# Software Engineering 2

## Software process

Dr. Petr SPACEK[1]

[1]petr.spacek@fit.cvut.cz
Department of Software Engineering
Faculty of Informatics
Czech Technical University in Prague

Winter Semester, 2020

# Outline



Intro

Connection to the previous course

Software process

Basics

Models of software process

Acpects

How to pick?

Conclusion

Summary

1 Intro

2 Software process

3 Conclusion

# Outline



Intro
Connection to the previous course
Software process
Basics
Models of software process
Acpects
How to pick?
Conclusion
Summary

1. Intro
   - Connection to the previous course
2. Software process
   - Basics
   - Models of software process
   - Acpects
   - How to pick?
3. Conclusion
   - Summary

# Connection to the previous course

## Software engineering



The 4 main activities of Software engineering:

- Specification
- Development
- Validation
- Evolution & maintenance

# Outline



Intro
Connection to the previous course
Software process
Basics
Models of software process
Acpects
How to pick?
Conclusion
Summary

1. Intro
   - Connection to the previous course
2. Software process
   - Basics
   - Models of software process
   - Acpects
   - How to pick?
3. Conclusion
   - Summary

# Basics



- Software process
  - set of activities required for developing software
- Basic activities
  - Specification - what to do?
  - Development
    - architecture - from which components?
    - design - how to create the components?
    - implementation - actual production of the software
  - Validation - does the result what is required?
  - Evolution & maintenance - how to changed the system?
- Software process model
  - description of the software process from a particular perspective

# Plan-driven or agility?



- plan-driven
  - activities are planned in advance
  - progress is measured by comparing the status of a plan
  - increased overhead in case of changes
- agile
  - planning is only small portions
  - easy to change "direction" in case of changes in customer requirements

# Outline



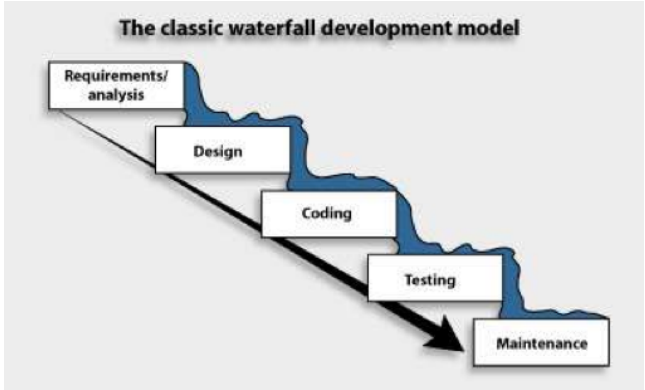
Intro
Connection to the previous course
Software process
Basics
Models of software process
Acpects
How to pick?
Conclusion
Summary

1. Intro
   - Connection to the previous course
2. Software process
   - Basics
   - Models of software process
   - Acpects
   - How to pick?
3. Conclusion
   - Summary

# Waterfall



- Separate phases
  - Requirements Analysis
  - Design
  - Implementation
  - Testing
  - Deployment and maintenance



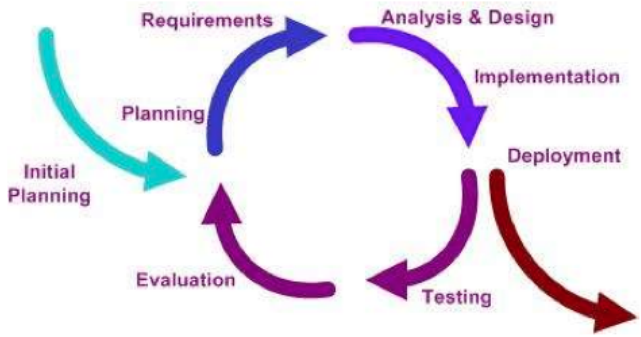

- Pros (+)
  - A clearly defined plan
  - Predictability (time, scope, price)
  - Easy coordination of work

- Cons (-)
  - It is necessary to understand what client wants from the begining
  - Reaction to change (requirements, deadlines, ...)
  - Speed of first delivery (when customers see something)

# Iterative



- Changes compared to the waterfall
  - Several versions of system
  - Each made using waterfall




- Pros (+)
  - A clearly defined plan
  - Predictability (time, scope, price)
  - Easy coordination of work
  - The client has access to versions / prototypes
- Cons (-)
  - It is necessary to understand what client wants from the beginning
  - ~~Reaction to change (requirements, deadlines, ...)~~
    - Better than the waterfall, but the changes are not delivered quickly
  - ~~Speed of first delivery (when customers see something)~~

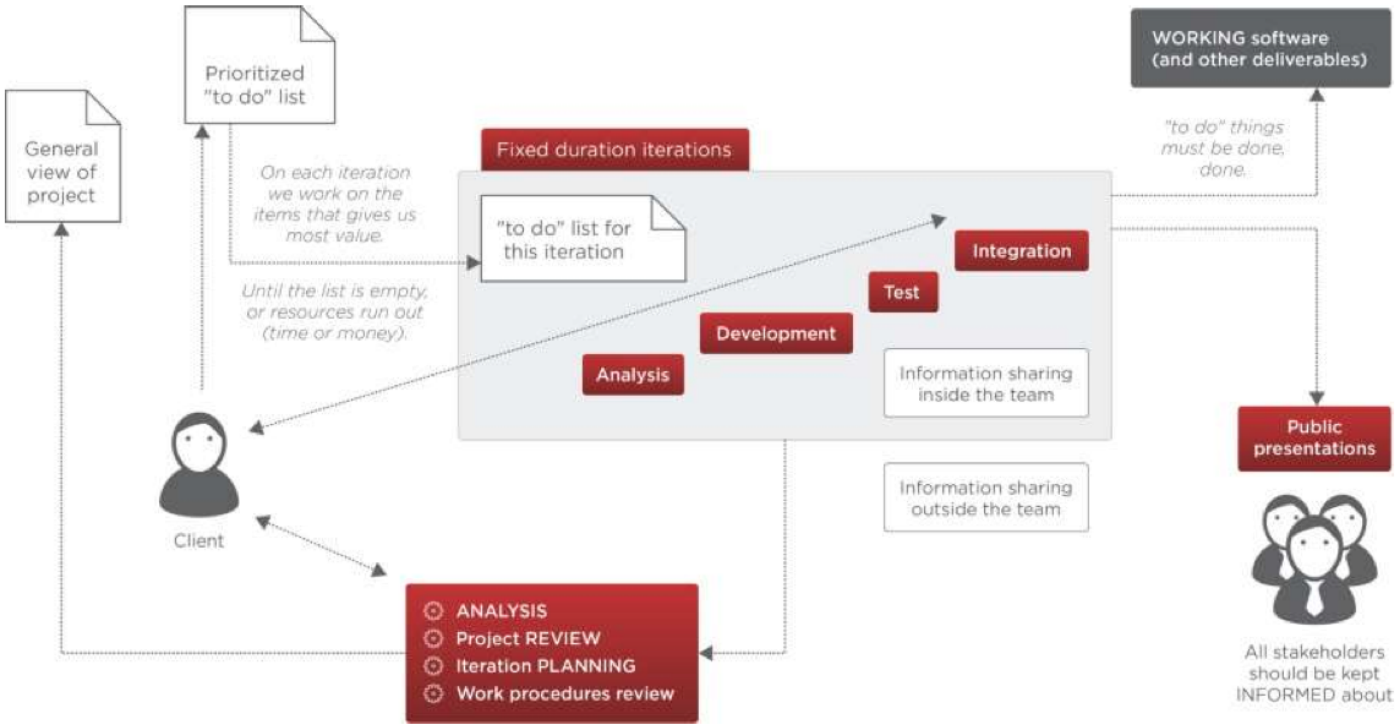
# Agile




Manifesto for Agile Software Dev.
AGILE
INDIVIDUALS AND INTERACTIONS OVER PROCESSES AND TOOLS
WORKING SOFTWARE OVER COMPREHENSIVE DOCUMENTION
CUSTOMER COLLABORATION OVER CONTRACT NEGOTIATION
RESPONDING TO CHANGE OVER FOLLOWING A PLAN

# Agile




Prioritized "to do" list
General view of project
On each iteration we work on the items that gives us most value.
Until the list is empty, or resources run out (time or money).
Fixed duration iterations
"to do" list for this iteration
Integration
Test
Development
Analysis
Information sharing inside the team
Information sharing outside the team
Client
ANALYSIS
Project REVIEW
Iteration PLANNING
Work procedures review
WORKING software (and other deliverables)
"to do" things must be done, done.
Public presentations
All stakeholders should be kept INFORMED about

# Outline



Intro
Connection to the previous course
Software process
Basics
Models of software process
Acpects
How to pick?
Conclusion
Summary

1. Intro
   - Connection to the previous course
2. Software process
   - Basics
   - Models of software process
   - Acpects
   - How to pick?
3. Conclusion
   - Summary

# Flexibility



- Flexibility
  - Response to Changes
  - Speed and cost

- Waterfall
  - Very inflexible
  - High cost
- Iterative
  - Changes could be incorporated into the next iteration
  - Lower costs than the waterfall
- Agile
  - Expects changes
  - The cost of changes is very low

# Predictability



- Predictability
  - Knowing what and when you get
  - How much will it cost
- Waterfall
  - High, going according to plan
- Iterative
  - High, we have a plan
- Agile
  - Low, we plan only for a short period
  - Knowing what we get for the next sprint

Easier Budget and Scope Guiding

Good

KindaGood

Bad

# Architecture and Design



- Architecture and Design
  - A well-designed system
  - Follow design principles

- Waterfall
  - High
- Iterative
  - High
  - The risk of contamination by problems in the next iteration
- Agile
  - Low
  - The risk of creating a problem at each sprint

# Implementation



- Implementation
  - Space for the delivery of quality work
  - Requirements for programmers


- Waterfall
  - Quality, enough space for QA
  - Revision, coding standards, ...
- Iterative
  - Quality, enough space for QA
  - Revision, coding standards, ...
  - The risk of contamination by problems in the next iteration
- Agile
  - Good team required (which may be problematic)
  - Risk of poor quality work when there is no space for revision

# Documentation

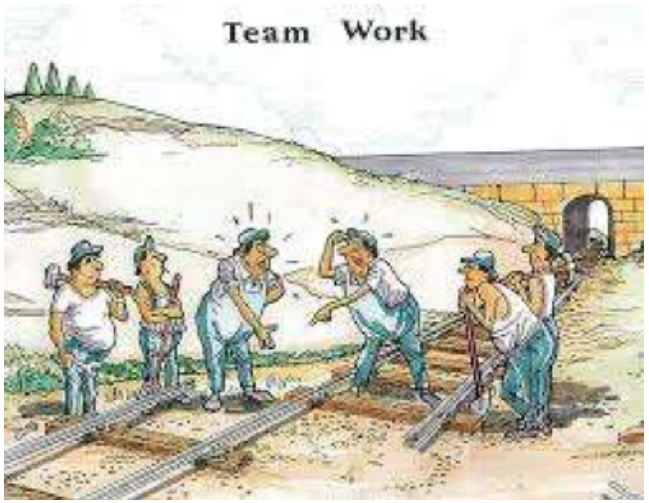


- Documentation
  - I have a high quality and consistent documentation of the system
- Waterfall
  - Typically OK
- Iterative
  - It should be consistent across versions (one True or more versions?)
- Agile
  - Low
  - Difficult to maintain across sprints

# Cooperation with clients



- Requirements for synergy
  - How often a client must participate on the project?




- Waterfall & Iterative
  - The well-defined moments
  - Can be planned easily
- Agile
  - Throughout all the project
  - High risk of failure, when poor synergy

required

# Outline



Intro
Connection to the previous course
Software process
Basics
Models of software process
Acpects
How to pick?
Conclusion
Summary

1. Intro
   - Connection to the previous course
2. Software process
   - Basics
   - Models of software process
   - Acpects
   - How to pick?
3. Conclusion
   - Summary

# How to pick a model?



- Waterfall
  - Usable for new kinds of software (original, innovative)
- Iterative
  - Most used for large-scale projects
  - Predictability is more important than flexibility
  - Good for development and also for maintenance
- Agile
  - Product development
  - Development within the company (a clear willingness of people to participate in it)
  - Rather smaller projects or certain parts of the process (e.g. only development)

# Outline



Intro
Connection to the previous course
Software process
Basics
Models of software process
Acpects
How to pick?
Conclusion
Summary

1 Intro
- Connection to the previous course

2 Software process
- Basics
- Models of software process
- Acpects
- How to pick?

3 Conclusion
- Summary

# Conclusion

## Summary



Intro

Connection to the previous course

Software process

Basics

Models of software process

Acpects

How to pick?

Conclusion

Summary