

Lecture 12: Scheduling, Estimation, and Prioritization (Part II)

Valentin Razmov

22 Jul 2005

CSE403, Summer'05, Lecture 12

Outline

- Scheduling
 - Being behind schedule, ahead of schedule
- Frequent scheduling and prioritization-related mistakes students make
- Best practices for project scheduling
- Scheduling in the context of your projects

22 Jul 2005

CSE403, Summer'05, Lecture 12

Resources

- *Rapid Development*, by Steve McConnell
- *Code Complete*, by Steve McConnell

22 Jul 2005

CSE403, Summer'05, Lecture 12

How These Three Concepts Tie Together (reminder)

- 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*.

22 Jul 2005

CSE403, Summer'05, Lecture 12

Scheduling

- If your project moves forward on budget and on schedule, you are in the minority...
- What can you do if that's not the case?

22 Jul 2005

CSE403, Summer'05, Lecture 12

Your Options If You Fall Behind Schedule

22 Jul 2005

CSE403, Summer'05, Lecture 12

Your Options If You Fall Behind Schedule

Which of these would you choose?

- n Negotiate an increase in the amount of time
- n Add more people to the team
- n Hope that you can catch up later
- n Be upfront about it
- n Hide it "under the rug" and move forward
- n Negotiate a reduction in the scope of the project

22 Jul 2005

CSE403, Summer'05, Lecture 12

Your Options If You Fall Behind Schedule (cont.)

- n Negotiate an increase in the amount of time
 - n Sometimes not an option, but it may be possible
 - n Increase by how much? By the slipped time or more?
 - n Avoids addressing the bigger problem that caused the delay
- n Expand the team
 - n "Adding people to a late software project makes it later." (Brooks, 1975)
 - n Why? Under what circumstances?
 - n There is a limit to schedule compression.

22 Jul 2005

CSE403, Summer'05, Lecture 12

Your Options If You Fall Behind Schedule

- n Hope that you can catch up later
 - n Statistics show this to be an illusion: you're more likely to fall further behind
- n Be upfront about it; don't try to "hide it under the rug"
 - n If you conceal the truth, you will lose the customer's faith in your team / company
- n Negotiate a reduction in the scope of the project
 - n Prioritize the optional and nice-to-have features, then drop the least essential ones
 - n Can you save time by providing similar (but perhaps rough) useful functionality?

22 Jul 2005

CSE403, Summer'05, Lecture 12

Your Options If You Are Ahead of Schedule

22 Jul 2005

CSE403, Summer'05, Lecture 12

Your Options If You Are Ahead of Schedule

- n Strive to maintain that edge
 - n Allows you to gently exceed expectations again and again
 - n Gives you a "cushion" in case difficulties arise in the future
- n Spend the earned capital by giving everyone on the team extra breathing room
 - n May help if the team really needs it, e.g., near the burn-out point.
- n Push to get even further ahead.
 - n May needlessly burn out the team early on
 - n If you can, your original schedule was probably too conservative.
- n Listen to what upper-level management says
 - n As much as you may not want to do that...

22 Jul 2005

CSE403, Summer'05, Lecture 12

Frequent Mistakes Students in Previous SE Classes Have Made

Scheduling and prioritization-related:

- n Not exploring all unknowns (risks) early on to create a realistic schedule
- n Not maintaining an up-to-date schedule with all remaining tasks and how they map to the resources (time, people) in the team
- n Leaving too few resources (people) for a critical task that can't be delayed
- n Not leaving enough "safety net" time before major releases in case something unexpected happens
 - n It often happens in the most inopportune moments.

22 Jul 2005

CSE403, Summer'05, Lecture 12

Frequent Mistakes Students in Previous SE Classes Have Made

Scheduling and prioritization-related:

- Underestimating the challenges of a new development environment
 - Overly relying on similarities to known environments
- Spending time on "cool" features that are not central to the needs of the users while delaying the development of promised features
 - A real project is not about what developers enjoy doing, it's about what brings value to customers.
 - Hopefully, the two are similar, but if not, the latter should take precedence.

22 Jul 2005

CSE403, Summer'05, Lecture 12

Best Practices in Project Scheduling

- Building in a margin of safety into the schedule
- Continuously measuring progress and re-estimating resources needed
 - The daily builds are the pulse of your project.
- Using multiple project estimation approaches and studying the differences between them
- Scrubbing the requirements

22 Jul 2005

CSE403, Summer'05, Lecture 12

Scheduling in the Context of Your Team's Project

- According to data in *Code Complete*, the breakup of development time for a 10-15KLOC project is:
 - 13% - architecture
 - 20% - detailed design
 - 20% - coding and debugging
 - 20% - unit testing
 - 12% - integration
 - 15% - system testing
- Is this reflected on your latest schedule?
- How far into each phase is your project?
- Whose job is it to take care of scheduling on your team? Who owns and manages the schedule?

22 Jul 2005

CSE403, Summer'05, Lecture 12

Scheduling in the Context of Your Team's Project (cont.)

- "Once you have a delivery date and a product specification, the main problem is how to control the expenditure of human and technical resources for an on-time delivery of the product." (Steve McConnell, *Code Complete*)
- Case:** Your team has a fixed delivery date and an existing product specification, as well as relatively fixed (but flexible) human resources. What aspects can you vary and where is your leverage if a project estimate suddenly reveals that you cannot deliver for another 6 weeks?

22 Jul 2005

CSE403, Summer'05, Lecture 12

Scheduling in the Context of Your Team's Project (cont.)

- Case:** Your team has a fixed delivery date and an existing product specification, as well as relatively fixed (but flexible) human resources. What aspects can you vary and where is your leverage if a project estimate suddenly reveals that you cannot deliver for another 6 weeks?
- Reduce / renegotiate functionality
 - Push to version 2 certain non-essential features
- Use outside technical resources
 - Reuse code
- Delegate tasks

22 Jul 2005

CSE403, Summer'05, Lecture 12

Favorite Related Quotes

- "Doing things at the last minute is much more expensive than just before the last minute." (Randy Pausch)
- "If you haven't got time to do it right, you don't have time to do it wrong."
- "Good judgement comes from experience. Experience comes from bad judgement."
- "Failing to plan is planning to fail."
- "Work expands so as to fill the time available for its completion." (Parkinson's Law, 1957)

22 Jul 2005

CSE403, Summer'05, Lecture 12