

# Script for interviews

Version 3.0

## Introduction:

**Scope:** *We are interested in your Scrum process and how it is supported by artefacts.*

**Time:** *about 1 hour*

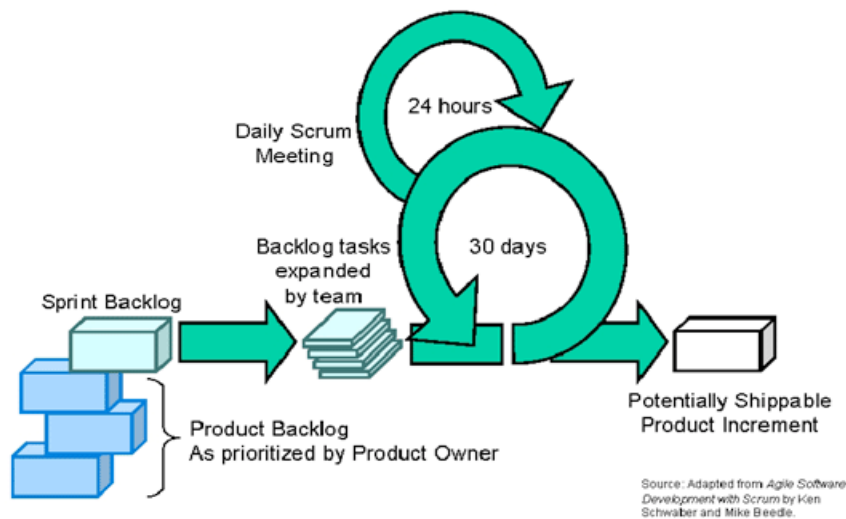
## Facts:

- How many years of scrum experience do you have (exact number +-1year)
- Job title
- How many projects?
- Many projects available □ choose one to talk about: current one or longest one or pick the one they prefer to talk about. Choose the most relevant for tools / last one (for actual informations). Ask for project duration.

## Scrum Diagram:

First of all I intend to understand the perception of Scrum you have in your team

and to find out how your development process is organized.



*(drawing theoretical version of the Scrum diagram on a whiteboard, by hand, black color)*

This is the Scrum process as we know it from theory. There is a product backlog from which the user stories for the current sprint are chosen and will go to the

sprint backlog. This is the sprint planning (participants: Scrum Master, Product Owner, Team).

There will then be a sprint in which the chosen features will be implemented and in the end of each sprint there will be a potentially shippable product increment. This sprint in theory has a fix duration of 30 days. A daily stand-up meeting called Daily Scrum Meeting (Scrum Master, Team) takes places, usually in the morning. At the end of each sprint, there will be a meeting with Scrum Master and the developers team in order to learn from faults committed with regard to the Scrum process (Scrum retrospective). Also there will be a sprint review (Team, Scrum Master, Product Owner).

What other meetings not yet mentioned in the diagram do you have in your process? Please add them to the diagram.

What are the roles you are using in your process? In which meetings do the different roles participate? What are the people of the different roles doing? Who is making decisions in which phase of the process?

*(In theory we have the well-separated roles of Scrum Master, Product Owner and the developers team.)*

How are non-functional (or not completely functional) requirements handled? Are there special stories, that are different from the usual user stories?

Knowledge sharing is an important factor of Scrum. How do you share knowledge in your process? Can you indicate in the diagram where the knowledge transfer is in your process? *{Backup: Compare other development processes, you have worked with in past, with Scrum. What are the similarities with regard to knowledge transfer? What are the differences to Scrum with regard to knowledge transfer? What are the benefits of Scrum with regard to knowledge transfer / communication?}*

*(New color blue on the whiteboard for artefacts)*

What artefacts do you use to support the different stages of the process?

*{Backup: How do you use the whiteboard/walltools/post-its? Do you use different tools for different aspects/stages?}*

□ for each artefact:

- Which version of the tool do you use?
- What are the points you like about this artefact? What are the points you do not like about this artefact?
- Which role does what in the tool?
- How did you adapt and/or configure the tool?
- Were you involved in the selection process of this artefact? If yes, how did you select this artefact?
- How does the team use the tools differently than the business people? Does

the tool handle these differences?

Is there data tracked in other artefacts (for example SVN, Post-its on the wall)?

If there are many artefacts/tools, let them rate each of them from 1 to 5.

*If there have a wall and a tool: How much overhead for synchronization of the wall and the tool is acceptable?*

*{Backup: Indicating one by one the different stages and asking in particular how the stage is handled}*

How do you handle versioning/ update of documents? How do you link information between different sources? Was the same information replicated in multiple places?

How do you differentiate between project level and product level? / How is long-term knowledge managed for the project?

How is the process visible to the developers/team? Does the tool handle this visibility or is it a need?

How does change management happen? How do you keep track of the decisions? How much reporting (*e.g. burndownchart*) was necessary to drive the project?

If using Scrum for more than 2 years in this team: Compare the process of the first year to the process of the current year. What are the differences? How did teamwork improve or progress over the project's duration?

What are the points you were able to improve? How did you refine the process from the first year to the current year?

*{Backup: What are the lessons learnt? Did you abandon some tools that didn't satisfy the needs? If yes, which needs they did not fulfill?}*

Indicate the painpoints of your process on the diagram (*red color*).

*{Explain what the problems are there.}*

Indicate the positive points on the process diagram (*green color*).

*{Explain how come they are perceived as positive.}*

What are your likes of the process diagram (*green color*)?

What are your dislikes of the process diagram (*red color*)?

How would you represent bug reporting in the process diagram? How is bug fixing implemented in the tool / process?

How do you estimate the stories? Did different estimation techniques improve the velocity?

*{Backup: Do bugs enter as new features in the product backlog?}*

How does the Scrum project fit in a bigger context? (*e.g. part of a bigger project, or long-term corporate projects*)

What could be automated, what was done manually (testing, report generation, etc.)? What would be good/you would wish to be automated?

How much discipline was necessary in the team? Would the productivity be improved if the people had more discipline?

What is your personal opinion about integrated platforms as for example Team Foundation Server (open question)?

**Final part:**

Do you have any more questions for us?

Can we contact you by e-mail for further questions?

Thank you very much for your time and your contribution.