

Agile Software Development with Scrum
Scrum FAQ by Ken Schwaber
(transcript of the podcast series)

2nd January 2006



Document Controls

Document History

Version	Date	Change	Author
1.0	02/01/2006	Document created	Howard van Rooijen

Circulation List

Name	Company
Agile Community	Conchango

Related Documents

- [Agile Software Development with Scrum Podcast Series.](#)



Contents

What is Agile?	5
What is the Agile Manifesto?	5
What is the Agile Alliance?	5
What is the Scrum Alliance?	5
How do I become a Certified Scrum Master?	5
What is a Scrum Practitioner?	5
What is a Scrum Trainer?	6
Where do I find a Scrum Master?	6
What is Scrum?	6
Why should I be thinking about adopting Scrum?	6
How will Scrum affect my organisation?	6
What are the benefits of using Scrum?	6
How can I adopt Scrum?	7
Where can I read more about Scrum?	7
How do you deal with fixed-price projects?	7
What is a Sprint?	8
What is a Product Backlog?	8
What is a Sprint Backlog?	8
Does Scrum need a Project Manager?	8
Does Scrum need a Project Plan?	8
Who attends the Daily Scrum?	8
How long is a Daily Scrum?	8
What are the Scrum Roles?	9
Who tells the Team what to do?	9
What does a Scrum Team look like?	9
Do I need very good developers for Scrum?	9
What does a Scrum Master do?	9
What does a Scrum Master do on a normal day?	10
Can the Scrum Master be a developer or an architect?	10
Can Scrum be useful for non-software projects?	10
Do I need to stick to a month time frame?	10
Why is everything in Scrum time-boxed?	11
Why does Scrum say nothing about engineering practices?	11
How do I measure the performance of the team?	11
How do I measure the performance of a team member?	11
What is the benefit of doing a Sprint Retrospective?	11
How to fix a bug within a Sprint?	12
What kind of reports do I supply to management?	12
Why would you cancel a Sprint?	12
How can you get buy-in for the use of Scrum?	12



Can Scrum work without fulltime commitment from one or more of the stakeholders?	12
Do Scrum teams need to be co-located?	13
Doesn't Scrum mean we have to throw away the best IT practices we've developed over the last 20 years?	13
What's the cost of change? Will the introduction of Scrum cause more problems than it solves?	13
Is there a downside to changing the make up of a team after the Sprint has started?	13
How do you scale Scrum to larger projects?	13



What is Agile?

Agile is a set of processes for software development, that uses iterative, incremental techniques and relies on self organising, self managing, cross functional teams. It heavily relies on people – it's less about process and more about what people can do. You can find parallels to Agile in Lean Manufacturing - Agile is to Lean Manufacturing as Traditional Software Development is to Traditional Manufacturing.

What is the Agile Manifesto?

In February of 2001 a number of people who had been using new approaches towards software development got together to talk about the similarities of their approaches and the differences between those and what Rational was proposing in terms of the Rational Unified Process. We found that we had a number of things in common and we documented those in terms of the [Agile Manifesto](#):

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

You can read more about the history of the Agile Manifesto [here](#) and there are also more articles at the [Agile Alliance](#) site.

What is the Agile Alliance?

The [Agile Alliance](#) is a non-profit organisation that was founded after the [Agile Manifesto](#) was signed and published, for people who were interested in promoting and knowing about Agile processes, to join, so that they could feel they were part of the movement.

What is the Scrum Alliance?

The [Scrum Alliance](#) is an emerging group of people who are [Certified Scrum Masters](#) who are working together to share their experiences and to move forward the understanding of the Scrum process. The primary manifestation of this is at Scrum Gatherings, which are held several times a year where the people get together and make progress on the Scrum process.

How do I become a Certified Scrum Master?

You attend a [Certified Scrum Master Course](#) – these are 2 day courses taught worldwide by people who have been qualified to teach these courses. The material constantly evolves and upgrades.

What is a Scrum Practitioner?

A Scrum Practitioner is a Certified Scrum Master who has practiced Scrum for at least a year and who has filled out a questionnaire describing at least one of their Scrum projects and how they have applied the Scrum process to it. You can become one by going to:

<http://www.controlchaos.com/certification/practitioners.php>



What is a Scrum Trainer?

A Scrum Trainer is someone who can train people to become [Certified Scrum Masters](#) by teaching a 2 day Scrum Certification Course. They are all Certified Scrum Masters who have passed through the Scrum Practitioner level and have taught a course with Ken Schwaber.

You can find out more at: <http://www.controlchaos.com/certification/trainers.php>

Where do I find a Scrum Master?

There are, at this time, over 3400 [Certified Scrum Masters](#), people who have been trained on how to be a Scrum Master, many of these people are excellent, some of them however were totally asleep during the class and aren't worth even talking to!

The best source of personal characteristics for a Scrum Master is to look for someone who is used to working in almost impossible situations to get something done. Someone who is used to being stubborn and persistent, and who is used to getting things done without totally offending everyone.

What is Scrum?

Scrum is an Agile process or framework for managing Agile projects. It is a project management process, certainly not a methodology, as that would be too heavy.

Why should I be thinking about adopting Scrum?

If you develop software and are unhappy with the turnover of staff, if you are unhappy with the quality of software being built, if your customers are unhappy with the way you're building software and are starting to send the work off shore to save money or getting someone else to build it, if you are unhappy with the productivity of your software developers, if your developers do not like coming to work in the morning – you should consider adopting Scrum. However adopting Scrum is a very difficult activity. So to the degree that you have been experienced problems with any of the items mentioned before; the pain of those must be greater than the pain of adopting Scrum.

How will Scrum affect my organisation?

Scrum will totally change your organisation. Organisations are organised around the processes they use. The traditional method is based around people telling other people what to do – this is a command and control structure and about the belief that the future is predictable. Scrum is based around the idea that people can manage themselves and that the future is unpredictable and the best we can do is to make the most intelligent adaptations to it that are possible to deliver something of value. Shifting the way an organisation works based on these different assumptions is massive and requires a significant change process. This is the difficulty of implementing Scrum.

What are the benefits of using Scrum?

The benefits of Scrum are a competitive advantage if you are an ISV or your ability to respond more quickly to your competition if you are a business organisation. The benefits are the ability to develop and implement high value, high priority software quickly to increase your return on investment, to reduce your staff turnover, to increase your staff satisfaction and to increase the quality of your product.



How can I adopt Scrum?

You can start using it. There is no roll out of Scrum; there is no detailed plan of implementing Scrum because every organisation is different. The way that any organisation, at any point in time, is dysfunctional is different based on the organisations history – so the way you adopt Scrum is going to be different. The impact of using it is going to be different. The only way you'll find out is by using it. Just figure out what the project needs and start doing iterative, incremental development – it's that simple. Organisations start using Scrum within 2 days.

Where can I read more about Scrum?

There are two books that are primary sources of information about Scrum;

- [Agile Software Development with Scrum](#)
- [Agile Project Management with Scrum](#)

There is also a Yahoo! egroup called [Scrum Development](#) that has tonnes of information between people asking questions about Scrum and people who are using Scrum responding.

[Control Chaos](#), [Mountain Goat Software](#), [Scrum-Master](#) are all great sources of information that are more dynamic, of course, than any book.

How do you deal with fixed-price projects?

Scrum works well when the customer and the [Product Owner](#) trust the [development team](#) and are working together collaboratively. When the customer has had enough projects fail that they don't trust the development organisation to do what they say, they usually enter into contractual relationships with the software development group to hopefully minimise their risk and to provide a way of suing that organisation if they fail to deliver. These are called fixed price fixed date contracts. These attempt to predict exactly the cost and the time at which the software that's been specified by the customer will be delivered.

Scrum does this the same way that any other process deals with fixed price, fixed date contracts – they detail everything the customer wants and estimates it in detail and come up with a time at which that can be delivered and contracts with the customer to do so. However, if you are using Scrum you start using iterative incremental development to deliver increments of software every month.

We also enter into 4 contractual modifications with the customer, which gives them the option, should they want to do so, to collaborate with the development teams during the project. These 4 contractual provisions are:

1. For any part of the requirements that the team hasn't started working on yet, the customer is free to change them with anything of equal value.
2. For any part of the requirements that the team hasn't started working on yet, the customer is free to reprioritise them
3. For any part of the work that has been delivered as increments already, the customer is free to ask the teams to implement them in addition to any call for implementations and they will be charged time and material to do so.
4. Since the team has given the customer, as part of the proposal, the list of requirements of the project in a prioritised list by value, the customer can often see the system they want emerging and may feel satisfied before the entire set of requirements are developed. In this case they can cancel the contract at this point in time and take delivery of the system that is adequate to their needs and is only charged some penalty (such as 25% of unbilled revenues) to take care of any overhead that the team has absorbed by entering into the contract.



The intention of this is to satisfy the customers distrust and at the same time offer an opportunity for them to enter into a trusting, collaborative, Agile relationship with the Scrum teams.

What is a Sprint?

A sprint is a monthly iteration.

What is a Product Backlog?

A product backlog is a prioritised list of things that a customer wants a development team or development teams to produce.

What is a Sprint Backlog?

Every iteration a [Product Owner](#) (customer) gets together with a team and tells them what [product backlog](#) items or requirements they want the team to work on for the next [Sprint](#) to turn into an increment of potentially shippable product functionality. The team takes these Product Backlog Items, as many as they think they can do in the next Sprint and they decompose them into the tasks required to produce the increment of potential shippable product functionality. These tasks are called the Sprint Backlog. This is an analogue to Lean Manufacturing where you only build a Sprint Backlog when a Product Backlog is presented to you and requested to be turned into an increment; so this is Just In Time Order Processing.

Does Scrum need a Project Manager?

The [Scrum Master](#) is the “Agile” Project Manager, but unlike the traditional Project Management role they aren't solely accountable for the success or failure of the project, only that the Scrum process is used properly during the project.

Does Scrum need a Project Plan?

All projects need a project plan. However the Scrum project plan is not detailed into tasks until the Sprint where we're turning the Product Backlog item into an increment occurs. Until that point the project plan stays at the requirements level and even the requirements are granular at the top level of priority and are pretty coarse at the bottom end of the priority. This is because the project plan is anticipating change and so not a lot of effort is put into the items which are most likely to change. So the project plan itself may layout and anticipate a date, however it puts in no penalty for changing the requirements or the date or the cost.

Who attends the Daily Scrum?

The [team](#) attends the Daily Scrum, to inspect what each other did the day before, what each other is planning to do and to make adaptations to maximise their chances of delivering what they've committed to for that Sprint. These are the “Pigs” of the Sprints, that is the ones who have skin in the game. Anyone else is free to attend but they are not allowed to say anything, make facial gestures that imply approval or disapproval – they are required to be invisible; they are there to listen only.

How long is a Daily Scrum?

15 minutes. Scrum uses time boxes throughout to stop things dragging on forever and to give regularity to the whole process.



What are the Scrum Roles?

There are only 3 Scrum Roles:

The Product Owner - who represents all customers and manages the Product Backlog. This is the person who is always able to present to the team what the highest priority pieces of work to work on next are.

The Team – there maybe many teams who work on a Scrum project, but each team should optimally be 7-9 people; they form cross functional teams that manage themselves. It's totally up to them, within the guidelines, standards and conventions of the organisation to figure out the work necessary to turn the Product Backlog that the customer wants turned into an increment and then to do the work to turn the Product Backlog into the increment of potentially shippable product functionality.

The Scrum Master – this role used to be called the Project Manager, but the Scrum Master doesn't have authority over the Product Owner, over The Team or over anyone, they are simply responsible for making sure that the Scrum process is used as it's described; they are like a process manager – to the extent that the Product Owner, to the extent that The Team don't know how to fulfil their roles within the process – the Scrum Master is responsible for teaching them how. Different analogies for a Scrum Master are a sheep dog with a flock of sheep or a parent with a child that's growing from relative immaturity to a mature state.

Who tells the Team what to do?

The Team tells itself what to do.

What does a Scrum Team look like?

A Scrum team looks like a group of people working together as a community. They look like people working together on a problem. If you look at a well functioning sports team – that is what a Scrum team looks like.

Do I need very good developers for Scrum?

You need very good developers for software development. You can do Scrum with terrible software developers and you will get terrible increments of functionality every Sprint. The difference is you will know it every Sprint end that you have terrible increments, because you have terrible developers.

If you have very good developers you will get very good increments at the end of every Sprint and that's a wonderful thing. You don't need them specifically for Scrum – you need the specifically for software development.

What does a Scrum Master do?

The Scrum Master is responsible for the Scrum process, for teaching the Product Owner and for teaching the team how to fulfil their roles within the Scrum process. They are also responsible for the changes within the organisation, to help the organisation optimise its ability to realise the benefits of Scrum, both in terms of productivity, in terms of quality and in terms of early realisation of benefit. So to the extent that an organisation is dysfunctional to building software quickly with quality the Scrum Master is responsible for going out and working with the organisation to make these changes happen. So they are a change agent.



What does a Scrum Master do on a normal day?

A Scrum Master does not have a normal day. As a minimum the Scrum Master attends the Daily Scrum; to listen, to hear what is going on, to see if the Team is self managing, to see if The Team is happy, to see if the Team is productive, to see if The Team is creative, to see if The Team is well on target toward meeting what they have committed to for that Sprint and if not to have The Team work with the Product Owner to add more work to the Sprint or to remove work from the Sprint so that they can meet their commitments.

Aside from that the Scrum Master is responsible for causing changes to happen within the Product Owner and the Team so that they optimally fulfil their responsibilities and roles. That is if the team doesn't know how to manage itself to teach them and help them understand how to, if the Product Owner doesn't know how to manage the Product Backlog to maximise the value of each increment, to teach the product owner how to. If the organisation doesn't know how to have one Product Owner to represent the entire group of stakeholders who are prioritising the Product Backlog for the team, to help them to figure out how to do so.

Above and beyond that, the Scrum Master is responsible for keeping any that would impede the progress of the team and the progress of the Product Owner out of way, and this means helping the organisation change so that they can take advantage of the benefits of Scrum. So this change process often fulfils much of the Scrum Master's day.

Can the Scrum Master be a developer or an architect?

The Scrum Master can come from anywhere – the Scrum Master is sometimes on the team of developers, the Scrum Master is sometimes the Product Owner, the Scrum Master can sometimes be the functional manager. Any role that the Scrum Master occupies in addition to just being a Scrum Master causes conflict, causes difficulty for the Scrum Master because there is a conflict of interest – a dual set of responsibilities they have. However this is unavoidable, so sometimes optimal Scrum Master is a lead engineer, this means that the lead engineer has signed up for work during the Sprint at the same time as they are committed to removing impediments for the team and if both of them take more time than is available it puts the Scrum Master / lead engineer into a conflict situation. However almost all Scrum Master's have conflicts: that's life.

Can Scrum be useful for non-software projects?

Scrum has been used for building hardware, for building embedded software, for managing public relations campaigns, for managing anything of complexity where we want to guide it to a successful or optimised outcome. Scrum is for managing complexity.

Do I need to stick to a month time frame?

The length of the iteration for Scrum is proposed and recommended to be a month long. The reason for this is that the Product Backlog is stated in terms of requirements; requirements are items that a customer can value as one being more important than another. If the Product Backlog is stated in terms of requirements that means during the Sprint we need to do analysis in addition to high level design, architecture, infrastructure, low level design, coding, testing and documentation and it tends to take about a month to do that, to complete 4 or 5 Product Backlog items – which tends to be the number that are selected every Sprint. This time has been empirically determined to be optimal across 15 years of experience using Scrum. If customers, if teams want to have longer or shorter Sprint lengths, we tend to discourage longer for a number of reasons. We tend to discourage shorter because that requires greater granularity of the Product Backlog items than requirements often into the level of specifications and then the question arises “where does the analysis occur?” and if it's occurring outside teams and being given



to them as documentation, then we run into the problem of lack of clarity and poor communications between the people who did the requirements and team that's acting on them.

The length of the time frame is stable; that is the size of the requirements may change, but the length of the Sprint is always the same once it's been fixed. This gives us some stability to the business in terms of always expecting to see an increment at the same time every month or every period, and also it gives the team a regularity that they know that, if they are the 10th day of the Sprint they should always be at a certain point. If you're always changing the Sprint length, then there is no regularity to them.

Why is everything in Scrum time-boxed?

To stop things from going on for unpredictable amounts of time; we want customers to be able to know that every month they can see the progress of their project. We want the team, once it starts going on a Sprint, that they know that they only have the length of the Sprint to complete their work. This puts pressure on them to not spend too much time doing analysis, not to spend too much time doing coding, but instead to have a regularity of operations so that they regularly deliver things of value to the customer. I heard a football coach once say that there wasn't a single that he would have lost if he hadn't run out of time. This is the same thing that is true of Scrum. We have these rules to give people a format to work within.

Why does Scrum say nothing about engineering practices?

Scrum is general purpose project management wrapper for complex projects. Engineering practices for software engineering are specific for their use in software development. Engineering practices for its use in building bridges would be very different to those for software development. When we use Scrum in software development projects we embrace and encapsulate the engineering practices from Extreme Programming because these encourage quality products; every iteration, every Sprint.

How do I measure the performance of the team?

You measure the performance of the team by its velocity, that is, its ability to turn requirements into increments of functionality in a Sprint. So its velocity is how many Product Backlog items per Sprint, or per \$100,000, or per 100,000 financial units, can it turn into an increment and that figure should be going up across time.

The primary measure we had seen used in the past was function points, however, function points require experts to measure them and they tend to be measured differently by the expert that's doing the measurement. When these were used we found that the industry average of function points per person, per month was 2. The optimal Agile team using all of the best Agile practices including co-location, XP engineering practices, managed 77 function points per person per month.

How do I measure the performance of a team member?

You don't – you only measure the performance of the team. It is a self-managing team that is seen as a unit, not as a group of individuals.

What is the benefit of doing a Sprint Retrospective?

Scrum is a process of self-improvement and process improvement – at the end of every Sprint the team gets together and talks amongst themselves about what things they liked, what things they didn't like and they are free to change what things they didn't like so that the next Sprint works better. The Sprint Retrospective is about looking back to see what they liked and what they didn't like and then modifying the process to fit their own needs.



How to fix a bug within a Sprint?

The only thing that the team works on within a Sprint are the Product Backlog items that they have selected and items that are specifically included as exceptions, for instance, it wouldn't be very wise for a team to refuse to fix severity 0 bugs, application failure situations, if they are responsible for maintaining existing products. We often tell a team to set aside time for to work on incidents like this if these types of situations occur. However, defects that aren't critical are not part of a Sprint unless specifically called for as part of the Sprint planning meeting.

What kind of reports do I supply to management?

There are many types of report you can provide, however the thing to remember at the start of implementing Scrum, that your customers and your management is used to seeing reports that are quite different from the ones given by Scrum. There has to be a transition period of them understanding and expecting to manage the project with these reports, to them expecting and understanding how they manage the project with Scrum reports.

The primary Scrum reports are baseline Product Backlog at the start of a Sprint and what the baseline Product Backlog is at the start of the next Sprint and a delta report that details why they are different, that is, what changes occurred between the start of one Sprint and the start of the next Sprint that caused the Product Backlog to be different. The other page that is part of this report is a Product Backlog Burndown which indicates the likely intersection of velocity of the team with the timeline based on the requirements and this is used for managing the anticipated end date and progress of the entire project.

If these 4 pages (baseline, baseline +1, delta report and Product Backlog Burndown) are collected and at the start of every Sprint and put into a manila folder and the project is n Sprints long, these n manila folders represent a completely auditable trail of the project.

Why would you cancel a Sprint?

You would only cancel a Sprint if you were unable to proceed and create anything of value during the Sprint. Instances of this happening are business conditions change in such a way that the Product Backlog items that you are working on no longer have value, which means that any work you have done on them, will have to be thrown away, as Scrum is a low ceremony, low documentation approach it's ok to throw it away as it has no value. Another instance of this is when the technology on which you've been trying to build this software proves untenable, however cancelling a Sprint should be the last thing you do and it's an act of desperation than a normal act that should occur.

If you are in synchronisation with other teams when you cancel a Sprint, the Sprint that you re-plan should be shorter so that you stay in synchronisation; that is, ending your Sprint at the same time as the other teams.

How can you get buy-in for the use of Scrum?

In general, if you start showing customers, the Product Owner, the stakeholders, deliverables at the end of Sprint and they don't show any interest in being engaged, looking at, collaborating with you about them, then it's probably not work using Scrum, nor is it worthwhile continuing with the project.

Can Scrum work without fulltime commitment from one or more of the stakeholders?

Scrum can work, of course, without the fulltime commitment of one or more business stakeholders, however they must be available to collaborate and drive at the start and end of the Sprint, they must be



available to the team on the nuances of the project during the Sprint. Sometimes they have to be available more because there's no one on the team with business domain knowledge, sometimes they don't have to be available hardly at all, because there's a lot of business domain knowledge within the development organisation.

Do Scrum teams need to be co-located?

A Scrum team is extremely high productivity if it's co-located, that is, the nuances of what everyone's thinking and working on is quickly communicated with minimal disruption. If people are not co-located, video conferencing, email, instant messaging – all those tools are good workarounds, they cut productivity but many, many teams Scrum and are not co-located; it's just the best they can do.

Doesn't Scrum mean we have to throw away the best IT practices we've developed over the last 20 years?

The IT project management processes and procedures that have been built up over the last 20 years are appropriate to the scientific management, Frederick Taylor, standard production line, predictive development where you have a repetitive production line. They are largely inappropriate to Lean Manufacturing and sometimes misleading to empirical process, inspect and adapt type of development which accepts and welcomes complexity and change. However the ones which are appropriate – keep them. Of course throw away the ones that are inappropriate as these certainly haven't lead to wild successes – these have led to our current problems.

What's the cost of change? Will the introduction of Scrum cause more problems than it solves?

To the extent that an organisation has difficulties delivering software and is dysfunctional at delivering software that's the degree of change that it has to go through to start delivering software regularly with improved ROI, with improved quality, with improved joy. Sometimes that transition is massive, painful and difficult. Sometimes it's relatively simple. Either way an organisation will only go through it if it has commitment to being a world class, high quality software organisation, is adequate to go through the degree of pain and change necessary. Of all organisations that attempt to become a world class software organisation only 1/3 will make it.

Is there a downside to changing the make up of a team after the Sprint has started?

Any change to a team once the Sprint has started causes problems. Sometimes the problems that it causes are less than the problems by not changing it; then it is worthwhile. However you should be totally aware and prepared for the difficulty that the new person / people on the team are going to have being introduced into the existing team. Just think of the problems associated to a child when its family moves town and the overhead that had to happen for that child to fit in.

How do you scale Scrum to larger projects?

The largest team / project that Ken has personally scaled Scrum to, is 3400 people overall and it lasted 2 ½ years. The techniques for doing this are not dissimilar for scaling any other project and the engineering practices for doing so, the differences is that the levels of management doing so are in the Scrum context.