Topics

- Software Development Processes (cont)

Homework

- HW2 due
- HW3 out
- Quiz 2 next Tues
LAST TIME: Code and Fix  [“Rapid Dev.”, McConnell]

Sometimes called the “Big Bang” model

The Process:

1. Start with a general idea of what to build
2. Code like hell
   - OPTIONAL: use combination of informal design, debug/test methods
3. Stop when you have a product ready to release
Waterfall Model

[“Rapid Dev.”, McConnell]

The waterfall model is document driven

"steps" or "phases"

How it works:

- progress **sequentially** through steps
- review held after each step (to advance)
- if mistakes, backtrack and redo
- “assembly line” development

S. Bowers
Q: Advantages?

- Each phase is clearly defined, with clear inputs/outputs (incl. requirements, design)
- Can ensure high-quality at each stage (“oversight”)
- Specialists can be experts at specific steps (analysts, architects, developers, QA team)

Q: Disadvantages?

- No software until late in project (release after system testing)
- Forces stable requirements early in project (changes mean backtracking)
- Considerable overhead (via documentation, review, management)
**How it works:**

- break project into series of “mini” projects, each addressing specific risks

- risks interpreted broadly:
  - poorly understood requirements, architecture, etc.
  - performance, security, etc.
  - missing features

**Q: Advantages?**

- flexibility, focus on quality (by removing risks), clear checkpoints

**Q: Disadvantages?**

- complicated for managers (so much flexibility)
Some Notes on Using Waterfall and Spiral Models

The term “Waterfall” is often a proxy for “Plan-Driven”

Waterfall for large, well-understood projects needing high quality/oversight

- well understood means: requirements are stable, high confidence is effort required, and/or completed similar projects in the past
- large refers to size of team and scope of project

Spiral for high-risk projects that aren’t well understood

- most projects aren’t like this … somewhere in the middle