Software Lifecycle Models and Software Process

• Software lifecycle basics
• Software lifecycle models
  – build-and-fix
  – waterfall
  – rapid prototype
  – incremental and iterative
  – spiral
• Software process examples
  – XP and RUP
• Process improvement
  – CMM & ISO9000
Software Lifecycle

• A series of steps through which a software product progresses
• Lifetimes vary from days to months to years
• Consists of
  – people!
  – overall process
  – intermediate products
  – stages of the process
What is a process?

• “Device” for producing a product (get job done)
• Level of indirection
  – Process description describes wide class of instances
• Humans create process descriptions to solve classes of problems
• Thus
  – software processes are “devices” for creating and evolving software products
Intermediate Software Products

• Objectives
  – Mark the “end” of phases
  – Enable effective reviews
  – Specify requirements for next phase
    • Note the abstract requirements/design cycle

• Form
  – Rigorous
  – Machine processible (highly desirable)

• Content
  – Specifications, Tests, Documentation
Phases of a Software Lifecycle

• Standard Phases
  – Requirements Analysis & Specification
  – Design
  – Implementation and Integration
  – Operation and Maintenance
  – Change in Requirements
  – Testing throughout

• Phases promote manageability and provide organization
Build-and-Fix

Build First Version

Modify until Client is satisfied

Operations Mode

Retirement
Gantt Chart

- Plan activities
- Allocate resources
- Define dependencies
- MS Project
Rapid Prototyping

Rapid Prototype
- Verify

Design
- Verify

Implementation
- Test

Operations

Retirement

Req. Change

---

CAL POLY
Incremental\textsuperscript{1} and Iterative\textsuperscript{2}


For each build: Perform detailed design, implement, test, deliver.

Increments add new features

Could be staggered

Or consecutive

Requirements

Verify

Arch. Design

Verify

Operations

Retirement
For each build: Perform detailed design, implement, test, deliver.

Iterative

Requirements

Verify

Arch. Design

Verify

For each build: Perform detailed design, implement, test, deliver.

Could be minimized or incorporated into iterations (common with agile approaches)

Operations

Retirement

Iterations may revisit features
The Spiral Model

Add a Risk Analysis step to each phase!

More recently extended into MBASE (Model-Based Architecting and Software Engineering)
Comparing Engineering Disciplines

• What is design? What is construction?
  – In civil and mechanical engineering, design makes up about 10% of the project
  – Design completed by creative, intelligent people
  – Construction completed by skilled but often less educated people

• Jack Reeves suggested that source code is the design document and compilers/linkers do the construction for free.
  – What are the implications of this view?
Comparing SD Processes

- Processes vary in complexity and control
Modern Process 1

- Rational Unified Process (RUP)
  - iterative and incremental
  - use-case driven
  - component-based architectures
  - visually modeled with UML
  - quality verification
  - tools based
Phases

Inception  Elaboration  Construction  Transition

Business Modeling
Requirements
Analysis & Design
Implementation
Test
Deployment
Configuration & Change Mgmt
Project Management
Environment

Iterations

Initial  Elab #1  Elab #2  Const #1  Const #2  Const #N  Tran #1  Tran #2

Click on an element for more information.
Modern Process 2

- eXtreme Programming (XP)
  - lightweight process
  - frequent iterations
  - Best practices “in the extreme”
    - continuous integration
    - pair programming
    - test-driven development
    - refactoring
  - popular agile method
XP Practice Coupling

Refactoring

- Improving the structure of code without changing its behavior
- Removing code smells
Comparing Modern Processes

- RUP vs. XP
Capability Maturity Model (CMM)

• CMM is not a software lifecycle model...
  – Strategy for improving the software development process regardless of the process “model” followed
    • Basic premise: the use of new software methods alone will not improve productivity and quality, because software management is, in part, the cause of problems
  – CMM assists organizations in providing the infrastructure required for achieving a disciplined and mature process ($$)

• Includes
  – technical aspects of software production
  – managerial aspects of software production
Capability Maturity Model (continued)

- Five maturity levels
  - 1. initial – ad hoc process
  - 2. repeatable process – basic project management
  - 3. defined process – process modeling and definition
  - 4. managed process – process measurement
  - 5. optimizing process – process control and dynamic improvement

- To move from one stage to the next, the SEI provides a series of questionnaires and conducts process assessments that highlight current shortcomings
ISO 9000

• Further attempt to improve software quality based on International Standards Organization (ISO)

• ISO 9000 = series of five related standards
  – within ISO 9000 standard series ISO 9000-3 focuses on software and software development

• Basic features:
  – stress on documenting the process in both words and pictures
  – requires management commitment to quality
  – requires intensive training of workers
  – emphasizes measurement
ISO 9000

- Adopted by over 60 countries (USA, Japan, European Union, ...)
- To be ISO 9000 compliant, a company’s process must be certified
Process Measures

- Quality (e.g. faults per KLOC)
- Functionality (e.g. function points)
- Productivity (e.g. function points per hour)
- Maintainability (e.g. effort to make change)
- Customer satisfaction (e.g. survey results)
Product Measures

- External Quality (e.g. faults)
- Internal Quality
  - Coupling, cohesion, complexity
- Size (e.g. LOC, #classes, #methods)
- Testability (e.g. tests per method)
- Scalability (e.g. users supported)
- Performance (e.g. transactions per second)