CIS 422/522

Software Life cycles and Process models I
Definition

• Software Life Cycle: evolution of a software development effort from concept to retirement
• Life Cycle Model: Abstract representation of a software life cycle as a sequence of 1) activities or phases and 2) products (usually graphic)
• Software Process (process model): institutionalized version of a life cycle model. Usually intended to provide guidance to developers.
Rationale

• Developed as a tool for gaining and maintaining control over complex software development processes

• Application of “divide-and-conquer” to software processes and products
  – Goal: identify distinct and relatively independent phases and products
  – Can then address each separately

• Intended use
  – Provide guidance to developers in what to produce and when to produce it
  – Provide a basis for planning and assessing development progress
A Simple Process Model

From van Vliet
A “Waterfall” Model
Phases and Products

• Requirements
  – Goal: implementation-independent specification of what the software must do and any constraints on its development
  – Product: Software Requirements Specification (SRS)

• Architecture
  – Goal: decomposition of the problem into components that together satisfy the requirements within the constraints
  – Products: specifications of components, relations, interfaces

• Detail Design
  – Goal: internal design of components (e.g., objects) to identify appropriate algorithms and data structures supporting the interface
  – Products: design documentation, pseudo-code
Phases and Products

• Implementation
  – Goal: realization of the design in a machine-executable language
  – Product: code

• Testing
  – Goal: validation and verification of the implementation against requirements and design
  – Products: test plan, test cases

• Maintenance
  – Goal: maintain deployed system
  – Products: bug fixes, patches, new versions
Issues with Life Cycle Models

• Application of “divide-and-conquer” to software processes and products
  – Goal: identify distinct and relatively independent phases and products
  – Can then address each separately

• Intended use
  – Provide guidance to developers in what to produce and when to produce it
  – Provide a basis for planning and assessing development progress

• Never an accurate representation of what really goes on. (Why?)
A “Waterfall” Model

What are the problems:
1. As a model of a real process
2. As a process model to guide development
Iterative “Waterfall” Model

- Requirements Analysis
- Architecture
- Design
- Coding
- System Integration and Testing
- Deployment
- Maintenance and Evolution
Also…

- Spiral model
- Prototyping models
- RAD models
- Extreme Programming
- Etc., etc.
The Joys of Faking It

From: Parnas & Clements “A Rational Design Process”
Design Processes are Idealizations

• Assertion: Design is an inherently “irrational” process
• Completely rational processes proceed by a sequence of optimal steps (the right choice each time)
• Real processes rarely proceed rationally from goals to products
• This is an essential characteristic of the design process
  – It’s a human process
  – We’re neither omniscient nor omnipotent
It Pays to “Fake it”

• Thesis: It is nonetheless useful to “fake” a rational design process
  – Follow the ideal process as closely as possible
  – Write the documentation and other work products as is we had followed the ideal

• Rationale
  – Idealized process can provide guidance
  – Helps come closer to the ideal (emulation)
  – Helps standardize the process (provide a common view of how to proceed and what to produce)
  – Provides a yardstick for assessing progress
  – Provides better products (e.g. final draft not first)
Idealized Process

• Establish and document requirements
• Design and document the module structure
• Design and document the module interfaces
• Design and document the uses hierarchy
• Design and document the module internal structure
• Write programs
• Maintain
Current Nomenclature

• This view of the ideal process is now common
• The Software Design Process is concerned with specifying:
  – software architecture (structure)
  – software components (interfaces)
  – interconnections among components (e.g., uses)
  – internal details of the components, (internal structure)

--- so that the resulting system satisfies its requirements and pre-determined design criteria (goals, assumptions, constraints)
How do we Choose a Development Process?

E.g., for your projects
Objectives

• Goal: proceed as rationally and systematically as possible (i.e., in a controlled manner) from a statement of goals to a design that demonstrably meets those goals with design and management constraints
• Understand that any process description is an abstraction so we must compensate for deviation
A Software Engineering Perspective

- SE view provides perspective on life-cycle activities (e.g., Design)
- Choose processes, methods, notations, etc. to provide an appropriate level of control for the given product and context
  - Sufficient control to achieve results
  - No more than necessary to contain cost and effort
- Provides a basis for choosing or evaluating processes, methods, etc.
  - Does it achieve our objectives at reasonable cost?
  - E.g., does this notation provide a handle on the properties of interest?
“Appropriate” Control

• What constitutes “appropriate” control will be vastly different for different types of developments
  – Large vs. small
  – New problems vs. old
  – High risk vs. low, and so on

• These are neither independent nor exclusive

• Development approaches vary in their assumptions about these issues
  – E.g., RAD vs. Spiral
  – In general, we will consider the “large system” product context common to business (multi-person, multi-version)
• Need to agree on kind of control you need and how you will accomplish it
• Process model (description) will then help keep everyone on track
  – Basis for planning and scheduling
  – Each person knows what to do next
  – Basis for tracking progress against schedule
• Should be one of the first products but expect it to evolve
Assignment

- Reading:
  - Text: Chapters 3 (Process Models)

- Project
  - Discuss which process model is appropriate for your project.