Agile Software Development - Scrum

Discover Scrum Software Development! Explore the different Scrum roles, workflows, and values.

Table of Contents

1. Scrum Software Development
2. Scrum Roles
3. Scrum Workflow
4. Sprint Planning
5. Daily Scrum
6. Sprint Review
7. Backlog Refinement
8. Scrum Artifacts
9. Scrum Limitations
10. Scrum Values
11. Scrumban
12. Scrum of Scrums
13. Large-scale Scrum
14. Exercise: Identify Scrum Values

Scrum Software Development

[Topic title: Scrum Software Development. The presenter is Colin Calnan.] In this video, I'm going to describe what Scrum software development is used for. Scrum software development is an agile framework and it's used for completing complex projects. Scrum was originally formed and created in 1993 and it was created by a gentleman named Jeff Sutherland. And in it, he borrowed the name scrum from an analogy that was put forth back in 1986 in a study in the Harvard Business Review. Where high performing cross functional teams were compared to the scrum formation used in rugby games. *[Scrum Alliance.]* As well as that, there's something called a scrum alliance and what that does is it exists to help transform the way that we tackle complex projects. So it brings the Scrum framework and agile principles beyond software development into the broader world of work. As well as that, it's important to note that Scrum is not a process or a technique for building products. But rather it's a framework employ various processes and techniques to help the project move forward. *[Scrum Features.]* So they're essentially three key Scrum features, and that are, that it's lightweight, simple to understand, but difficult to master. *[Sprint.]* So when we look at one of the core components of Scrum, it's the sprint. And basically, the sprint is a time box period in which we perform work. And in order for the sprint to happen, we run through what's called sprint planning. And sprint planning is the work done at the beginning of a sprint. And it's usually time boxed to a maximum of eight hours for a one month sprint. So a sprint should last for one month or less. One sprint will always follow another. So sprints are sequential, that just once one ends, the other starts. And basically, sprints include a flow of sprint planning. Where, in the planning meeting, the team members will pull from the backlog to decide and prioritize which work will be done in the next sprint. Then they'll meet and have daily scrums, where they would talk about the work that has been done and the work that will be done and any impediments. Then there's the development work that happens through the sprint. Then there's the sprint review, which is generally held at the end of the sprint to inspect what happened and adapt the product backlog, if needed. And then finally, there's the sprint

retrospective, which is an opportunity for the team to inspect itself and the processes. And then create a plan for improvements that they can have during the next sprint. So that is a introduction to Scrum software development and what it's used for.

Scrum Roles

[Topic title: Scrum Roles. The presenter is Colin Calnan.] In this video, we're going to identify the roles in the Scrum framework. *[Product Owner Role.]* So the first role we're going to look at is the product owner role. So what is the job of the product owner? Well, the product owner is probably one of the most important jobs in Scrum. And some of the tasks and responsibilities of the product owner are to do things like develop the product vision. So the product vision being what the stakeholders delegate, or inform the product owner of what they want it to do. And then the product owner has to develop the vision of what that will look like. Then they do things like collect and prioritize requirements. So they're collecting requirements and prioritizing them based on the needs of the stakeholders and on the capabilities of the development team. They also control budget and oversee return on investment. So they are caught between both the, I guess, marketing and stakeholder side, as well as the developer side. In terms of controlling budget for tasks and overseeing whether or not the features that are being built are generating a return on investment. They also work on things like product validation and quality verification. So they will look at and make sure that what's being built is actually something that the customer needs. And then they also make sure that the quality that's being generated matches what the customer's expectations are. They interface with all stakeholders, so they're in between both the development team, the financers, the marketers, and everybody else involved with the product. They have varying responsibilities, so not just looking after the development team or working at marketing, but they work in all aspects of the product. And then they're also responsible for the success of the product and of the teams. *[Scrum Master Role.]* So the next role that we would have is the Scrum Master role. So the Scrum Master is generally an advisory role. Someone within the team who can give some advice on things like correct methods and technologies. They're not so much a leader but a coach to the team. So providing them with support when they need it, helping them understand how to execute properly, etc. They ensure that the Scrum rules are enforced. So some of the rules that happen within the daily Scrums that have to be adhered to, they make sure that those are adhered to and that people don't stray outside of those. They're also not involved in the development in any way. So they're usually not a developer and outside of the development team. So they just look at it objectively from the outside. And then they advise on correct methods and technology. So they can give support on the way to do things within Scrum, how to use technologies for that. *[Other Scrum Master Responsibilities.]* Other Scrum Master responsibilities would be do things like eliminate impediments. So if there is something that's stopping one of the members on the team from doing something, then the Scrum Master would help to eliminate any impediments that are there by working together with the team to solve that. As well as that, they can work to solve problems that cannot be solved by the development team. So again, this objective eye from the outside might be able to help solve problems that can't be solved from the inside of the development team. Then they can work to overcome organizational and strategic roadblocks. For example, if there is some hierarchy or some issue internally that needs to be resolved, instead of the development team getting involved in that, the Scrum Master would take that on. They're also good at managing stakeholder expectations. Obviously, the product owner has a role in that. But with the Scrum Master, the expectations are generally around things like what can be done within a certain timeframe? And then they also develop and maintain optimal conditions for performance. So they make sure that the team has what they need, they make sure that all the systems are set up the way they need to be. And in general, they just make sure that everything is set up so that the product can be delivered. *[Scrum Development Team.]*

Then we have the Scrum Development Team. Obviously, one of the very important parts of Scrum as well is the team that are building out the codes and the software. And generally, one of the traits of this team is that they're self-organized. So they don't lead or the Scrum Master doesn't really organize them. The

Scrum Master's just there to help them out when they need some help. And the goals of the team are generally self-determined, so that the team themselves determine their goals. They're self-dependent in that they don't need the Scrum Master to get things solved for them. They don't depend on the Scrum Master in any way. They also help determine what requirements can be met during a sprint. So in this idea of things like planning poker or just making decisions about estimating, they decide what requirements can be met during a specific sprint. And they don't need Scrum Master management, it's not the job of the Scrum Master to micromanage them or people manage them in anyway. They are an independent team that can self-organize and work on a sprint as a needs be. So that those are some of the roles that we can identify in the Scrum framework.

Scrum Workflow

[Topic title: Scrum Workflow. The presenter is Colin Calnan.] In this video, we're going to describe the Scrum workflow. *[Product Backlog.]* So the Scrum workflow starts off with the product backlog. So this is a, essentially, list of features or requirements that the product has to adhere to. And they are prioritized and put in a backlog for the team to pull from. *[Sprint Planning.]* Then the next step would be sprint planning. So this is a meeting where all of the members of the sprint would get together to sit down to go through the product backlog. And determine from the product backlog what can be done within the confines and constraints of a sprint. Then from that, a sprint backlog is created. So the items from the product backlog are moved into the sprint backlog. Where they can be estimated, they can be assigned to individuals and there can be discussion that happens around them. They're a place to go to look at what is the work that needs to be completed on this sprint. Then once the sprint starts, we have the, the sprint would start and it's usually maybe a month or less of time in which the sprint backlog would be completed. And then everyday there is a daily Scrum, a short standup meeting. Where everybody talks about what they have done, the day before and what they are going to work on that day and any impediments that they may have. *[Sprint Review and Sprint Retrospective.]* And then once the sprint has been completed, we have a sprint review, which will look at the work that's been completed within the sprint. And a sprint retrospective, which looks at what happened during the sprint, what processes were used, what impediments were happened upon. And how those can be improved and worked upon for the next sprint. *[Requirement Delivery.]* And then finally, we have requirement delivery. So once things have been reviewed in the sprint and everything looks good, then they're basically pushed into the product. They're delivered into the product. Those requirements have been completed. And then what happens is it starts another increment. We go back to the start again to do a sprint retrospective. And then back into sprint planning, where we pull things out of the backlog and start again with our sprint. So that is an overview of the Scrum workflow.

Sprint Planning

[Topic title: Sprint Planning. The presenter is Colin Calnan.] In this video, I'm going to describe sprint planning and its importance. So let's start with an overview of sprint planning. So the work to be performed in the sprint is basically planned at the start of the sprint, and that is how we do sprint planning. So the plan is basically created by the scrum team through collaborative work within the team. Sprint planning, the planning for the sprint is timeboxed and it's usually timeboxed to a maximum of eight hours for a one month sprint. And basically in sprint planning, what it does is it answers the questions and helps determine what can be delivered in the increment resulting from the upcoming sprint. So that's what we call the result of the sprint is the increment, and then it basically asks how will the work needed to deliver that increment be achieved? So those are some of the answers that sprint planning can provide. *[Sprint Delivery.]* So then we can look at things like determining sprint delivery. So what can be done in the sprint? Well, the development team works to forecast the functionality of the things that can be developed.

As well as that, the product owner discusses the objectives that's been come up with. And then works to see how the capacity, what the project capacity for developing that is. As well as that, the product owner does things like discusses the sprint objectives and works though them with their team. And then as well as looks at the product backlog to see how many of those items could be achieved within the sprint and could achieve the sprint goal. As well as that, the entire scrum team collaborates on understanding the work required to be done during the sprint. *[Sprint Goal.]*
So then we have the sprint goal and basically the sprint goal is an objective set for the sprint, that then could be met through implementation of the product backlog. So that's how the objectives are met is by implementing what's in the product backlog. As well as that, the goal provides guidance to the development team on why it's building that increment and then it's usually created during the sprint planning meeting so the goal is created during that meeting. As well as that, the sprint goal gives the development team some flexibility around functionality that's implemented within the sprint. And then the product backlog, basically the selected product backlog items deliver one coherent function, and that's generally the sprint goal. As well as that, if the work turns out to be different from what the development team expected, they can work with the product owner and negotiate the scope of the sprint backlog within the current sprint. *[How to Achieve Outcome.]* So then, how do we work towards determining to achieve the outcome? How to achieve the outcome? Well generally, having set the sprint goal and selected the backlog for the sprint, the development team then decides how it will build that functionality into like a done product or a deliverable increment. So the product backlog items are basically selected for the sprint. Then with the plan for delivering them, that's what we call the sprint backlog. So you take the product backlog items with the sprint plan and then that's what we call the sprint backlog. As well as that, the development team usually would start by designing the system and the work needed to convert the product backlog into the working product increment. As well as that, the work that's planned for the first few days of the sprint by the development team can be basically decomposed by the end of the meeting often into like smaller units of maybe a day or less. And then finally again, if the development team determines that it has two much or too little time, or too much or too little work. It can negotiate the product backlog items with the product owner to change or modify what's going to be done within and the product that's going to be created by the increment at the end of the sprint. So that was description of sprint planning and its importance within the scrum framework.

Daily Scrum

[Topic title: Daily Scrum. The presenter is Colin Calnan.] In this video, I'll describe the purpose of a daily scrum. So what is a daily scrum meeting? Well, a daily scrum meeting is a meeting that is typically held in the same location and at the same time each day. Ideally, a scrum meeting is held in the morning and that helps set context for the coming day's work. Generally, the scrum meetings are time-boxed, strictly time-boxed to 15 minutes. This helps keep the discussion brisk and relevant. All team members are required to attend the scrum meeting. *[Daily Scrum Features.]* So some of the features of daily scrum. So it's important to note that the scrum meeting is not used as a problem solving or issue resolution meeting, definitely not. Issues that are raised would generally be taken offline and dealt with by the relevant people immediately after the meeting. As well as that, the daily scrum is not a status update meeting in which a boss is collecting information about who is behind schedule or whatever. But it's generally a meeting in which team members would make commitments to each other. As well as that, it's used to synchronize events and create a plan for the next 24 hours. As well as that, it helps the development team determine their progress towards the sprint goal and inspect that in each scrum. So what are the things that during daily scrum each team member needs to focus on? These are kind of the questions that each team member would need to answer. One, what did you do yesterday? So what were yesterday's accomplishments? Two, what will you do today? Focus on those things. Three, are there any impediments in your way stopping you from getting to where you need to get to by the end of today? And then alternatively, there's a possibility to discuss the product backlog. So that's the purpose of daily scrum.

Sprint Review

[Topic title: Sprint Review. The presenter is Colin Calnan.] In this video, I'm going to describe the sprint review and retrospective. So a sprint review is an informal meeting, not a status meeting, that is held at the end of the sprint. And it's used to inspect the increment and also adapt the product backlog, if necessary. So it's based on the idea that any changes to the product backlog during the sprint, that attendees collaborate on the next things that could be done to optimize value. So there's some time for collaboration in there. It's also intended to elicit feedback and foster collaboration within the team. And it's typically four-hour time boxed for one month sprints. For shorter sprints, it could be even shorter. *[Sprint Review Elements.]* So some of the elements of a sprint review, are that the attendees include the Scrum team and the key stakeholders that might be invited by the product owner. The product owner, it's themselves, who would explain what product backlog items have been done, and what product backlog items have not been done. So a discussion about the backlog. As well as that, the development team would discuss what went well during the sprint, what problems it ran into, how the problems were solved. The development team would also demonstrate the work that was done, and answer any questions about the increment. As well as that, the entire group will collaborate on what to do next. There's also a review of the timeline, the budget, the capabilities and the marketplace for the next anticipated release of the product. So there's a look at a number of things, including marketplace and when the next release might be. *[Sprint Retrospective.]* Then we have the sprint retrospective. And that generally occurs after the sprint review but prior to the next sprint itself, or to the next sprint planning. And it basically, the purpose of it, is to allow the Scrum team to inspect how the last sprint went, in regards to things like people, relationships, processes and tools. And then it can help the team to create a plan for implementing improvements to the way that the Scrum team does its work. It's a three-hour time boxed meeting for one month sprints. And the Scrum master participates as a peer team member in this meeting, as well. So from an accountability standpoint over the Scrum process, they're treated as a peer within the meeting. By the end of the sprint retrospective, the Scrum team should have identified improvements that need to be made, that can be implemented in the next sprint. And usually, you would have done this, by inspecting how the last sprint went. Then identifying any, and ordering any items that maybe need improvement. And then finally, adapting the process to ensure that those improvements are achieved. So that's an overview of the sprint review and the sprint retrospective.

Backlog Refinement

[Topic title: Backlog Refinement. The presenter is Colin Calnan.] In this video, I'm going to describe what backlog refinement is. *[Backlog Grooming.]* So backlog refinement is also known as backlog grooming. And basically it's an activity that occurs on a regular basis, and is something that might be an official secluded meeting or an ongoing activity. It usually happens when the product owner or some or all of the product team, or the scrum team, review the items that are on the product backlog. It helps to ensure that the backlog contains appropriate items. That those items are prioritized correctly and appropriately, and that the items at the top of the backlog are ready for delivery. *[Actions During Backlog Refinement.]* So some of the actions that happen during backlog refinement and some of the activities that occur would be creating new user stories and that would be in response to newly discovered needs for the product. Removing user stories that are maybe no longer relevant to their priority for the product, as well as that, reassessing the priority of existing stories. Creating or developing estimates for stories which haven't yet been estimated. Adjusting estimates based on newly discovered information, and then splitting user stories which might be high priority but might be too coarse to fit into one iteration. So the story might be too large to fit into one iteration that they would be broken down into smaller stories that could then be implemented into the next iteration. *[Benefits of Backlog Grooming.]* So what are the potential benefits of doing this backlog grooming? Well, one would be to ensure that the backlog remains populated with items

that are still relevant. As well as that, to ensure that the backlog items are detailed and estimated to a degree that's appropriate with the priority of those items. As well as that, unlike a more formal requirements document, the backlog serves as a dynamic body of information. And at any time there are a sufficient number of stories in there that could be used for scheduling in the next few iterations. As well as that, backlog grooming can help avoid the idea of scope creep on requirements that maybe haven't been flushed out. And then that avoidance of scope creep can also help avoid things like cost and schedule overruns. So that is an overview of backlog refinement or what's also known as backlog grooming within Sprint and Scrum. In this video, I'm going to describe what backlog refinement is. So backlog refinement is also known as backlog grooming. And basically it's an activity that occurs on a regular basis, and is something that might be an official secluded meeting or an ongoing activity. It usually happens when the product owner or some or all of the product team, or the scrum team, review the items that are on the product backlog. It helps to ensure that the backlog contains appropriate items. That those items are prioritized correctly and appropriately, and that the items at the top of the backlog are ready for delivery. So some of the actions that happen during backlog refinement and some of the activities that occur would be creating new user stories and that would be in response to newly discovered needs for the product. Removing user stories that are maybe no longer relevant to their priority for the product, as well as that, reassessing the priority of existing stories. Creating or developing estimates for stories which haven't yet been estimated. Adjusting estimates based on newly discovered information, and then splitting user stories which might be high priority but might be too coarse to fit into one iteration. So the story might be too large to fit into one iteration that they would be broken down into smaller stories that could then be implemented into the next iteration. So what are the potential benefits of doing this backlog grooming? Well, one would be to ensure that the backlog remains populated with items that are still relevant. As well as that, to ensure that the backlog items are detailed and estimated to a degree that's appropriate with the priority of those items. As well as that, unlike a more formal requirements document, the backlog serves as a dynamic body of information. And at any time there are a sufficient number of stories in there that could be used for scheduling in the next few iterations. As well as that, backlog grooming can help avoid the idea of scope creep on requirements that maybe haven't been flushed out. And then that avoidance of scope creep can also help avoid things like cost and schedule overruns. So that is an overview of backlog refinement or what's also known as backlog grooming within Sprint and Scrum.

Scrum Artifacts

[Topic title: Scrum Artifacts. The presenter is Colin Calnan.] In this video, I'll describe the three Scrum Artifacts and the role that they play. So firstly, the three Scrum Artifacts are Product Backlog, Sprint Backlog and the Increment. So Scrum's Artifacts represent work or value to provide transparency and opportunities for inspection and adaption, which we looked at earlier. Product Backlog is basically an ordered list of everything that might be needed in the product. It's the single source of requirements for any changes that need to be made in the product. Also, a Product Backlog is never complete, it's constantly evolving as the product in the environment change and evolve. As well as that the product owner is responsible for the Product Backlog, and that includes its content, availability, and ordering. As well as that, Product Backlog refinement is also part of the Scrum team's responsibilities. So the act of adding detail and estimates, etc., and ordering items is also done by the Scrum team. As well as that, higher order Product Backlog items are usually clearer and more detailed than lower order ones. So the higher priority, the more detail and the clearer they are as opposed to the lower priority ones. So here's some common components of the Product Backlog. So the Product Backlog lists all features, functions, requirements, enhancements and fixes that make up changes to be made to the product in future releases. *[Common Components.]* Product Backlog items have attributes of a description, order, estimate and value. So those are the things that can be stored within the Product Backlog, in relation to the items that are in there. So then the Sprint Backlog. So the Sprint Backlog is then a set of the Product Backlog items selected for a specific sprint. So generally, the Sprint Backlog is also a plan for delivering the next product

increment and it's the realization of the sprint goal in a backlog form. The Sprint Backlog is also a forecast by the development team about the functionality that they're going to deliver in the next increment. And then the work that needs to be done to deliver that functionality into a done increment. So the backlog is essentially a functionality forecast of the work that needs to be done. Then as well as that, the Sprint Backlog is a plan with enough detail that changes in progress can be understood in the daily Scrum. And as well as that, the Sprint Backlog is a dynamic, highly visible, real-time picture of the work that the development team plans to accomplish during the sprint. The product Sprint Backlog belongs solely to the development team. And it's generally used to monitor progress because at any point in time in a sprint, the total work remaining in the Sprint Backlog can be summed up to show us what's left.And then finally, the final Scrum Artifact is the Increment. And the Increment is basically the sum of all the Product Backlog items that are completed in a sprint, and the value of all of the increments of all other previous sprints. So at the end of the sprint, the new increment must be done, which means that it's in a usable condition and meets the Scrum team's definition of done. It must be in a usable condition regardless of whether the product owner decides to actually release it. So that's the Increment. So there we have the three Scrum Artifacts. That is the Product Backlog, Sprint Backlog, and the Increment.

Scrum Limitations

[Topic title: Scrum Limitations. The presenter is Colin Calnan. Limitations of Scrum.] In this video we're going to identify the limitations of Scrum. So Scrum doesn't, or in some cases, doesn't satisfy top management's need for budget, scope and schedule control. Generally, they get exactly what they want but they can be unaware of what it will cost or how long it will take to reach a finished product. So it doesn't provide detailed estimates of time and cost that management likes. So it generates uncertainties surrounding cost, completion date, because it focuses on ballpark estimates of time and cost, that leads to this uncertainty. It also leads to uncertainty surrounding the final product. As well as that, for traditional organizations that are built on hierarchical structures, Scrum kind of contradicts that structure. In that, it uses self-organizing teams and intense collaboration, which can be incompatible with those corporate cultures. It can also be sometimes difficult for the Scrum master to plan structure and organize a project that lacks clear definition. Because with Scrum, only the next backlog items in the iteration will be defined clearly. And then the daily Scrum meetings and the frequent reviews can require substantial resources. So that is something that could be a limitation. So then we've also got Scrum, some of the cons of Scrum. Well, Scrum often leads to scope creep due to the lack of some definite end dates. As well as that, the chances of project failure are high if individuals aren't committed and cooperative. Adopting the Scrum framework in large teams is also challenging. The framework can be successful but only with experienced team members. Daily meetings can be sometimes frustrating and onerous for team members. And then if any team member leaves the project in the middle, it's going to have a huge impact on the project. As well as that, quality is always hard to implement, until the team goes through some aggressive quality testing processes. So that's another con. So those are just some of the limitations of the Scrum process.

Scrum Values

[Topic title: Scrum Values. The presenter is Colin Calnan.] In this video, I'll identify the Scrum values and what they mean. So the Scrum values we're going to cover are commitment, focus, openness, respect, and courage. *[Commitment.]* So let's look at the first one, commitment. So a commitment, a commitment to the team to collaborate and learn the best we can. To be committed to the sprint goal. To be committed to the selected product backlog items that we've created. A commitment on scope isn't possible because in the highly complex, highly unpredictable world of software development, a commitment on scope is just not possible. So commitment is about dedication, and that applies to the actions and effort, not the final

results. So we commit to actions and efforts. As well as that, we commit to creating working software. *[Focus.]* So the next value we'll look at is focus. We can look at focus, in something like scrum, which is an iterative and incremental approach, we use timeboxing that helps us focus. We can also focus on what's important now without being bothered by considerations of what might be important at some point in time in the future. We also focus on what we know now and the acronym YAGNI, or you ain't going to need it. So we focus on what we know and things that we're not going to need, that helps us retain focus. As well as that, we focus on what's nearby in time and recognize that the future is highly unpredictable. We also want to learn from the present, so that we can gain experience for the future. So focus helps do that. *[Openness.]* Then we have openness. So scrum requires transparency and openness to work. As well as that, we want to inspect reality in order to be able to adapt. We are open about our work, our progress, and our problems. We are also open for people and working with people. We're open to collaboration across disciplines and across skills. And we're open in sharing feedback and learning from one another. As well as that, we're open for change, as the organization in the world that it's in, change unpredictably, unexpectedly, and frequently and constantly. *[Respect.]* So as well as that, we want respect. We show respect for people. We value their experience and their background. We respect diversity and accept different points of view. We respect the framework, the Scrum framework. We respect our wider environment by not behaving as if we're some sort of isolated island in the world. We also respect the stakeholders. We respect the accountabilities of the Scrum rules. *[Courage.]* And finally, we have courage. And courage, we show courage by building what people want and need. We also show courage in admitting that requirements will never be perfect and not everything can be planned because of complexity and reality. As well as that, we recognize the courage to consider that change is a source of good and inspiration and innovation. The courage to not deliver incomplete or unsatisfactory software. The courage in sharing all of our information, so transparency. The courage to change direction when needed and to share risks and the benefits of those risks. And then finally, we show the courage to support the Scrum values, the ones that we've just covered. So those are the Scrum values, commitment, focus, openness, respect, and courage.

Scrumban

[Video description begins] *Topic title: Scrumban. The presenter is Colin Calnan.* [Video description ends]

In this video, I'll describe Scrumban and how it is used. So here's an example of a Scrumban board.

[Video description begins] *The example shows a table that contains six columns: Backlog, To-do, Designing, Developing, Testing, and Done.* [Video description ends]

You can see that there are some terms in here that are familiar to Scrum such as the backlog, developing, testing, as well as user stories. You can see that's very familiar to a Kanban style board

[Video description begins] *The Kanban style board example includes a table containing three columns: TO-DO, DOING, and DONE.* [Video description ends]

where you've got to do items and done items. So we know Scrum and Kanban as flavors of Agile. Scrum's best suited for product development and web development projects. Kanban is best for production support. So Scrumban combines the best features of both Scrum and Kanban, and it's used a lot in maintenance projects. Scrumban is becoming very popular in the service industry, where we have both development and maintenance projects.

[Video description begins] *Overview of Kanban.* [Video description ends]

So let's have a quick look overview of Kanban. So what's Kanban? Well, it's workflow visualization. It splits work into smaller items. Items are usually written on a card and then put on the wall. And we use named columns to illustrate where each item is in the workflow. As well as that, we limit work in progress. So we assign explicit limits on how many items may be in progress in each state, in each workflow state. And then we measure the lead time, which is basically the average time to complete one item or the cycle time. And then optimize the process to make the lead time as small and predictable as possible. So then we have Scrumban, Scrum and Kanban. And that basically uses the kind of prescriptive nature of Scrum to help it be agile, so to create agility. As well as that, we use the process improvement of Kanban to allow the team to continually improve its process. We also use Kanban's pull system, so our flows becomes smoother as our capability improves. As well as that, the average lead time and cycle time will become the primary focus of performance. And now since the team pulls work into a small cube before it pulls into the work in progress, the team's perspective of the iteration backlog is that it always contains something worth doing. So there will always be something worth doing next in the backlog. And rather than going through the trouble of estimating the scope of work for every single iteration, in Scrumban, we just fix the size of the backlog. And that we plan that it will occasionally run to zero before our planning interval ends. As well as that, within Scrumban we can do iteration planning at regular intervals with review and retrospective.

[Video description begins] *Benefits of Using Scrumban*. [Video description ends]

So the benefits of using Scrumban are just in time which means decisions and facts are just when they're needed. We can also reduce the lead time. And we can create process improvement by adding some other values of Scrum when needed. As well as that, this is the idea of kaizen or continuous improvement. We improve quality and we also minimize and reduce waste, in that everything that is not adding value to the customer we take out.

[Video description begins] *When to use Scrumban*. [Video description ends]

So when should you use Scrumban? Well, sprint teams who are focused on new product development could use it. Projects with frequent changes or unexpected user stories. Projects with frequent programming errors. If Scrum is maybe challenged by workflow or resource challenges, you could implement Scrumban in its place. For projects managing improvement communities after roll out as well as working on maintenance projects.

[Video description begins] *Best Practices for Scrumban Backlog*. [Video description ends]

So some of the best practices for the Scrumban backlog would be on demand prioritization. So that's kind of like the ideal work planning process. It should always provide the team with the best thing to work on next. No more and no less, so everything, it's just prioritizing on demand. As well as that, the backlog should be event driven. And we also want to assure an adequate level of analysis before starting development. As well as that, we want to avoid creating too many user stories, and we want to avoid analyzing too many user stories, because all of that creates waste.

[Video description begins] *Difference Between Scrum and Scrumban*. [Video description ends]

And then, finally, here's a table of differences between Scrum and Scrumban.

[Video description begins] *Artifacts*. [Video description ends]

So, if we look at the artifacts, in Scrum we might have boards, backlogs and burndowns. In Scrumban we just have boards.

[Video description begins] *Roles*. [Video description ends]

For roles, in Scrum you would have product owner, Scrum master, the development team. In Scrumban you have the development team and other roles.

[Video description begins] *Meetings*. [Video description ends]

Then for meetings there's, in Scrum, a daily stand up planning meetings, review and retrospectives. And then Scrumban is essentially daily with others as needed.

[Video description begins] *Teamwork*. [Video description ends]

In Scrum, the teamwork is collaborative, and in Scrumban we have this idea of a swarm.

[Video description begins] *Iterations*. [Video description ends]

For iterations within Scrum we have iterations. Within Scrumban, there are no iterations.

[Video description begins] *WIP*. [Video description ends]

Work in progress, so there's the sprint content for the Scrum. And in Scrumban we have essentially the workflow state that's it's in.

[Video description begins] *Estimates*. [Video description ends]

Estimates exist in Scrum. There are no estimates in Scrumban.

[Video description begins] *Changes*. [Video description ends]

Changes that occur will be planned for the next sprint in Scrum. And then they're done as needed in Scrumban.

[Video description begins] *Product Backlog*. [Video description ends]

And then we have product backlog. That is something that's ordered and estimated within Scrum but it's just in time on Scrumban.

[Video description begins] *Impediments*. [Video description ends]

And then, finally, impediments. Those are immediately managed within Scrum but in Scrumban those are completely avoided. So that is basically Scrumban, the merging of Scrum and Kanban together, and how it's used within an organization.

Scrum of Scrums

[Topic title: Scrum of Scrums. The presenter is Colin Calnan.] In this video, I'll describe how Scrum of Scrum helps improve team coordination. So what is Scrum of Scrums? Well, the Scrum of Scrums is basically a technique that's used to scale up to large groups, so over a dozen people. Where it consists of dividing the groups into agile teams of 5 to 10 individuals. So each daily scrum, within a subteam, ends by designating one member as an ambassador to then go and participate in a daily meeting with ambassadors from other teams. So that's called the Scrum of Scrums. So as well, depending on the context, ambassadors might be technical contributors or the scrum master for each team, or even the

manager of each team. So the Scrum of Scrums basically works as a normal daily meeting. The ambassadors from each team report on completions, next steps and impediments on behalf of the teams that they represent. The Scrum of Scrums is expected to focus on the resolution or solving of coordination challenges between teams. And then the Scrum of Scrums will track these items via a backlog of its own where each item contributes to improving the between team coordination. So that's the Scrum of Scrums, and how it helps improve team coordination.

Large-scale Scrum

[Topic title: Large-scale Scrum. The presenter is Colin Calnan. LeSS Framework.] In this video, I'll describe the two different Large Scale Scrum frameworks or LeSS Framework. So what's LeSS? Well, LeSS is basically a scaled up version of one team Scrum. And it maintains many of the practices and ideas of one team Scrum. But in LeSS, you'll find a single product backlog because it's for product not for a team. You'll find one definition of done for all teams. You'll find one single potentially shippable product increment at the end of each sprint. You'll find one single sprint, one single product owner, and then many complete cross-functional teams. In LeSS, all teams are in a common sprint to deliver a common shippable product every sprint. So how does it work, *[Sprint Planning 1.]* well, we have Sprint Planning 1. So in this spring planning, in addition to the product owner, it's going to include people from all teams. The idea is to let teams members self-manage to decide their division of the product backlog items. *[Sprint Planning 2.]* Then we have Sprint Planning 2, and that's held independently by each team. And sometimes for simple coordination in learning, two or more teams may hold it in the same room. *[Daily Scrum.]* So daily scrum, this is also held independently by each team, though a member of a team A may observe team B's daily scrum to increase information sharing. *[Coordination.]* So then there's coordination, so talking, communicating in code, open spaces, communities. *[Product Backlog Refinement.]* As well as that, the overall, so if we look at the overall product backlog refinement, there might be optional or short overall product backlog refinement meetings. And that would include one product owner, and people from all teams. And the purpose would be to decide which team are likely to implement which items. And therefore select those items for a later in-depth single team product backlog refinement or PBR. So then the PBR itself, the only requirement in LeSS is that there is a single team PBR, that's the same as one team scrum. And so basically, there maybe a variation to that. And that, there might be a multi-team PBR, where you'd have two or more teams in the same room. And that would help increase learning and coordination. *[Sprint Review.]* As well as that, there's sprint review. So in addition to the product owner, it's going to include people from all teams, and relevant customers, users and other stakeholders. *[Overall Retrospective.]* And then there's an overall retrospective, and that's a meeting that's not found in one team scrum. And the purpose is to explore improving the overall system, rather than focusing on one team. So the maximum duration of it is 45 minutes per week of sprint. And it's going to include the product owner, the scrum masters, and then representatives from each team. So basically LeSS provides two different Large-Scale Scrum frameworks, LeSS frameworks. One of them is up to eight teams of eight people each, and then the other is up to a few thousand people on one product. So you can see that these are built for scaling to really, really large teams. So that's the LeSS Framework and how it might be used.

Exercise: Identify Scrum Values

[Topic title: Exercise: Identify Scrum Values. The presenter is Colin Calnan.] In this exercise, you will identify the roles in the Scrum framework. Describe sprint planning and its importance. Describe what the backlog refinement is. Describe the 3 Scrum artifacts and the role they play. Identify the limitations of Scrum. Identify the Scrum values and what they mean. And describe Scrumban and how it is used. So pause the video, find your answers to some of these questions, and then we'll come back and look through

them. *[Roles in Scrum.]* Okay, so the first one we look at is identify the roles in the Scrum framework. Well, that's pretty straightforward, there are three roles in the Scrum framework. Product owner, Scrum master and Scrum development team. Those are the primary roles in the Scrum framework. *[Sprint Planning.]* So describe sprint planning and its importance. So sprint planning is the work to be performed in the sprint, and it's planned at the sprint planning. So at the sprint planning, it's when we plan all of the work that needs to be done in the sprint. The plan is created through collaborative teamwork on the entire Scrum team. And generally, sprint planning answers the questions, what can be delivered in the increment resulting from the upcoming sprint? And how will the work needed to deliver the increment be achieved? So it's what is being delivered and how can that be delivered. That's usually what's covered in the sprint planning. *[Backlog Refinement.]* Describe what a backlog refinement is. Well, backlog refinement, or what's also known as backlog grooming, is when the product owner and some of the team, usually all of the team, review the items in the backlog. To ensure that the backlog contains appropriate items that need to be done, that they're prioritized. And that the items at the top of the backlog are ready for delivery. So that's what backlog refining or grooming is. *[Scrum Artifacts.]* Describe the 3 Scrum artifacts and the role that they play. So there are 3 Scrum artifacts, product backlog, sprint backlog, and increment. The product backlog is a list, an ordered list, of everything that might be needed in the product. And it's the single source of requirements for any changes that need to be made to the product. At any point in time, the total work remaining to reach the goal can be summed up in the product backlog. So the sprint backlog, the sprint backlog is basically a set of those product backlog items that are selected for the sprint. There's also a plan for delivering the product increment and realizing the sprint goal. The sprint backlog makes all of the work that the development team sees as necessary to meet the sprint goal visible. And then finally the increment. The increment is basically the sum of all the backlog items that have been completed during a sprint and the value of the increments of all other previous sprints. So at the end of a sprint, the increment must be done. Which means that it must be in a usable condition and meet the Scrum team's definition of what is done. It basically must be in a usable condition regardless of if the product owner decides to release it or not. *[Limitations of Scrum.]* So the next question, identify the limitations of Scrum. So there are some limitations, Scrum doesn't provide detailed estimates of time and cost that management sometimes likes. Instead, it focuses on ballpark estimates of time and cost. This can also lead to uncertainty around cost completion date and final product. And the regular meetings that happen within Scrum also can consume substantial resources, which can be a bit of a limitation. *[Scrum Values.]* The next question, identify Scrum values and what they mean. So the Scrum values are commitment, focus, openness, respect, and courage. So they're pretty straightforward. Commitment is commitment to the team, to the sprint goal, commitment to the backlog items. Commitment to the actions and efforts, as well as creating working software. So commitment to the process. Focus, focus is achieved through time-boxing. It helps focus on what's important right now and on what is known. And recognizing that the future is unpredictable. Openness, things like transparency and openness around work, progress and problems. Realizing that people are not resources and that they require openness. Collaboration, feedback, learning, and change. So openness surround all of these things. Respect, so show people due respect. Value diversity and value peoples' experience. Respect the framework, respect the Scrum framework. Respect the environment that you're working in, respect stakeholders, and respect the fact that accountability is necessary. And then finally, courage. The courage to build something that people want and actually need. The courage to accept that change is good, the courage to not deliver incomplete or unsatisfactory software. The courage to be transparent and to share risk. And the courage to support the values, these important values of Scrum. *[Scrumban.]* And then finally, to look at, describe Scrumban and how it is used. So Scrum and Kanban are two types of Agile, two methodologies. Scrum is best suited for product and development projects. Kanban is mostly used in kind of production support. Scrumban combines the two and uses the best features of both, and it's generally used in maintenance projects. Scrumban uses the prescriptive nature of Scrum to be agile. And it uses the processing improvement of Kanban to allow the team to continually improve its process.