CIS 422/522

Software Processes Part 2
- Process Models
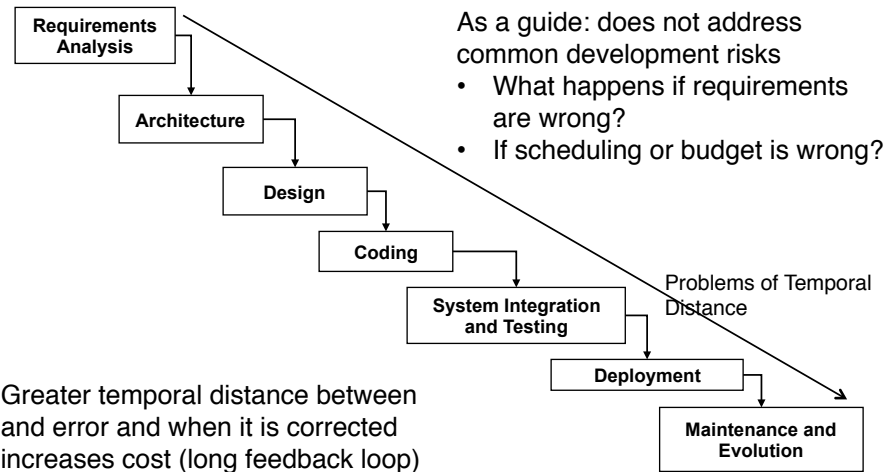- Choosing a Process
- Project workspaces/start

Software Processes

- Developed as a conceptual tool for organizing complex software developments
- Organize the work
- Address developmental risks
- Intended use (idealized)
    1. *Model* of development (what does or should occur)
   2. *Guide* to developers in what to produce and when to produce it

A "Waterfall" Model*

Requirements Analysis

Architecture

Design

Coding

System Integration and Testing

Deployment

Maintenance and Evolution

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

Problems of Temporal Distance

Greater temporal distance between and error and when it is corrected increases cost (long feedback loop)

CIS 422/522 © S. Faulk   3

Common Process Models

Waterfall
Prototyping
Iterative
Spiral
Agile

CIS 422/522 © S. Faulk   4

Characteristic Model: Prototyping

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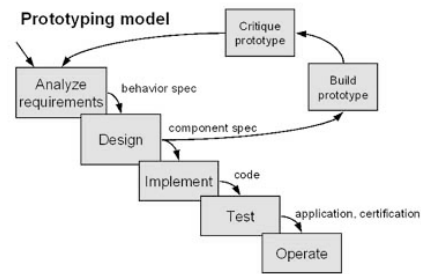
Characteristic Processes:
The Iterative Model

- Process is viewed as a sequence of iterations
  - Essentially, a *series of waterfalls*
  - Each iteration builds on the previous one (e.g., adds requirements, design components, code features, tests)
  - Each iteration produces complete set of work products deliverable software
  - Customers provide feedback on each release
  - There is no "maintenance" phase – each version includes problem fixes as well as new features
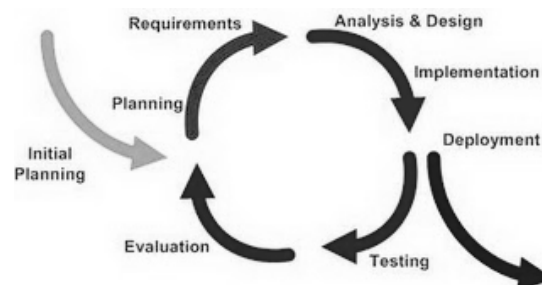
Iterative Model

- Also called "incremental development"
- Addresses some common waterfall risks
  - Risk that software cannot be completed – build incremental subsets
  - Risk of building the wrong system – stakeholders have opportunities to see parts of the software at each increment
  - Each iteration provides checkpoint for feasibility, schedule, budget and others issues

# Advantages of Incremental Development

- Customers get usable functionality earlier than with waterfall
- Early feedback improves likelihood of producing a product that satisfies customers
  - Reduces risk: e.g., if customers hate the product, find out before investing too much effort and money
- The quality of the final product is better
  - Core functionality is developed early and tested multiple times
  - Only a relatively small subset of functionality added in each release: easier to get it right and test it thoroughly
  - Detect design problems early and get a chance to redesign

Characteristic Processes:
The Spiral Model

- Process viewed as repeating cycles of increasing scale
- Identify risks and determine (next set of) requirements
- Each cycle builds next version by extension, increasing scale each time

Spiral Model

determine goals

Risk evaluation and Mitigation

plan next phase

development
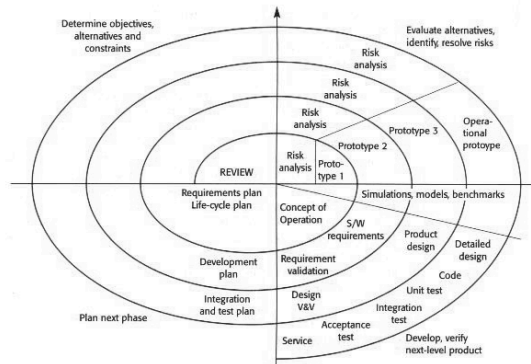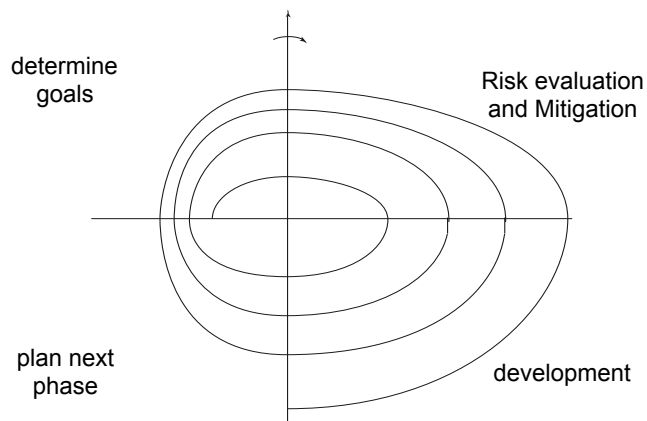
Spiral Model Characteristics

- Response to lack of explicit risk analysis and risk mitigation in "waterfall" process
- Includes risk analysis and mitigation activities at each phase (e.g., prototyping)
- Explicit Go/No-Go decision points in process
- "Heavy-weight" process: substantial overhead not contributing directly to end products

Characteristic Processes:
Agile (e.g. scrum)

- Process viewed as nested sequence of builds (sprints)
    - Each build adds very small feature set (one or two)
    - Small team, daily meetings
    - Nightly build/test, frequent customer validation (preferably on site)
    - Focus on delivering code, **little or no time spent on documentation**

Copyright © 2005, Mountain Goat Software

How do we Choose a Development Process?
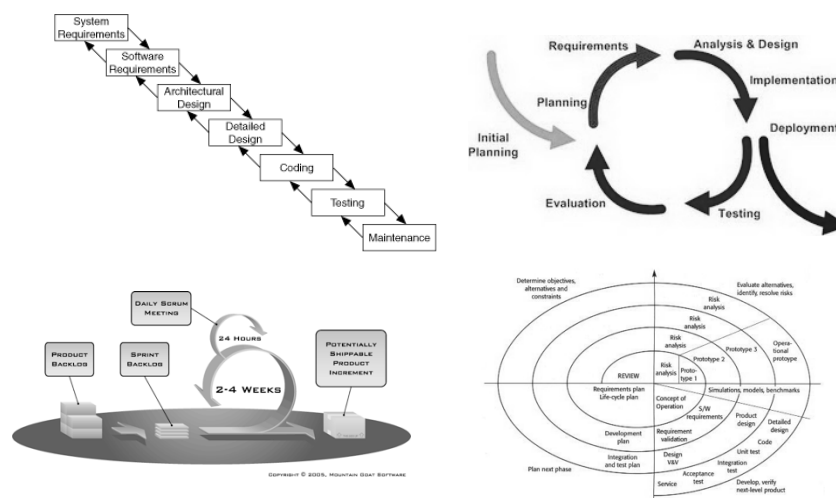
E.g., for your projects

Objectives

- Goal: proceed as systematically as possible (I.e., in a controlled manner), from a statement of goals, to a design that demonstrably meets those goals within design and management constraints
  - Understand that any process description is an abstraction
  - Real developments must compensate for deviation from the ideal (e.g., by iteration, backtracking, etc.)
  - Still important to have a well-defined process to follow and measure against

Choosing a Process

- Question of control vs. cost: processes introduce *overhead*
- Choose a process to provide an appropriate level of control for the given product and context
  - Sufficient control to mitigate risks, 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?
  - Does it address the most important developmental risks?
- To make a rational choice: need to agree on kind of control you need and how best to achieve it

Exercise: Which Model?

Exercise: Project Processes

- Discuss: Which process is the best fit for your projects and why?
- For each process you do not select: Which process characteristics do not fit well with our project?
- For the process selected
  - How do the process characteristics align with project needs?
  - How does it help address project risks?

Take-away

- Expected to know standard processes and their rationale
- Understand how and why people use different development models
- Understand how to choose an appropriate model for a given developments
  - Often poorly understood in industry

Project Preparation

Project Requirements
Assembla Worksites

Assignment: Project

- Goal: get clear on what you should build
  - Carefully review the Address Book requirements
  - Start creating a mock-up (on paper) of your application
  - Generate questions for instructor on detailed behavior
- Assembla: start drafting a project plan
  - Understand how to edit the Wiki pages
  - Identify team roles (even if they might change)
  - Lay out initial cut at your schedule (look at the class website Schedule page for deliverable due dates)
  - Create first meeting notes, developer logs
- For Monday: video lecture (link on Schedule page)

Questions?