

Software Engineering for Capstone Courses

Richard Anderson
CSE 481b
Winter 2007

Announcements

- HW 2, Due Thursday, Jan 19
- Presentations, Tuesday Jan 23
 - 15 minute presentation + 3 minutes discussion
 - PowerPoint slides
 - Group order: A, B, C, D

Today's lecture

- Software Engineering vs. Computer Science
- Software Life Cycle
- Requirements
- Risk Analysis

Key aspects of software engineering

- Large scale projects
- Long lasting projects
- External constraints
 - E.g., Make money, Support a business process, Don't kill anyone
- Life cycle that includes many non-programming activities

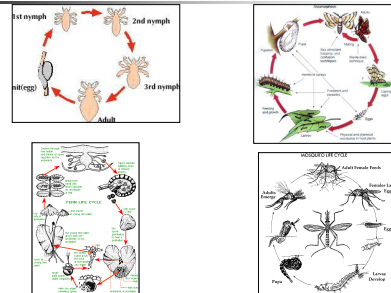
SE for Capstone Courses

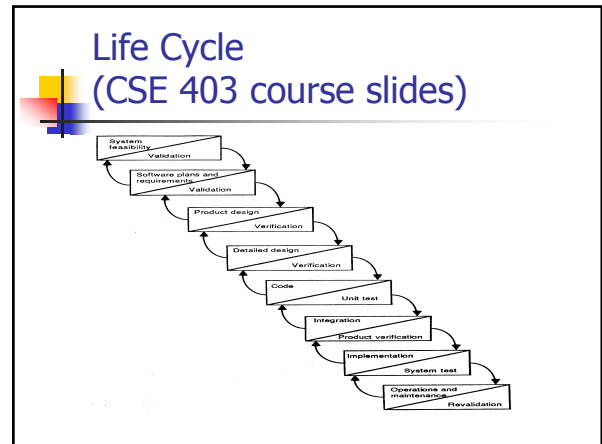
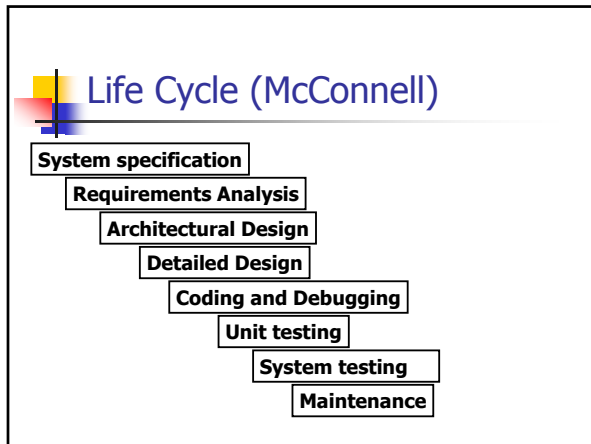
Computer Science

Software Engineering

- Process Useful
- Partial Life Cycle
- Realistic Challenge
- Meet a Delivery

Life Cycle





- ## Requirements on requirements
- Who are they for?
 - What are they for?
 - Pitch to management
 - Fodder for market study
 - Basis for legal contract
 - Easy to understand, concise, complete, unambiguous, . . .

- ## Requirements
- "Gather and document the functions that the application should perform for the users in the users' language and from the users' perspective"
 - Requirements should neither constrain nor define methods of implementation

- ## Customers
- (Almost) every large software project has a customer who is paying the bills
 - Project requirements driven by this customer

- ## Approaches to requirements
- Personas
 - Scenarios
 - Use cases

Project Pitch

- What you are going to do
- What value it delivers to the customer
- What is the novelty
- What are the risks

Is dog food good for you?

- Dog food (verb), to use your own software



Risk

- Exposure to the chance of injury or loss
- For a software project:
 - Failure to deliver on time
 - Exceeding resource limits
 - Not meeting quality threshold

Risk analysis

- Types of Risk
 - Code Development
 - External Dependency
 - Technology
 - Personnel
 - Requirements Change

Sources of Risk I

- Development risks
 - Code harder to develop than thought
 - Learning curve on new facilities
 - Expected facilities not available
 - Need to iterate on requirements / design
 - Performance Issues
 - Trigger other bugs

Sources of Risk II

- Integration risks
 - Parts don't fit together
 - Integration reveals bugs
 - Integration reveals design errors
 - Need to rewrite code after integration
 - Code left out

Sources of Risk III

- Testing risks
 - Bugs will be found
 - Bugs won't be found
 - Complexity of testing matrix
 - Deployment beyond development machines
 - Difficulties in test automation and test tools
 - UI and Workflow feedback

Sources of Risk IV

- Requirements Risks
 - New requirements introduced
 - Change in Specification
 - Inconsistencies in requirements
 - User feedback
 - Market conditions
 - Platform and technology changes

Sources of Risk V

- Deployment Risks
 - Packaging distributable
 - Rights and licensing of components
 - Legal signoff
 - Marketing signoff
 - Systems configuration

Sources of Risk VI

- People risks
 - Unexpected Loss of Personnel
 - Illness
 - Vacation
 - Other demands on time
 - Group friction
 - Inaccurate evaluation of skills
 - Drop in performance

What to do with risk analysis

- Avoid the risk
- Transfer risk off the critical path
- Buy information
 - Bring in outside help
 - Prototype
- Publicize risk
 - The sky is falling
- Schedule to accommodate some risk
- Monitor risks as project progresses