Software Design

*The Journey of an Idea to Invention*

(Week One)
Learn By Doing

- Develop a system through which consumers can receive coupons that are both temporally and geographically relevant. - e.g. coupons for Potbelly's at lunchtime.
Quick Review

- Questions/Assumptions?
- Use cases?
- Customers?
- Actors?
- Actions?
- Events?
Who Am I?

- Cybersecurity-focused Engineer for USG
- B.S. CSCI 2005 (GW)
- M.S. CSCI 2006 (GW)
- Taught Computer Network Defense @ GWU
Goals

- Practice software design
- Understand design considerations
- Identify the value of software design
- Think critically
Poor design leads to...
Poor design leads to....

I could restructure the program’s flow
or use one little ‘goto’ instead.

Eh, screw good practice. How bad can it be?

goto main_sub3;

*compile*

[Image: A comic strip showing a sequence of events related to poor programming design. The first panel shows a person suggesting to restructure the program’s flow or use a goto instead. The second panel shows another person dismissing good practices. The third panel shows a code line, ‘goto main_sub3’, and the fourth panel shows a person coding with their feet up on their desk.]
Software Failures

- Mars Climate Orbiter
- Y2K (Y2K38?)
- F22-Raptor
- Your last CSci 3212 assignment
More realistically...
The Challenge

“...And that, in simple terms, is what’s wrong with your software design.”
What is “Software Design”? 
What is “Software Design”? 

- “Software design is a process of problem-solving and planning for a software solution.”
- “Software design is the process by which an agent creates a specification of a software artifact, intended to accomplish goals, using a set of primitive components and subject to constraints.”
What is “Software Design”?  

- “The final goal of any engineering activity is some type of documentation.”
- “…the only software documentation that actually seems to satisfy the criteria of an engineering design is the source code listings.”
Design Considerations
Initial Approaches
Initial Approaches

- Start with the *most important thing*
  - Often it's the customer (e.g. the user interface)
  - Might be a critical component
  - Whatever it is that pays the bills
- "Begin with the end in mind."
  - *Seven Habits of Highly Effective People*
Learn By Doing 2

FoodFriends: Request “live” advice from friends on local restaurants and bars.
What is the “Main Idea”? 

- “I want to ask my friends if they know good food and drinking spots near me at any given time. I should get a graphical representation of their responses, in a chart.” 
- How should this work? 
- What features are needed? 
- Can you restate the problem?
Initial Use Case

Me

Request

Friend

Can we do better?
Maybe This?

Hungry Person

Food request

Voters

But can we identify components?
Even Better...

Request Originator

Request Acceptor Service

Requests Advice

Request Acceptor Service

Requests Notification

Notification Service

Request

Notification Service

Voters
What next?

Does this suggest any components?
Components

Client App

- Vote Requestor
- Vote Receiver
- Request Processor
- Notification System

Server App

- Vote Request Processor
- Vote Response Processor
- Messaging System
- History Store
Workflow w/ Components

- Draw on board....
Putting it together...
Requirements

- **Functional**
  - Input → Output (and behavior)
  - Example: Request for suggestions → geolocation data sent to server

- **Non-Functional**
  - Judge *operation* of a system
  - Examples: security, speed
Specifying Requirements

- Iterative process of refinement:
  - Problem Statement
  - High-level requirements
  - Use cases
  - Workflow
  - Functional requirements
  - Non-functional requirements
  - Components
More work on the board...
Message

- Organized, iterative, thoughtful, refinement
  - High level
  - Use cases
  - Workflow
  - Requirements
  - Rinse
  - Repeat
UPDATE DOCUMENTATION AS YOU GO
Tools

- Visio
- Omnigraffle
- OpenOffice
- Gliffy
- More?
When do you stop?
And eventually....
Homework!

- Due: Before class begins on 11 September 2013
- Develop a specification document that describes users, use cases, workflows, features, and structure
- Prepare a short screencast that explains your specification document to a developer
Turning plans into reality

TO DO LIST:
1. WAIT FOR TONIGHT
2. TRY TO TAKE OVER THE WORLD!
Waterfall Model

- Requirements
- Design
- Implementation
- Verification
- Maintenance
V-Model
Spiral Model
Iterative & Incremental
Agile Development

Agility is...

VALUES

adaptability
transparency
simplicity
unity

Strategy
charter
funding
estimation

Release
goals
vision
backlog
review

Iteration
release plan
iteration plan
standup

Daily
retrospective
acceptance testing

Continuous
TDD
refactoring
integration
collaboration

Delivery
working software

visibility

Fall 2013
Design I
Scrum
Next Steps

- Do homework
- Create low-level workflows
  - UI mockups
  - Algorithms and data structures
  - Control flow
  - Code structure
- Task List
- Code