

# Software Life Cycle

Professor Domanski

CSc430 – Software Engineering

# Types of Software Life Cycle Activities

1. *Feasibility* – is proposal worthwhile
2. *Market Analysis* – is there a market?
3. *Requirements* – functionality?
  1. From the user?
  2. What structures & tasks are common
4. *Project Planning* – how to develop the software
  1. Cost Analysis – find cost estimates
  2. Scheduling – build the schedule
  3. Quality Assurance – what will help ensure quality
  4. Work Breakdown – what are the subtasks?

# Types of Software Life Cycle Activities (cont'd)

1. *Design* – how software will provide functionality
  1. Architecture – system structure
  2. Interface – specify interfaces between different system parts
  3. Detailed – Algorithms
2. *Implementation* – Building the software

# Types of Software Life Cycle Activities (cont'd)

1. Testing - Executing with data
  1. *Unit Testing* – by original developer
  2. *Integration Testing* – testing when different parts are put together
  3. *System Testing* – testing in a close-to-actual environment
  4. *Alpha Testing* – by customer at the developers' site
  5. *Beta Testing* – by customer at the customers' site
  6. *Acceptance Testing* – to satisfy the purchaser
  7. *Regression Testing* – Saving tests from past versions to ensure new version retains all the previous capabilities

# Types of Software Life Cycle Activities (cont'd)

1. *Delivery* – providing customer with an effective software solution
  1. Installation – at customers' site
  2. Training – Teaching users how to use it
  3. Help Desk – Answering User Questions
2. *Maintenance*
  1. Updating & improving the software to ensure continued usefulness.

# Typical Documents

1. *Statement of Work* – Preliminary description of desired capabilities, by user
2. *Requirements Specification* – What the finished software will do
  1. *Object Model* – identifies main objects / classes
  2. *User Case Scenarios* – possible user behaviors

# Typical Documents (cont'd)

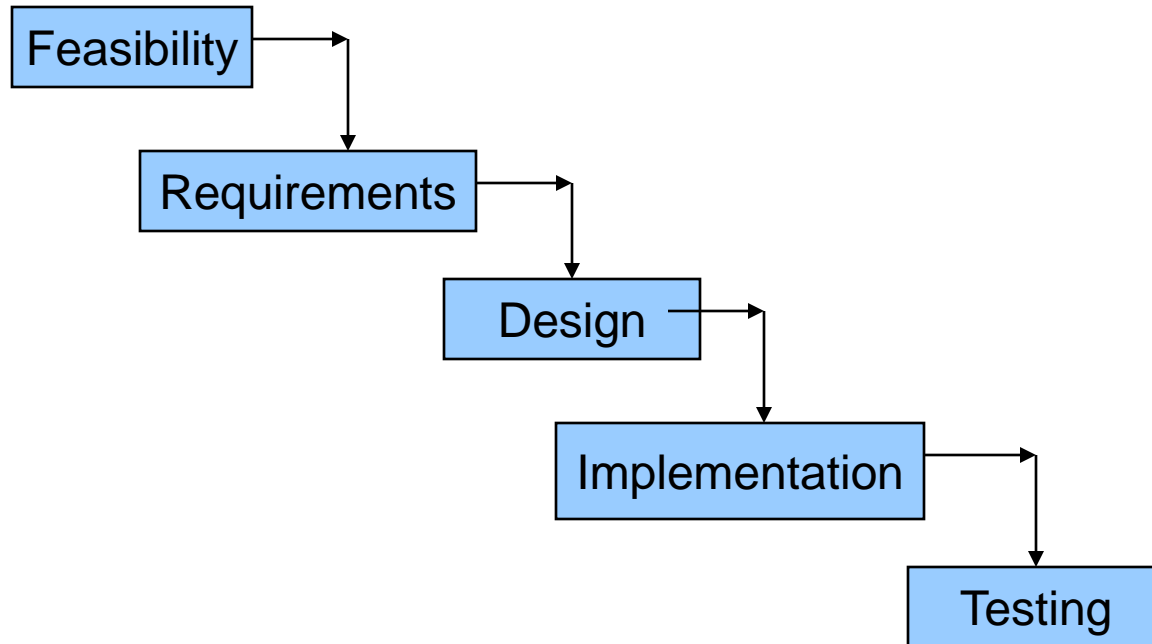
1. *Project Schedule* – order of tasks and time estimates
2. *Test Plan* – how will it be tested
  1. Acceptance Tests – by customer to determine acceptance
3. *Software Design* – software structure
  1. Architectural – high-level with interconnections
  2. Detailed – low-level modules or objects

# Typical Documents (cont'd)

1. *Software Quality Assurance (SQA) Plan* – describes activities that will ensure quality
2. *User Manual* – How to use the software
3. *Source Code* – The actual product code
4. *Test Report* – What tests were done & how the system behaved
5. *Defect Report* – describes customer dissatisfaction, specific system behaviors/failures/errors

# Software Life Cycle Models

## *Linear Sequential - Waterfall*



# Software Life Cycle Models (cont'd)

- **The Prototyping Model**
  - throwaway version to test concepts
  - Gain customer approval
  - Avoid developing unnecessary features
- **Incremental Model**
  - deliver a minimal set of functions
  - iterate and add minimally
- **Boehm's Spiral Model**
  - A spiral in the middle
  - Constantly revisits customer communication, planning, risk analysis, engineering, construction & evaluation