Topics
• Basic Ideas in Software Engineering (cont)
• Quiz on Thurs

Homework
• HW1 due today
• HW2 out (next Tues/Thurs)

Reminder
• CEDE Fall Workshop Meeting after class tomorrow
Basic Ideas in Software Engineering (Last Time)

Product vs Process vs Project

The four “big” ideas in software engineering

Why? ... problem to solve (e.g., business objectives)
Who? ... customers and users (stakeholders)
What? ... product to build (requirements)
How? ... the plan (process + schedule + management)

Business Objectives (the “Why”)

• desired outcomes of the client/company creating the product
• quantified, including by when (e.g., “Increase profits by 2% after 2 months of release”, “Reduce processing time by 20% in first 6 months”)

Requirements (the “What”)

• what the product should do or how it should work
• features/functions and characteristics of desired product
Functions vs Features

- a “function” is something a user accomplishes with product
  - e.g., make a phone call
- a “feature” is how users accomplish functions with product
  - e.g., use a keypad to dial a phone number

Exercise 1: Come up with another example of a function and supporting features

Exercise 2: Identify a potential needed function for your project

Exercise 3: Identify a potential business objective (desired outcome) for your project

Q: What does it mean to have a successful project?
- A successful project should meet the business objectives (the “Why”)

Q: Is this all?
- Also within time and budget constraints

Q: What does it mean to have a successful product?
1. users: product does what is needed / is useful
2. client: enables the company to achieve the business objectives

Typically 2 doesn’t happen unless 1 happens
Successful vs Challenged vs Failed

A successful project

- delivers the “correct” product on time and within budget

Two notions of “correctness” in software engineering

- Verification: was it built correctly? (e.g., satisfies requirements)
- Validation: was it the correct thing to build? (e.g., satisfies users needs)

A challenged project

- either late, over budget, or only partially correct
- e.g., issues with validation (only partially useful)
- or issues with verification (e.g., buggy/low-quality)

A failed project

- either canceled (e.g., too over budget and/or too late)
- or not adopted (e.g., not useful or too low quality)
How well do Software Engineering Projects do?

From the "CHAOS Report"
- Self-reported (survey) information by select companies
- Not a comprehensive survey of companies

[From: Standish Group, 2016]
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)