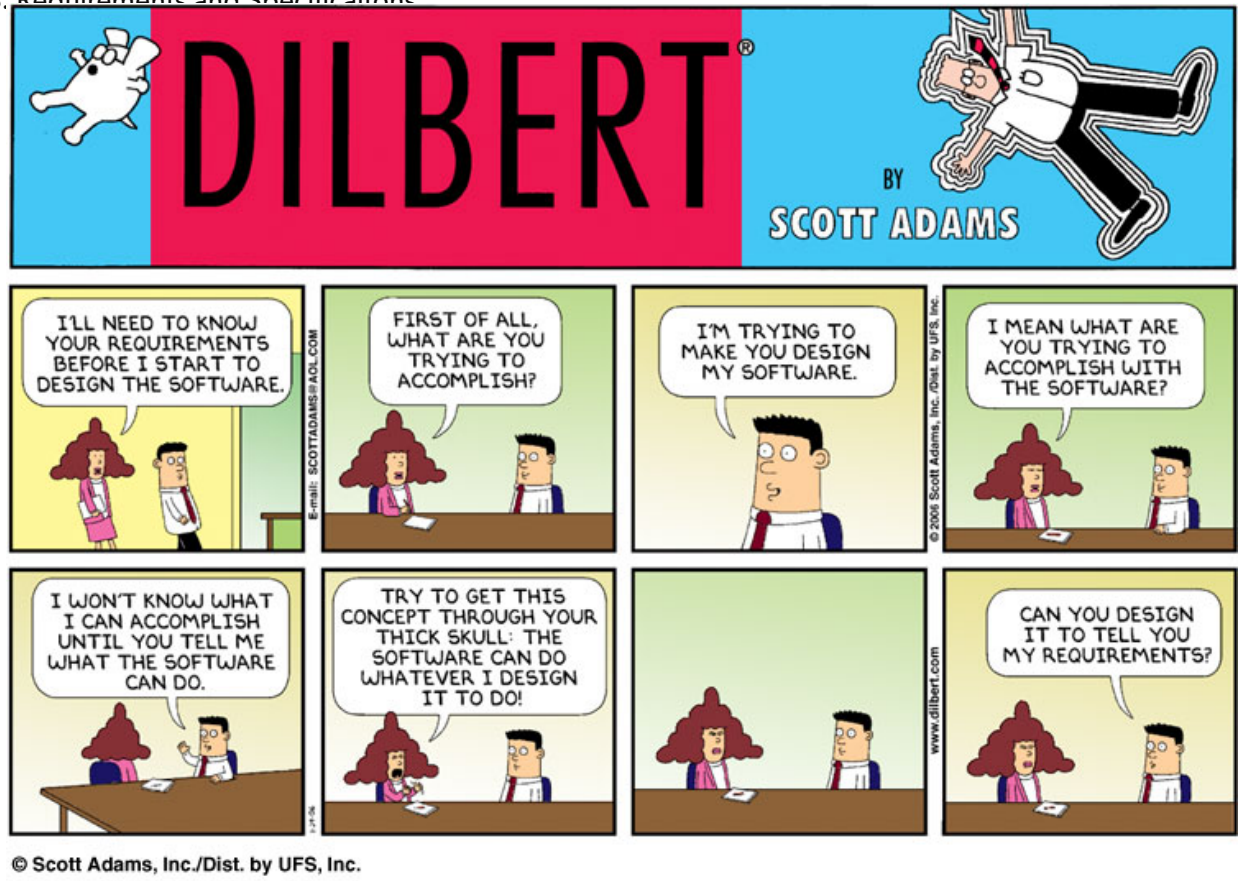
CS1320
Creating Modern Web and Mobile Applications

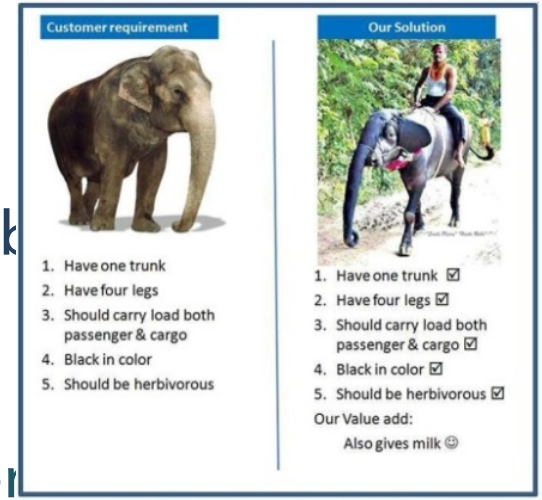
Lecture 8:

Requirements and Specifications



Motivation

- You have to know what to build before you k
  - Understanding the user's needs
  - Understanding what would meet those needs
- Requirements is the definition of the proble
  - Not a definition of the system
  - Or a definition of what will be built
  - Or a definition of how the system is built
  - These come later



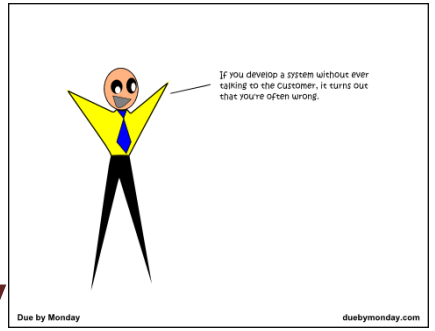
Importance of Requiremen

- Many software projects fail
- Poor requirements cause over half of these failures
  - 13% fail due to incomplete requirements
  - 12% fail due to lack of user involvement
  - 10% fail due to unrealistic expectations
  - 9% fail due to changing requirements
  - 7% fail because the system is no longer needed

Requirements Process

- Requirements Analysis is an on-going process
  - You need a good first approximation
  - Think of it as a dialog with your stakeholders
  - Understand what they want
  - Their understanding what you can do
- Things are going to change
  - People change their minds
  - Markets change
  - New opportunities arise

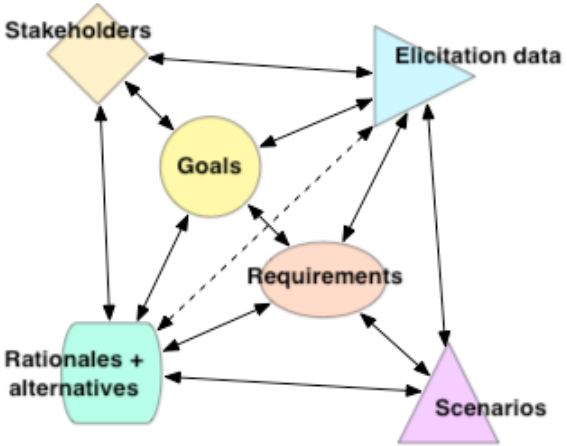
Requirements are Difficult



- Communications difficulties, e.g. different v
- Cultural differences
- Can't anticipate all questions
- You won't understand the user's needs a
- Building the system creates opportunitie
- Users are fickle (don't know what they re
- Users don't understand what is possible/difficult

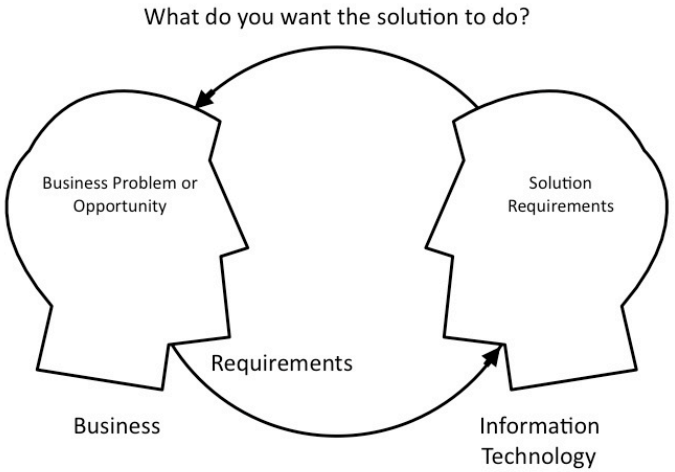
Requirements Goals

- Define the *problem* from the user's point of view
- Assess the need for a solution
- Determine the outlines of the best solu
- Determine what is required and what is
- Determine what is feasible (resource lir
- Determine acceptance criteria
- DO NOT WORRY ABOUT IMPLEMENTATION



Creating Requirements

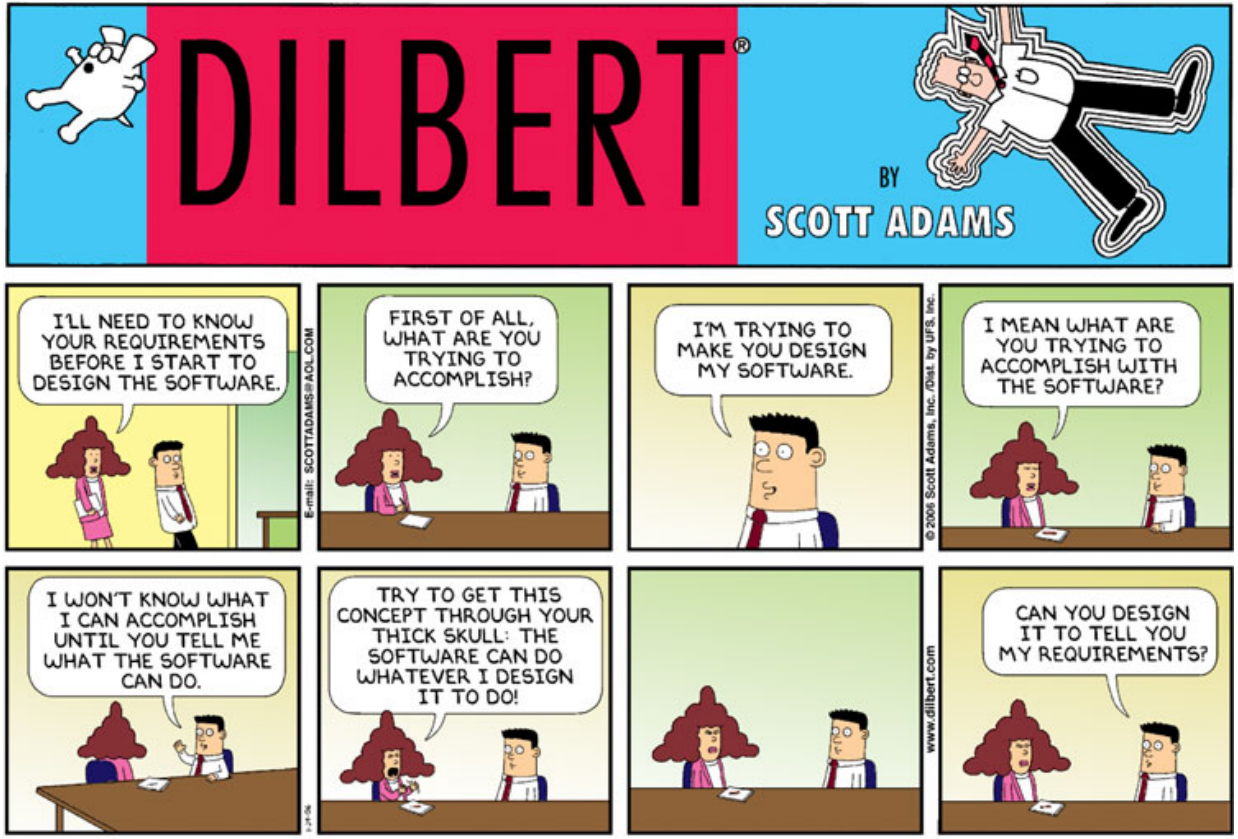
- Put the USER first
  - Don't assume the final system
  - Don't assume a particular solution
- Assess the needs for the projec
  - Get the data from the potential users
  - But what data is required?
  - But how to get the data?



Who is the User (Stakeholder)?

Consider a shopping site

- Is it the person doing the shopping
- Is it the person who does the packing and shipping
- Is it the person who does the ordering of products
- Is it the person who sets up the catalog
- Is it the person owning the store
- Is it the person owning a competing store
- Is it the person who is paying you to build the site



Talking to Your Client

- Understanding what they want
  - And why they think they want it
  - There should be a reason
- Understanding what they have done
  - Do they have a current system to interface with
  - Have they talked to users?
    - Do they know their potential users (other than themselves)
    - You might have to do it for them
    - Do they know who their potential users are
  - Do they have a current user base
    - Have they solicited feedback
    - Can you get access to the base for interviews, surveys, …

Good and Bad Clients

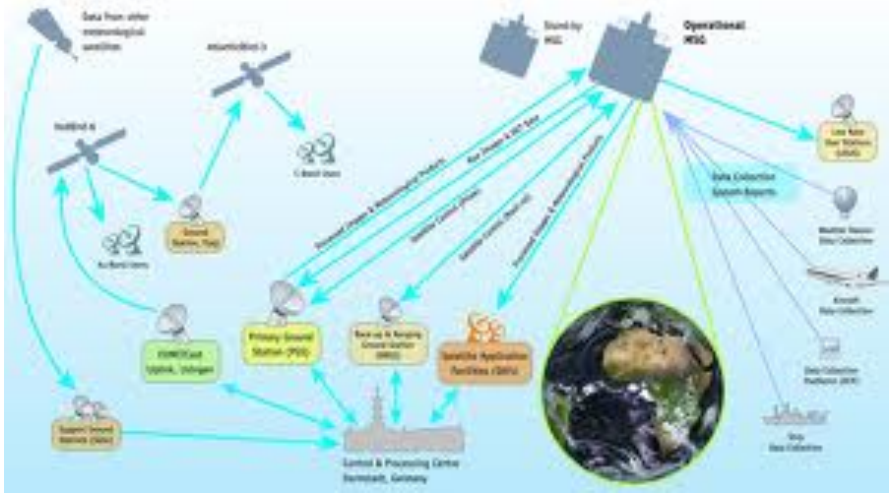
- Not all clients are equal
- Good clients
  - Provide appropriate information
  - Keep in touch, answer questions, communicate
  - Provide appropriate feedback throughout the project
  - Provide direction
  - Have "done their homework"
- Bad Clients

What's Worse than a Bad Client

- He who represents himself has a fool for a client
  - Holds in law
  - Holds in medicine
  - Holds in software development
- Clients emphasize the user and the user's needs
  - Programmers emphasize the programmer
  - Easier to code doesn't mean easier to use
  - You might not be a typical user
- Those of you working on your own project
  - Do real user studies
  - Find a set of potential users ***outside*** your group and ***listen*** to them

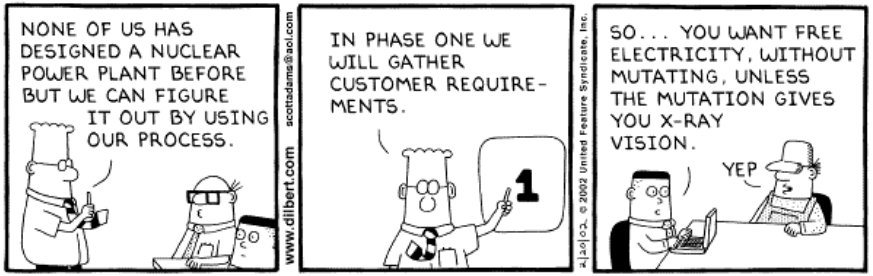
Getting USER Information

- Talking to users
  - Interviews
  - Focus Groups
  - Questionnaires (Qualtrics, Gforms)
  - Facebook, Mechanical Turk
- Observing users



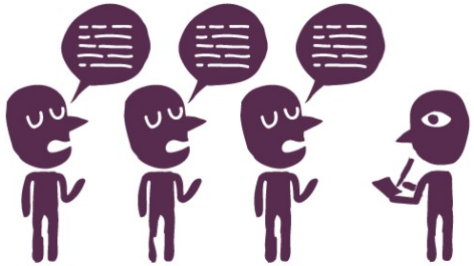
Talking to Users

- Understanding what they w
  - And why they think they want it
- Understanding what is important
  - Features, performance, …
- Vocabulary
  - Don't be afraid to ask questions
- Who is in charge



Interviewing Potential Users

- Structured versus unstructured interviews
- Interviews require considerable preparation
  - DO YOUR HOMEWORK
  - Determine what information is needed
  - Find out about the interviewees
  - Decide on questions and organization
- Process
  - Move from general to specific to general (open) questions
  - Summarize the interview
    - In terms of stories, use-cases, scenarios
      - » Examples of the use of the system
    - In terms you can feed back to the client
  - Follow up



Users don't know what they want.

Questionnaires and Surveys

- Who is your target audience (stakeholders)
  - Particular users or the general public
  - Others that may be affected by the system
  - Avoiding bias in audience selection
- Needs a lot of thought
  - What information is needed
  - Phrasing questions so as not to bias the outcomes
  - Determining how to interpret the results
- Likert scales
  - Range of 1-7 (or 1-5) where … rate X
  - Provides a means of compiling results
- End with open questions (won't be answered by everyone)

Qualtrics
Survey Solutions for Success

Stories and Scenarios

- Used in requirements and specifications
  - Also called use cases
  - Also used in design, development, testing
- Descriptions of common use cases
  - Think about how your system will be used
  - A story describes a common use
    - From start to end
    - Should be as specific as possible (realistic)

Stories and Scenarios: Example

Sam hears an oldie ("Both Sides Now") on the radio and decides he wants to buy a CD containing that song. He is interested in getting a CD with other songs of interest and one that doesn't duplicate songs in his current collection. He decides to use your system. He starts by typing in the name of the song into the query box. …

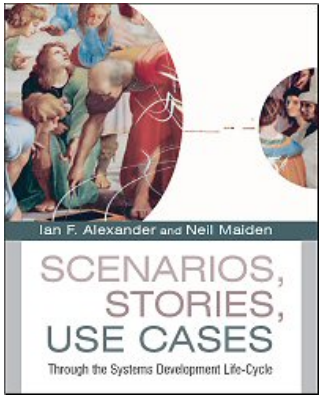
- Does he start by logging in or not needed?
- Is he successful or not?
- How many queries does it take?
- Does he need to refine the queries?
- How does he find CDs that wouldn't duplicate his current songs?
- Can he buy the CD from the web site; save for later; …?
- What does he do when done?

Stories and Scenarios

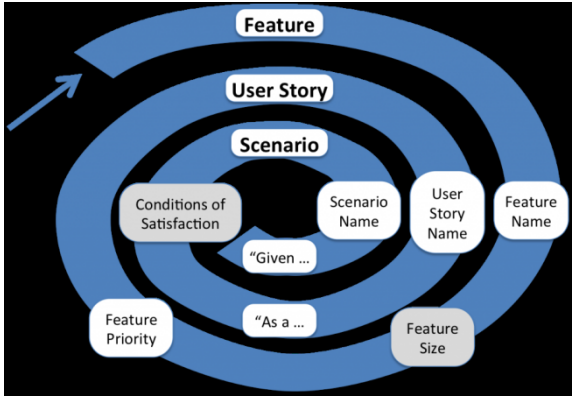
- Detailed description of task
  - Use actual names
  - Be specific
  - Include motivation
  - It is a story
- Pictures of your application for each step
  - Can be crude
  - Should give sense of what is going on

Stories and Scenarios

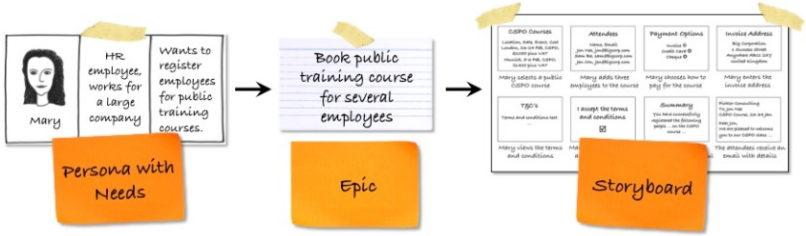
- Form a basis for getting user feedback
  - Give the user a sense of what your app can/will do
  - Give the user a context for providing input
  - Give the user a chance to change the story
    - What happens if …
- Ensure multiple stories are consistent
  - With each other
  - With your overall application



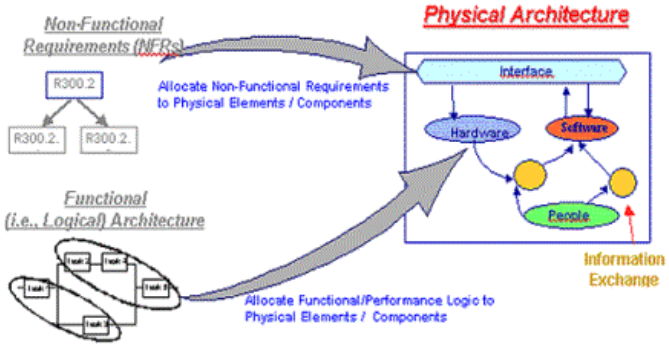
Stories and Scenarios



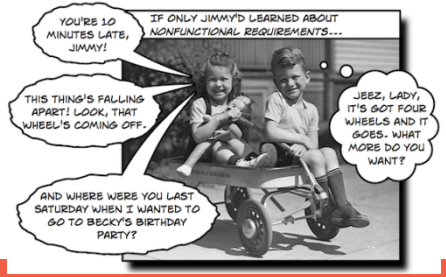
- Should cover the principle tasks
- Should cover a diverse set of tasks
  - With a diverse set of users
- Should give you a sense
  - Of what you will need to do
  - Of how the application will be used



Other Requirements

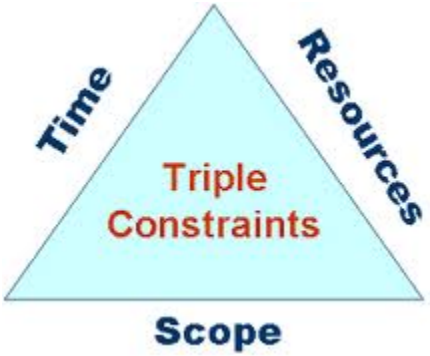


- Security constraints
- Privacy constraints
- Anticipated average and maximum loads
  - What should happen if these are exceeded
- Universal Access: Internationalization and accessibility
- Validity constraints
  - A product should only be ordered if the user charge is valid
  - Users should only be charged when the product is shipped
  - Avoid duplicate orders; avoid dropped orders
  - Users must have valid email addresses
  - Users must be previous registered



Other Requirements

- Legal issues (money, privacy, children, health, accessibility …)
- Reliability (up time, data safety)
- Performance
- Time and cost limits
- Adoption strategy
  - Meeting all stakeholders needs
  - Especially important for social/rating apps
- Testing strategy



Completeness

- The more you can state, the better off you'll be
- The requirements are going to change
  - Clients will want different things as the project proceeds
  - New options and possibilities will arise
  - Some things might be too difficult
  - Your designs might not appeal to users
    - Even if they appeal to the client
- Keep these up to date

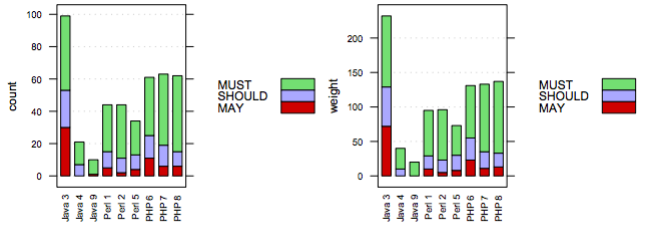
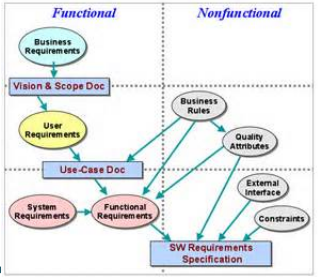


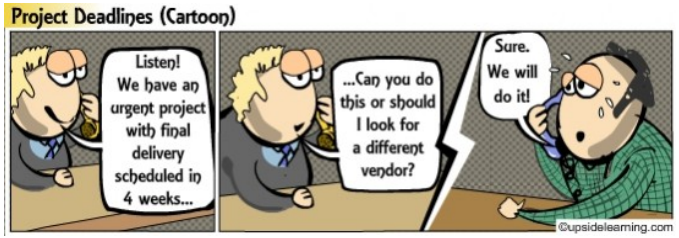
Figure 4.4: Completeness of solutions in terms of the UI-related requirements implemented that are working correctly (grades 2,3,4). LEFT: Number of requirements. RIGHT: Ditto, weighed by effort.

Specifications



- Next step after understanding what the user wants
- Also known as Requirement Specifications
- Different from requirements
  - Requirements look at things from USER's point of view
  - Specifications look at things from PROGRAMMER's
- Detail WHAT will be built (not HOW)
  - Essentially a contract
  - We'll go over these in more detail briefly next Wednesday

Project Deadlines



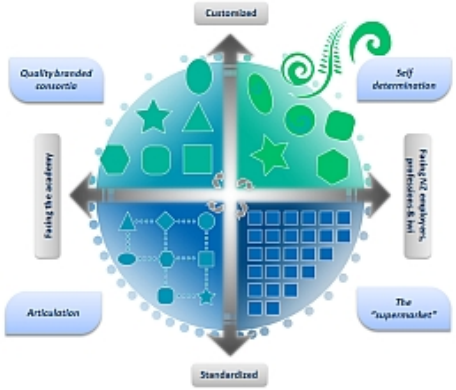
- By 2/20 you should have talked to your clients
  - Have a good understanding of the project
  - Think of several stories you can work with
  - Provide a short report detailing this (meeting notes)
- By 2/25 you should have
  - Interacted with potential users for defining requirements
  - Gotten client and user feedback on their needs
  - Determined the outline of a potential solution
  - Handed in an initial specification

Next Time

- LAB on Monday

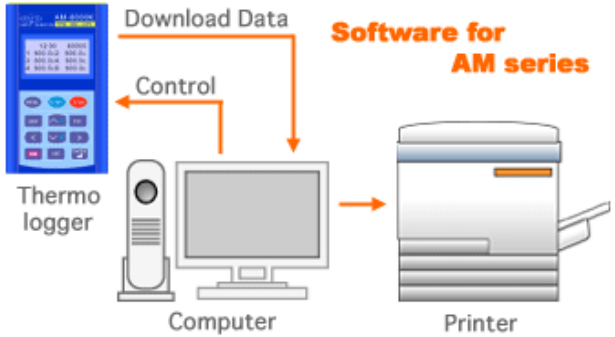
Specifications

- Specify what will be done
  - **Scenarios**
  - Lists of features to implement
  - Note optional versus required (priority)
- Define the user experience
  - Sketches of web pages (not final design)
- Identify interface to existing systems
  - Servers, databases, etc.
- Outline of web site and pages
  - List of what pages are needed



Specifications

- Detail what the application will do
  - From the programmer's point of view
  - Can talk about other systems, components, modules
  - More likely to talk about commands, inputs, outputs
  - **WHAT** not **HOW**
- Define the inputs and outputs
  - What information is needed
  - What information is used
  - Where does this information come from
  - Where does this information go
- Specifications Document due 2/26



Specifications Do Not

- HOW not WHY
- Identify particular technologies to use
  - Unless mandated by outside requirements
  - Back ends, front ends, databases, …
- Identify how tasks are done
  - Front end, back end, database
  - Particular algorithms or processing
- Provide detailed web site designs

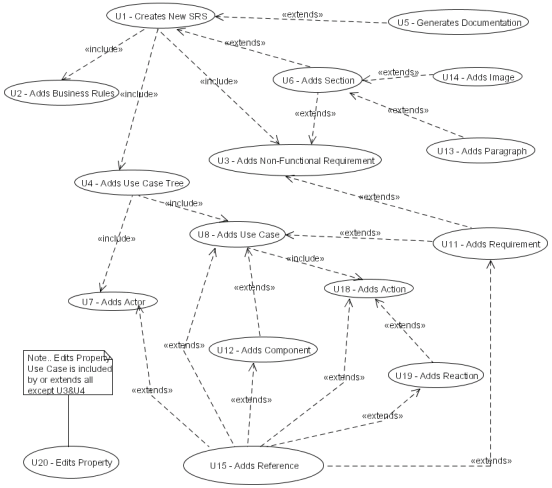
Question

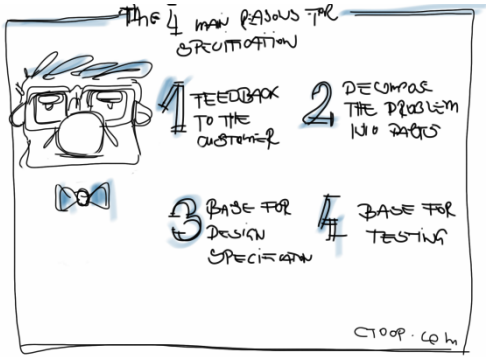
Specifications do not include:

A. Analysis of user surveys or interviews
B. Diagrams describing the user interface
C. Descriptions of user commands and what they do
D. Description of how system features relate to user requirements
E. Descriptions of existing systems the new system will have to interact with

Creating Specifications

- How might you do this?
  - What would you do with twitter data application



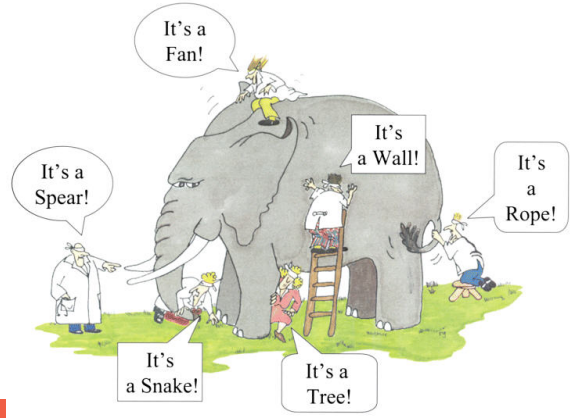


Major Causes of Software Failure

- Poor User Input
- Stakeholder Conflicts
- Vague Requirements
- Poor Cost and Schedule Estimation
- Skills that do not Match The Job
- Hidden Costs of Going “Lean and Mean”
- Failure to Plan
- Communication Breakdowns
- Poor Architecture
- Management Problems

Obtaining Requirements

- How might you do this?
- Example:
  - CD Query Data (from previous lab)
  - How would you find out what is wanted?
  - Who would you ask?
  - What questions would you ask?
  - What do you really want to know?



What's Worse than a Bad Client

- He who represents himself has a fool for a client
  - Holds in law
  - Holds in medicine
  - Holds in software development
- Clients emphasize the user and the user's needs
  - Programmers emphasize the programmer
  - Easier to code doesn't mean easier to use
- Those of you working on your own project
  - Do real user studies
  - Find a set of users outside your group and listen to them

Designing the User Experience

- Should be done interactively with your clients
  - And with perspective users
- Provide them with 3 (or more) alternative designs
  - To show the range of available options
  - To avoid commitment on your part
- Be prepared for multiple iterations

Create A Scenario

- For either of the previous labs
- For your final project
- For using banner

Sample Interviews

Question

What phase of development involves understanding what the system one is writing is going to do?

A. Requirements
B. Scenarios or use cases
C. Specifications
D. Human factors
E. Web site design