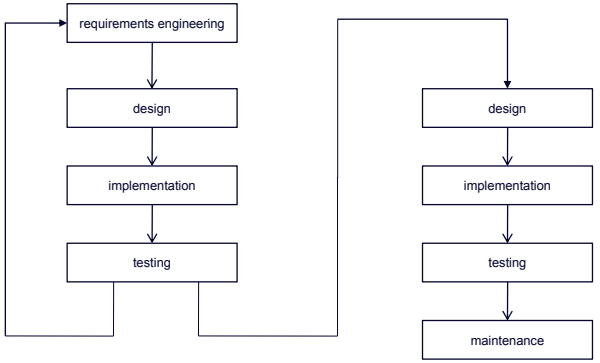
Vragen

- Noem de belangrijkste activiteiten in een software engineeringsproject
- Welke vormen van onderhoud kan men onderscheiden?
- Karakteriseer het waterval model
- Geef voor- en nadelen van *agile* ontwikkeling

Software development model

Prototyping



Software development model

- Advantages of prototyping
  - Resulting system is easier to use
  - Resulting system has less features
  - User needs are better accommodated
  - Design is of higher quality
  - Problems are detected earlier
  - Resulting system is easier to maintain
  - Development costs less effort

Software development model

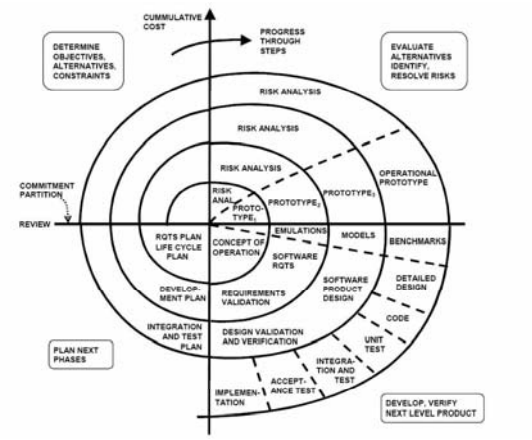
- Disadvantages of prototyping
  - Resulting system has more features
  - Design is of lower quality
  - Performance of resulting system is worse
  - Resulting system is harder to maintain
  - Team members should be more experienced

Software development model

- Prototyping
  - Is useful in situations with unclear or ambiguous requirements
  - Is useful in when emphasis is on user interface
  - Users and designers must be aware of pitfalls
  - Must planned and controlled

Software development model

- Incremental development



Software development model

- Incremental development
  - First focusing on essential features
  - Additional functionality is only included if needed
  - Resulting systems are leaner but provide sufficient support to the user

Software development model

- Rapid application development (RAD)
  - Similar to iterative development process models
    - User involvement
    - Prototyping
    - Reuse
    - Automated tools
    - Small development teams
  - Time boxing

Software development model

- RAD has four phases:
  - Requirements planning
  - Application design
  - Construction
  - Cutover (testing, training, installation)
- MoSCoW:
  - Must haves
  - Should haves
  - Could haves
  - Won't haves

Software development model

- Dynamic systems development method (DSDM)
- Builds on RAD
- 5 phases:
  - Feasibility study
  - Business study
  - Functional model iteration
  - Design and build iteration
  - implementation

Software development model

- Extreme programming (XP) principles:
  - Rapid feedback
  - Simplicity
  - Incremental change
  - Embracing change
  - Quality work

Software development model

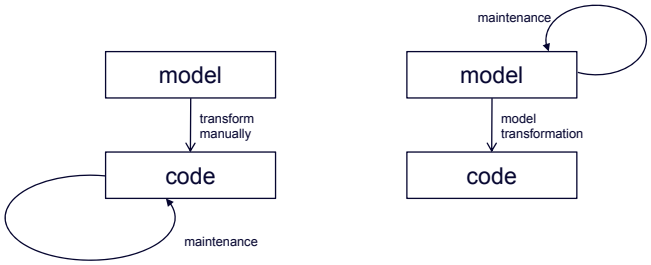
- Extreme programming in practice:
  - Planning game
  - Small releases
  - Metaphor
  - Simple design
  - Testing
  - Refactoring
  - Pair programming
  - Collective ownership
  - Continuous integration
  - 40-hour week
  - On-site customer
  - Coding standards

Software development model

- Rational Unified Process (RUP) in practice:
  - Iterative development
  - Requirements management
  - Architecture and use of components
  - Modeling and UML
  - Quality of process and product
  - Configuration and change management
  - Use-case driven development
  - Process configuration
  - Tool support

Software development model

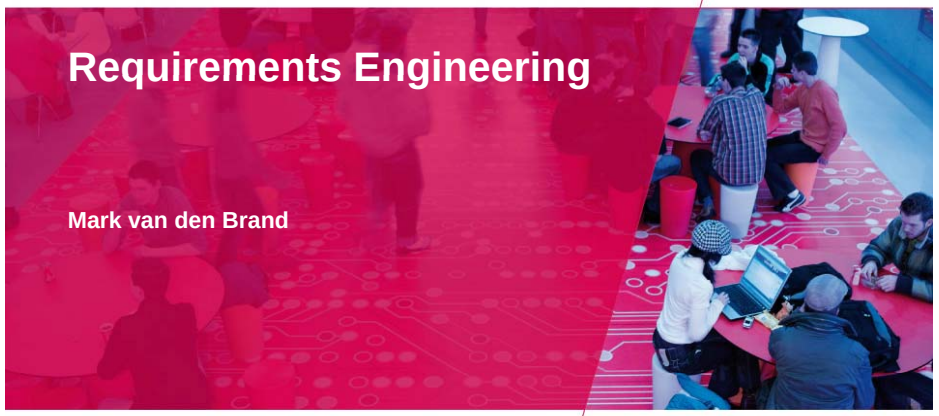
- Model-driven architecture (MDA)
  - Traditionally models are manually transformed into code
  - MDA advocates model transformations



Software development model

- Several models
  - CIM (computation independent model)
  - PIM (platform independent model)
  - PSM (platform specific model)
  - Code

Requirements Engineering

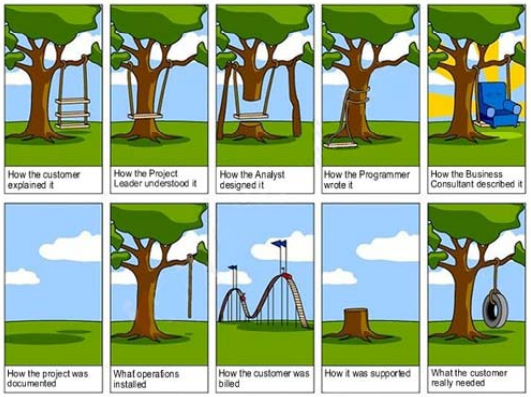


Mark van den Brand

Technische Universiteit Eindhoven University of Technology

Where innovation starts

Requirements Engineering



Domain Analysis

- The process by which a software engineer learns about the domain to better understand the problem:
  - The *domain* is the general field of business or technology in which the clients will use the software
  - A *domain expert* is a person who has a deep knowledge of the domain
- Benefits of performing domain analysis:
  - Faster development
  - Better system
  - Anticipation of extensions

Domain Analysis document

A. Introduction
B. Glossary
C. General knowledge about the domain
D. Customers and users
E. The environment
F. Tasks and procedures currently performed
G. Competing software
H. Similarities to other domains

Example of domain analysis document

- http://www.site.uottawa.ca/~laganier/seg3700/cemdomain.htm

Starting Point for Software Projects