CS 160: User Interface Design

The Design Cycle & Brainstorming 01/24/11
Font Selection in Keynote
Font Selection in PowerPoint
TipCalc2

Check Amount:

Tip Amount
- 10%
- 15%
- 20%

Split: 1 Person

Total Each:

$0.0
Enter bill amount:

$0.00
Review

Course overview

Project theme

Course mechanics
Due Last Friday

Wiki account
Course petition
Enrollment

1. Decisions were sent on Sunday; we could not admit everyone.

2. Not sure you will stay? Drop by this Friday and email cs160@imail.

3. Official database update has latency.
Due Today (before class)

Reading comment
(1 per lecture; cs160/cs160Reading)
Reading Response: Lewis & Riemann (1/24)

How does the Lewis & Riemann design cycle compare to the three stage (design, prototype, evaluate) cycle described in lecture?

This article was written at a time when most software was sold "shrink-wrapped" in stores. Does their argument about the design process hold equally for web applications? If not, what has changed?

Where do you disagree with Lewis & Riemann? Why?

#reading_response

Last updated by Bjorn Hartmann 5 days ago

It looks to me that the basics of "Design, Prototype, Evaluate" are in L&R's design cycle, but they seem to blow up each of the steps into extra detail. Much of the advice holds true today. However, in the last couple steps, L&R talk about "maintaining their sales," and how their products may fall out of favor with users in a little while. Their approach seems dated in a way.

One of the assumptions L&R make is that to stay cool, you must capitalize on new trends in culture. There is one big problem with that, and that's the danger of feature bloat. In the age of web apps, successful web apps pick a very narrow focus, and make sure to do it right and well. They avoid latching onto fads that may only just weigh down and over-complicate the application.

Shrink-wrapped applications are large bundled packages that companies buy to solve all of their needs, as cheaply as possible (in terms of bundle pricing, as well as tech support subscription). Web apps are best served in byte-sized portions, where a consumer mixes and matches several small, discrete, and well polished services to make up his or her experience.

Take for example, the difference between Yahoo and Google. While Yahoo mashes all of its products together on its home page to the point where it can't decide what it is, Google sticks to the philosophy of keeping its apps separated for simplicity and elegance. Yahoo has had a sort of identity crisis, fueled either by depression that it's losing market share, or the denial that people seem to only use it because they're too lazy to change their email account. Google knows it is first and foremost, a search engine, and its home page makes that clear. Google doesn't flaunt its other apps, but instead lets the Internet hype draw users to its other pages, which are equally focused and styled to the designed task.

Last updated by Chris Woytowitz 4 days ago
Amanda Ren (4 days ago) - 1. How does the Lewis & Rieman design cycle compare to the three stage (design, prototype, evaluate) cycle described in lecture?

Both cycles start with observing the current habits of the user. You want to consider what the user's current practices are, and what in mind what real tasks your users will use the system for. Then the cycle from lecture tells us to do rapid prototyping. The Lewis & Rieman cycle tell us to think carefully about all tasks that the system should have to support before creating a prototype. Then finally, both cycles tell us to test the prototype with users, obtain feedback, and iteration through the design process once again.

2. This article was written at a time when most software was sold "shrink-wrapped" in stores. Does their argument about the design process hold equally for web applications? If not, what has changed?

If we were using web applications, then we should iterate between "building the design, tracking the design, and changing the design" because we could easily update our product to meet consumer needs once it is brought to market, which isn't easy for "shrink-wrapped" software. So for web applications that can be easily changed, we can give different subsets of users different versions of a final product, and based on the feedback, release just one version as the absolute final product. Obviously, we cannot easily do this for "shrink-wrapped" software.

3. Where do you disagree with Lewis & Riemann? Why?

In the "rough out the design" step, they say to discard any new features that do not support any of the representative tasks. I disagree because I feel that the new feature may actually potentially help the user, even if it does not support one of the current representative tasks.

Joe Crnkovich (10 hours ago) - Lewis and Riemann do mention another option in dealing with the aforementioned new features with no immediate relevancy, as you may have seen. They advise going back to your initial task list and making additions if the new feature could in fact help the user. I believe they want you to take this extra step in going back to the original list so that you keep the focus on tasks in your design. The task-centered design process seems to be a great way to maintain focus and applicability in designing user interfaces.

You make a good point in discussing an easier testing method for various versions of a product. As you kind of mentioned, it is but the work of a moment to provide updates to web applications. "Shrink-wrapped" software takes considerably more time, making it necessary for the consumer/tester to buy the new software and then install it. This process is also much more costly to implement, with the costs of CDs, gas, etc.

As you say, the two cycles are very similar to each other, with the task-centered design process attending to more details. It is interesting to note that the Lewis and Riemann cycle stresses looking for old code or patterns. This can cut down the time and energy necessary for a project by a great deal. This is something that I plan on utilizing as much as possible for obvious reasons. The goal-oriented approach tends to work pretty well in my experience.
Proposal

Use the **answer field** to **summarize** the most important definitions and arguments from the reading – treat it as a wiki, edit the answer together.

Use the **follow-up discussion + comments** for **individual replies**.

Both answers and responses to answers count. But your responses must be as detailed and well-reasoned (“Yeah, what he said” doesn’t count.)
Reading Response: Norman (1/26), Design of Everyday Things

Reading Summary (post as “Answer” - optional but very useful for the midterm): Summarize the main concepts in Norman’s chapter (e.g.: affordances, visibility, mental model, mapping, feedback, paradox of technology).

Individual question
(post as “Follow-up discussion” - required):
1) Give an example of a physical device (an “everyday thing” as Norman would call it) with bad design that you have had to use. Do not think about software! Think about household appliances, sports equipment, cars, public transportation, etc.) Which of Norman’s design principles did this device violate? How would you re-design it to solve the problem?

2) Are there any differences in affordances of physical devices versus affordances of software user interfaces? In this context, what does Norman mean when he mentions “perceived” versus “actual” affordances?

Tips and Tricks?

Auto-complete slow in Eclipse?
Hey guys, When I updated with my fresh install of the Android SDK in Eclipse (Helios), sometimes the auto-complete feature would stop.

Workaround for Unix problem
Hi Everyone, In the second section we faced problems with many machine not running the virtual devices with an error saying NAND.

mac target error FIX!
Howdy fellow mac users if you were having problems specifying a target when you go to make a new AVD, do the following: eclipse -> - Instructor thinks this Note is good.
Due Friday 5pm

Individual Design Exercise
Programming Assignment 1
Design Exercise

The point is NOT to implement one of the examples I listed in the assignment.

Brainstorm at least 12 ideas – go for breadth (radically different ideas).

Then pick the best idea, prototype and evaluate.
Help with Programming Assignment

Extra office hours tomorrow?
Section on Thursday (~24hrs before deadline)

Alternative: Follow these two Android tutorials.
Hello World
developer.android.com/resources/tutorials/hello-world.html
Hello Views – Form Stuff
developer.android.com/resources/tutorials/views/hello-formstuff.html
Concepts you will need to learn

Creating layouts and widgets
Edit XML file or use Eclipse GUI editor
Referring to widgets in Java code
  e.g.: findViewById(R.id.foobar);
Adding event listeners
  code that gets called when the user interacts with widgets
Showing output on the screen
  e.g., TextView.setText()
Topics for Today

1. The Design Cycle

2. Brainstorming
The Design Cycle
Design

Evaluate

Prototype
The Art of UI Design

A soufflé is eggs, butter, milk & flour, but the difference between soaring and sinking is in the execution.
The Design Process [Koberg & Bagnall]

1. Acceptance
2. Analysis
3. Definition
4. Ideation
5. Idea selection
6. Implementation
7. Evaluation
The Design Process [Koberg & Bagnall]

1. Acceptance
2. Analysis
3. Definition
4. Ideation
5. Idea selection
6. Implementation
7. Evaluation
Acceptance

Getting started
Because of a deadline
Because of possible reward
Because you are forced to

Commitment
Time
Resources
Responsibility

Key is to set motivation
Analysis

Understand Users and Tasks
Who are the users?
What are their tasks?
Observe and test, don’t guess

Tools
Notebook
Tape recorder
Camera
Video camera
Definition

Focus on the problem
Choose appropriate level of detail

Not “bicycle cup-holders” but “helping cyclists to drink coffee without accidents”
Ideation

**Brainstorming**
- Stretch mental muscles
- Loosen up with simple games
- Do homework
- Seed with related ideas/objects

**Get physical**
- Sketch
- Make models
- Act out

**IDEO rules**
- One conversation at a time
- Stay focused
- Encourage wild ideas
- Defer judgment
- Build upon idea from others

*Aim for quantity*
Idea Selection

Define importance of each idea
Does it address problem
Will target users like it
Is hardware available
Is software available
What is the cost
Market window
...

Rank ideas according the your criteria

Pick top N
Choices depend on resources and stage of the project
Implementation

Scale up low → high fidelity
Implementation

Scale up low → high fidelity

Low-fidelity (quick, cheap, dirty)
sketches, paper models, foam core, …
Implementation

Scale up low $\rightarrow$ high fidelity

Low-fidelity (quick, cheap, dirty)
sketches, paper models, foam core, ...

Medium fidelity
(slower, more expensive)
Flash, JavaScript, AJAX, ...

Design Process

1. Acceptance
2. Analysis
3. Definition
4. Ideation
5. Idea selection
6. Implementation
7. Evaluation
Implementation

Scale up low ➔ high fidelity

Low-fidelity (quick, cheap, dirty)
- sketches, paper models, foam core, …

Medium fidelity
- (slower, more expensive)
- Flash, JavaScript, AJAX, …

High fidelity
- (slowest, most expensive)
- The full interface
Implementation Example: Web Design

Sites created at multiple levels of detail
Sites iteratively refined at all levels of detail
Iterate quickly to see what works

Site Maps → Storyboards → Schematics → Mock-ups
Evaluation

Many types of evaluation:
Prototype walkthroughs
Think-aloud studies
Wizard-of-Oz
Performance comparisons

Type of evaluation chosen depends on the level of implementation, etc.
Evaluation Example: Paper Prototype Walkthrough

Observer (or video camera)

User

“Computer”

Interface

Interface elements

Design Process

1. Acceptance
2. Analysis
3. Definition
4. Ideation
5. Idea selection
6. Implementation
7. Evaluation
Design Cycle Over Project Lifespan

1. Acceptance
2. Analysis
3. Definition
4. Ideation
5. Idea selection
6. Implementation
7. Evaluation

Number of Ideas under consideration

Project timeline
Design Cycle Over Project Lifespan

Evaluation reveals problems with design. Re-design requires cycling the process.
Prototype implementations eventually increase in fidelity to reach final product.
Comparison

Who will use?
What are their tasks?
Plagiarize
Rough out a design
Think about design
Create a prototype
Test it with users
Iterate
Build a production version
Track use
Evolve the design

[Lewis & Rieman]

[Koberg & Bagnall]

Design Process

1. Acceptance
2. Analysis
3. Definition
4. Ideation
5. Idea selection
6. Implementation
7. Evaluation
Comparison

Who will use? [2]
What are their tasks? [2]
Plagiarize [4]
Rough out a design [4,6]
Think about design [5]
Create a prototype [6]
Test it with users [7]
Iterate [7->1]
Build a prod. version [6]
Track use [7]
Evolve the design [7->1]

[Lewis & Rieman]

[Koberg & Bagnall]
Waterfall Model (Soft. Eng.)

Initiation

Application Description

Analysis

Requirements Specification

Design

System Design

Implementation

Product
Comparison

**Focus differs**

WF has no feedback

High cost of fixing errors: increases by 10x at each stage

Iterative design finds problems earlier

True for modern web applications?
How well do they follow the cycle?
What do they do for each step of the cycle?
How many cycles do you think they went through?
Brainstorming
The Psychology of Creativity

Conformity: the enemy of creativity

Groups and organizations encourage conformity

Part of “brand” or “corporate identity”
The Psychology of Creativity

Pressure to conform affects judgment and perception:
The emperor’s new clothes
McCarthyism: if you’re not one of us, you’re one of them…

People in minority will adopt majority opinion and even manufacture their own explanation of it.
Creativity and Dissent

Authentic dissenters – people who really disagree with group – can enhance group creativity.

Their opinion needn’t be right – but they can free the group from stagnant thinking.

The originality of the minority stimulates the majority.
Creativity and Dissent

Authentic dissenters – people who really disagree with group – can enhance group creativity

Their opinion needn’t be right – but they can free the group from stagnant thinking.

The originality of the minority stimulates the majority
Dissent and Authenticity

The benefits of dissent are weakened if

Dissent is not real: A deliberate “devil’s advocate” in the group can actually stifle dissent, because the majority know the opinion is manufactured.

Dissent is not encouraged: Polite or pro-forma acceptance is not enough.
Thinking outside the box:
Draw a series of 4 straight lines through all the points below, without lifting pen from paper:
Why Is This Hard?

We adopt expectations about the solution
Based on conventions
Based on what we believe the questioner expects
IDEO’s Brainstorming Rules

1. Sharpen the Focus
2. Playful Rules
3. Number your Ideas
4. Build and Jump
5. The Space Remembers
6. Stretch Your Mental Muscles
7. Get Physical

Aim for quantity
Hope for quality
Sharpen the Focus

Posing the right problem is critical – neither too narrow, nor too fuzzy

Not “bicycle cup-holders” but “helping cyclists to drink coffee without accidents”
Number Your Ideas

Obvious but very useful

Helps keep track of them when the brainstorm is successful (and 100 or more ideas are in play)

Allows ideas to take on an identity of their own
Build and Jump

**Build to keep momentum on an idea:**
“shock absorbers are a great idea; what are other ways to reduce coffee spillage on bumps?”

**Jump to regain momentum when a theme tapers out:**
“OK, but what about hands-free solutions?”
Concept Refinement

Premature idea rejection is a serious barrier to good design.

One big differentiator between good designers and great ones is the latter’s ability to successfully develop unusual ideas

This requires a strong instinct to be able to distinguish fatal vs. minor flaws in an idea
The Space Remembers

Covering whiteboards or papering walls with text is extremely useful in group work.

It’s a very effective form of external (RAM) memory for group

Even better, its shared RAM. Helps group share understanding
Stretch your Mental Muscles

Warmups: word games, puzzles

Get immersed in the domain: go visit the toy shop, or the bicycle shop, phone shop etc…

Bring some examples of the technology to the brainstorm
Get Physical

Sketch
Make models
Act out

Moggridge, Designing Interactions, p.732
Next Time

Lecture Topic: Sketching, Storyboarding & Critique
Bring pens+paper! (Ideally unlined).

Reading:
Don Norman, The Design of Everyday Things.

Don’t forget!
Read, then write a comment on Piazza
Programming Assignment & Design Review, Due this Friday, 5pm