Topics

• Quiz 1
• Basic Ideas in Software Engineering (wrap up)
• Software Development Processes

Homework

• HW2 out (next Tues/Thurs)
Senior Design Lab in Paccar 106

Teams can check out equipment for use in the lab ...
- Space is somewhat limited ... 20 or so monitors
- Plus shared workspace and a printer

To check-out equipment ...
- See Jason Schnagl (HERAK 103E)
- Computers (Mac and PC) plus additional equipment (e.g., phones) available
- Jason can also set up a development server for your team to use

Please see the lab rules, ask if you have questions ...
- By using the lab, you are agreeing to the rules

How well do Software Engineering Projects do?

![Modern Resolution for All Projects Chart]

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011–2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

[From: Standish Group, 2016]
• From the “CHAOS Report”
• Self-reported (survey) information by select companies
• Not a comprehensive survey of companies
Some success factors from the survey (most to least impact)

- Executive management support
- User involvement
- Optimization (of scope ... limit to high value features)
- Skilled resources
- Project management expertise
- Agile process
- Clear business objectives
- Emotional maturity (bad: over-ambition, prestige, absence, ...)
- Execution (planning, measuring progress, managing change)
- Tools and infrastructure (limit unnecessary overhead)
The Quality Triangle

**Scope**: amount of work team has to do
- e.g., the features that need to be implemented

**Time**: how much time you have
- may be different than how much time you need!

**Budget**: amount of money you have to spend
- in software, typically translates to the size of the team

**Q**: How do you think these three things relate to software “Quality”?
The work the team has to do for the project (e.g., which features to implement)

The time you have to complete your project

The amount of money budgeted to the project

When cost, scope, or time change, the quality of the product is affected

Related to the saying “fast, cheap, good; pick 2”

Q: How does this relate to your project?

• You have fixed cost and time ... so high quality depends on scope
• You also need a useful product ... so find & focus on the MVP!
Software Development Processes (aka Lifecycles)

Define development steps/phases and order to carry them out

The main steps (phases) found in most process models:

- **Requirements** ... what to build
- **Architecture & Design** ... technologies, components, detail
- **Implementation** ... primarily coding
- **Testing** ... unit, system, integration, user
- **Deployment** ... make “live”, install, train
- **Maintenance** ... updates, bug fixes, upgrades

We’ll look at a few different “classic” processes
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
Q: What are possible disadvantages of code-and-fix?

- potentially risky!
  - imaging use this process to build a bridge, house, or to fence a yard
- could lead to rework
  - halfway through find design flawed, have to start over
  - customer doesn’t find it useful / not what envisioned
- hard to stay on a schedule

Q: What are possible advantages of code-and-fix?

- no overhead ... just pure coding!
  - but upstream mistakes 10-100x more expensive to fix downstream
- (optionally) can show signs of progress (unless team “goes dark”)
- requires little expertise (just need to program)
- good for small bug-fix (maintenance) type projects