# Software processes

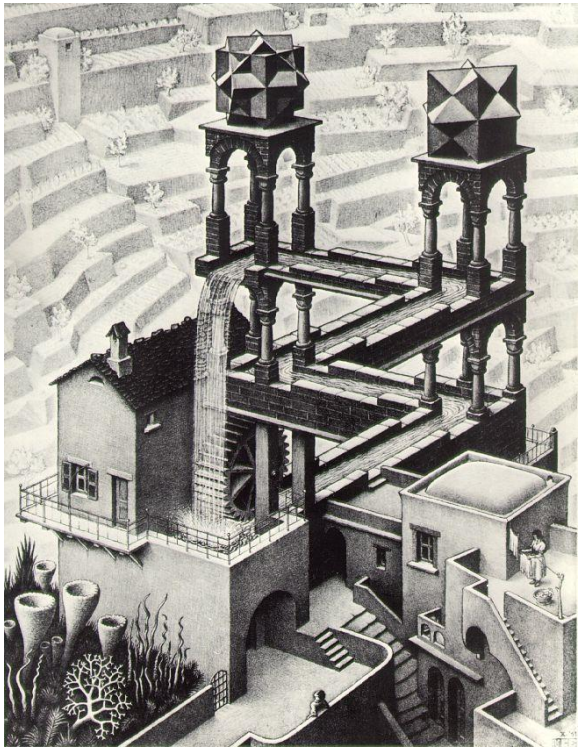
Programming 4

KdG
University of Applied
Sciences and Arts

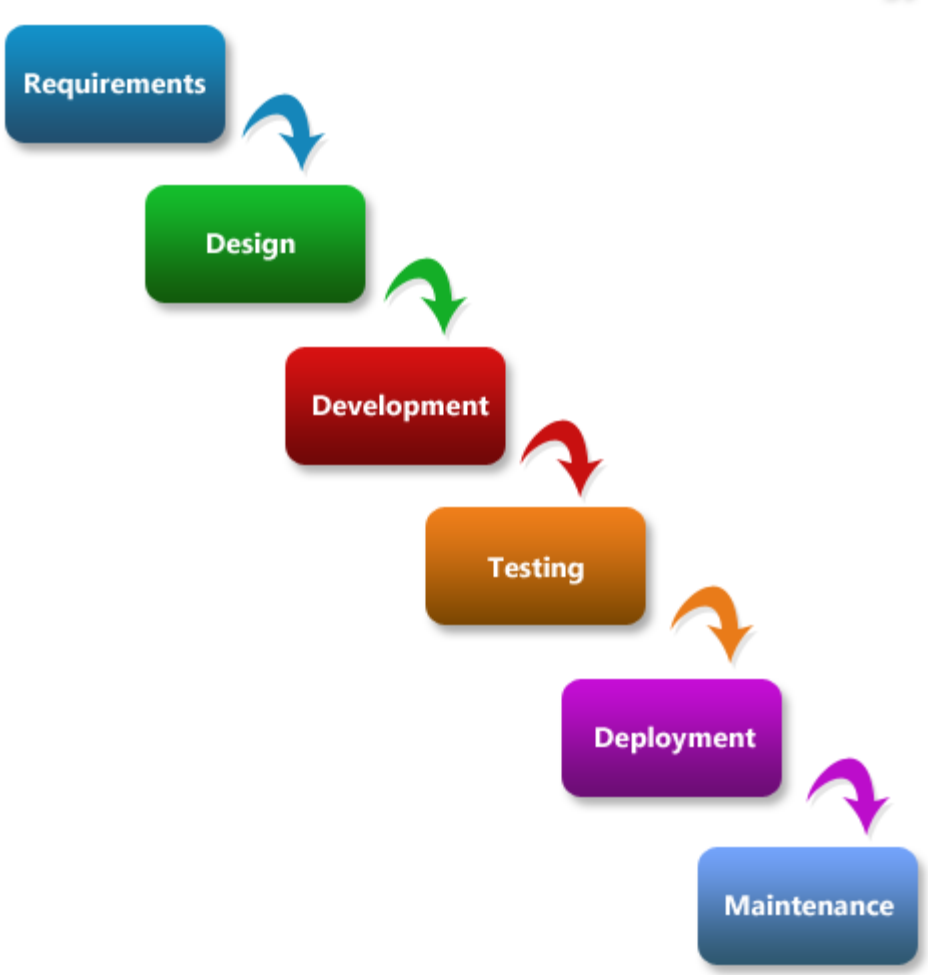
# Agenda

1. Waterfall
2. Agile and Scrum

# Waterfall



# Waterfall

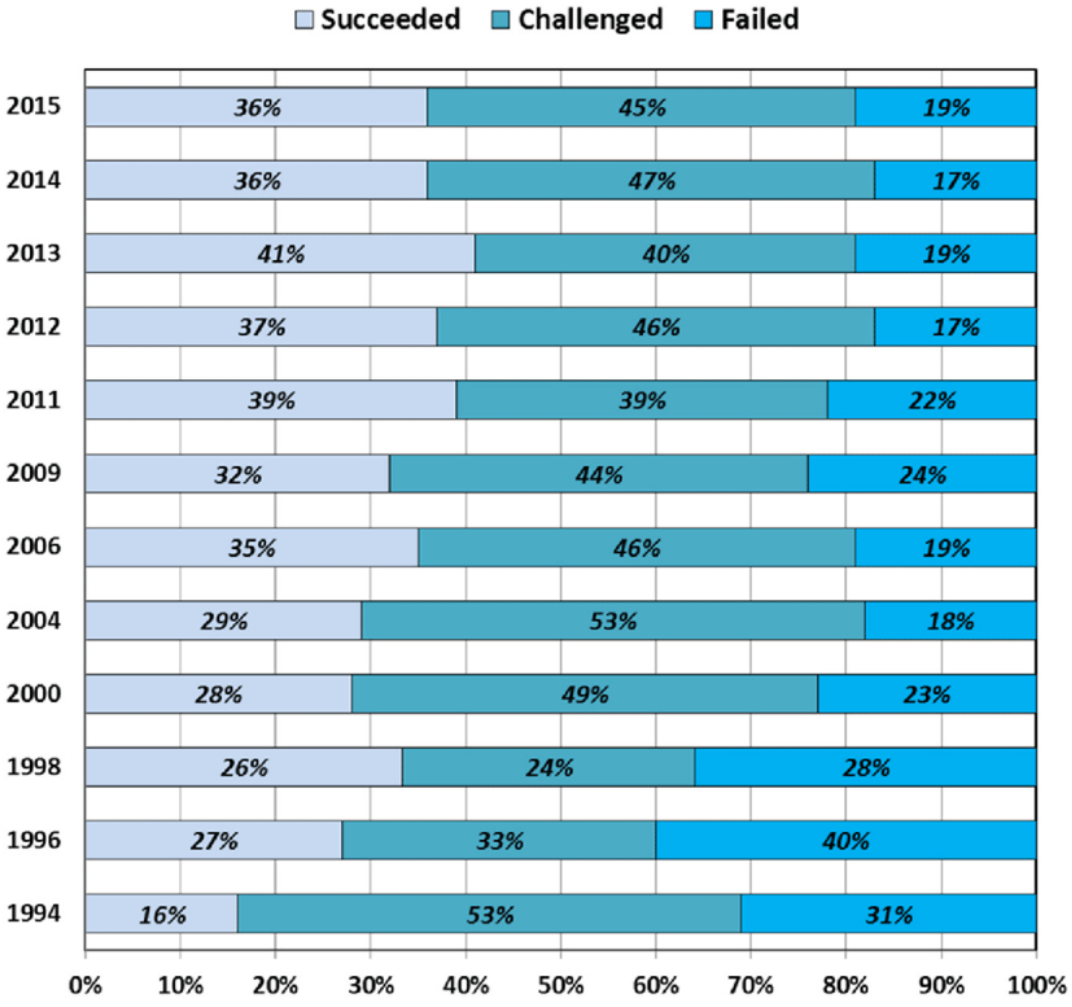

Requirements
Design
Development
Testing
Deployment
Maintenance

# Waterfall: a successful approach for Project Management…

- Sequential tasks
  - *Task 2 starts when task 1 is finished*
- Succesfull  project approach in many disciplines… (engineering, business…)

PMBOK®  PRINCE2®

# Waterfall: … but subpar for Software Projects



# … especially large software projects

**CHAOS RESOLUTION BY PROJECT SIZE**

| | SUCCESSFUL | CHALLENGED | FAILED |
|---|---|---|---|
| Grand | 2% | 7% | 17% |
| Large | 6% | 17% | 24% |
| Medium | 9% | 26% | 31% |
| Moderate | 21% | 32% | 17% |
| Small | 62% | 16% | 11% |
| TOTAL | 100% | 100% | 100% |

*The resolution of all software projects by size from FY2011–2015 within the new CHAOS database.*

# Waterfall

## Strengths

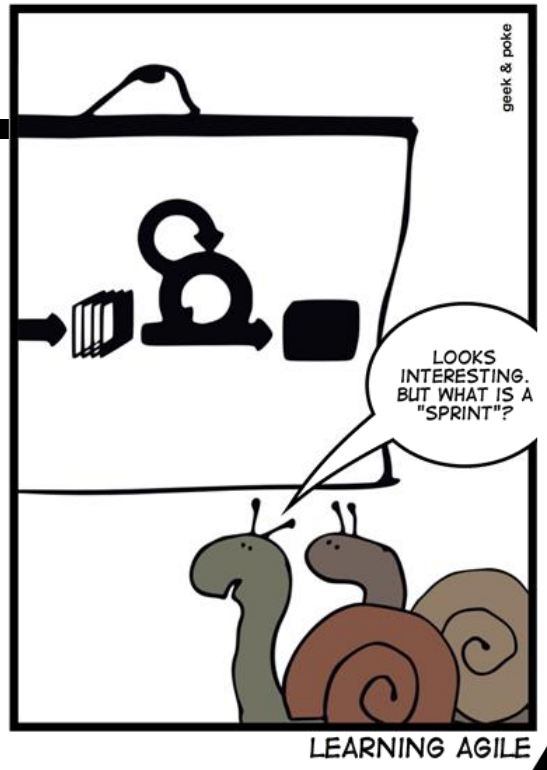
- Requirements, budget and planning known in advance
- Good for small, simple projects
- Extensive project management, procedures and documentation

## Weaknesses

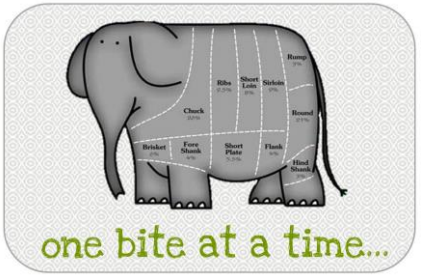
- Estimation of budget and planning is hard
  - Complexity increases exponentially with size
- Little interaction between phases
  - Late reality check of progress and requirements
    - Hard to deal with changing requirements
    - Tests at the end
  - Procedures consume resources and are rigid

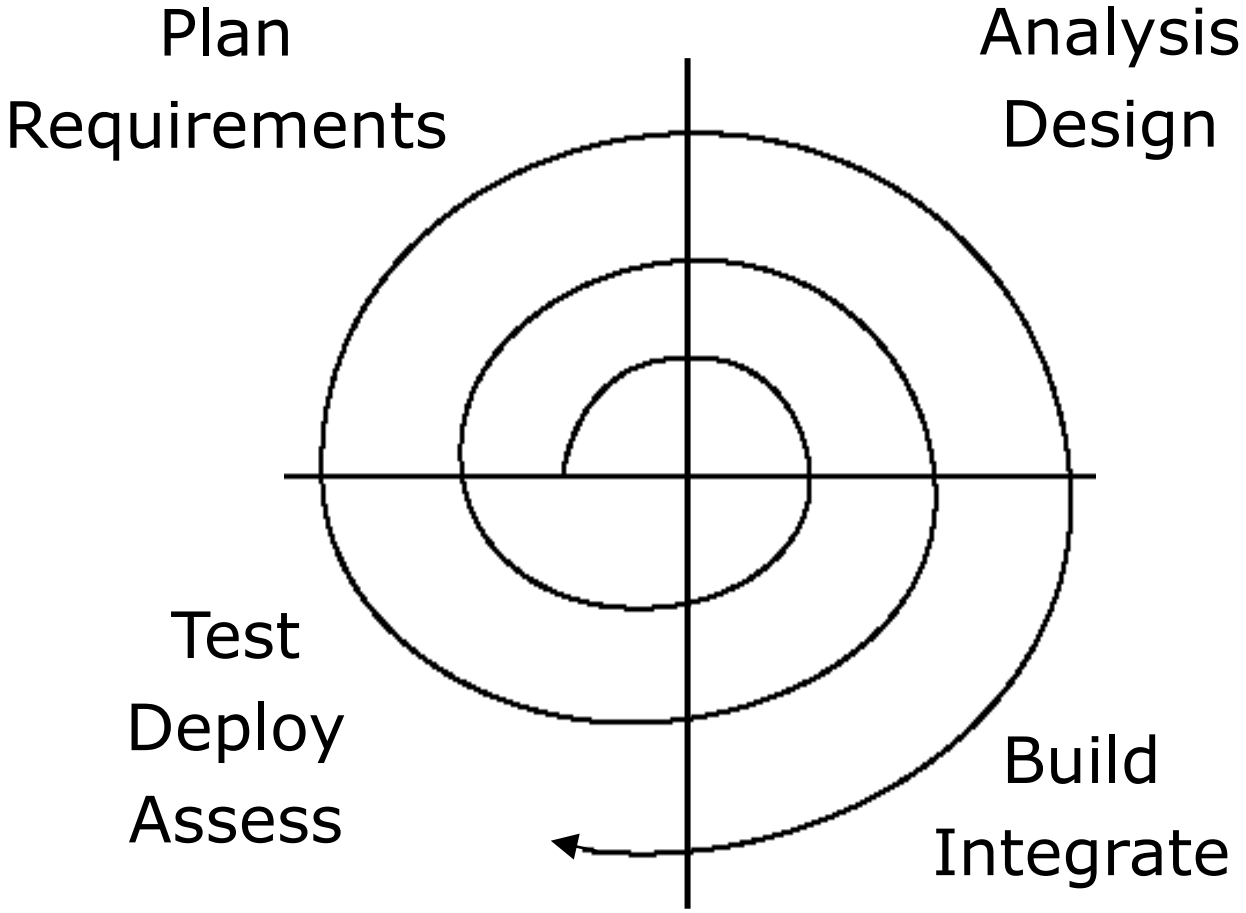
# Agile


THE SCRUM
PRIMER


geek & poke
LOOKS INTERESTING. BUT WHAT IS A "SPRINT"?
LEARNING AGILE

# Iterative process




- The Unified Process (UP) is an example of an iterative process.

## 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:

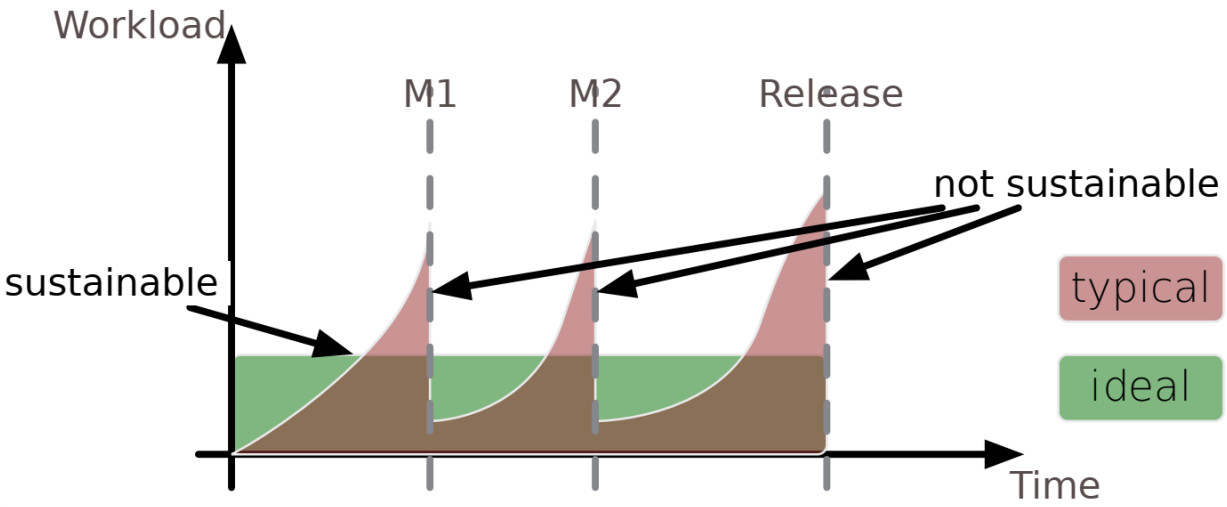
Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

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

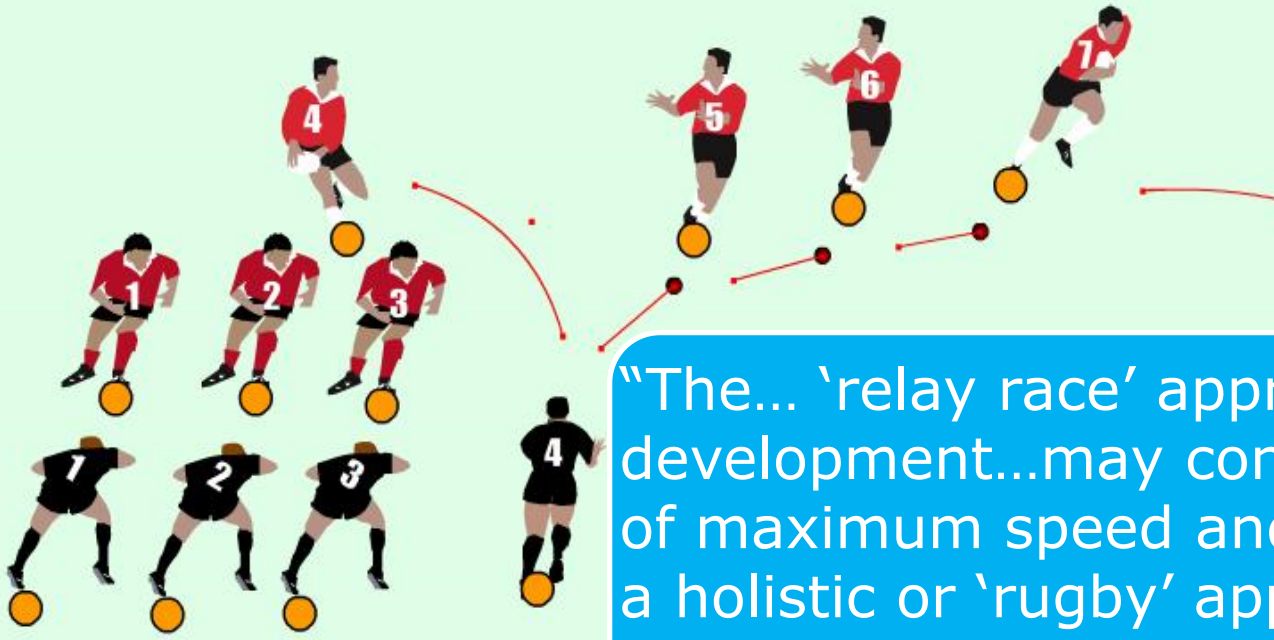

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

# Agile principles

- Strong focus on communication
- Short iterations with a fixed length
- Incremental delivery, evolving software
- Adaptive planning
- Sustainable development



# Scrum: an agile method




"The… 'relay race' approach to product development…may conflict with the goals of maximum speed and flexibility. Instead a holistic or 'rugby' approach—where a team tries to go the distance as a unit, passing the ball back and forth—may better serve today's competitive requirements."

Hirotaka Takeuchi and Ikujiro Nonaka, "The New New Product Development Game", *Harvard Business Review*, January 1986.

# Scrum: an agile method

Convention: sources marked with the syllabus icon are an integral part of the subject matter



https://scrumprimer.org/

## Roles

- Product owner
- ScrumMaster
- Team

## Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

## Events

- Sprint planning
- Sprint
- Daily scrum
- Sprint review
- Sprint retrospective

# Roles

- Product owner
  - Represents customer
  - Responsible for maximising the business value that is produced
  - Define, clarify features
  - Prioritise features
  - Accept or reject work results
- Scrum master
  - Enact Scrum values and practices
  - Remove impediments
  - Coach
  - is NO project manager. Does not decide who should do what.

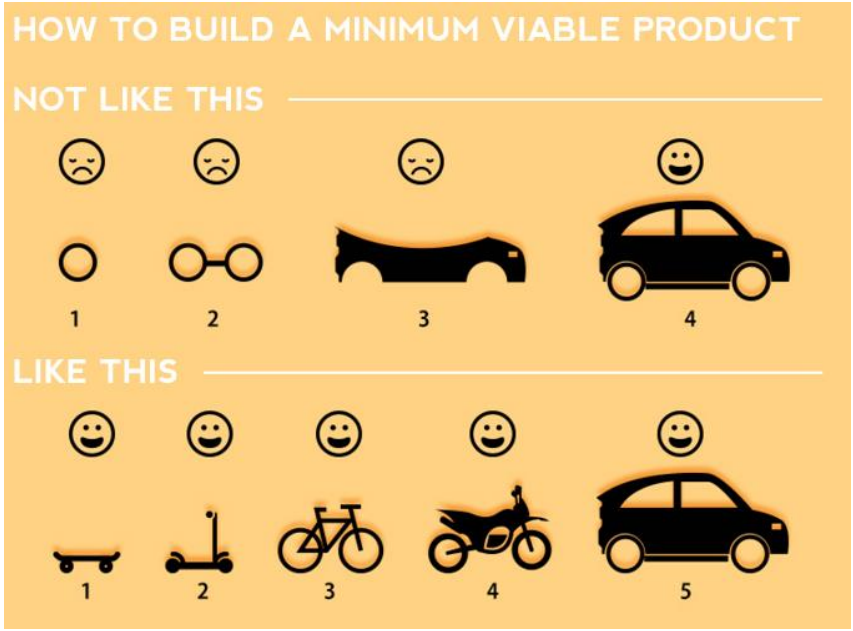
# Roles

- Team
  - Two pizza teams

    *You should be able to feed the team with 2 large pizzas (Jeff Bezos – Amazon)*
  - Typically 4-8 members
    - Smaller = less capacity
    - Bigger = less efficiency
  - Self organising
  - Cross-functional and interchangeable: programming, analysis, architecture, testing, user interface design…
  - Collective responsibility

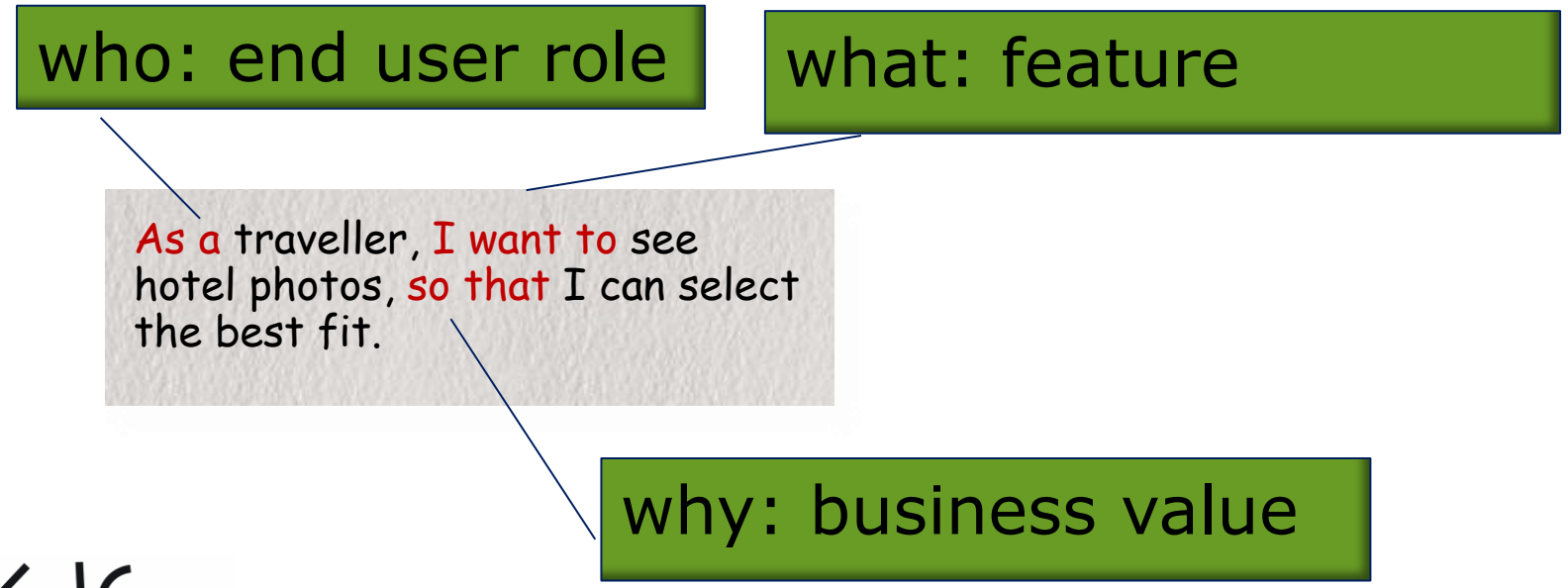
# Scrum overview

- Each sprint (iteration) production quality software is built.
- Product owner prioritizes features based on business value

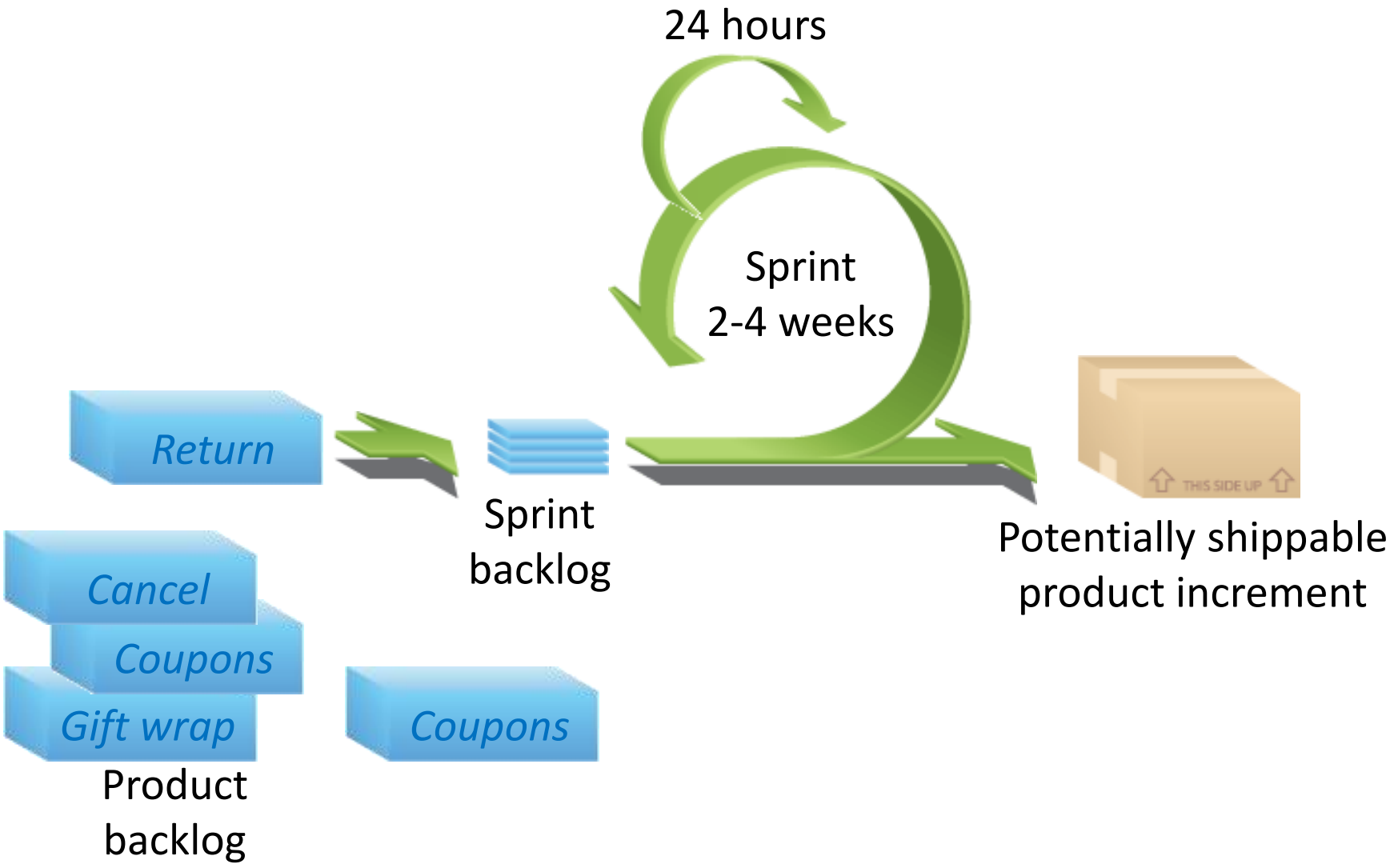


# Scrum requirements: user stories

- Product backlog: list of user stories remaining to be built



# Scrum proces



# Sprint Planning Meeting

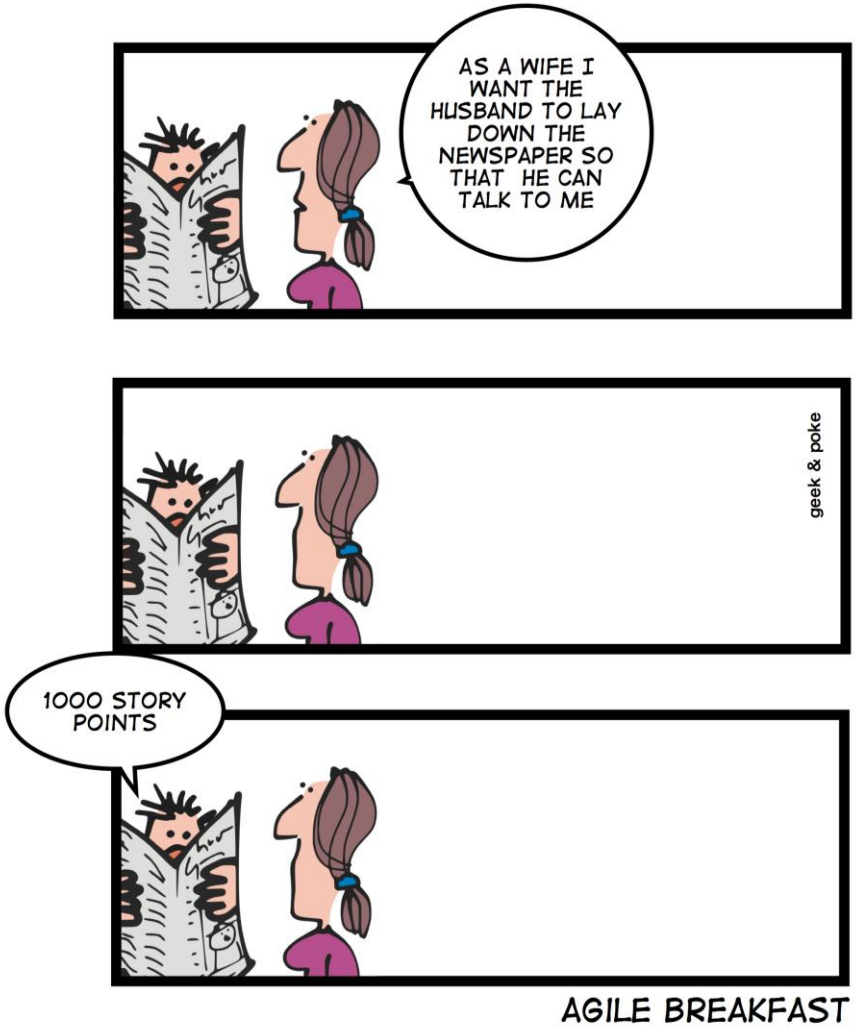
- Sprints have a fixed diration within a project (between 1 and 4 weeks)
  - All Scrum activities are time boxed
- At the start of a  Sprint there is aplanning meeting (time boxed, typically 2-8 hours)
- User stories with highest priority are considered
  - Sprint theme: look for a theme in the high priority stories, lower priority stories within the theme may be selected too
  - If there is a dependency on a user story, that one is selected as well
  - Selected user stories go in the sprint backlog

# Sprint planning meeting

- Team splits stories into tasks
  - A task is done by one person (exception: pair programming)
  - Preference: tasks are subfunctions of the story.
  - Can include analysis, design, refactor existing functions
  - Uer interface, business logic, persistence, interaction with other systems Acceptance criteria can be a
- Team estimates task effort (consensus)
- Number of selected user stories depends on estimation and team capacity for the sprint

# Planning Poker

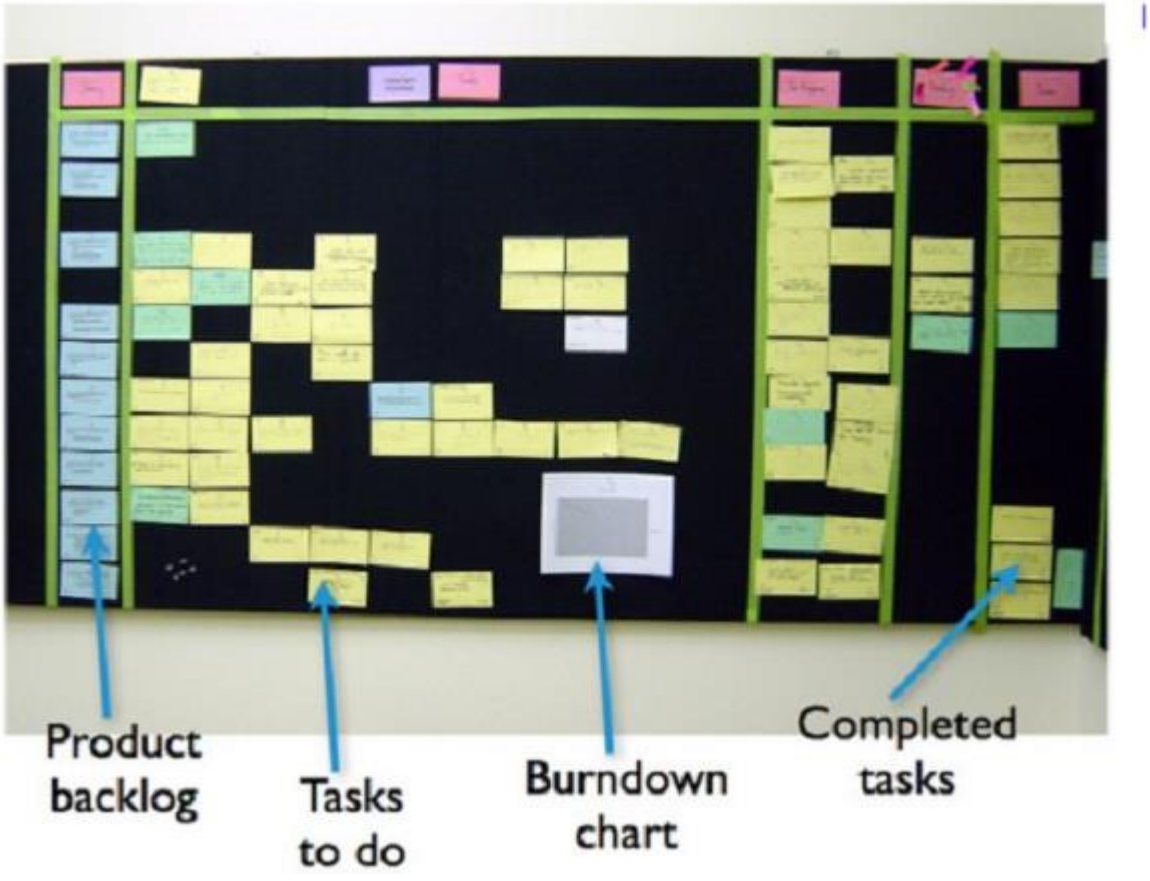
- User story estimation [story points, ideal days…]
- Planning poker by team
  - Everybody votes simultaneously with a card
  - Discuss to reach consensus
    - high and low estimates explai
    - If no consensus: play another round

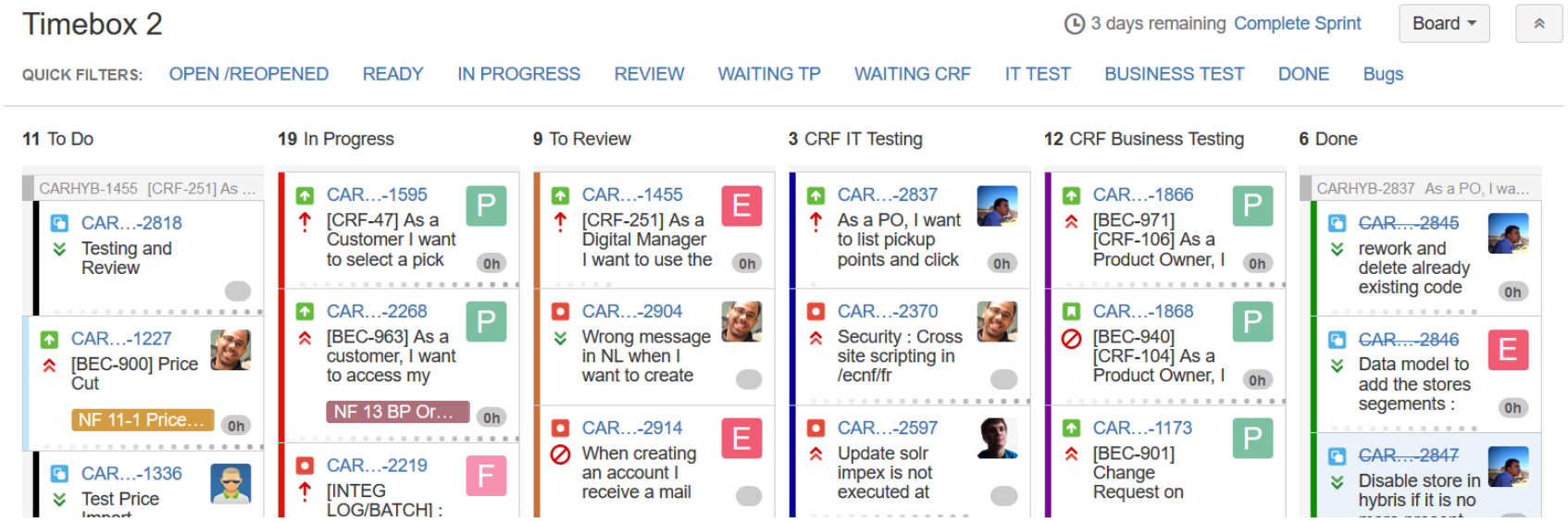
# Scrum process

- Daily standup
  - <= 15 minutes
  - Each team member answers 3 questions:
    1. What did you do the previous day?
    2. What will you do the next day?
    3. Are there any impediments for working smoothly on these tasks
  - Tasks statuses are adapted on the scrum board (kanban)
- Scrum Master + team + Product Owner

# Scrum board (Kanban)

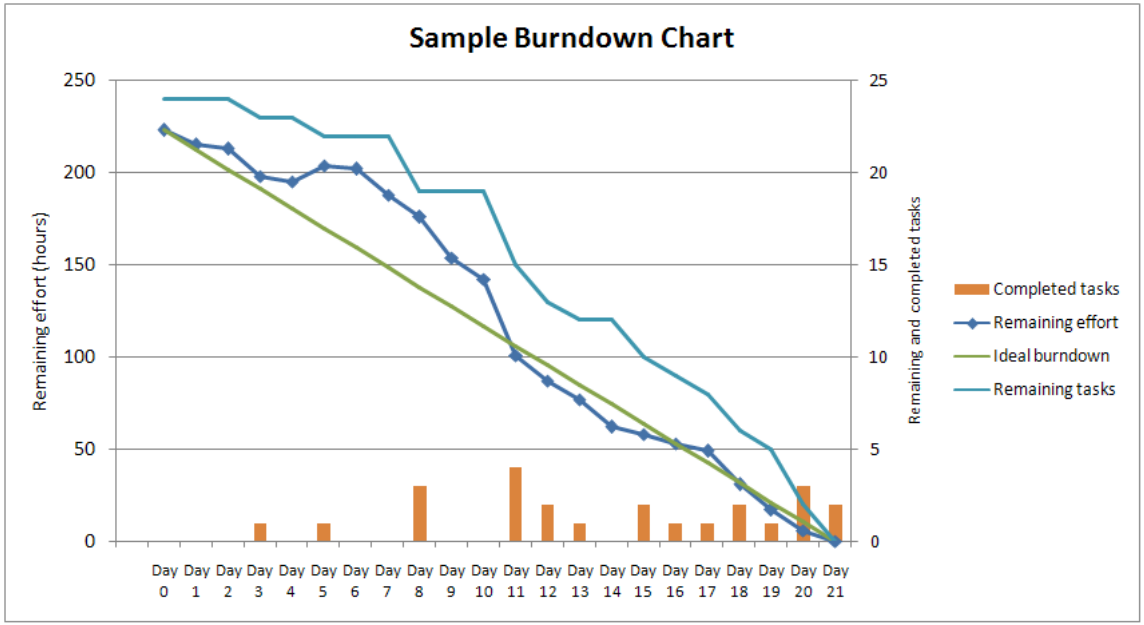


# Scrum board (Kanban)



# Sprint burndown chart

- estimated work remaining
  - You can optionally add a count of completed tasks
  - Tracks team progress (not individual performance)
  - Burn down of previous Sprints are used as a measure of the velocity of the team
    - Velocity: work that can be done in a sprint



# Sprint review meeting

- At the end of the sprint the team presents the user stories that are finished
- <= 4 hours
- Product owner evaluates the user stories
  - Accepted stories can go in production
  - Refused stories return to the product backlog (with an indication of what is done)

# Sprint retrospective meeting

- Evaluate procedures and techniques used by the team
- Scrum is an agile and self adapting process. Discuss how your team should apply Scrum
- <= 2 hours
- What should the team
  - Stop doing?
  - Start doing?
  - Continue doing

# Product Backlog Grooming / scrum add-on

- Elaborate user stories that are candidates for the next sprint to make them READY for development
- By Product Owner and some team members
- Typically one sprint in advance

# Release planning / scrum add-on

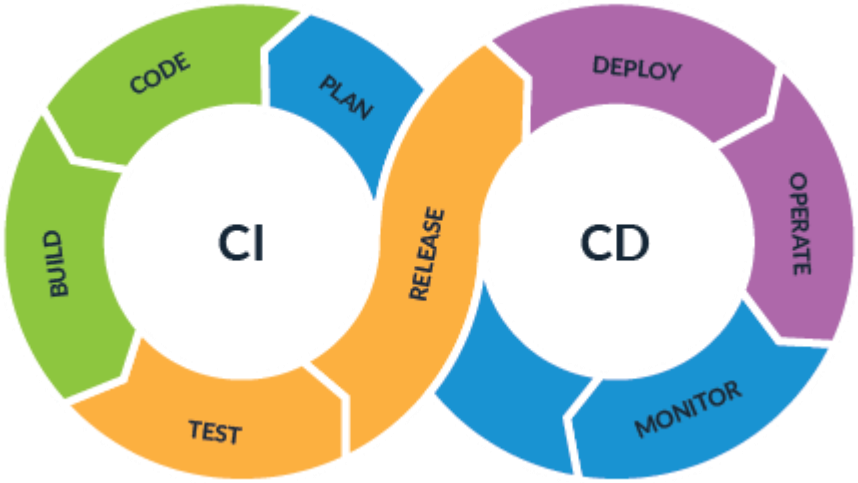
- High level planning for milestones consisting of multiple sprints (~ quarter)
  - To give the cusommer an idea of what is comming
  - Subject to change

# *Potentially* shippable product

- Features usable by the end user
  - Features do not always comprise a consistent, finished product
- Sometimes, only milestones are deployed
- Sometimes, every successful sprint is deployed

Read [DevOps & CI/CD (Continuous Integration/Continuous Deployment)]

  - Enabled by cloud; was not possible in the waterfall era



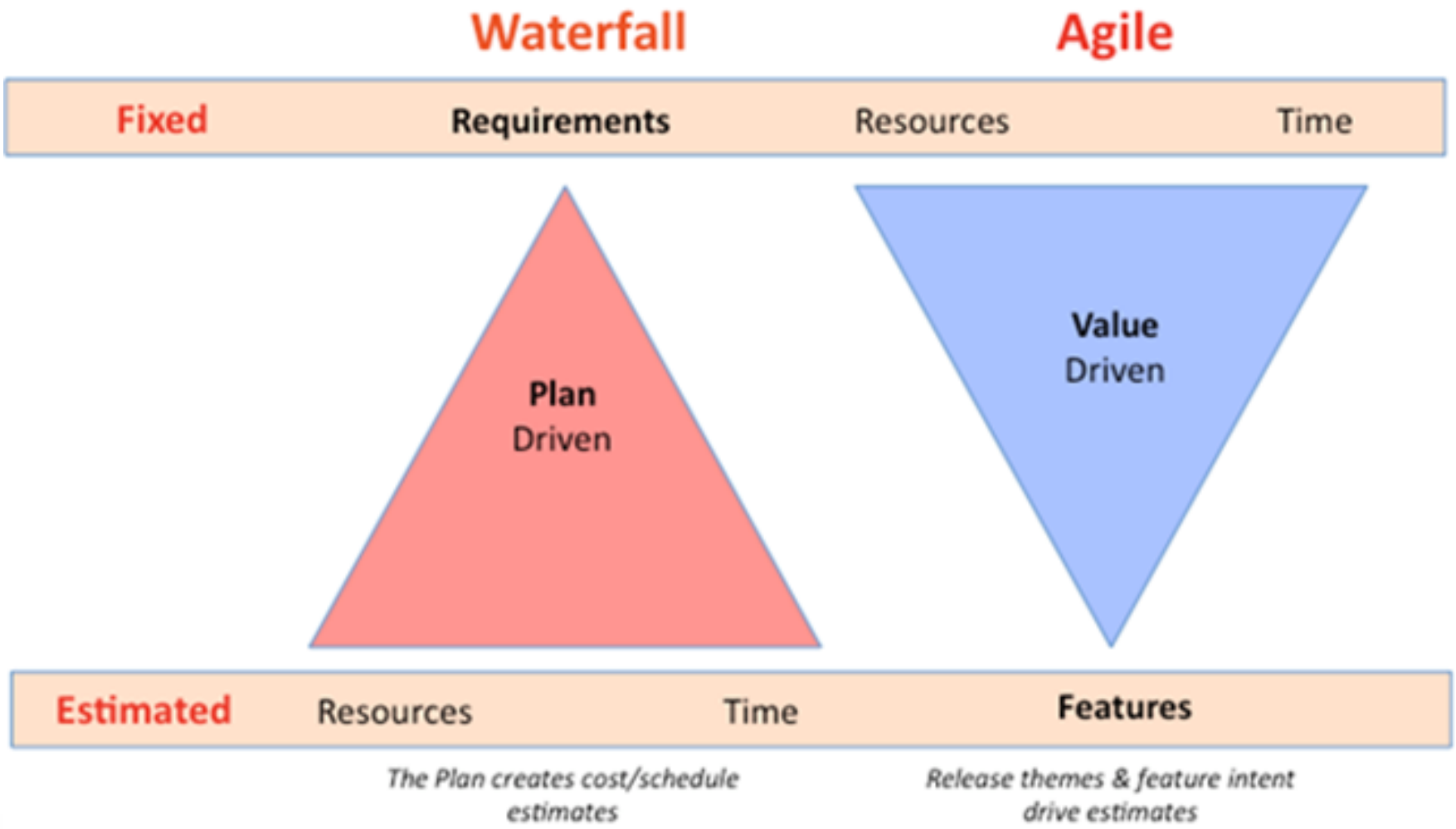
# Spike / Scrum add-on

- Sometimes, estimating a user story is tricky because the team lacks knowledge/experience
- A spike is a user story to try something out
  - Exceptional, not the rule
  - Time boxed (in sprint planning)
  - Experimental implementation (not production-ready software)
  - Mitigate risk and develop an understanding
  - Report/demo in Sprint review.
    - Can you estimate the user story now?


Spikes in Scrum

# SCRUM variations

- Scrumban: Combines Kanban en Scrum
- Scrum for large international teams
  - LeSS: Large Scale Scrumm (Craig Larman & Bas Vodde)
  - DA: Disciplined Agile (PMI / Scott Ambler)
  - Nexus (scrum.org / Ken Schwaber))


Waterfall
Agile
Fixed
Requirements
Resources
Time
Plan
Driven
Value
Driven
Estimated
Resources
Time
Features
The Plan creates cost/schedule estimates
Release themes & feature intent drive estimates

# Agile

## Strengths

- Flexible
- Little overhead
- Effective progress is known at any time
- Fast production verification of features
- Fast Return On Investment

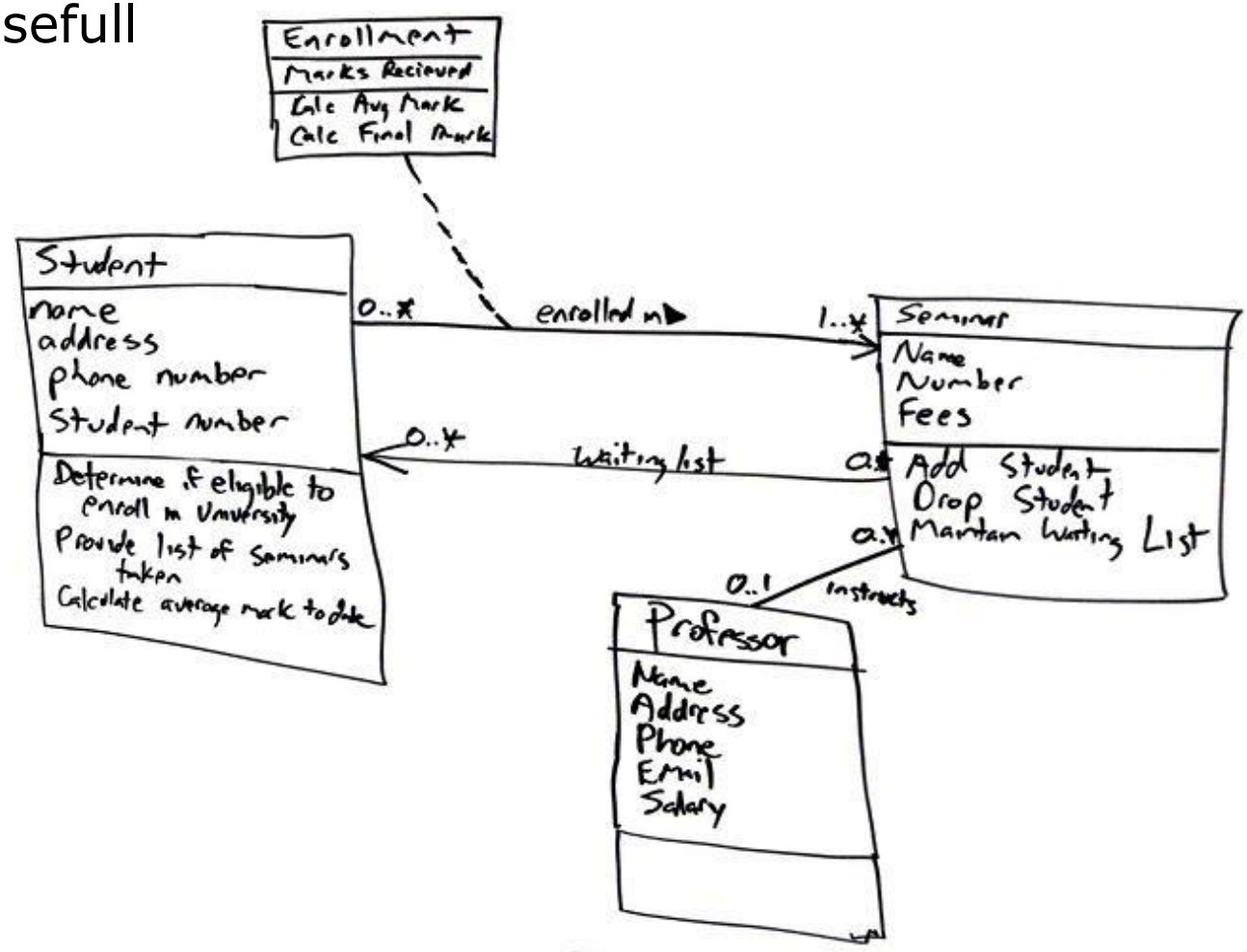
## Weaknesses

- Hard in non-agile organisations
  - Less predictable
  - Hard for fixed-form price projects
  - Needs daily customer involvement
  - Limited project metrics
- Limited to managing the development process in a project
- Harder for distributed teams

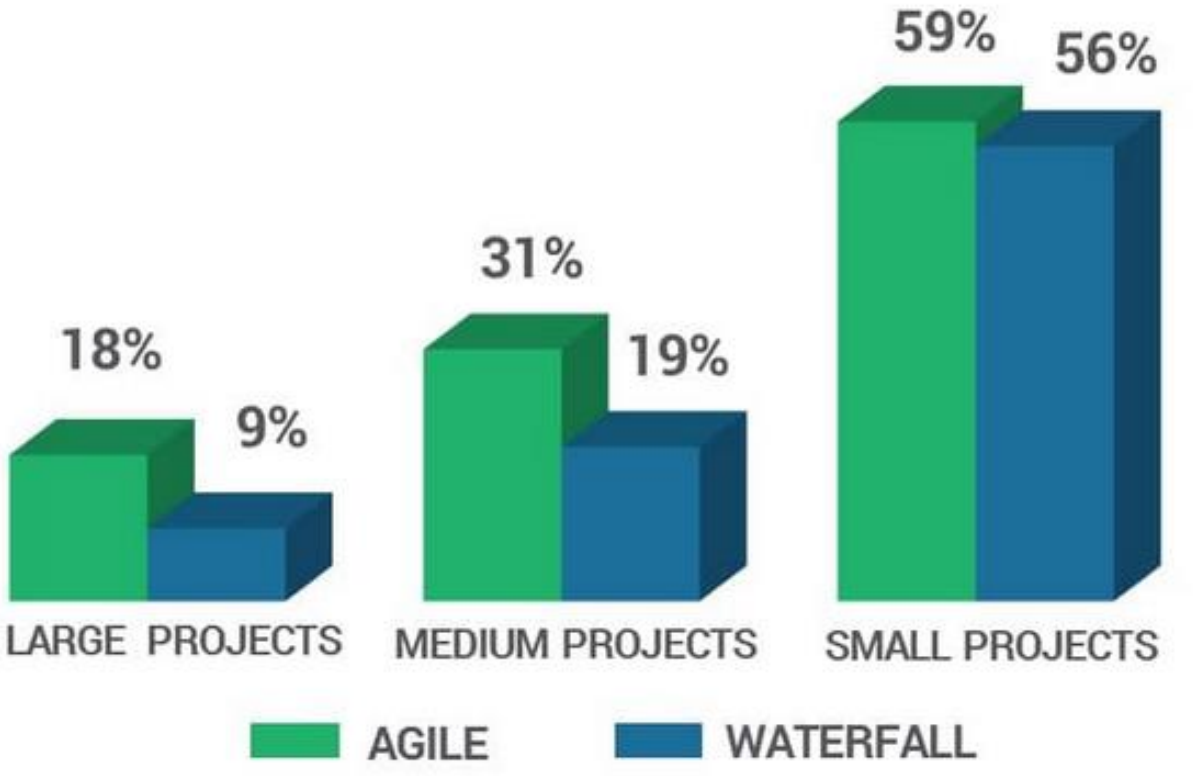
Exercise 2

# Agile Modeling

- Modeling as a communication tool
- Limit to what's usefull
- Model in team



# Project methode en succes




*Source: Standish Group, Chaos Studies 2013-2017*

# Summary

1. Waterfall
2. Agile and Scrum