System Development Management

Lecture 3: Scrum

Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

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

That is, while there is value in the items on
the right, we value the items on the left more.

Starting with scrum

- No long preparations. It’s a philosophy and a mindset you can adopt directly.
- Accept that functionality is flexible which can not defined at forehand in detail. Accept that flexibility requires frequently feedback.
- Scrum is not only a change in the development process but also in the management of software development. It’s a change in culture. Especially for the development team which gets more own responsibility.
- Start working with scrum in the beginning according to the book. Later on, when you have more experience, you can adapt it to your own situation. No Scrino, use Agile/Scrum coach!!

Characteristics

- Embrace change (in contrary to traditional Project Management)
- Short development cycle (2-4 weeks), a fixed completion time, called a *sprint.*
- Development team is fully flexible for future sprints, but totally inflexible for current sprint.
- Feedback is essential to develop the right product. The earlier, the better.
- In general: you should practice the things you find difficult. The same for releasing software; do it as often as possible.
- Since a sprint is a fixed short planning cycle you will become better and better in planning the amount of work you can do. So you will keep your promises to the customer better.
- Choose per sprint the functionality which is most essential delivering the most added value to the customer.
- The customer is the one who can tell what functionality delivers the most added value. Therefore, the customer should be available for contact (i.e. being the product owner).

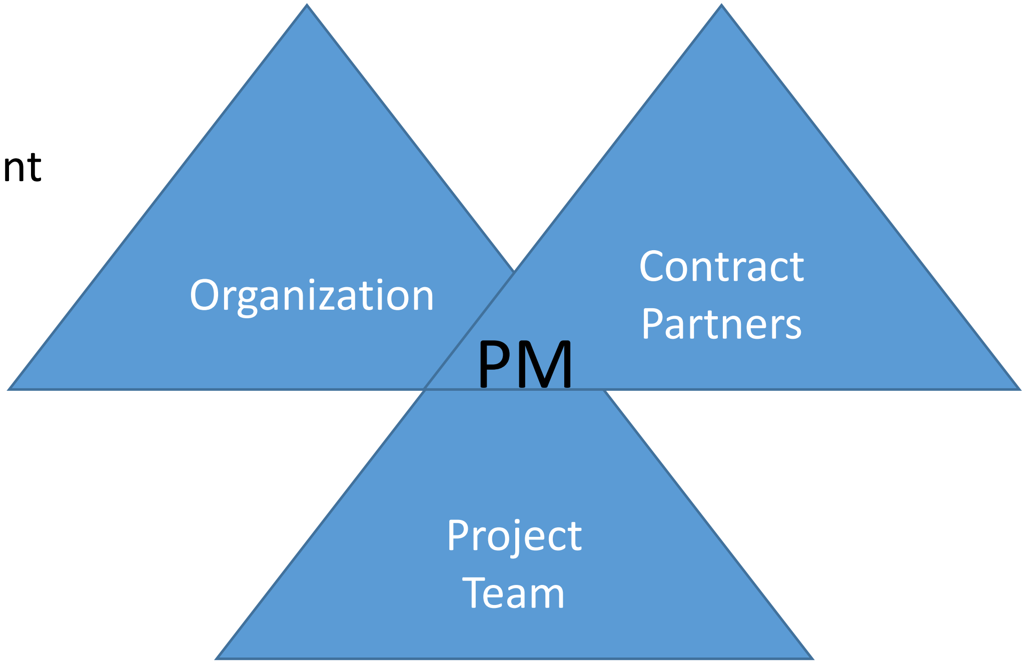
Roles and responsibilities

- The *development team* is responsible for the result and therefore manages itself. The team choose there own working process and tools. The team decides how to make the product. The team is also responsible for planning, progress and reporting of a sprint.
- The *product owner* prioritizes with all stakeholders what work the development team has to do in a sprint. Furthermore de product owner makes a list of future sprints called the *product backlog*. The product backlog is prioritized on added value.
- The *scrum master* is responsible for the sprint process (that it is efficient and effective). A scrum master is not the boss.
- The scrum process itself does not require a project manager. A project manager can be helpful to manage the external environment and to coordinate reports, steering committees, intakes and escalations.

Focus of project manager within traditional project?

- Focus of project manager should not only be on the project execution but also on the environment.

- Board of Management
- Line management
- Staff
- Users



Organization

Contract Partners

PM

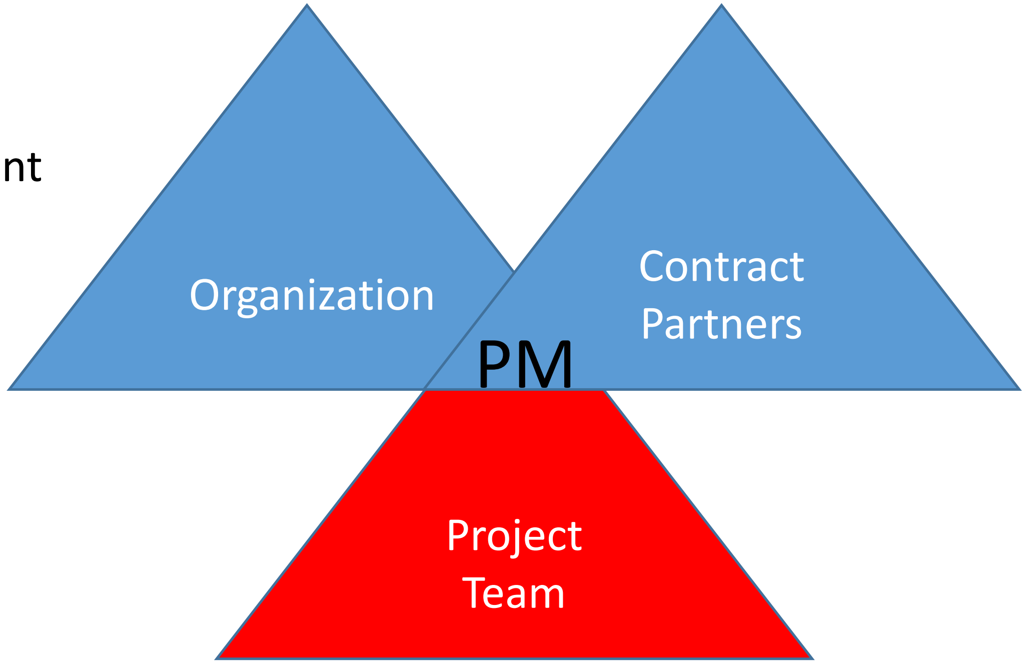
Project Team

- Supplier
- Financial partners
- Consultants
- Unions

Focus of project manager within scrum team?

- Focus of project manager should not ~~only~~ be on the project execution but ~~also~~ mostly on the environment.

- Board of Management
- Line management
- Staff
- Users



- Supplier
- Financial partners
- Consultants
- Unions

The scrum process

- After each sprint a working product is released.
- A scrum team exists of 5 – 9 persons so you can work in one room together.
- A sprint start with discussing the *sprint backlog* (the work to be done in this sprint). The team should be question the product owner until it has answers to all the questions they have so they know exactly what is expected from them.
- In the *sprint planning meeting* the sprint backlog is split in separate tasks called *work items*. A work item describes a demonstrable result which can preferably be achieved in 2-4 hours but never more than 2 days. If it is hard to split work into smaller items it often means that it is not clear enough what should be achieved.
- The scrum team divides the work and delivers a achievable planning. Each member of the team gives his/her commitment.
- The scrum master organizes a daily stand up with three questions for each participant: (1) what did you achieve yesterday? (2) what will you achieve today? (3) what is blocking you?
- An *impediment* is a blockade for the team to achieve the result of the sprint. The scrum master is responsible for removing this blockade before the next stand up.

Planning and adjustment in scrum

- *Velocity* is the amount of work a team can do in one sprint. During each sprint the velocity is measured. Velocity is used to estimate if the desired work is achievable in a sprint.
- *Planning Poker* is an estimation technique in scrum. It’s based on the consensus within the group on the duration of time of an amount of work. Discussion during the planning poker is very important and the arguments of each other give you inside the work to do.
- By comparing the estimation time to the real development time helps the team to become more predictable in future sprints. The *burn down chart* shows the hours “burned” in relation to the amount of work done.
- The scrum board is used for mutual adjustments. On the scrum board you can see in one eyesight work ToDo, Busy and Done. Work items are put on the board from highest priority to lower priorities. The highest priority items are those which are worked on as first.

Definition of Done

1. The software is tested.
2. The software meets the standard of coding.
3. The software is reviewed by (at least) 1 colleague.
4. The help file and user manual is adjusted.
5. The software can be deployed directly.

Finishing the sprint

- The *end of sprint* demo or *review meeting* is a live demonstration of the current status of the product. It is a real product. Not a PowerPoint nor a UI demo.
- The product owner has to give the team decharge of the sprint.
- As a scrum team you want to improve constantly. Therefore, the *retrospective* is a very important meeting in which the team evaluates the sprint defining measures for what went wrong and keeping what went right. The retrospective leads to concrete actions leading to structural improvements in the development process. The retrospective should not be a ritual dance!

How to manage a scrum team

- Give the team the confidence and space to organize themselves. Provide them the boundaries within they should work.
- Adjudge the team there own learning process. Do not intervene!
- Make sure that all the results are transparent all the time.
- Manage on the value for the customer.
- Be an example for the team by working transparently on your backlog.
- Help the team by removing obstacles and roadblocks.

Pitfalls for working with Scrum

- Seeing mistakes and failures as something bad and wrong.
- Not acting after you have a painful learning experience.
- Not starting with something if you believe it is difficult.
- Being afraid to make mistakes or punishing making mistakes
- Hiding mistakes so nobody can learn form it.
- Using transparency and openness against people.
- Managing Scrum teams via command and control.
- Not measuring the benefits of Scrum because its seems difficult.
- Only measuring how “happy” everybody is.
- Implementing Scrum only half.
- Adjusting Scrum before using Scrum.
- Accepting that a piece of work can not be split in more details.
- Starting a sprint with to much work to end with an executable product.

Literature

- De kracht van Scrum – Rini van Solingen & Eelco Rustenburg (in Dutch)
- De Bijenherder – Rini van Solingen (in Dutch)
- Scrum voor managers – Rini van Solingen & Rob van Lanen (in Dutch)
- Agile: geschikt/ongeschikt – Fred Heemstra, Luuk Ketel & Erik van Daalen (in Dutch)
- Agile for Project Managers – Denise Canty