

Chapter 9

The Sprint

All of the development, planning, retrospectives, and reviews that happen in Scrum happen within the scope of a Sprint. It's the most important part of Scrum.

Sprints are the heartbeat of Scrum, where ideas are turned into value.

— 2020 Scrum Guide page 7

Fixed Length Sprints

A Sprint has a fixed length of one month or less.

Not to sound pedantic, but 'fixed' means the length of a Sprint can't be changed once it's been started.

Sprints are fixed-length events of one month or less to create consistency.

— 2020 Scrum Guide page 7

Sometimes the Scrum Certification exam will ask you when it's possible to change the length of a Sprint.

The temptation is to say 'during Sprint planning,' but Sprint planning happens during the Sprint, so if you were to change the duration of the Sprint, it would be the length of the next Sprint you'd be changing, not the current one.

So the discussion about the length of the Sprint and whether the Sprint should be shortened or extended are best to take place during the Sprint Retrospective. That way if you do decide to change the Sprint length, the change can apply to the Sprint that starts after the Retrospective finishes.

Having said that, the length of the Sprint shouldn't be changed too often. It should remain relatively consistent throughout the project.

One Month or Less

The other key point here is the fact that a Sprint is a month or less.

- It's not 28 days
- It's not 4 weeks

The length of a Sprint is one month or less.

You'll be tested on that, so commit it to memory.

The Start of the Sprint

A new Sprint starts immediately after the conclusion of the previous Sprint.

We've already discussed how the next Sprint starts immediately after the current Sprint ends.

The Scrum Guide re-states this point again here, which is a good clue for you to pay attention to it. You'll be tested on this fact several times on the exam.

Test Yourself

True or False: Sprint Zero can last for up to 49 days.

This is false for two reasons.

First, there is no such thing as 'Sprint Zero' as far as the Scrum Guide goes.

- There are no Release Sprints in Scrum
- There are no UAT Sprints in Scrum
- There is no QA Sprints in Scrum
- There are no Infrastructure Sprints in Scrum
- There is no Sprint Zero in Scrum

Whenever you see the term 'Sprint Zero' on the Scrum Master Certification exam, it's usually a red herring.

More to the point, this statement is false because the maximum length of a Sprint in Scrum is one month. That's the max. Often it is shorter.

Test Yourself

True or false: A Sprint can be any length a team decides is best, so long as it is less than a month.

This is true.

The Scrum Guide does not specify the minimum length of a Sprint, and when the Scrum Guide doesn't explicitly define something, then it's up to Scrum Teams to figure things out for themselves.

I could not imagine fitting a Sprint Planning meeting, Sprint Retrospective, and Sprint Review into a 24-hour Sprint, but nothing in the Scrum Guide forbids it.

What Happens During the Sprint?

It's always assumed that things will change unpredictably from day to day during a Sprint. This is

why Scrum advocates constantly inspecting and adapting.

All the work necessary to achieve the Product Goal, including Sprint Planning, Daily Scrums, Sprint Review, and Sprint Retrospective, happen within Sprints.

During the Sprint:

- No changes are made that would endanger the Sprint Goal;
- Quality does not decrease;
- The Product Backlog is refined as needed; and,
- Scope may be clarified and renegotiated with the Product Owner as more is learned.

— 2020 Scrum Guide page 7

However, some things that should not change during the Sprint

1. No changes are made that put the Sprint Goal at risk
2. No changes are made to decrease the quality of the existing product

The Scrum Guide doesn't have many hard and fast rules, but those are two of them.

Test Yourself

During development, the Scrum Team has found a Product Backlog Item is much more complex than initially anticipated.

The developers believe this PBI should be broken into multiple Product Backlog Items that should be scheduled across multiple Sprints. What should you as the Scrum Master do?

- ☐ Have the Scrum Developers clarify the requirements and renegotiate the scope of the Product Backlog Items with the PO
- ☐ Expand the length of the Sprint so the Development team can complete the complex Product Backlog Items
- ☐ Break the Product Backlog Item into multiple PBIs and add them to the Product Backlog
- ☐ Rewrite the User Story so the work related to the Product Backlog Item can be completed within the current Sprint

A is correct.

If the developers have issues with the size, scope, complexity, and clarity of the work they are doing, they simply contact the Product Owner so requirements can be clarified and scope can be renegotiated.

Sprints are a fixed length, so they can't ever be extended, which makes B incorrect.

The Scrum Master can never add items to the Sprint Backlog or the Product Backlog, so C is incorrect.

And the Scrum Guide makes zero references to User Stories, so any talk of User Stories on the

Why do we have Short Sprints in Scrum?

Some people often wonder why Scrum emphasizes short sprints.

The answer is simple. The longer the Sprint, the less predictable things become.

It's hard enough to predict four days into the future, let alone four weeks.

Short Sprints make things more predictable, and they also allow more regular interactions with stakeholders as Sprint Reviews, which occur at the end of the Sprint, take place more often.

Sprints enable predictability by ensuring inspection and adaptation of progress toward a Product Goal at least every calendar month.

When a Sprint's horizon is too long the Sprint Goal may become invalid, complexity may rise, and risk may increase.

Shorter Sprints can be employed to generate more learning cycles and limit the risk of cost and effort to a smaller time frame.

Each Sprint may be considered a short project.

— 2020 Scrum Guide page 7

The Scrum Guide places a great deal of emphasis on a short Sprint, and there are plenty of reasons for that.

Remember, a Sprint Review happens at the end of a Sprint. With a 2 week Sprint, the stakeholders get to review the product twice. With a 4 week Sprint, they only get to see it once.

A short Sprint makes it easier to manage risk. If the team is doing something wrong, they'll find out sooner from the stakeholders in a 2-week Sprint than in a month long one.

It's arguably easier to estimate work for two weeks, as opposed to three or four, and the project is less likely to go completely sideways in two weeks as opposed to two months.

Test Yourself

The development team is working with a new technology, which has introduced a great deal of risk and uncertainty into the project.

The Product Owner wants to reduce the Sprint time down to 2 or 3 weeks from the current Sprint length of one month.

What should the Scrum Master do?

- ☐ Act on the Product Owner's recommendation and reduce the Sprint length to 2 weeks
- ☐ Coach the team on how shorter Sprints reduce risk and ask them to consider reducing the

Sprint length to 2 or 3 weeks.

- ☐ Explain to the Product Owner that the Sprint length cannot be changed after development has started.
- ☐ Plan a week-long 'learning Sprint' where developers learn the new technology and don't try to deliver an Increment.

The correct answer is B. As a Scrum Master, your job is to teach people about Scrum practices and Scrum Theory and coach them on how to apply Scrum in a manner that works best for them.

Neither the Scrum Master nor the Product Owner can arbitrarily set the length of a Sprint in Scrum. This is a decision that must come about through the consensus of the team.

C is incorrect because the length of the Sprint can be changed if the team agrees a change is needed. However, the Sprint length cannot be changed mid-sprint.

D is incorrect because there is no 'learning Sprints' in Scrum. Furthermore, it is a requirement that every Sprint works toward the creation of an Increment. So Sprints are planned where the result is not a usable increment.

Empiricism Trumps Scrum Metrics

Various practices exist to forecast progress, like burn-downs, burn-ups, or cumulative flows.

While proving useful, these do not replace the importance of empiricism.

In complex environments, what will happen is unknown. Only what has already happened may be used for forward-looking decision-making.

— 2020 Scrum Guide page 7

The Scrum Guide is light on metrics.

The only Scrum metrics referenced in the guide are mentioned here:

- Burn-down charts
- Burn-up charts
- Cumulative flows

What's more, the guide kinda talks them down, saying that while they are useful, they're not as good as empirical analysis, which is a bit of a diss.

Still, for the exam, you do need to know what each of these charts does. You don't need to understand them in depth, but questions will arise about what they can tell you.

Burn-Down Chart: A burn-down chart is a graphical representation of the amount of work remaining in a project versus time. It tracks the progress of a project by showing the remaining work that needs to be completed on the vertical axis and the time on the horizontal axis. The chart starts with the total amount of work to be completed at the beginning of the project, and as work is

completed, the line on the chart gradually moves down until it reaches zero at the end of the project.

Burn-down charts are useful because they help teams track progress over time and adjust their approach as necessary. For example, if the team is falling behind schedule, they can identify this early on and make changes to get back on track.

Burn-Up Chart: A burn-up chart is similar to a burn-down chart but shows progress differently. Rather than showing the remaining work on the vertical axis, a burn-up chart shows the amount of work completed over time. The chart starts with zero completed work at the beginning of the project, and as work is completed, the line on the chart moves up until it reaches the total amount of work to be completed at the end of the project.

Burn-up charts are useful because they show progress towards a goal and can help teams identify whether they are on track to complete the project on time.

Cumulative Flow Chart: A cumulative flow chart is a graphical representation of the flow of work in a project. It shows the amount of work in progress at any given time, as well as the rate at which work is being completed. The chart has a horizontal axis representing time and a vertical axis representing the number of tasks.

The chart starts with a backlog of tasks at the beginning of the project and shows how tasks move through different stages of completion over time. For example, a task might start in the "to do" column, move to "in progress", and then finally move to "completed".

Cumulative flow charts are useful because they help teams identify bottlenecks in their workflow and optimize their processes to increase efficiency. By tracking the rate of completion over time, teams can also identify whether they are on track to complete the project on time.

You Can't Predict the Future with Scrum

Managers and team leaders always want metrics that will help predict when a given project will be finished, or when a product will hit a given milestone. Scrum is quick to downplay the various charting techniques used to predict the future.

The Scrum Guide is quick to point out that there are limits to the accuracy of charts and metrics, and there is no replacement for empiricism.

While proving useful, these do not replace the importance of empiricism.

In complex environments, what will happen is unknown. Only what has already happened may be used for forward-looking decision-making.

— 2020 Scrum Guide page 7

The Scrum Guide is quick to point out that there are limits to the accuracy of charts and metrics, and there is no replacement for empiricism.

Empiricism insists that experience and evidence should form the basis for belief and knowledge; more so than a chart does.

Importance of Empiricism

For example, a chart might say a team is completing 20 backlog items a week. Does that mean you complete 20 backlog items next week?

You might empirically know that two developers will be on vacation, or perhaps one developer is coming down with the flu.

The trend on the chart may indicate that the team will complete 20 backlog items next week, but you empirically know that sickness or vacation times will likely cut that number in half.

That's an example of empirical knowledge being a much better, and a much more informed indicator of future performance than a burn-down chart.

Who Can Cancel a Sprint?

The Scrum Certification guide will ask you who has the power to cancel a Sprint, and under what conditions the Sprint can be canceled.

A Sprint could be canceled if the Sprint Goal becomes obsolete. Only the Product Owner has the authority to cancel the Sprint.

— 2020 Scrum Guide page 7

These two points surprise people for the following three reasons.

1. It is indeed possible for an active Sprint to be canceled
2. The only reason to cancel a Sprint is if the Sprint Goal becomes obsolete
3. Only the Product Owner can cancel a Sprint, not the Scrum Master or any stakeholder

Just by the authoritative nature of the name, people think a Scrum Master would have the authority to cancel a Sprint.

Scrum Masters don't.

A Scrum Master can't cancel a Sprint. Only the Product Owner can.

And there is only one reason why a Sprint can be canceled: the Sprint Goal has become obsolete.

That means none of the following reasons are justifications to cancel the Sprint:

- The lead developer quit
- The building flooded and there's no power
- The Scrum Master has gone on vacation
- The CEO wants a different feature developed

There are plenty of reasons why someone in the organization might want to cancel the Sprint, but the only reason a Sprint can be canceled is if the Sprint Goal has become obsolete, and the only person allowed to make that call is the Product Owner.

What isn't said about the Sprint

People want the Scrum Guide to answer all of their questions about how to run a big project. Scrum doesn't do that.

Scrum provides a set of basic rules written out in a guide that's less than 4000 words. I wrote college papers longer than that.

If Scrum doesn't provide a rule or guideline around something, then it's up to the group of self-managing, highly motivated professionals on that team to figure it out on their own.

That can be frustrating.

When to Start the Sprint?

The Scrum Certification exam will try to trick you into applying some arbitrary rule that doesn't exist.

For example, say a product has two Scrum Teams working on it, which happens quite regularly.

- Should those two teams start their Sprints on the same day?
- Should those two teams end their Sprints on the same day?
- Should those two teams have Sprints of equal length?

On the surface, all three of those assertions seem reasonable. It seems like a nice way to keep the two Scrum Teams in sync.

At the same time, that approach might create chaos. Imagine stakeholders having to go to two Sprint Reviews on the same day. That might be too much for them.

Or maybe two teams who finish their Sprints on the same day would end up pushing their increments into production on the same day, which might cause panic for system administrators?

So maybe it'd be better to stagger the Sprints? Have one team start a 4 weeks Sprint on the first of the month, and have the other start the Sprint in the middle of the month?

No Rule Means No Rules

The fact is, the Scrum Guide doesn't care what those two teams do. The teams are allowed to figure it out on their own.

If the Scrum Guide doesn't speak directly to the topic, then there are no rules about it.

The Scrum Guide **does** say two teams working on the same project must share the same Product Backlog and the same Product Owner.

- Should their Sprints be the same length?
- Should their Sprints start on the same day?

Scrum doesn't provide any guidance on these topics either way. It's assumed your team will use

empirical measures and apply lean thinking and come up with a plan that is best for them. Scrum wants you to figure out what's best for you.