NC STATE UNIVERSITY

Software Lifecycle and Task Planning

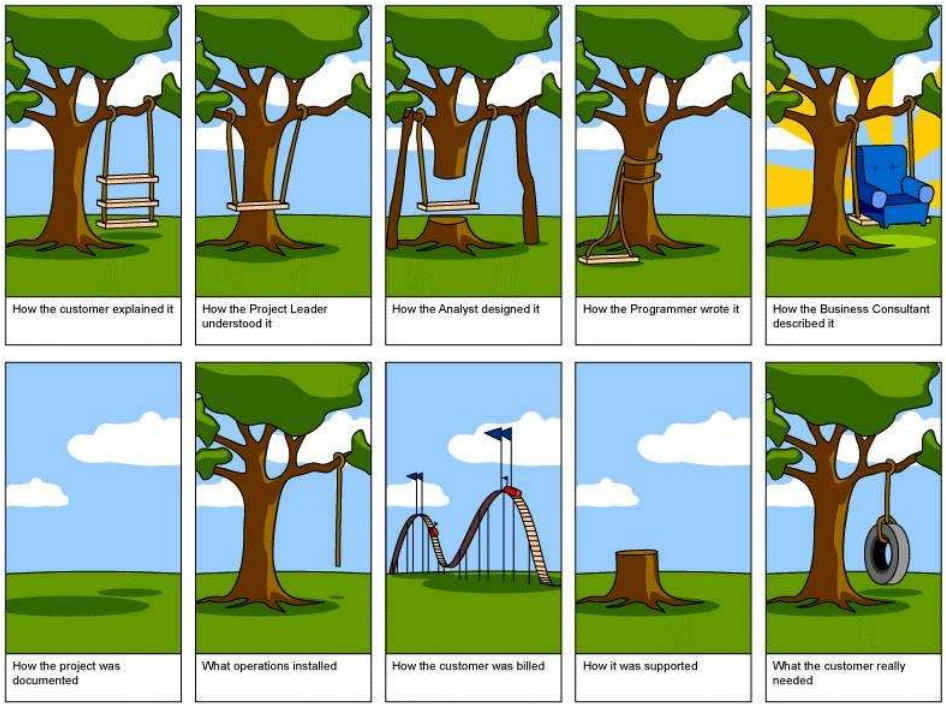
CSC 216

2

NC STATE UNIVERSITY

SOFTWARE LIFECYCLE

3



Software Process

- Software Process Development Phases
  - Requirements/Analysis
  - Design
  - Implementation/Code
  - Testing
  - Integration
  - Maintenance
- Software Artifacts: any document, file, or tangible creation related to the software or its creation or maintenance.

NC STATE UNIVERSITY



Requirements/Analysis

- Goal: Understand customer requirements for the software system
- Very difficult to “get right” the first time through
- Requirements may evolve over the course of developing a software system.
- Software Artifacts: requirements documents, use cases, user stories

6

NC STATE UNIVERSITY



Design

- Goal: decide the structure of the software and the hardware configurations that support it.
- How individual classes and software components work together in the software system
  - Small programs: ~ 10 classes and interfaces
  - Medium programs: 1000s of classes and interfaces
- Design for single user applications different from web applications and mobile applications
- Software Artifacts: design documents, class diagrams, other UML diagrams

7



Implementation/Code

- Goal: translating design into a concrete system (i.e. code)
- Can use any language, but some languages are better suited to certain types of programs than others
- Software Artifacts: source code, documentation, configuration files, media, executables, bug database, source code repository



Testing

- Goal: Execute software with intent of finding errors
- While you can't test until there is code to run, you can start planning testing when your analyzing the requirements
- Software Artifacts: test code (including test harnesses, scaffolding, etc.), bug database, test database, test inputs and outputs, documentation



Integration

- Goal: to bring the entire software system together
- Individual programmers or pairs work together on parts of the system. Integration brings all the pieces together into a working whole
- Software Artifacts: APIs, interfaces, integration documentation



Deployment

- Goal: to deliver software to a customer
- Software engineers package software and the associated documentation for delivery to customers
- Artifacts: executable(s), documentation, install instructions, APIs

NC STATE UNIVERSITY

Maintenance

- Goal: upgrade and fix the software after delivery
- Just because you deliver your software doesn't mean you're done with it
  - Work on the next version
  - Fix bugs in the current version
- Software Artifacts: all!

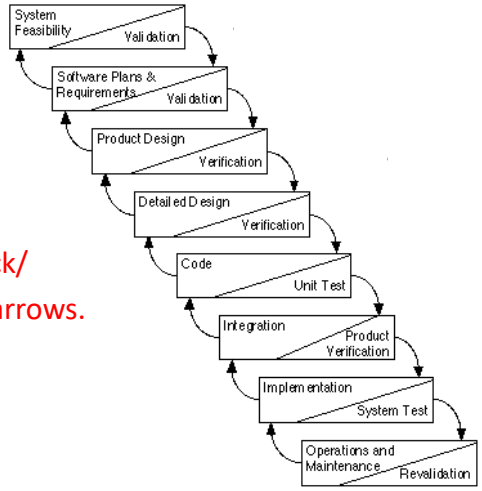
NC STATE UNIVERSITY

Software Models

- Software Models are generalized approaches to software development that incorporate the phases in different ways
  - Waterfall (Plan-Driven)
  - Iterative (Agile)

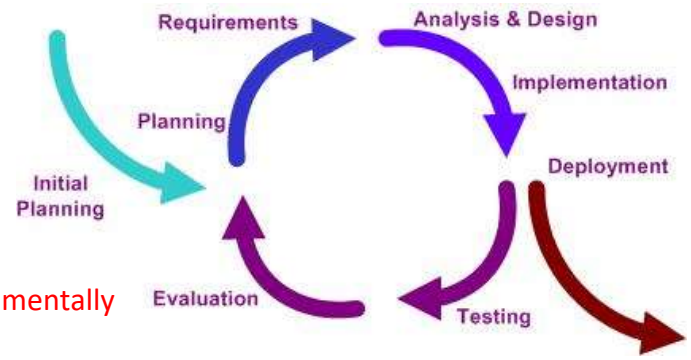


The Waterfall Model of the Software Life Cycle



Notice back/ feedback arrows.

Iterative Model

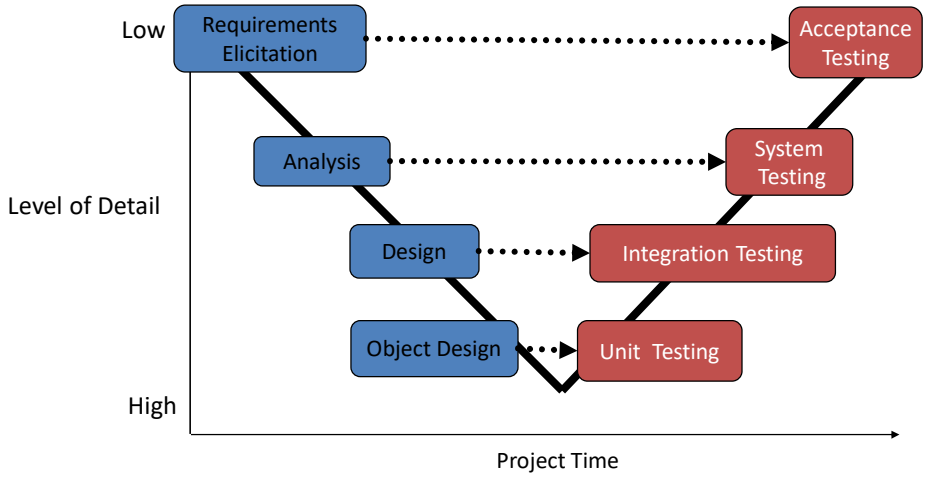


Iterative fundamentally means "**redo**."

Output from an iteration is examined for modification.

NC STATE UNIVERSITY

Development and Verification Activities



16

NC STATE UNIVERSITY

Software Engineering Best Practices

- Best practices have been shown empirically to:
  - Increase programmer productivity
  - Reduce software faults and failures
  - Support collaboration
  - Support the development of high quality software
- Support the major phases of development
- May be used in waterfall or iterative process models
- May be supported by tooling

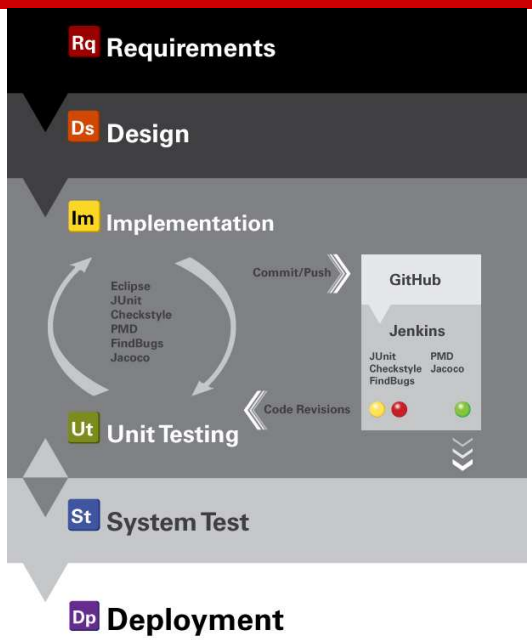
17

NC STATE UNIVERSITY

Best Practices

- Design Practices
- Integrated Development Environments
- Programming Languages
- Reviews and Inspections
- Static Analysis
- Testing Practices
- Code Coverage
- Version Control
- Continuous Integration

NC STATE UNIVERSITY



CSC 216/217 Lifecycle

TASK PLANNING

Task Planning

- Project Management Activity
  - Preferably mentor guided
  - Typically conference room with white board
- Task Identification and Breakdown
- Estimates for task completion
  - Time (ideal hours)
  - Story point (unit-less relative measure)
- Estimates are put together to create a schedule



NC STATE UNIVERSITY

Benefits of Task Planning

- Definition of Project Scope
  - Identify expectations of instructors
- Illumination of Project Deliverables
  - Evolve culture of software development beyond “writing code”
- Accountability for teams
  - Reduces conflict by identifying tasks
  - Increases leadership (including technical leadership and ownership)

NC STATE UNIVERSITY

Task Identification and Breakdown

- Start with ***deliverables*** and brainstorm the activities needed to complete them
  - Distinct activities become tasks
  - Most deliverables will consist of several tasks
    - Some explicitly stated
    - Some implied

z4bco6

Project Part 1 Deliverables and Tasks

Ds Design Document and Rationale

- ☑ – Read requirements and identify candidate classes, state, and behavior
- ☑ – Consolidate candidate classes, state, and behavior into a class diagram that represents a potential solution
- ☑ – Write a rationale that addresses questions and rubric items.

St System Test Plan

- ☑ – Identify the inputs and outputs to the system
- ☑ – Write five tests that provide explicit inputs and outputs
- ☑ – Create overview describing how to run the tests

Project Part 2 Deliverables

Rq Project solution that
Ds implements teaching staff
Im design and requirements

- Ut – Solution code
- Dp – JUnit tests for 80% statement coverage of non-GUI classes
- – Meaningful, complete comments and generated Javadoc
- – Removal of static analysis notifications
- – Passes teaching staff unit tests (meets the provided design)

Im Process Points

- – Skeleton
- – Partial solution

St System Test

- – Revise as appropriate
- – Report actual results of execution

☑ Project 2 Part 2 Tasks

- Read the project writeup
- Create Eclipse project with correct name
  - Commit to provided repository
- Create code skeleton
  - Create packages
  - Create classes
  - Copy in provided code
  - Create fields
  - Create methods with default return values (0, null, false, etc.)
  - Push to GitHub and fix any compiler errors
- Javadoc!
- Implement and unit test
  - There will be a suggested order of implementation
- Use teaching staff test feedback to write tests for debugging
  - Debug/test/debug/test…
- Generate Javadoc
- Run BBTP
- Deploy!

Estimating Size

- Ideal Time: the amount of time “something” takes when stripped of all peripheral activities
  - Example: American football game = 60 minutes
- Elapsed Time: the amount of time that passes on the clock to do “something”
  - Example: American football game = 3 hours
- Story Point: relative, unit-less measure expressing the size of the task, relative to all other tasks

NC STATE UNIVERSITY

Scheduling

- Incorporate the tasks and your estimates into your schedule
  - Look at the rest of your classes
  - Identify chunks of time that you can work on the project
    - Schedule them and stick to it!
  - Prioritize tasks based on dependencies
    - When starting a work session know what you'll be working on next!
  - Build in contingency time
  - Start early and ask questions

NC STATE UNIVERSITY

References

- http://www.di.net.au/blog/index.php/2009/05/the-lighter-side-of-the-software-process/
- Laurie Williams, Process Models slides: http://agile.csc.ncsu.edu/SEMaterials/ProcessModels.pdf
- Margaret Heil, Mentored Task Planning resources
- Laurie Williams, Planning & Estimating slides