University of Zurich

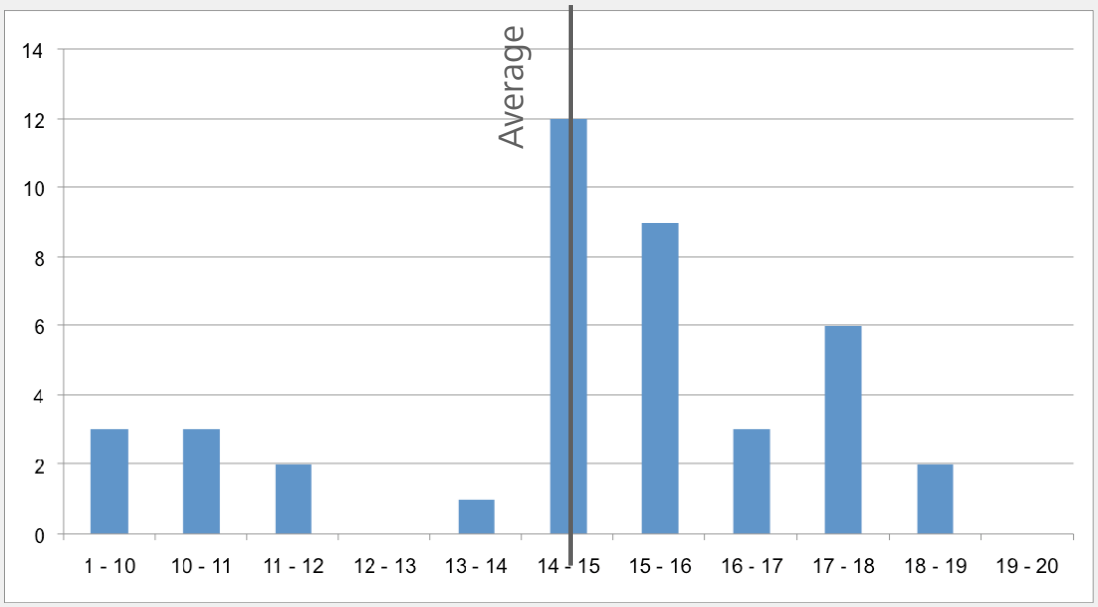
# Discussion SE Exercise 3

Dustin Wüest and Cédric Jeanneret

Requirements Engineering Research Group
Department of Informatics
University of Zurich

University of Zurich

## SE Exercise 3 Results



# Ex 1: Modularization

University of Zurich

Basic modularization of the **solution**

- Reduction of complexity (smaller parts)
- Development process planning (e.g.: milestones)

Best shown using a diagram!

- Modules can be recursively decomposed
- Relationships between modules

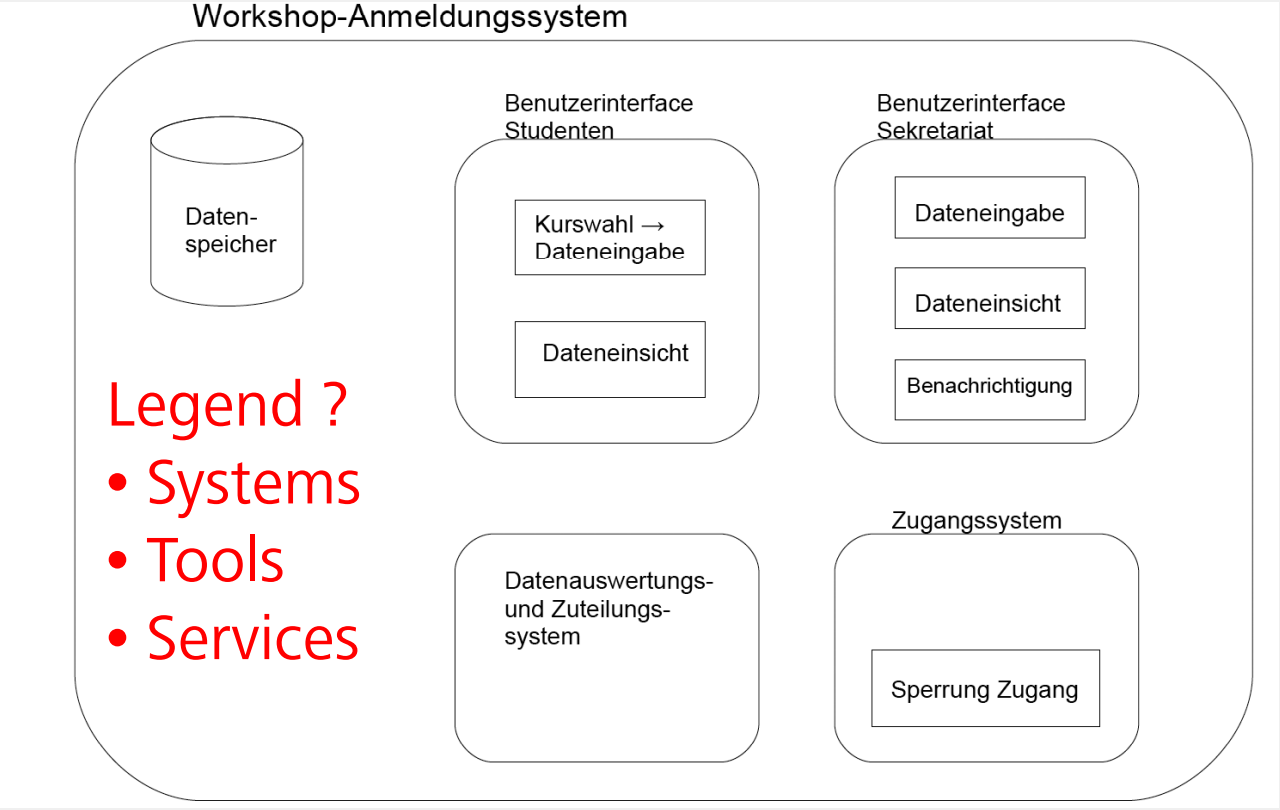
Difference between modules and use-cases:

- Use cases: scenario from an user viewpoint
- Modules: structural elements of the solution



# Ex 1: Modularization
## *Students solution*

University of Zurich



## Ex 2: Software Processes
## *Phase Model*

University of Zurich

*A system is first conceived then realized… (project life-cycle)*

Prerequisites:

- Definable requirements (but not necessarily defined)
- Low development risk

Advantages:

- Good for project management

Disadvantages:

- Late delivery of the running solution
  - Risks related to adequacy !
  - Technical risks



## Ex 2: Software Processes
## *Evolutionary model*

University of Zurich

*A system grows up… (system's behavior)*

Prerequisites:

- The system can be introduced in several steps
  1. Students registration
  2. Students notification
  3. Workshop assignment

Advantages:

- Very good for project management
- Early start of operation
  - early feedback from users
  - increase motivation of stakeholders

Disadvantages:

- Danger of poorly structured piece of software

## Ex 2: Software Processes
## *Agile Software Development*


Prerequisites:

- Involvement of a customers representative who can take decisions
- Experienced software architect
- Intensive software quality assurance at the source

Advantages:

- Involvement of the customer

Disadvantages:

- May turn to "Cowboy programming"
- Danger of poorly structured piece of software



## Ex 2: Software Processes


| | Blasco Art | ABS | VR |
|---|---|---|---|
| Project's Risk | Medium (adequacy risk) | Low (even if safety-critical) | High (innovation) |
| Definable Requirements? | Yes | Yes (even defined and stable) | No |
| Stepwise Introduction | Yes / No | No | Yes |
| Software Process | Evolutionary Phase Model | Phase Model | Evolutionary Agile |

## Ex 2: Software Processes
## *Remarks*


A software system can be extended after its development, no matter the process used to develop it.

All processes aim at "uncover errors as early as possible" and reduce the risk.



## Ex 3: Milestones

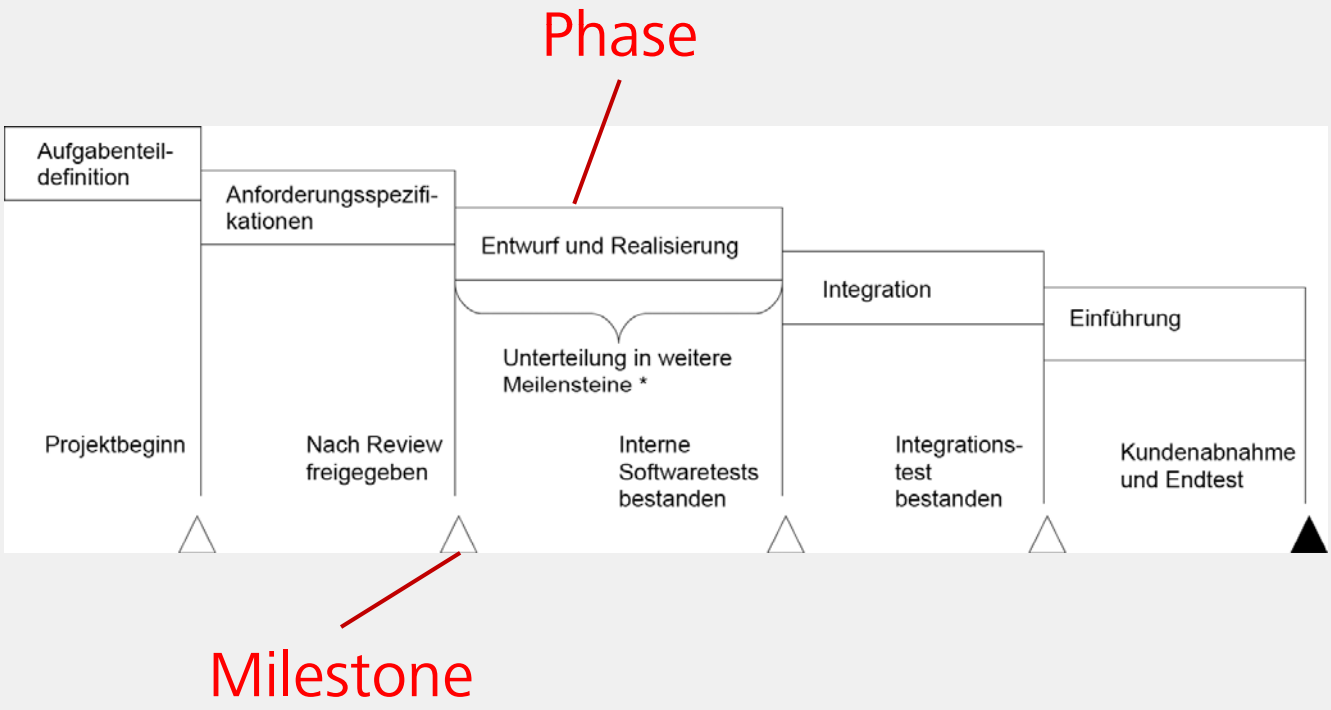

Purposes

- Structure the process (e.g.: deliveries)
- Evaluate the progress

Criteria

- Consistency with your chosen process model
- Verifiable milestone (review, tests)

## Ex 3: Milestones
## *Student Solution – Phase Model*






## Ex 4: Prototyping

Purpose: Early detection (and resolution) of problems.

Explorative

- Demonstration: the feasibility and usefulness of a system
- Prototype in narrower sense: Adequacy of requirements, fitness of an intended solution

Experimental (laboratory prototype)

- Investigate the feasibility of a critical part of a system
- Evaluation of design alternative

Evolutionary (pilot system)

- Related to an evolutionary process model

# Ex 4: Prototyping
## *Correction*


Phase and module

Kind of prototype

Justification

Meaningful use of prototype !

- Demonstration during project definition phase (phase model)
- Pilot system (evolutionary model)

- Prototypes in narrower sense for GUIs (requirements / conception).

Throw-away prototypes cost time and money!