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
  - Identify distinct phases of development and distinct products
    - Requirements phase – understand the problem to be solved
    - Product – Software Requirements Specification
  - Assumption: Simpler to address each phase separately
    - E.g., Elicit, specify, and validate requirements before doing design
    - True to the extent dependencies between phases and products are limited

Common Process Models

- Waterfall
- Prototyping
- Iterative
- Spiral
- Agile

A “Waterfall” Model

What are the issues:
1. As a guide to how software should be developed?
2. As a model of any real development?
A “Waterfall” Model

1. As a guide: does not address some common development risks
   - What happens if requirements are wrong?
   - Is scheduling or budget is wrong?

2. As a model: unrealistic as a model of any real development
   - How do real developments differ?

Waterfall Model Variations

There have been many variations attempting to address these issues

Characteristic Model: Prototyping

- Waterfall variation
- First system versions are prototypes, either:
  - Interface
  - Functional
- Which waterfall risks does this try to address?

Characteristic Processes: The Iterative Model

- Process viewed as a sequence of iterations, each iteration produces an increment of the working software (sequence of waterfalls)
  - Build minimal useful subset, test, validate
  - Build next version by extending last iteration
  - After first iteration, always have working software
Iterative Model

• Also called “incremental development”
• Addresses some common waterfall risks
  – Risk that software cannot be completed – build incremental subsets
  – Risk of building the wrong system – stakeholder have opportunities to see the software
  – Also, feasibility, schedule, budget and others to some extent

Characteristic Processes: The Spiral Model

• Process viewed as repeating cycles of increasing scale
• Identify risks and determine (next set of) requirements, build next version by extension, increasing scale each time
• Early iterations may be prototypes

Spiral Model Goals

• Response lack of risk analysis and risk mitigation in “waterfall” process
  – Make risk analysis standard part of process
  – Address risk issues early and often
• Explicit risk analysis at each phase
• Framework for explicit risk-mitigation strategies
  – E.g., prototyping
• Explicit Go/No-Go decision points in process
Characteristic Processes: Agile (scrum, RAD, XP)

- Process viewed as nested sequence of builds (sprints)
  - Each build adds small feature set
  - Customer in loop, code centered (little or no documentation)
  - Problem detection and correction though daily team meetings (scrum)

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
  - Always must compensate for deviation from the ideal (e.g., by iteration)

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?
Project Relevance

- 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 you produce but expect it to evolve

Contents of a Process Specification

- Details depend on the purpose of the specification
- In general terms [Parnas & Clements]
  - What product we should work on next
  - Equivalently – what decision(s) must we make next
  - What kind of person should do the work
  - What information is needed to do the work
  - When is the work finished?
  - What criteria the work product must satisfy
- In personal terms, answers the questions
  - Is this my job?
  - What do I do next?
  - What do I need to do the work?
  - Am I done yet?
  - Did I do a good job?

Project Processes

- Discuss: what process elements are appropriate for your project?
- What are the products?
- What aspects of traditional models are irrelevant?
- What are the constraints?
  - Which aspects can’t be changed?
  - Which can be?
- What are the major risks?
- What are appropriate strategies to address the risks?

Project Planning
From Process to Plan

- Process definition manifests itself in the project plan
  - Process definition is an abstraction
  - Many possible ways of implementing the same process
- Project plan makes process concrete, it assigns
  - People to roles
  - Artifacts to deliverables and milestones
  - Activities to tasks over time
- Project plan should be one of the first products but expect it to evolve

Project Plan

- Minimal plan contents
  - Risks and mitigation strategies
    - Evolves with progress and understanding
  - Tasks to be performed
  - Person(s) assigned to roles and tasks
  - Deadline for each task
  - Sequencing among tasks
    - Task dependencies
    - Development plan
- Usually owned by team manager
- Updated as project proceeds

Project Plan Template

- Use the template provided in your Assembla team workspace (under the Wiki tab)
- This should be a living document
  - Changed as the project progresses
  - For the reader, reflects both the planned activities and provides a snapshot of the current project state

Work Breakdown Structure

- This is a technique to analyze the content of work and cost by decomposing it into its component parts. It is produced by:
  - Identifying the key elements
  - Decomposing each element into component parts
  - Continuing to decompose until manageable work packages have been identified. These can then be allocated to the appropriate role/person
- The WBS is used to allocate responsibilities
- For the software, the WBS depends on the software architecture (discuss next)
Milestone Planning

- Milestone planning is used to show the major steps that are needed to reach the goal on time
- Milestones typically mark completion of key deliverables or establishment of baselines
  - Baseline: when a work product is put under configuration management and all changes are controlled
- Often associated with management review points
  - E.g., Requirements baseline, project plan complete, code ready to test
- Can use Gantt charts

Assignment

- Reading:
  - Chap. 8
- Project
  - Familiarize with assembla
  - Begin filling out project plan
The Joys of Faking It

From: Parnas & Clements "A Rational Design Process"

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)

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