Week 6 Lecture 16

Business

What's in this lecture?

- Project Management
- Avoiding Overcommitments

The story so far...

- Know how to break down a project
- Covered an Agile approach to development
- Can estimate time to complete a component

Management

- But what about the intangibles:
  - How do you manage expectations?
  - How do you delegate responsibly?
  - How do you communicate bad news to client?

Expectations

- Fundamental rule: people don't like surprises
- 'Good' surprise (component finished early):
  - You deliberately under-promised
  - You deliberately over estimated
- 'Bad' surprise (nothing works; functionality incomplete):
  - Poor communication
  - Talk doesn't match walk
  - Bush league

Expectations

- Surprises aren’t the causes of the ‘good’ and ‘bad’ -- the miscommunication is
- Goal with managing expectations is to address them early and consistently

Ahead of Schedule

- Managing expectations with words:
  - ‘Making good progress’
  - ‘May wrap up component a few days early’
  - ‘Have basic functionality, need to finish testing and bug fixes’

Alert!

- Mistake I always make:
  - Magnitude between my delivery date and deadline dictates how happy the client is
  - Problem: choppy delivery sets wrong expectations for next time
- What really should be done:
  - Consistent delivery

Behind Schedule

- When scope and externalities are accounted for, delivery falls squarely on you to communicate
- Unlike behind ahead of schedule describing situation with specifics is extremely important
- Prioritize by
    - effect on component delivery
    - effect on team members
    - effect on deadline

Alert!

- My default attitude is that if I give up a night's rest, or my weekend, I can get back on track.
- This is seriously bad:
  - Takes toll on your productivity and happiness
  - Team is left in the dark
  - If you succeed, you'll do it again

My Rule

- YMMV:
  - If I fall behind on a day's goals:
    - Let my internal team/coworkers know immediately
    - If blocking, push those goals to next day
    - If not caught up by close of business on second day, alert extended team/client

Delegation

- On the receiving end:
  - Don’t let somebody else set your schedule without your input
  - People will take as much rope as you give
  - Repeat schedule back to them for input

Delegation:
Good Practices

- If you need something done
  - Open a dialogue to discuss the issue
  - State your intentions and needs
  - Ask for their opinion on how to best accomplish the goals

Delegation: Good Practices

- Never leave somebody with backlog hanging over their head
- Work with them to ameliorate any issue
- Accept and appreciate their contributions
- If they are trying but failing, don’t get angry
- Nobody should be yelled at for trying to help

Delegation: Know your Role

- Developers are territorial:
  - over their code
  - over their goals
  - over their hours
- Shoveling work on another to mask your own problem is disingenuous and wrong

Communication

- Something bad may happen:
  - data is lost
  - deadline are blown
  - clients are mishandled
- Your goal: communication and damage control

Damage Control

- Quickly inventory what went wrong and why
- Identify possible fixes
- Outline plan of action to fix

Commitments

- Extremely easy to overcommit
- Extremely hard to say ‘no’ in person
- Without a feedback loop involving your schedule, you are flying blind

Making Good Commitments

- Ask to examine schedule and get back with a good estimate
- Writing out an idea forced you to think harder than if you speak it
- The limits of your schedule need only one voice in their creation: yours

Communication

- Plan-in-hand, call the client or team
- State that there is a problem, and what it is
- Go quickly through why it happened
- Focus majority of energy on how to fix it
- If you messed up: admit it and apologize
- Note: Only after you fully understand the problem and/or have a solution in place -- don't rush.