

Vragen

- Geef een definitie van Software Engineering?
- Waarom is Software Engineering noodzakelijk?
- Noem een aantal stakeholders?
- Wat zijn de kwaliteitsattributen?

Internal Quality Criteria

- These:
 - Characterize *aspects of the design* of the software
 - Have an effect on the external quality attributes
- E.g.
 - The amount of commenting of the code
 - The complexity of the code

Short Term Vs. Long Term Quality

- Short term:
 - Does the software *meet the customer's immediate needs*?
 - Is it sufficiently efficient for the volume of data we have *today*?
- Long term:
 - Maintainability
 - Customer's future needs
 - Scalability: Can the software handle larger volumes of data?

Activities Common to Software Projects

- Requirements and specification
 - Domain analysis
 - Defining the problem
 - Requirements gathering
 - Obtaining input from as many sources as possible
 - Requirements analysis
 - Organizing the information
 - Requirements specification
 - Writing detailed instructions about how the software should behave

Activities Common to Software Projects

- **Design**
 - Deciding how the requirements should be implemented, using the available technology
 - Includes:
 - *Systems engineering*: Deciding what should be in hardware and what in software
 - *Software architecture*: Dividing the system into subsystems and deciding how the subsystems will interact
 - *Detailed design* of the internals of a subsystem
 - *User interface design*
 - *Design of databases*

Activities Common to Software Projects

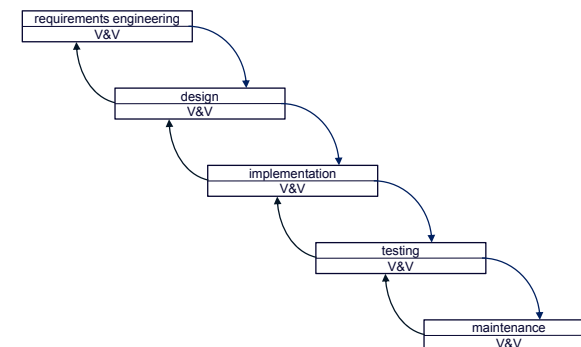
- **Modeling**
 - Creating representations of the domain or the software
 - Use case modeling
 - Structural modeling
 - Dynamic and behavioral modeling
- **Programming**
- **Quality assurance**
 - Reviews and inspections
 - Testing
- **Deployment & maintenance**
- **Managing the process**

Software Engineering Projects

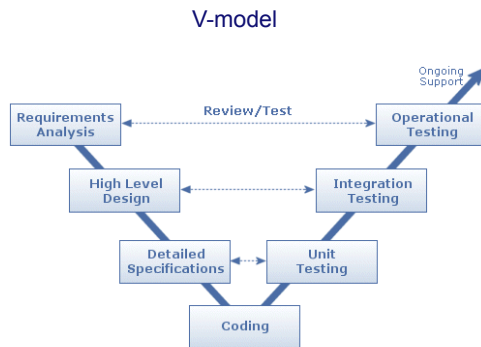
- Most projects are *evolutionary* or *maintenance* projects, involving work on *legacy* systems
- Corrective projects: fixing defects
- Adaptive projects: changing the system in response to changes in
 - Operating system
 - Database
 - Rules and regulations
- Enhancement projects: adding new features for users
- Reengineering or perfective projects: changing the system internally so it is more maintainable

Software development model

Waterfall model



Software development model



Software development model

- **Waterfall model**
 - Document oriented
 - Suited for (very) large projects (> 50 people)
 - Too many design activities during coding and testing

Software development model

- **Agile methods**
 - Individuals and interactions are more important than processes and tools
 - Working software is more important than comprehensive documents
 - Customer collaboration is more important than contract negotiation
 - Responding to change is more important than following a plan

Software development model

- **Agile methods**
 - No extensive architectural or design phase
 - No energy spend on documentation