Karlsruhe Institute of Technology

Softwaretechnik II

Oliver Hummel, IPD

Topic 2

Exercises Software Process Models

SOFTWARE DESIGN AND QUALITY GROUP
INSTITUTE FOR PROGRAM STRUCTURES AND DATA ORGANIZATION, FACULTY OF INFORMATICS

sdq.ipd.kit.edu

KIT – University of the State of Baden-Wuerttemberg and
National Research Center of the Helmholtz Association

www.kit.edu

Understanding Process Models

1. Discuss the following statement in the context of the RUP
   - inception = requirements
   - elaboration = architecture and design
   - construction = implementation

2. Discuss the pros and cons of the following iterations lengths –
   - 5 days
   - 4 weeks
   - 6 months
   - 2 years

   ➔ and briefly justify which you would prefer for an iterative and incremental project of four years.

3. What does timeboxing mean?

Project Planning

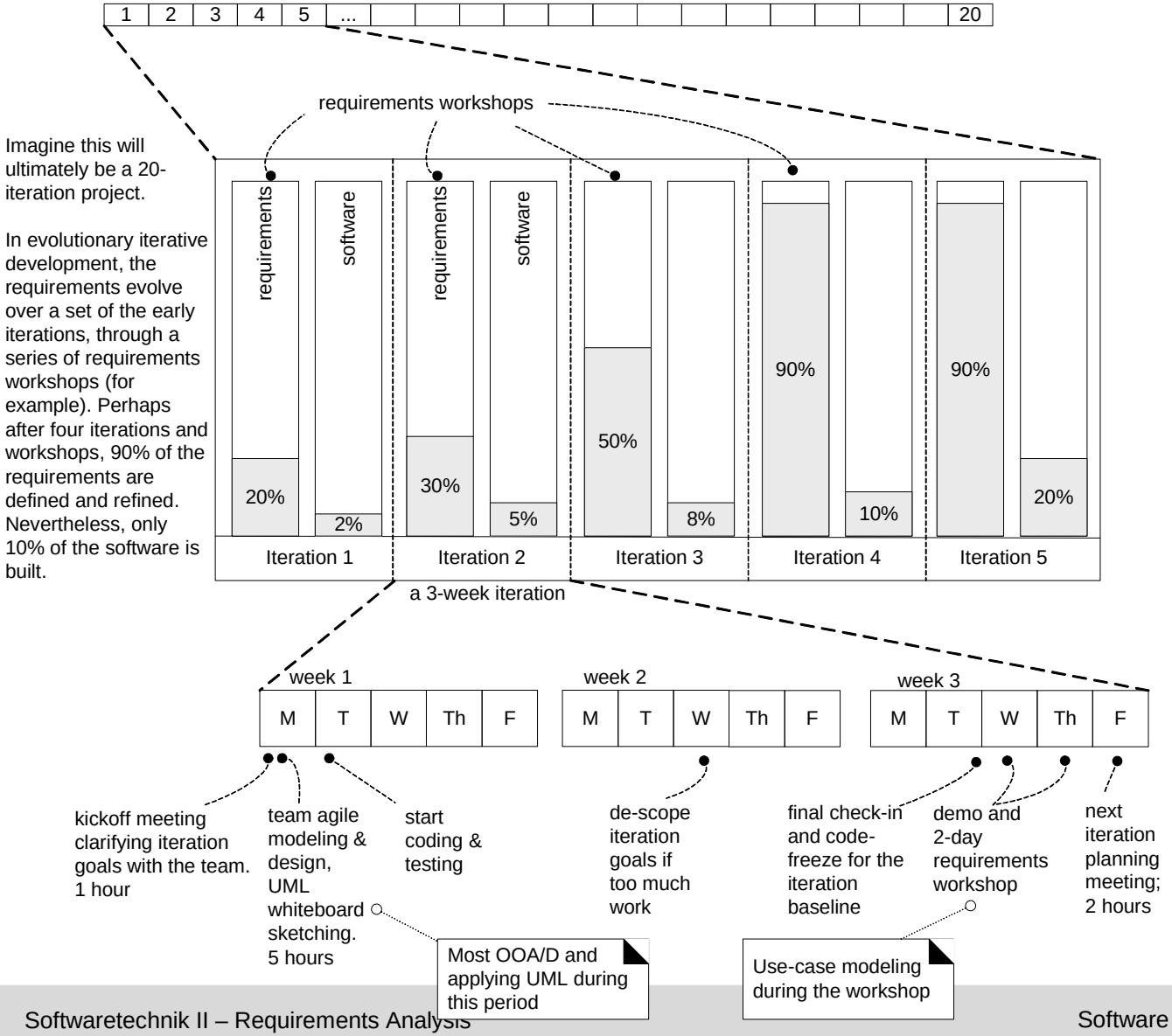
1. Come up with a reasonable (but simple) project plan for a small development project of about 5 months
   a) following a classical waterfall approach
   b) in an iterative and incremental style

   Please focus on the Activities and the Deliverables
   - and pay special attention to the amount of requirements that is elaborated and roughly how much code is roughly implemented at the end of each phase or iteration

RUP Instantiation Example

- How much of which activity should be performed at what time?
  - the following **example** demonstrates how an iterative project with about 20 iterations could look like [Larman]
  1. a two day (timeboxed) requirements workshop is hold before iteration 1
     1. start with **high-level requirements** collection (half a day)
        - just use case or feature names, key non-functional aspects
     2. ask chief architect and important business people to pick about **the 10% most important use cases**; in terms of –
        - architecture significance, business value and risk
     3. **refine the selected** use cases in the remaining time
  2. select the most important aspects of these use case to be implemented in a planning meeting at the beginning of iteration 1
  3. do iteration 1 (timeboxed to e.g. 3 weeks)
     - the goal is to learn and to establish collaboration guidelines in the team
     - do modeling, programming and testing, and a demo of the 1st increment
     - do another requirements workshop (review old and refine new use cases)
       - and plan the next iteration

RUP Example as Picture



You Know …

- … you **did not understand** iterative development when [Larman]
  1. you try to fully define –
     - most of the requirements before starting design and implementation
       - i.e. you write all use cases in detail during inception
     - most of the design or architecture before starting to program
  2. you spend weeks in UML modeling before programming a single line
  3. you still think –
     - inception = requirements
     - elaboration = architecture and design
     - construction = implementation
  4. you believe a suitable iteration length is 3 month rather than 3 weeks
  5. you try to plan a project in detail from start to finish
     - i.e. you speculatively try to predict all iterations
       - and what should happen in each one
  6. you defer testing until near the end of the project

Discussion

1. The lecture has presented numerous presumed advantages for iterative and incremental process models, however, it is likely that there are drawbacks, too.

   Hence, please name and discuss at least two potential drawbacks of I&I development methods.

Discussion

1. The lecture has presented numerous presumed advantages for iterative and incremental process models, however, it is likely that there are drawbacks, too.

   Hence, please name and discuss at least two potential drawbacks of I&I development methods.

Potential answers:

- Constant changes to the code base require a lot of refactoring effort
- Domain experts or customers must be „always“ available to clarify open issues

➔ Assumption, this would be different in sequential process models

Let‘s get Technical

- … and create a first overview of technical (UML) models that can be used for systematic **agile software modeling**
  - bringing together I&I as well as agile development
    - and modelling
  - mainly based upon –
    - RUP
    - [Larman]
    - [Ambler]