Chapter 17

The Increment

> An Increment is a concrete stepping stone toward the Product Goal.
>
> Each Increment is additive to all prior Increments and thoroughly verified, ensuring that all Increments work together.
>
> In order to provide value, the Increment must be usable.
>
> — 2020 Scrum Guide page 11

An increment must be add to what's already been done, and it must be able to fully integrate and work with all previous increments.

If an increment isn't compatible with past increments, then it's not a valid increment. If an increment stands along and doesn't increase the value of the product built to date, then it's not a valid increment.

On the certification exam, there will be many questions about integration, which is confusing, because there is no talk in the Scrum Guide about integration.

- The are no integration Sprints in Scrum
- There are no integration teams in Scrum
- There are events where integration happens

However, whatever you create today must be able to 'integrate with' or 'work with' whatever you created yesterday for it to be a valid increment.

Imagine you were sent to buy a door to put on the front of the house you were building.

You went and bought:

- A door that's too large
- A door with incompatible hinges
- A door that's the wrong color
- A door that's the wrong material

Did you just create a usable increment of work that adds to the existing project? No, you didn't.

Now, imagine you bought the perfect door that just slides onto the pre-existing hinges. Does an increment of work now exist?

No, it doesn't because it hasn't been integrated into the house. The door has to be hung on the hinges before it can be considered a usable and integrated increment.

And let's say it's super-easy to hang the door on the hinges. When should you hang the door?

- During the Daily Scrum

- During the Sprint Review
- During the Sprint Retrospective
- Right now!

Of course you should hang it up right now.

And then you move onto the next peice of work that needs to be done and you integrate that in.

And you keep doing that over and over and over again until it's time for the Sprint Review. All of which reinforces the next part of the Scrum Guide:

Test Yourself

Which of the following is correct

- ▢ Scrum is incremental but not iterative
- ▢ Scrum is iterative but not incremental
- ▢ Scrum is both incremental and iterative
- ▢ Scrum is neither incremental nor iterative

Option C is correct.

Scrum is iterative in that a set of steps get repeated each sprint, and it is incremental in that each Sprint produces an increment of work that gets added to all past increments.

> Multiple Increments may be created within a Sprint.
>
> The sum of the Increments is presented at the Sprint Review thus supporting empiricism.
>
> However, an Increment may be delivered to stakeholders prior to the end of the Sprint.
>
> The Sprint Review should never be considered a gate to releasing value.
>
> Work cannot be considered part of an Increment unless it meets the Definition of Done.
>
> — 2020 Scrum Guide page 12

Some people get the impression that the only time stakeholders should be allowed to see what developers have done is at the Sprint Review.

If the developers create an increment they want the stakeholders to see, there is nothing stopping them from presenting it to them at any time throughout the Sprint.

And why would you wait? If the developers have created something they are proud of, and the stakeholders are anxious to see it working, why would the development team wait a week or two until the Sprint Review to show it to them?

Commitment: Definition of Done

> The Definition of Done is a formal description of the state of the Increment when it meets the quality measures required for the product.
>
> The moment a Product Backlog item meets the Definition of Done, an Increment is born.
>
> — 2020 Scrum Guide page 12

The definition of done must be clearn and understood by everyone on the Scrum Team and by the stakeholders and the organization. It creates a common understanding about what has to be done in order for a feature to be complete.

Test Yourself

True or false: The Definition of Done is an informal understanding of what needs to be done to turn a Product Backlog item into an Increment.

This is false

The Definition of Done is a formal description, not an informal understanding.

Test Yourself

The application has failed to scale beyond a single processing core, and this has causes various production issues. What is the best way for the Scrum Team to proceed with this issue.

- ▢ Assign multi-core processing support to the DevOps team
- ▢ Create a testing team to implement multi-processor support
- ▢ Add multi-processor support to the Definition of Done
- ▢ Inform that QA team that they should test the app on multiple processors

Option C is correct.

If there is a quality metric that the product must support, it is the development team's responsiblitity to support it. If the requirement is added to the Definition of Done, the developers will not be allowed to release or integrate any increments that don't meet the multi-processor requirement.

Remember that there are no sub-teams in Scrum, and there is no talk anywhere about DevOps, UAT or QA teams anywhere in the Scrum Guide.

> The Definition of Done creates transparency by providing everyone a shared understanding of what work

---

> was completed as part of the Increment.
>
> If a Product Backlog item does not meet the Definition of Done, it cannot be released or even presented at the Sprint Review.
>
> Instead, it returns to the Product Backlog for future consideration.
>
> — 2020 Scrum Guide page 12

Test takers get killed on this point, so pay attention.

What a Product Backlog item is not completed during a Sprint, it is put back ito the Product Backlog.

- It's not presented at the Sprint Review
- It is not 'partially presented' at the Sprint Review
- Partial points are not assigned to it (No 'points' in Scrum!)
- The feature is not automatically added to the next Sprint's Sprint Backlog
- The feature is not partially released

If a Product Backlog item does not meet the Definition of Done during the Sprint, it is thrown back into the Product Backlog as though nobody had ever even been working on it. Even if it's 90% complete, it's thrown back into the Product Backlog for the Product Owner to prioritize it through backlog ordering.

An important, critical feature is 99% done at the time of the Sprint Review. What should the team do?

- ▢ Present the work completed for stakeholders to see at the Sprint Review
- ▢ Assign 99% of the points to the current Sprint and assign 1% of the points to the next Sprint
- ▢ Automatically add the Product Backlog item to the next Sprint's Sprint Backlog
- ▢ Return the item to the Product Backlog and do not present it at the Sprint Review

Option D is correct.

The Scrum Guide is clear. If a feature does not meet the definition of done, it is not presented at the Sprint Review, and it is returned to the Product Backlog.

> If the Definition of Done for an increment is part of the standards of the organization, all Scrum Teams must follow it as a minimum.
>
> If it is not an organizational standard, the Scrum Team must create a Definition of Done appropriate for the product.
>
> — 2020 Scrum Guide page 12

Who creates the Definition of Done?

It’s created by the Scrum Team, unless there is already an existing orginzational standard.

If the Scrum Team wants to create a more rigorous Definition of Done, they are certainly allowed to do that, but under no circumstances

Who creates the Definition of Done?

- ▢ The Product Owner if there is no organizational standard
- ▢ The Scrum Master if there is no organizational standard
- ▢ The Scrum Developers if there is no organizational standard
- ▢ The stakeholders if there is no organizational standard

Option C is correct.

The Scrum Team creates a Definition of Done if an organizational standard does not exist.

> The Developers are required to conform to the Definition of Done. If there are multiple Scrum Teams working together on a product, they must mutually define and comply with the same Definition of Done.
>
> — 2020 Scrum Guide page 12

There are only a few instances in the Scrum Guide where multiple teams working on the same project are discussed.

The rules the Scrum Guide requires for multiple teams working on the same project are few, but you will be tested on all of them:

- Each team shares the same Product Owner
- Each team share the same Product Backlog
- Each team shares the same Product Goal
- Each team shares the same Definition of Done
- Teams may invite other team members to their Sprint Planning

There is nothing in the Scrum Guide that asserts multiple teams working on the same project must:

- Start their Sprints at the same time
- Have Sprints that are the same length
- Have the same number of team members
- Have the same Scrum Master

There will be a number of questions about multiple teams working on the same project on the certification exam, and those questions will provide options that seem reasonable but are incorrect.

When answering questions in the the certification exam, focus on answers that map as closely as

possible to what is in the Scrum Guide. Don’t hunt for pragmatic answers. Hunt for correct answers.

What must shared between multiple teams working on the same product?

- ▢ The Product Owner
- ▢ The Scrum Master
- ▢ The Sprint Backlog
- ▢ The Product Backlog

Options A and D are correct.

Each Scrum team has its own Sprint Backlog.

A Scrum Master can be shared between multiple teams, or a Scrum master can dedicate 100% of their time to one team. the Scrum Guide guide doesn’t advise against either scenario.

When multiple teams work on the same project together:

- Each team shares the same Product Owner
- Each team share the same Product Backlog
- Each team shares the same Product Goal
- Each team shares the same Definition of Done
