Today ...

- Soft. Eng. Topics (cont.)
- Classic mistakes

Readings

- project charter handout

Homework

- Proj 1 due
- Proj 2 out
- HW 1 out
Warm up ... Q: what is the difference between these?

- **Failed project**: cancelled or never used
- **Challenged project**: late, over-budget
- **Successful project**: on time & budget, met customer needs

Q: Approximately what % of projects are in each category?

McConnell, 2006:

\[ \approx 25\% \text{ delivered on-time} \quad \text{only 1 in 4 projects!} \]

\[ \approx 25\% \text{ cancelled} \]

\[ \approx 50\% \text{ delivered late, over budget, or both} \]
Q: What is the major **cost** in developing software?

- people’s (developer’s) time
- software dev. is labor-intensive
- and the labor is not cheap (specialized skills)

... time usually measured in “**person (developer) days/months**”

Q: What does it mean that a project is **over budget**?

- required more dev. days (effort) than estimated (or at least assigned)

... could be because it took longer than expected
... or to complete on time, had to assign more developers (more later)

Note that **estimation** is critical for project success
... and we’ll spend a good amount of time on estimation
Product vs. Project vs. Process

Software (product) is created by a project following a development process.

Software engineering focuses on successful projects not successful products:
- Dev team not usually held responsible for marketing, market “forces”, whether it was the right thing to build, etc.

Projects are (should be):
- Temporary ... with a defined start and end
- Measurable ... create a product that can be measured
- Progressively elaborated ... you learn more about project as it progresses

A project is not an ongoing process!
- but a project is

A process:
- is similar to a running algorithm
- takes inputs, produces outputs
- involves going through a series of steps
- they are often ongoing (e.g., testing software vs. testing a particular product)
Which are **not** projects?

1. Stapling programs for the play this weekend
2. Repairing a flat tire on your car
3. Watering your plants twice a week
4. Baking a cake for your friend’s birthday
5. Shelving books at the library
6. Running an IT support department

Software development projects:

- Almost always involve one or more project **teams**
- Follow a **software development process** (whether intentional or not!)
Classic Mistakes

Excercise

1. Find as many mistakes as you can in Case Study 3-1 (Classic Mistakes)

2. What do you think were the “biggest” 5 mistakes?

3. What would you have done to avoid these?
Mistakes too good not to make

Need to rescue a project that’s behind schedule?
  – Add more people!

Need to shorten/reduce your schedule?
  – Schedule more aggressively!

A key contributor aggravating the rest of the team?
  – Wait until end of project, then fire him/her!

Have a rush project to complete?
  – Take whatever developers available now
  – And get started as soon as possible
Four dimensions of development “speed”

People
- individual speed
- overall team dynamics

Product
- vaguely or clearly defined

Technology (language, tooling, etc.)
- may help or hinder development efforts

Process
- may utilize people’s time
- or create one stumbling block after another