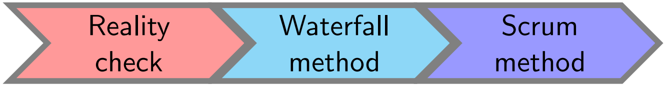
Software Development Part II—Scrum

Rasmus Dahlberg, Eivind J. Nordby, Martin Blom, and Tobias Pulls

Learning outcome based on syllabus

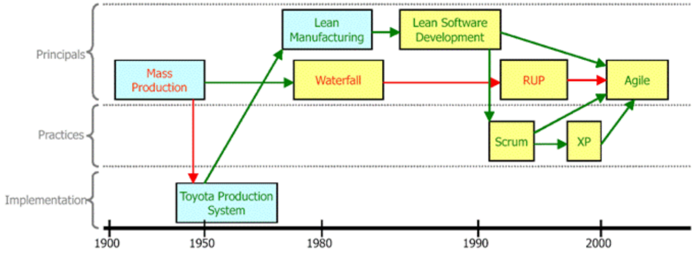
- Give an account of different ways to develop software (ISGA01)
- Describe different ways of developing software (ISGA06)
- Explain the development process of an information system (ISGA90)



Reality check

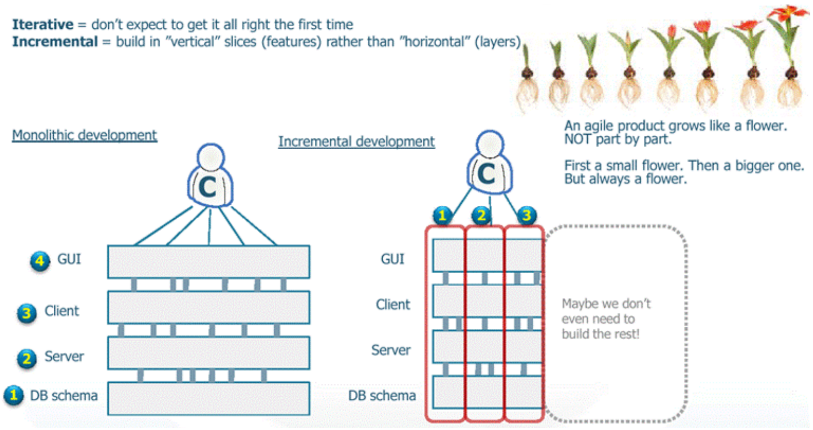
- Three things we wish were true
  - Customers know what they want
  - Developers know how to build it
  - Nothing changes at the course of a project
- Three things we have to live with
  - Customers figure out what they want
  - Developers figure out how to build it
  - Many things change at the course of a project

Timeline of different approaches towards product development



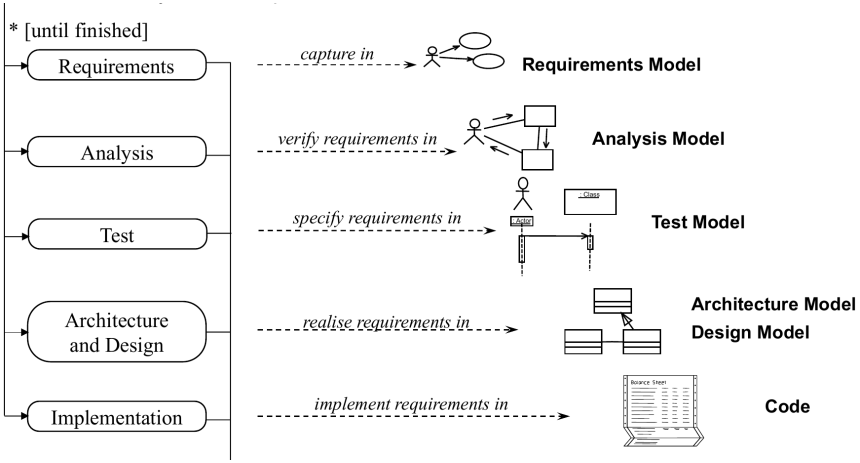
Our scope: Agile and Scrum, briefly Waterfall and XP

Monolithic vs Iterative and incremental


Iterative = don't expect to get it all right the first time
Incremental = build in "vertical" slices (features) rather than "horizontal" (layers)
Monolithic development
Incremental development
C
4 GUI
3 Client
2 Server
1 DB schema
GUI
Client
Server
DB schema
1
2
3
An agile product grows like a flower. NOT part by part.
First a small flower. Then a bigger one. But always a flower.
Maybe we don't even need to build the rest!

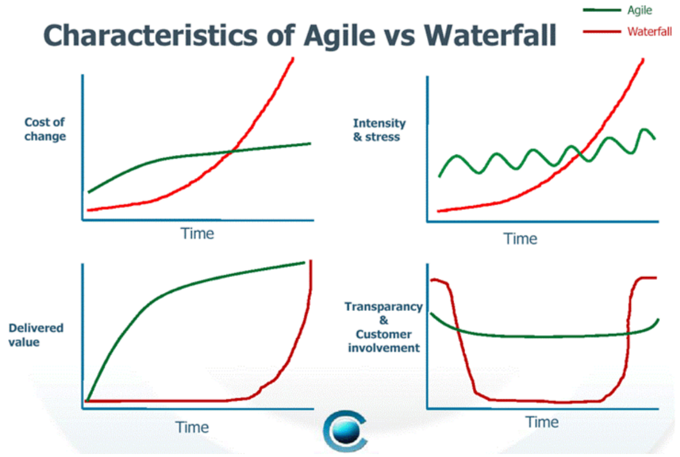
Agile development is all about feedback cycles

Each iteration is a mini project that involves all diciplines

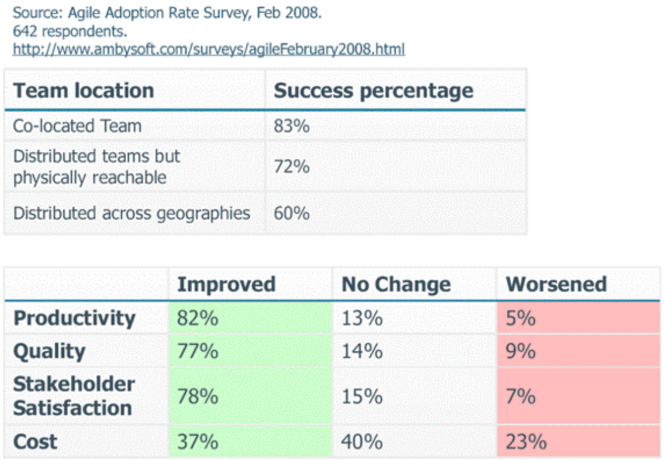


Note: not necessarily in this order!

Agile vs Waterfall



Does it work? Results from a survey on agile software development



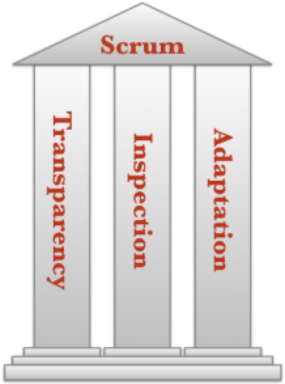
Source: Agile Adoption Rate Survey, Feb 2008.
642 respondents.
http://www.ambysoft.com/surveys/agileFebruary2008.html

Team location	Success percentage
Co-located Team	83%
Distributed teams but physically reachable	72%
Distributed across geographies	60%

	Improved	No Change	Worsened
Productivity	82%	13%	5%
Quality	77%	14%	9%
Stakeholder Satisfaction	78%	15%	7%
Cost	37%	40%	23%

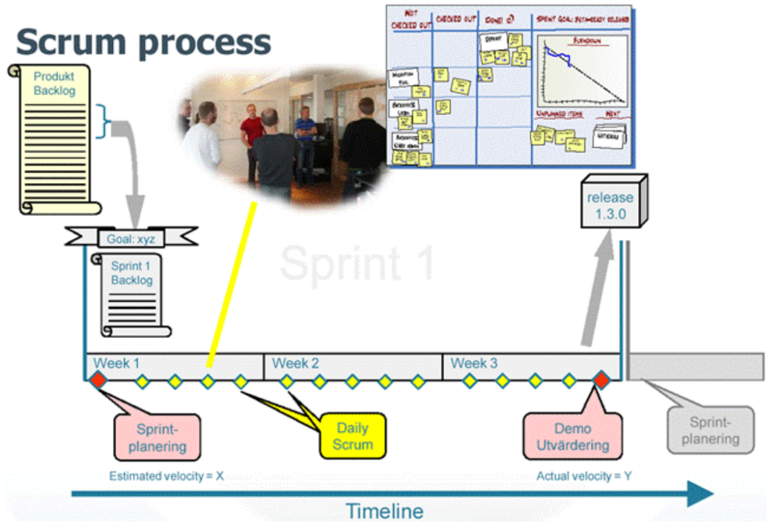
Scrum properties

- Emperical—progress based on real-world observations rather than fictious plans
- Identify problems early
- Prioritize strictly
- Plan for change and continuous improvement
  - Short feedback loop
  - Ship working software frequently
  - “Planning is needed, but always wrong”
- Cross-functional and self-organizing teams
- Pull-scheduling
- Timeboxing
- Simple tools
- ...



https://www.scrum.org/resources/blog/three-pillars-empiricism-scrum

An overview of the Scrum process



Roles:

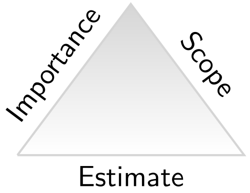
- Product owner
- Scrum master
- Developer

A typical sprint

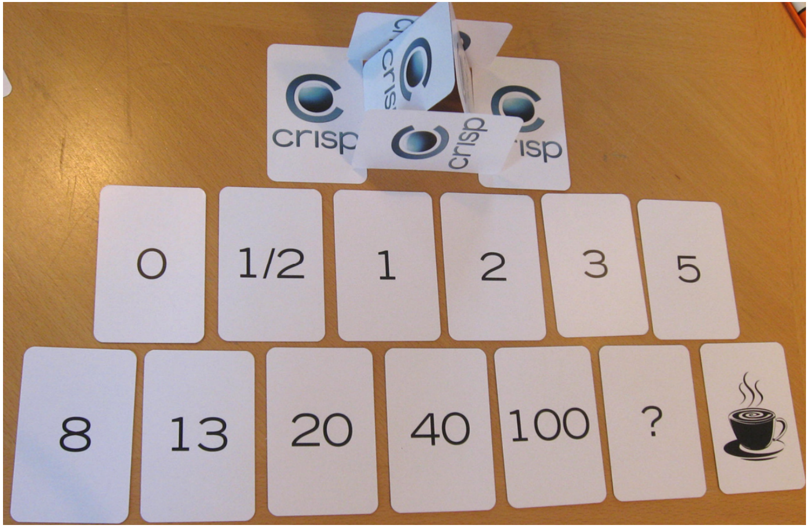
	Timeboxed
Sprint planning	
▶ Update and prioritize features in the product backlog	4h
▶ Add top-priority features to sprint backlog and divide into tasks	4h
Sprint execution	2–4 weeks
▶ Daily scrum—a short stand-up meeting	15m
▶ What did you do yesterday?	
▶ What will you do today?	
▶ Any problems?	
Sprint review	4h
▶ Team holds a demo for product owner and stakeholders	
Sprint retrospective	2h
▶ The good and the bad?	
▶ How can we improve as a team?	

Sprint planning—what should we work on the next couple of weeks?

- Dialog between product owner and development team
  - PO: present&adapt priority features in product backlog
  - Team: how much can be done
    - Story points
    - Sprint velocity
    - Poker estimates are common
- Concrete output of this meeting?
  - A sprint backlog and definitions of 'done'
  - A set of tasks for each feature in the sprint backlog
  - A sprint goal, a demo date, and how to demo
  - A time and place for daily scrum

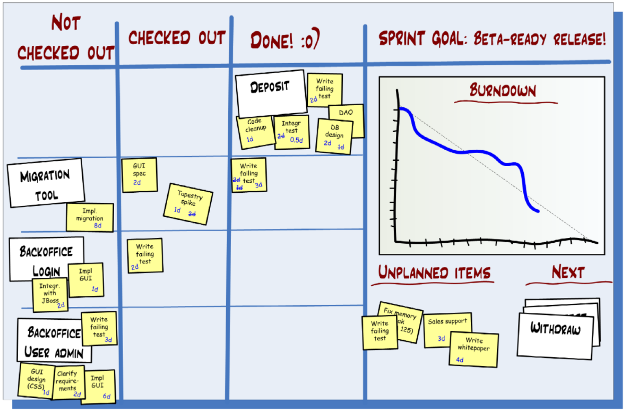


...using cards!



Daily scrum—a short stand-up meeting that repeats every day

- Purpose—keep **team members** up-to-date
- What did you do yesterday?
- What will you do today?
- Any problems?



A board is used to track progress

Sprint review—show-case the latest prototype and start a dialog

- Date and time already defined—**unconditional**
- All roles attend, including stakeholders if invited
- A demo of the prototype shows that the sprint goal is achieved
  - How to demo? Sprint planning...

Wait w000t: what if we are not done?
This is identified early on and solved accordingly!

Sprint retrospective—ensure that the team improves internally

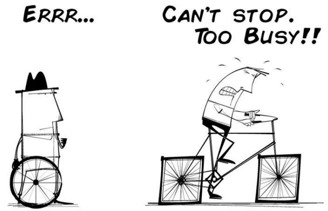
- Product owner is excluded from this meeting
- The goal is to improve the team productivity
  - What did we do right?
  - What did we do wrong?
  - How can we improve? **Choose one!**

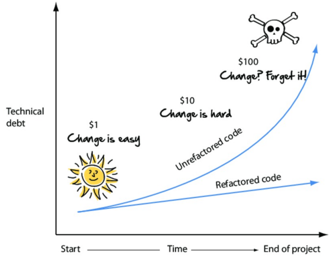
Make lists and perhaps magnet vote

What does ‘done’ really mean?

- **Can be delivered to the customer**
  - A given feature is implemented
  - Code follows good engineering practises
  - Code is documented and refactored
  - ...or anything else defined at sprint planning
- **If your estimates turn out to be wrong**
  - Work harder, longer and/or smarter
  - Lower quality by skipping design, testing, integration and/or documentation
  - Reduce and/or remove features
  - What are the pros and cons?

Consequences of the ‘hurry-up’ and ‘lowered-quality’ approaches

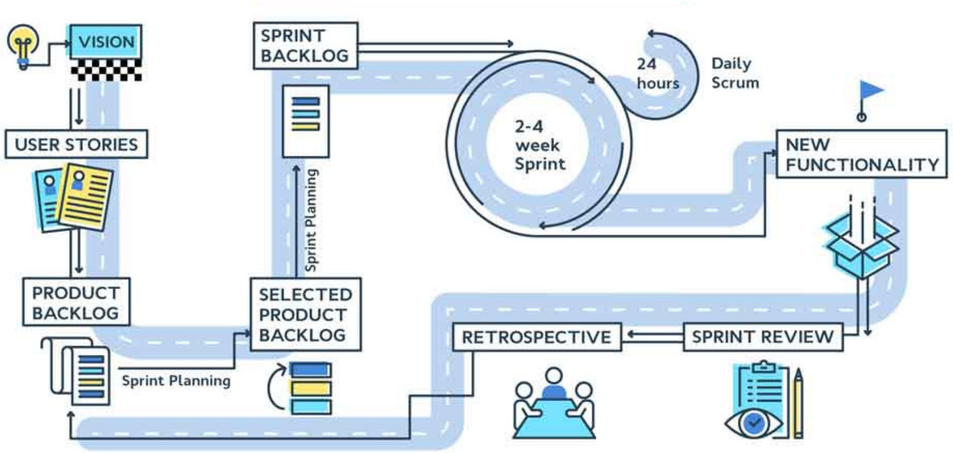




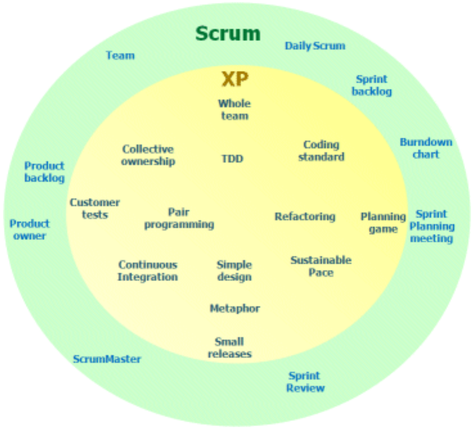
- **Hurry-up**—work overtime, skip breaks, add more people, ...
  - Burnout
  - Errors
  - ‘More junk in short time’
- **Lowered quality**—leads to technical debt and thus reduced efficiency
  - Harder to re-use code
  - Harder to add functionality
  - Harder to meet future goals

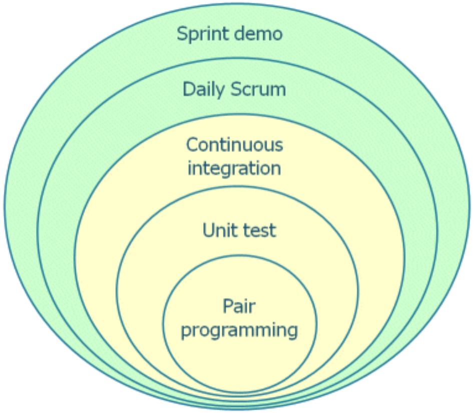
In other words: involve product owner and go with option three

From start to finnish—putting it all together



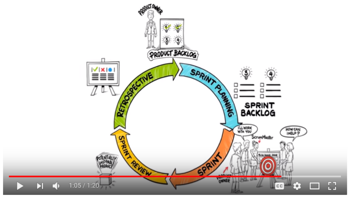
Scrum and eXtreme Programming (XP)





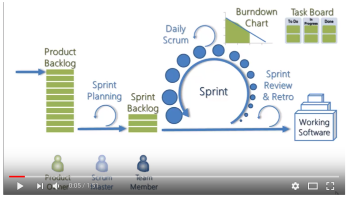
- Scrum can be viewed as a team-to-stakeholder interface
- The team is self-organizing, but it **could** work using XP practises

Can't get enough? Review the concepts or dig into the details (Optional)



What is Scrum?

https://www.youtube.com/watch?v=TRcReyRYIMg

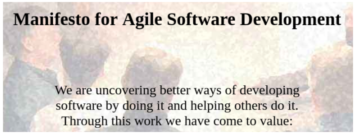


Explaining Scrum in less than 120 seconds

https://www.youtube.com/watch?v=WxiuE-1ujCM

#axosoft #agile #scrum
Intro to Scrum in Under 10 Minutes

https://www.youtube.com/watch?v=XU0llRltyFM



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:

http://agilemanifesto.org/

The Scrum Guide™

The Definitive Guide to Scrum: The Rules of the Game

https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf

An agile war story

Scrum and XP from the Trenches

How we do Scrum

http://wwwis.win.tue.nl/2R690/doc/ScrumAndXpFromTheTrenchesonline07-31.pdf

Any questions?