Lecture 11: Scheduling, Estimation, and Prioritization

Valentin Razmov

Outline

- Software project estimation
- Prioritization

Resources

- *Rapid Development*, by Steve McConnell
- *Code Complete*, by Steve McConnell
- *Software Requirements*, by Karl Wiegers

How These Three Concepts Tie Together

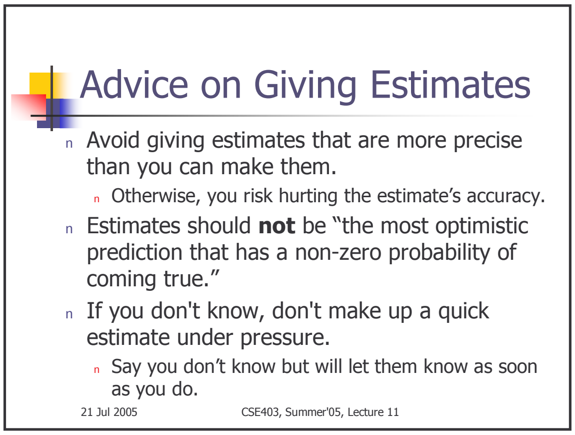
- You need an up-to-date *schedule* to keep you on track in the project.
- Items on the schedule must be continuously *estimated* (both in length and in start / completion times).
- Items on the schedule must have realistic *priorities*.

Software Project Estimation Approaches

- Estimate pieces of the project, then add the pieces together.
- Refer to estimate data from previous projects.
- Use an algorithmic approach (e.g., COCOMO).
- Use scheduling software.
- Have outside experts to the project estimate.
- and many others...

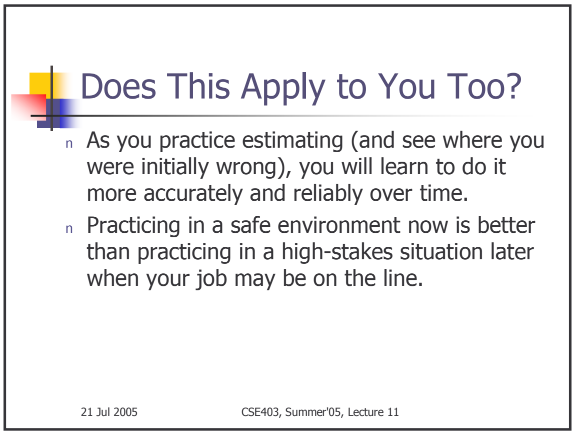
How Are Software Projects Estimated?

- Software project estimation, as software development, is done through gradual refinement.
  - You will know better when you have seen more.
- An estimate is best represented as a range of values with an associated confidence level.
  - Not a single point
- There is a trade-off between estimation accuracy and project control.



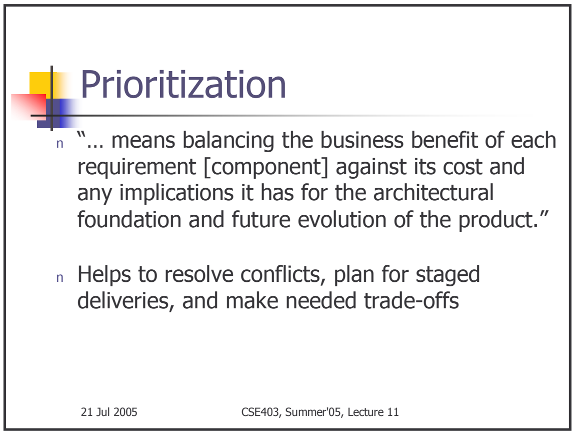
Advice on Giving Estimates

- Avoid giving estimates that are more precise than you can make them.
  - Otherwise, you risk hurting the estimate's accuracy.
- Estimates should **not** be "the most optimistic prediction that has a non-zero probability of coming true."
- If you don't know, don't make up a quick estimate under pressure.
  - Say you don't know but will let them know as soon as you do.



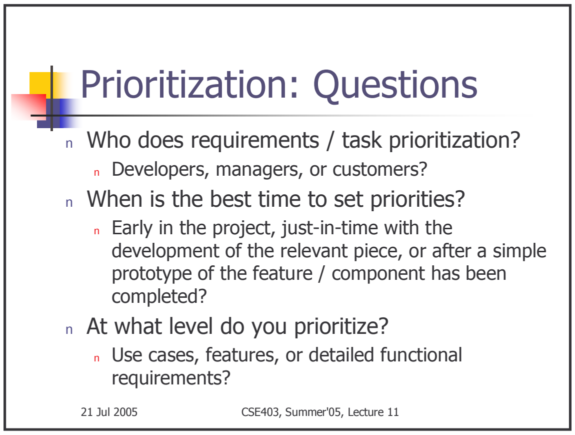
Does This Apply to You Too?

- As you practice estimating (and see where you were initially wrong), you will learn to do it more accurately and reliably over time.
- Practicing in a safe environment now is better than practicing in a high-stakes situation later when your job may be on the line.



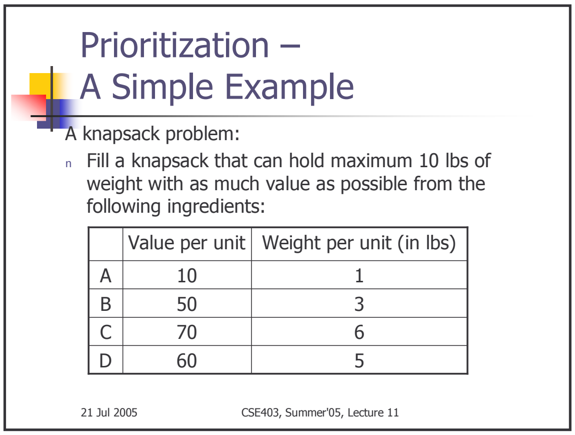
Prioritization

- "… means balancing the business benefit of each requirement [component] against its cost and any implications it has for the architectural foundation and future evolution of the product."
- Helps to resolve conflicts, plan for staged deliveries, and make needed trade-offs



Prioritization: Questions

- Who does requirements / task prioritization?
  - Developers, managers, or customers?
- When is the best time to set priorities?
  - Early in the project, just-in-time with the development of the relevant piece, or after a simple prototype of the feature / component has been completed?
- At what level do you prioritize?
  - Use cases, features, or detailed functional requirements?

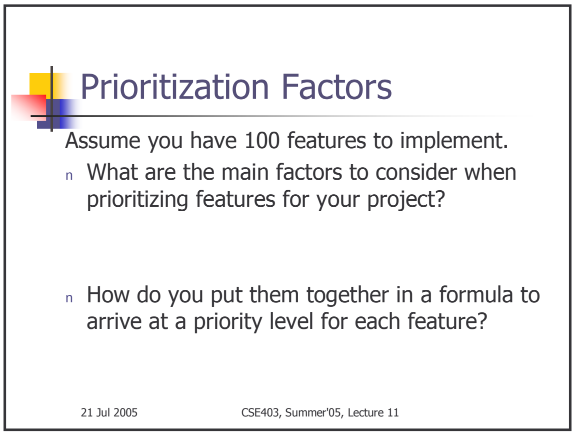


Prioritization – A Simple Example

A knapsack problem:

- Fill a knapsack that can hold maximum 10 lbs of weight with as much value as possible from the following ingredients:

	Value per unit	Weight per unit (in lbs)
A	10	1
B	50	3
C	70	6
D	60	5



Prioritization Factors

Assume you have 100 features to implement.

- What are the main factors to consider when prioritizing features for your project?
- How do you put them together in a formula to arrive at a priority level for each feature?