

Vragen

- Noem de belangrijkste activiteiten in een software engineeringproject
- Welke vormen van onderhoud kan men onderscheiden?
- Karakteriseer het waterval 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

- See <http://agilemanifesto.org/>
- **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

- **Processes and tools**
 - Fixed hierarchy, roles and team structure
 - Many, many rules
 - Management of process, not people
 - Emphasis on process, not customer value

Software development model

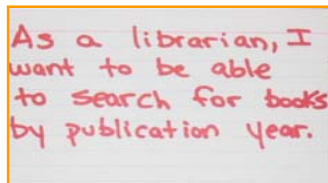
- XP practices
 - Co-location
 - Self organizing teams
 - Pair programming
 - Collective code ownership

Software development model

- Comprehensive documentation
 - **ESA Life cycle:**
 - User Requirements Document
 - Software Requirements Document
 - Architectural Design Document
 - Detailed Design Document
 - Software Transfer Document
 - Project History Document
 - Software Project Management Plan
 - Software Configuration Management Plan
 - Software Validation and Verification Plan
 - Software Quality Assurance Plan
 - Meeting minutes
 - Progress reports
 - ...

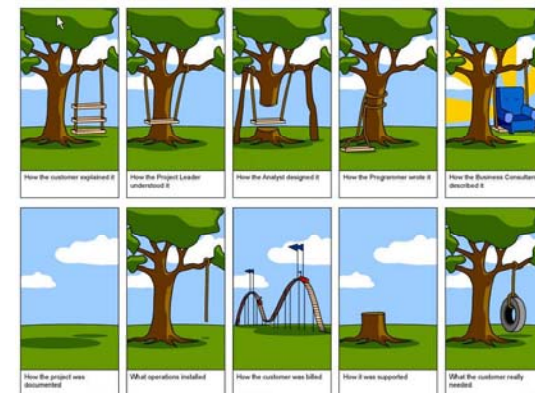
Software development model

- Working software
 - User stories
- Unit testing
- Continuous build and integration



As a librarian, I want to be able to search for books by publication year.

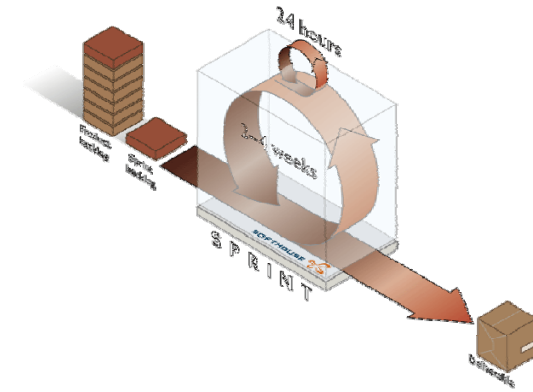
Software development models



Software development model

- **Customer collaboration**
 - **Scrum: Prioritized backlog of user stories**
 - **Must be accessible for questions**
 - **Frequent delivery of working software**
 - **Acceptance testing**

Software development model



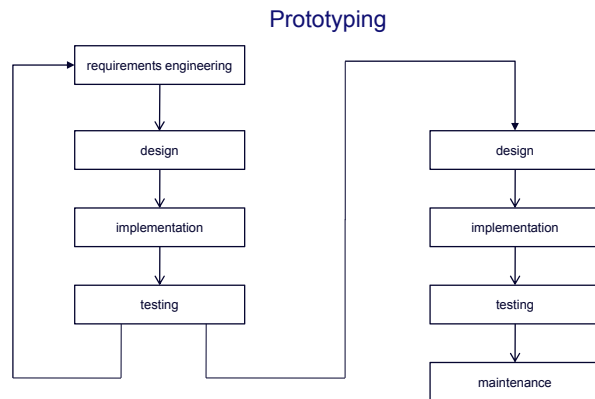
Software development model

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

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



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

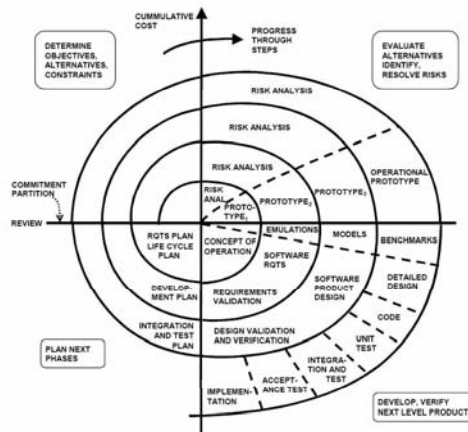
- **Disadvantages of prototyping**
 - Resulting system has more features
 - Design is of lower quality
 - Performance of resulting system is worse
 - Resulting system is harder to maintain
 - Team members should be more experienced

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**