System Development
Scrum 1

Datamatiker /Computer Science
2nd Semester

Spring 2017

Agenda for Scrum Days

Scrum is a management framework that describes how teams can work together to develop a product

Scrum Day 1

- Overview of Scrum

Scrum day 2

- Product backlog (PBL)

Scrum Day 3

- Sprint planning

Scrum Day 4

- Scrum tools & work on PBL for Fog project

Scrum Day 5

- Group review & discussion of PBL version 1

Learning Objectives for Scrum

- Knowledge of Scrum basic process model
  - How to document and estimate customer requirements
  - How to turn these requirements into an operational format the developers can use to control their daily work
  - How to monitor and manage the development effort
  - How to calculate team velocity, meaning how much work a team can handle in time-boxed period
  - How to work in an iterative manner where software is build piece by piece

Main literature

Henrik Kniberg *Scrum and XP from the Trenches*

https://www.infoq.com/minibooks/scrum-xp-from-the-trenches-2

Pages : *pp. 1-13* *(day 1)*
pp. 14-50 *(day 2)*
pp. 51-68, 75-92 *(day 3)*

How to Develop an IT System?

Traditional waterfall project example
To build a house!

Phase 1 – idea/analysis

Phase 2 – design

Phase 3a – fundament

Phase 3b – walls

Phase 3c – root

Phase 3 – construction

Phase 4 – test

Traditional Waterfall vs. Iterative Approach

General comparison of two methodology paradigms

Comparing Waterfall To Iterative

Waterfall	Iterative
Risk averse	Actively attacks risk
Subjective measurement of progress	Objective measurement of progress
Delays integration and testing	Continuous integration and testing
Nothing runs until the end	Something "runnable" produced every iteration
Difficulties at the end of the project	Difficulties at the start of the project

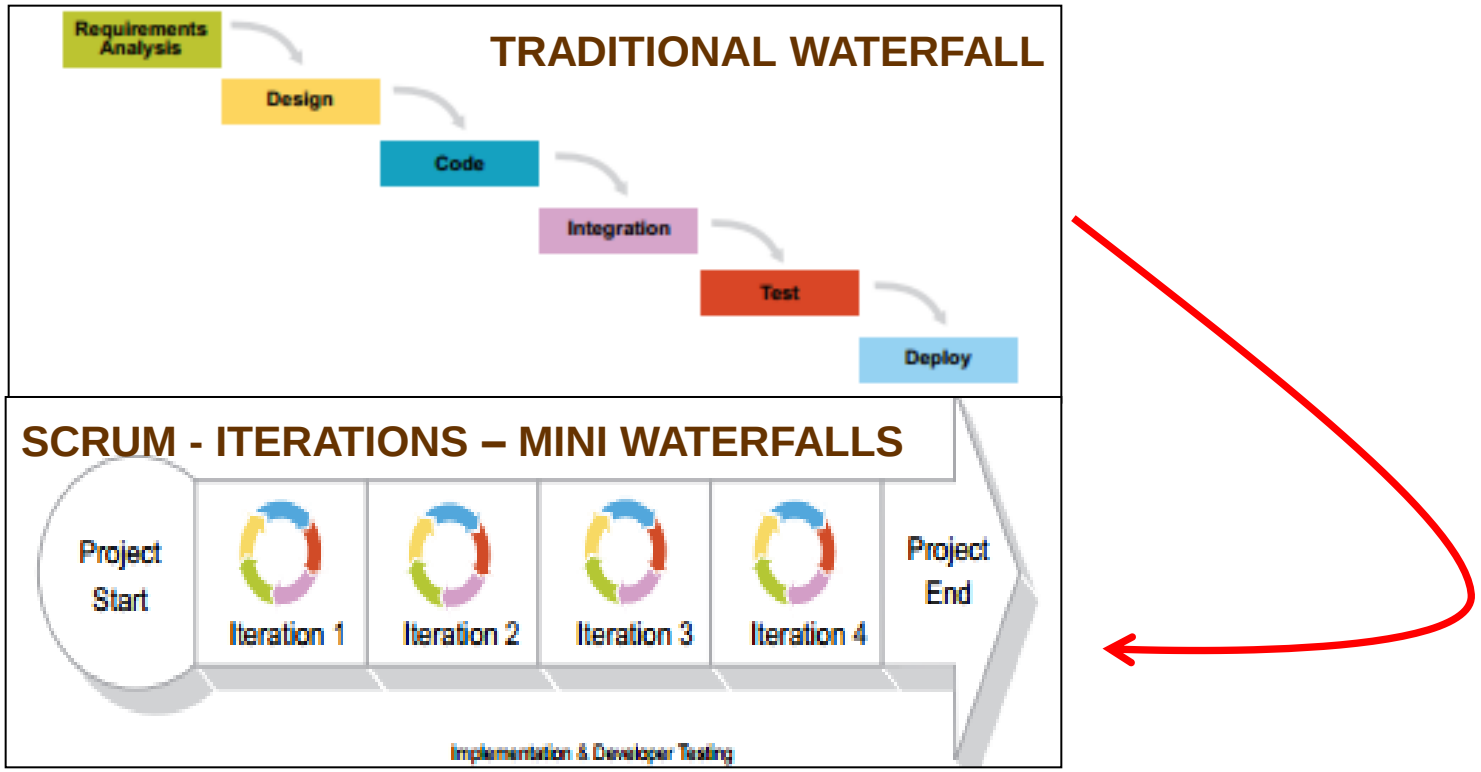
Traditional "waterfall" development depends on a perfect understanding of the product requirements from the beginning and minimal errors made in each phase.

Source: http://www.ibm.com/developerworks/rational/library/4029.html

Scrum in a Nutshell

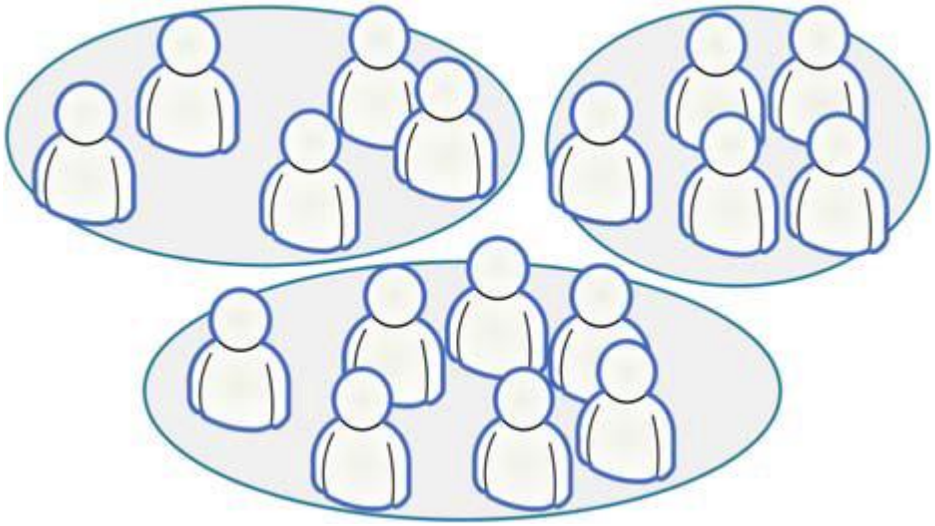
Scrum in a Nutshell 1

- The Scrum is iterative process
  - Many small water falls, usually called **sprints**



Scrum in a Nutshell 2

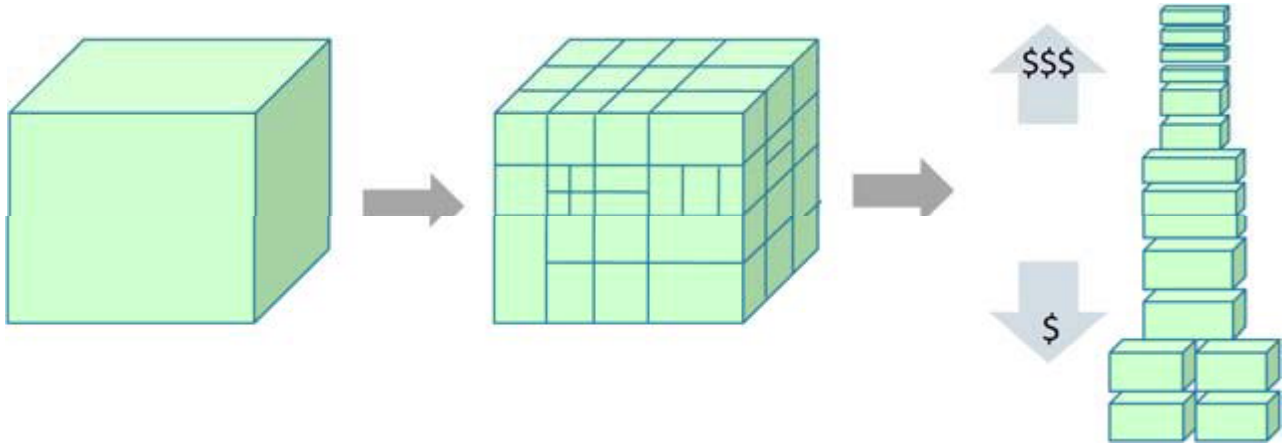
- Split your organization into small, cross-functional, self organizing teams.



Source: Kniberg ” KANBAN AND SCRUM – MAKING THE MOST OF BOTH”

Scrum in a Nutshell 3

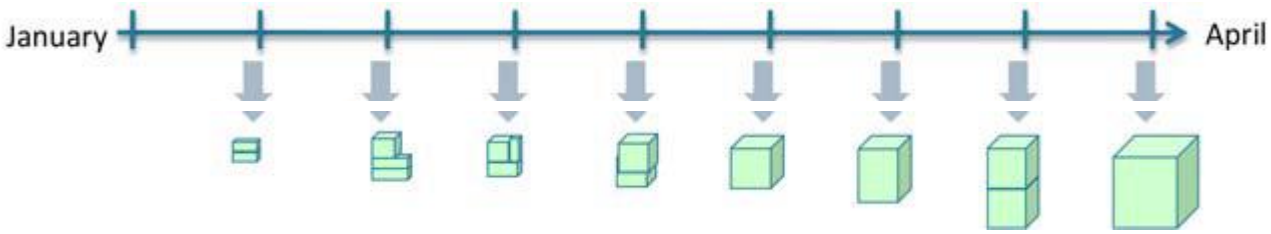
- Split your work into a list of small, concrete deliverables.
  - Sort the list by priority
  - Estimate the effort of each item



Source: Kniberg " KANBAN AND SCRUM – MAKING THE MOST OF BOTH"

Scrum in a Nutshell 4

- Split time into short fixed-length iterations (usually 1 – 4 weeks), with potentially shippable code demonstrated after each iteration.



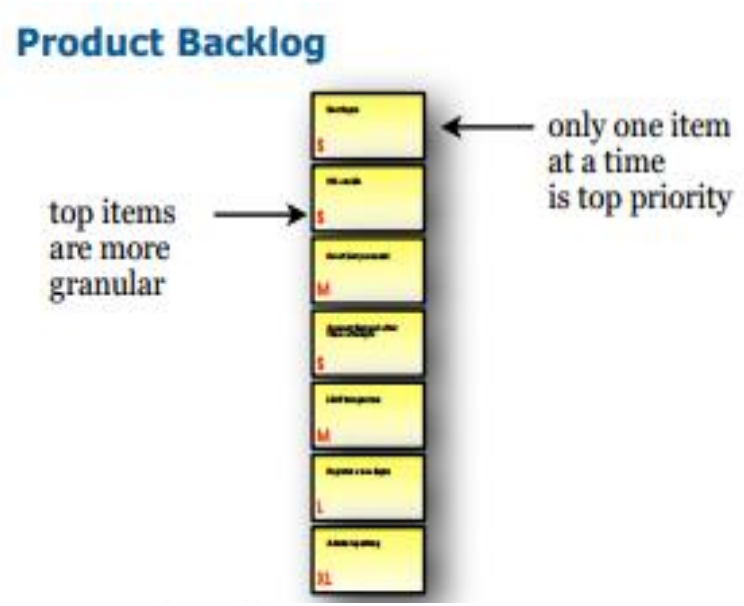
Scrum in a Nutshell 5

- After each iteration …
  - Optimize the release plan and update priorities in collaboration with the customer, based on insights gained by inspecting the release
  - Optimize the process by having a retrospective after each iteration.

Source: Kniberg ” KANBAN AND SCRUM – MAKING THE MOST OF BOTH”

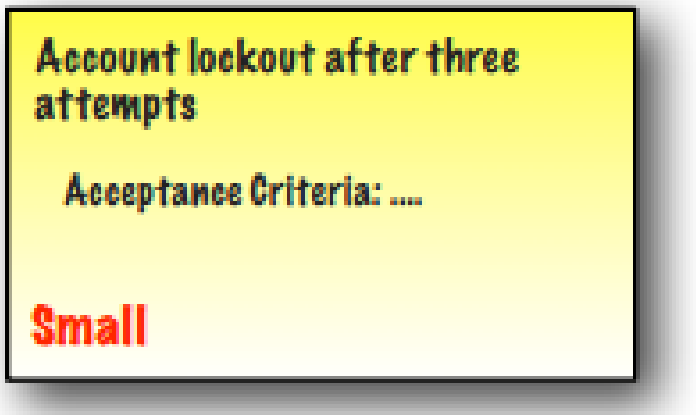
The Product Backlog

- A prioritized list of everything that might be needed in the product
  - requirements, features etc.
  - Things that the customer wants, described using the customer's terminology.



Product Backlog Item

- Often called (user) story, or just PBI.
- Example:



User Story

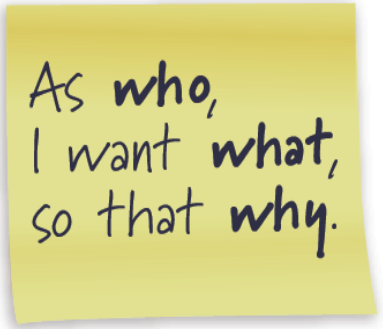
- … is short, simple description of a feature told from the perspective of the person who desires the new capability (typically user or customer)

- User stories can be written informally:

  Registered users can reset their password

- Or use a more formal template

  As a registered user,
  I want to reset my password,
  so that I can get back into the site if I forget my password

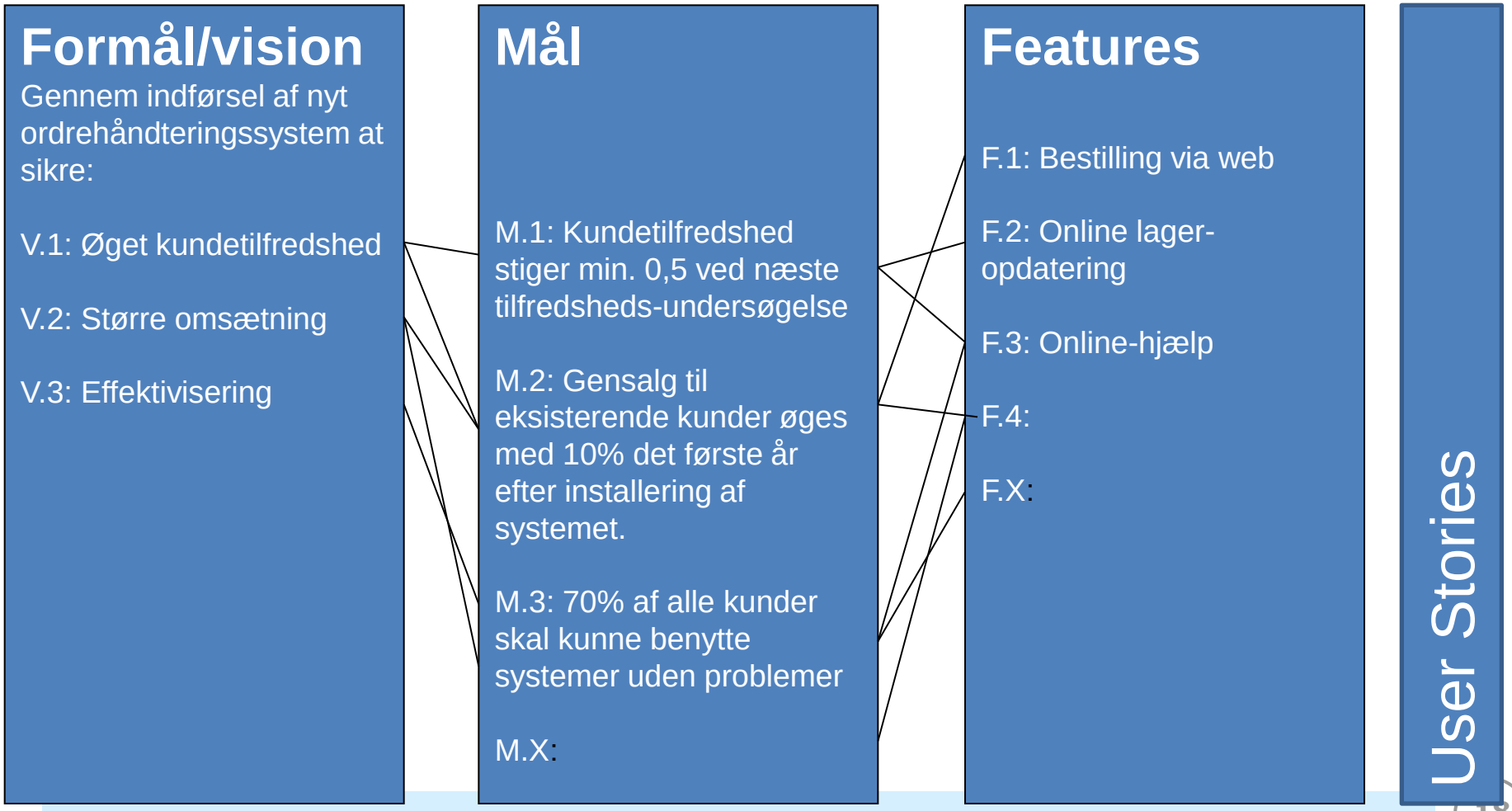


Product Backlog Example

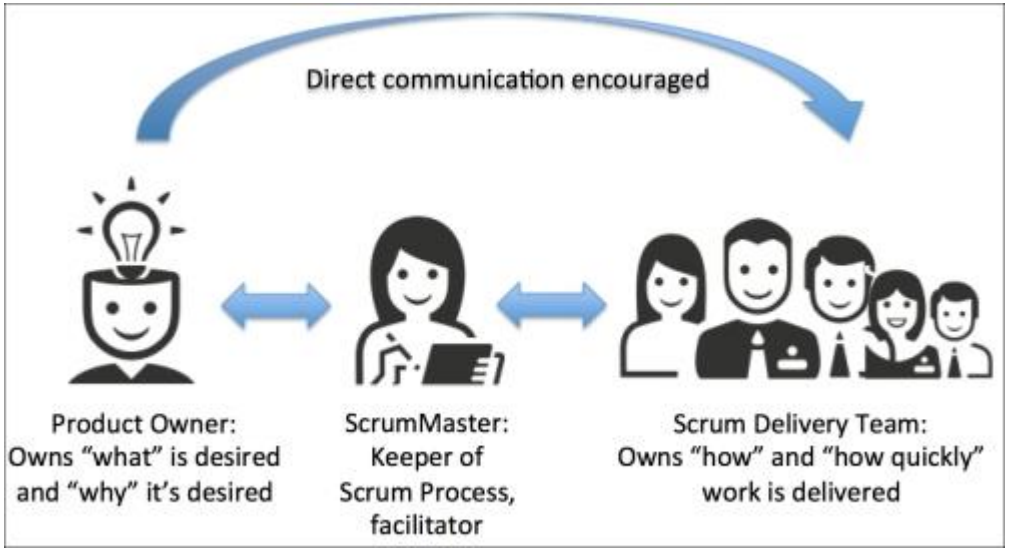
- Story example where feature description is specified in "How to demo" field, i.e. test steps description (acceptance criteria) (Kniberg p. 10)

PRODUCT BACKLOG (example)					
ID	**Name**	**Imp**	**Est**	**How to demo**	**Notes**
1	Deposit	30	5	Log in, open deposit page, deposit €10, go to my balance page and check that it has increased by €10.	Need a UML sequence diagram. No need to worry about encryption for now.
2	See your own transaction history	10	8	Log in, click on "transactions". Do a deposit. Go back to transactions, check that the new deposit shows up.	Use paging to avoid large DB queries. Design similar to view users page.

The traceability model & scrum!



Scrum Roles


Direct communication encouraged
Product Owner:
Owns "what" is desired
and "why" it's desired
ScrumMaster:
Keeper of
Scrum Process,
facilitator
Scrum Delivery Team:
Owns "how" and "how quickly"
work is delivered

Scrum Role Responsibilities

- **Product Owner**: responsible for the business value of the project
- **Development Team**: self-organized to get the work done
- **Scrum Master**: ensures that the team is functional and productive

Product Owner

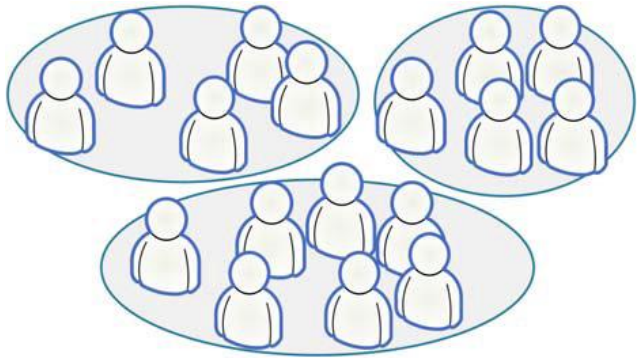
- Represents the stakeholders (= customer voice)
- Is responsible for maximizing product value
- Is responsible for managing the PBL:
  - Create Product Backlog items (user stories)
  - Prioritize Product Backlog items
  - Ensure the teams understands items

Scrum Master

- The Scrum Master is the process owner
  - responsible for ensuring Scrum is understood and enacted
  - Helps the team perform at their highest level (coach)
  - Protector of the team

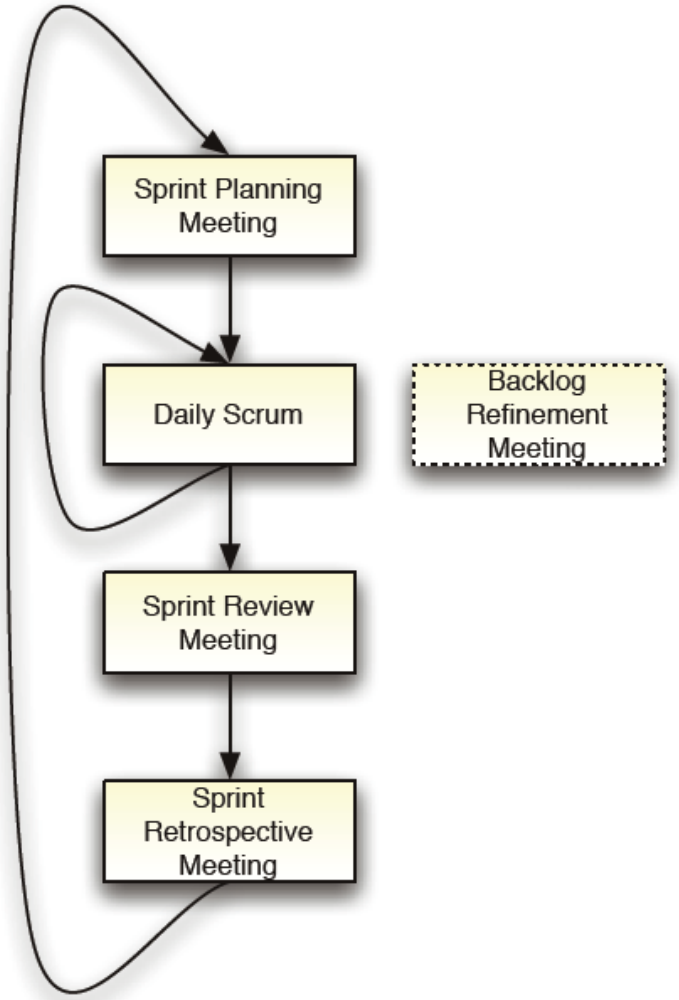
Scrum Team

- Cross functional
- Self-organizing
- Negotiates commitments with the Product Owner, one sprint at a time
- Has autonomy regarding how to reach commitments
- Collaborative
- Co-located
- 7 ± 2 members



Scrum Activities

- Scrum meetings



Scrum in a nutshell

- Let's "attend" a backlog refinement meeting by watching a video (13 minutes++):

  http://scrumtrainingseries.com/

- We will see Product Owner, Scrum Master and Team in action!

Agile Product Ownership in a Nutshell

- Home work: Watch 15 minutes video by Henrik Kniberg

  http://blog.crisp.se/author/henrikkniberg

Team Contract Work

- Make team contract
- Use ”TEAM CONTRACT” template on github for inspiration