the Scrum team, stakeholders, sponsors, customers, and interested members of other teams.
The conversation is focused on reviewing the just-completed features in the context of the overall development effort. Everyone in attendance gets clear visibility into what is occurring and has an opportunity to help guide the forthcoming development to ensure that the most business-appropriate solution is created.

Sprint Review

A successful review results in bidirectional information flow. The people who aren't on the Scrum team get to sync up on the development effort and help guide its direction.

At the same time, the Scrum team members gain a deeper appreciation for the business and marketing side of their product by getting frequent feedback on the convergence of the product toward delighted customers or users.

The sprint review therefore represents a scheduled opportunity to inspect and adapt the product.

Sprint Retrospective

The second inspect-and-adapt activity at the end of the sprint is the **sprint retrospective**. This activity frequently occurs after the sprint review and before the next sprint planning.
Sprint retrospective is the last activity in a sprint.

Whereas the sprint review is a time to inspect and adapt the product, the sprint retrospective is an opportunity to inspect and adapt the process.
During the sprint retrospective the development team, Scrum Master, and product owner come together to discuss what is and is not working with Scrum and associated technical practices.

Sprint Retrospective

The focus is on the continuous process improvement necessary to help a good Scrum team become great. At the end of a sprint retrospective the Scrum team should have identified and committed to a practical number of process improvement actions that will be undertaken by the Scrum team in the next sprint.

After the sprint retrospective is completed, the whole cycle is repeated again— starting with the next sprint-planning session, held to determine the current highest value set of work for the team to focus on.