CSE 403

Requirements, Part II

This week

- Requirements due 10/8, 11:59PM
- Architecture/Design/Tools/Process out; due 11/15, 11:59PM
- Reading: "Hints for Computer System Design," "Five Paper Prototyping Tips"
- Skim: "UML Class Diagrams," "UML State Diagrams," "UML Sequence Diagrams"

Requirements logistics

- Project managers are required to submit this assignment via Dropbox. Only project managers should do so.
- PM's are to submit a text document that contains:
  - PM name
  - The name of your project
  - A single URL of the requirements document. If you have created multiple documents for each section or believe that giving us multiple links to various sections makes it easier for us/you to read/manage your requirements document, you should still include a single URL -- that URL will contain links to the other documents or sections.
- You should make sure the TAs/instructor have access to these documents. Include instructions in this text document.

Next: Architecture, Design, Tools, Process (Team, Schedule)

- Architecture
  - Data/Information architecture, components and interfaces
  - UI/UX in requirements doc
- Tools and runtimes
  - Include physical topology
  - Rational for your choices
- Development process
- Schedule
- Organization

Express what you do in one sentence

(tip of the day)

- What is your project in one sentence?
- Simple explanation to others
- Simple explanation to yourself
- One sentence, one paragraph, one document
- Let's try...

Requirements

What are you going to build?

- Tell "what" not "how"
- Provide vision to show "why"
- Describe "who"
- Explain business drivers

Conclusion:

Specifying requirements is hard!

Why specifying requirements is hard

- Missing vision (can't describe the big picture)
- Ambiguity
- Not enough technical understanding from PM
- World changes quickly

Why specify requirements?

- Formulate, understand, and communicate what is being built
- Set priorities on when things should be developed and deployed
- Ensure that product is aligned with overall company vision and goals

Outline

- What are requirements?
- How do we gather requirements?
- How do we specify requirements?

What are requirements?

- Vision and background
- Users and use cases
- Functional (feature) requirements
- User Interface/User Experience/User Design
- Product deliverables

Users and use cases

- Actors
- Preconditions
- Triggers (what starts the use case)
- Minimal/success guarantees (end condition)
- List of steps to a successful scenario
- Failure end conditions
- Extensions, alternative paths

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
- **Scenario**
  - 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

Feature/functional specifications

Functionality/Features

- Usually bundled into the next release cycle
- Broken down into features
- Cognizant of time frame and resources
- Harder in the beginning? Because you are making decisions that have long term implications

How to formulate functional requirements

- Divide and conquer from vision
- Listen to feedback
- Brainstorm
- mutually exclusive, completely exhaustive (?)

Examples

- Display listing information for every nonprofit (approximately 1.5MM)
- Periodically get (new) nonprofit listing information from irs.gov
- Allow visitors to leave reviews, update listing information
- Allow nonprofits to "claim" listing page by registering with CB.
- Allow nonprofit to update listing information
- Allow visitors to make a donation to help a specific nonprofit
- Require nonprofits to register/login when claiming listing

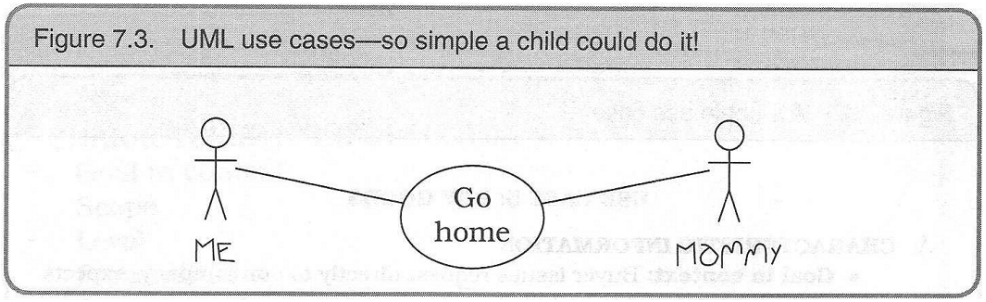
How specific should requirements be?

- Unfortunately, as usual, "it depends"
- How "technical" is the person writing requirements?
- How experienced is the team?
- How well do they work together?
- What does the engineering team want?
- Who is operating in "CYA" mode? (bad sign)

Drill down: Login and registration

- A nonprofit must be able to register and login to use the system

Are formal methods good?



- There are standard templates for requirements documents, diagrams, etc. with specific rules. Is this a good thing? Should we use these standards or make up our own?
  - Good; standards are helpful as a template or starting point
  - But don't be a slave to formal rules or use a model/scheme that doesn't fit your project's needs.

Note: Kinds of requirements

- Marketing requi (MRD)
- Product r ents (PRD)
- Fu requirements (FRD)

My opinion: it's all largely BS!

User Design: Part of Requirements?

- User interface, user experience, information architecture
- Mock up screens and flows
- Fonts, colors, icons,...
- Brand expression (logo, name, vocabulary)
- Low fidelity vs. high fidelity

User Design: Part of Requirements?

Vision/Requirements

Fonts

Logos/Iconography

User interface

User experience

(Photoshop) Mockups

HTML/CSS/JavaScript(?)

Holy grail of requirements specification

You know you've done a good job when your product...

Tastes like chocolate

Cures cancer

Costs a dollar

Take aways

- Requirements are hard -- ambiguity is the enemy
- Requirement specifications probably can't sufficiently capture everything
- User definitions, use cases, **functional specifications** are primary concepts
- Design: Is this part of requirement (UI/UX)?
- Paper & pencil, "storyboarding" go a long way (stay tuned!)
- Who's responsible? Product manager?

Project requirements: Part I

- Vision and background
- Users and use cases
- Functional (feature) requirements
- User Interface/User Experience/User Design
- Product deliverables