

CSE 403

Midterm Review

Are you succeeding?

- Your project metric is a quantitative measurement of how well your project is going
- Can you evaluate your project at each milestone based on this metric?
- If not, how will you know if your project is succeeding?
- How will we grade your project at each milestone?
- You need to specify this
- If you don't we'll have to "completely make it up" -- ad hoc

Transparency and “Participation”

- Are you making your project docs available to the other teams/classmates?
Your repos?
- What about our qualitative evaluations of your project? The quantitative evaluation?
- Seeing your peers' work is valuable insight -- you'll learn more and have more successful projects
- Your call -- however, your “participation” grade will be partially based on if you open up your project docs, repos up, and our qualitative evaluations
- Let your TA know (and if you don't an explanation why not)

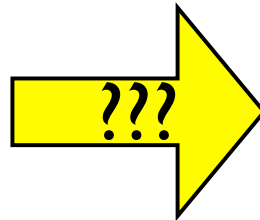
Schedules

- Feature lists for each of the three milestones
- High level costing for each feature
- Prioritization of each feature
- Low level detailed costing for each feature for 11/05 release
- Personnel assignments for each 11/05 release
- Costing numbers match resource numbers
- What is delivered at each release at a high level: A theme (is the site live, high level functionality)
- Risk assessment/issues

Example of loose coupling (a contrived example)

Topix wants to offer scholarships

- No idea of how much or how many
- No idea of the procedures
- No idea how to promote it
- No idea about restrictions
- No idea of selection process



- \$\$\$
- CSE Dept Procedures
- Application forms?
- partners@topix.com

Students want scholarships (money!)

- What's available?
- What are the qualifications?
- How do you apply?
- What are the restrictions?

Women in (Software) Technology

Guest Speaker:



Sharon Wienbar, Scale Venture Partners
Managing Director

11/02/12

<http://www.scalevp.com/team/sharon-wienbar>

A lesson in teams, organizations, and individuals

- Does a team need a leader/manager?
- Is there a difference between leadership and management?
- Is there a difference between “official” and “unofficial” leadership?
- Where does authority come from?
- What is the role of a manager?
- What’s this have to do with your project?

Observations

- There is no official authority in any of the teams
- Who makes decisions? Who divides work? What do you do when a person isn't "pulling her/his weight?"
- Project manager's have some unofficial authority in most projects
- Leadership by persuasion (a good thing)
- We're going to make some tweaks...
- An "org change"...

“Promotions”

- The following project managers have been promoted to be manager of their projects
 - Ryan Emmett, Instafeed
 - Caylan Lee*, Bulletin
 - Siavash Baratchi, Change My Mood
 - Yukio Maeda, Event Hub
- Congratulations!

Engineering manager is responsible for product deliverable

- Manages engineering team
- Responsible for all aspects of engineering output (features, timeliness, quality)
- Figures out the engineering process
- Recruiting engineering team
- Schedule
- SW development, quality insurance, operations
- Technically involved?
- Project management?
- Motivation

What does this mean?

- How will things change?
- What about Full House, Bulletin, Tile to the Top?
- Risk: We haven't talked much about management -- what's involved here?
- Management must come with some formal authority
- You should have questions. Managers (and Janette and Grace) should (de)brief their teams
- Other questions? See me or your TA

Mid term logistics

- Closed book, closed notes, no electronic devices
- Bring a pencil
- True/False but with an explanation
- Short answer
- Extra office hour at 11:30 today, in-and-out of the office this afternoon

Midterm format

- **Section 1. True/False.** Answer each of the following questions either True or False by circling your answer. You may optionally give an explanation for the answer. Incorrect answers with reasonable explanations may receive partial credit. Correct answers with substantially incorrect explanations may not receive full credit.
- **Section 2. Short Answer.** Answer the following questions in 1 to 3 sentences. For questions requiring examples (or reasons), answer in (approximately) 1 sentence for each example.
- **Section 3. Long Answer.**
- Haven't decided on the "point value" for each question yet (but the same for each section). Pay attention.

Major topics

- Teams
- Lifecycle
- Requirements
- Architecture
- Design

Key concepts

- Divide and conquer
- Abstraction
- Modules
- Coupling and cohesion
- Parallelism
- Examples in software, your project, and in real life

Teams

- Why work in teams?
- More people->more productivity
- More people->less productivity
- Organizational structures
- Roles
- Individuals

Lifecycles

- What are the components of a software lifecycle?
- What are the different kinds of lifecycles?
- When does software get released?

Requirements

- How do we specify requirements?
- Why is specifying requirements hard?
- Who specifies requirements?
- Why do we specify requirements?

Design

- What is the design stack?
- How do we do design?
- How do we specify designs?
- How does UML fit in?
- Who does design?

Architecture

- Why architecture?
- What are the “-ilities” and why are they important?
- Choosing tools: How and why?
- Modularity -- why is this good?
- Web, MVC, pipelines, layer cakes

Definitions

Divide and conquer, waterfall model, pipelining, product manager, project manager, user interface, information architecture, loose coupling, cohesion, agile, scrum, UML, use cases, modules, interface, implementation, metrics, abstraction, information architecture, runtimes, use cases

True/False Example

- **Question: True or False.** Requirements specifications should usually describe “what” not “how.”
- **Answer:**
 - **True.** (Specifications should not dictate how something should be implemented.)
 - **False.** Projects often fail when requirements completely ignore what is possible with regards to the architecture or tools (e.g. run time stack) used by a product.

Short answer example

- **Question:** What is a module?
- **Answer:** A module is a software abstraction -- a component of a software system. A module consists of an (public) interface (how other modules access the module) and a (private) implementation of that interface.

Short answer multi-part question

- Question: What are three properties of modularity?
- Answer:
 - Decomposability
 - Composability
 - Understandability
 - (Has continuity, protected/safe)
- Grading: If 6 points:
 - 1 point for 1 correct answer, 3 points for 2, 6 points for 3
 - Probably

baskin BR robbins



Baskin-Robbins

- Retail ice cream store
- Often found in malls
- Famous for serving 31 different flavors of ice cream
- Serves primarily ice cream snacks -- typically ice cream in a “cone” or a cup
- Imagine one of these moving into the HUB

Why does this matter?

This situation is the background for questions about use cases in the midterm.

Hint: You'll probably specify a use case.

Example use case: Jane makes a donation

- **Actor:** Jane, a visitor; **Scenario:** Visitor makes a donation; **Precondition:** NA; **Trigger:** Jan wants to make a donation; **Success condition:** Donation made; **Failure condition:** Donation not made
- **Steps** (*I labeled this as Scenario previously*)
 - Jane browses or searches for a nonprofit to make the donation
 - She reviews information on the nonprofit as well as the terms of service at Charity Blossom
 - She specifies the amount and payment method for the donation and commits to making the donation
 - She is presented with a receipt for the donation
- **Exceptions:** Jane can't find the nonprofit, Jane is a registered user, Jane's credit card fails to get approved, nonprofit can't take donations

If this still doesn't make sense...

