Lifecycle

CSE 403, Winter 2003
Software Engineering

http://www.cs.washington.edu/education/courses/403/03wi/
Readings and References

• Reading
  » Chapters 5-10, *The Mythical Man-Month*, Brooks

• Other References
  » *Rapid Development*, Steve McConnell
    • Chapter 7, Lifecycle Planning
  » *Anchoring the Software Process*, Barry Boehm, USC, 1995
    • [http://citeseer.nj.nec.com/boehm95anchoring.html](http://citeseer.nj.nec.com/boehm95anchoring.html)
The dreams of yesterday

• Boehm [1995]
  » "For a few golden moments in the mid-1970's, it appeared that the software field had found a sequence of common anchor points"
  » "a sequence of milestones around which people could plan, organize, monitor, and control their projects"
A Lifecycle

• The main function of a lifecycle model is to establish order in which project events occur.

• Typical events include:
  » specification, prototype, design, implementation, test, deliver, and do it again.

• But they usually don't happen in nice clean little stages like this:
  » so we develop various models and tweaks to try to maintain the benefits and still be realistic.
"good enough" now vs "perfect" later

• The goal is often
  » not to achieve what you said you would at the beginning of the project
  » but to achieve the maximum possible within the time and resources available
    » Sherman 1995, reference in McConnell

• Do deliver a small and useful tool on time

• Don't deliver a monster way too late
  » Fancy doodads have a tendency to be junk anyway
Classic Waterfall

1. Software Concept
2. Requirements Analysis
3. Architectural Design
4. Detailed Design
5. Coding and Debugging
6. System Testing
Characteristics

• Orderly sequential model
• Stages are disjoint
  » they don't overlap and you can't go forward until you've completed the current stage
  » you can't go back except with extreme difficulty
• Reviews at each stage to determine if ready to advance to next stage
• Document driven
  » specific documents will be complete at each stage
    • yeah, right
Some issues

- Very difficult to specify all requirements completely and correctly all at once
  - completely → lots and lots of detail
  - correctly → every single detail is correct
  - produces masses and masses of detail that will be irrelevant if some early decision changes
- Difficult to accurately say everything at once
- Gold plating requirements is tempting
- Inflexible solutions based on invalid detail
More issues

• Since so much is resting on getting it right before leaving each stage, the reviews tend to be massive affairs
  » a lot of work goes into preparing for each review
  » that makes it even more expensive to change direction if a review shows problems
  » if a review is delayed or problems are found, the entire project sits in a loop while the problems are resolved - $$$
Salmon lifecycle model

You can go upstream, but it's hard
Code and fix
Code and fix is dangerous

• No means of assessing progress
  » nasty surprises are not a good thing for your career
  » Yes: "I'm not worried, I know where they are in the project and they always deliver a useful product on time."
  » No: "I don't know. You remember the project when they were 95% complete for three months and then cancelled?"

• Risk of complete project failure right up to delivery
Spiral Model

- Oriented towards phased reduction of risk
- Take on the big risks early and make some decisions
  - are we building the right product?
  - do we have any customers for this product?
  - is it possible to implement the product with the technology that exists today? tomorrow?
  - does the company want to be in this business?
    - should the company be in this business?
FIGURE 1. The Win-Win Spiral Model

1. Identify next-level Stakeholders

2. Identify Stakeholders’ win conditions


4. Evaluate product and process alternatives. Resolve Risks

5. Define next level of product and process - including partitions

6. Validate product and process definitions

7. Review, commitment

*Anchoring the Software Process*, Barry Boehm, USC, 1995
Spiral steps

- Determine objectives, alternatives, constraints
- Identify and resolve risks
- Evaluate alternatives
- Develop the deliverables for the iteration and verify that they are correct
- Plan the next iteration
- Commit to an approach for accomplishing the next iteration or cancel the project
Spiral early and often

• The spiral model is especially appropriate at the beginning of the project when the requirements are still fluid

• Risk reduction is the key element
  » early cancellation of bad projects is a major benefit
  » confidence that you're building the right product is a major benefit
Milestones

• Key elements of project milestones
  » stakeholder concurrence on the system's objectives
  » determination and validation of system architecture

• Traditional
  » Requirements review, preliminary design review, final design review, acceptance test

• Boehm Spiral
  » Lifecycle Objectives, Lifecycle Architecture, Initial Operating Capability
Elements of Lifecycle Objectives (LCO)

- **Operational Concepts** - *What is it?*
  - Top level system objectives and scope
- **System Requirements** - *What does it do for us?*
  - Essential system features at an appropriate level
- **System and software architecture** - *How?*
  - Support analysis of feasibility at this level
- **Lifecycle plan** - *Who wants it? Who'll support it?*
  - Identification of the major stakeholders now, future
- **Feasibility Rationale** - *Is this really true?*
  - Evaluate conceptual integrity and compatibility
Elements of Lifecycle Architecture (LCA)

- **Operational Concepts** - *What is it?*
  - Elaboration of objectives and concepts
- **System Requirements** - *What does it do for us?*
  - Functions and interfaces, identify TBDs
- **System and software architecture** - *How?*
  - What is the actual design selection. Any risks?
- **Lifecycle plan** - *Who wants it? Who'll support it?*
  - Elaboration of who does what over the lifecycle
- **Feasibility Rationale** - *Is this really true?*
  - Evaluate conceptual integrity and compatibility
Initial Operational Capability (IOC)

• Software preparation
  » Are we really ready to go live?
  » Good release, support software, docs, data, ...

• Site preparation
  » Facilities, equipment, supplies, commercial off-the-shelf software (COTS) in place, ...

• User, operator, maintainer preparation
  » training, team building, for everyone who will be actually working with the darn thing