Human Computer Interaction
Issues

Prof. Summet
Problems with design

- Design problems may strike anywhere.
- Becomes a human factors issue when it leads a human to perform an unintended action.
- If the unintended action is positive, or results in a positive consequence, it's usually not spotted as an issue.
- When something negative occurs, an issue is present.
What was the problem?

- New zoning rules require one toilet for every 1000 square feet of retail space.
- Does not specify that they must be separated from one another by a partition.
- Issue with the implementation (letter of the law) of this new zoning requirement.
- But not an HCI issue (no computers, yet).
How many problems can you have with one product?

- Or rather, the product's documentation manual?
Illiteracy can be hazardous to your baby's health.

8. Heaters should be kept away from children and not be placed in a child’s room without supervision.
So, guess what this picture means...
Yep, that's right.

9. String out cords on top of rugs or floors. Placing anything on top of the cord may damage it.
But at least they help out Spanish speakers

- Manual is bilingual, as is the back side of the box.
One safety feature

- The heater has a built in “danger, hot” indicator that changes color to warn you when the grill is too hot to touch.
- The English side of the box tells you how it works:
So does the Spanish side...

- If you only read Spanish, you still get all of the pertinent information. Well, mostly...
Bad design issues can be hazardous, costly.

- If you have a printing mistake on 100,000 cardboard boxes and manuals, you have to make a decision. Do you ship the merchandise, or re-print the boxes and manuals, plus hire workers to re-box everything?
- Or do you sell your stock at a discount to Woot.com to be purchased by technology savvy individuals who are assumed to be literate English speakers?
- But so far we haven't really looked at HCI issues.
- Surely since software is infinitely mutable, fixing errors will be much less costly, so errors will be less likely to occur.
- Well, maybe...
Problem with a gas pump credit card payment terminal

- Either the hardware or the software engineers didn't read the updated user interface panel design specifications.
- Or the software was reused from a different terminal design.
- But really, how long does it take to find the YES and NO buttons on the keypad? Maybe 10 seconds...not a costly mistake.
Or is it?

- 10 extra seconds for a novice customer. (Repeat customers have it all figured out and remember what to do).
- Average time to gas-up goes from 180 to 184 seconds across all customers.
- Average profit per customer fill-up is 32 cents.
- Extra time at pump leads to longer wait times (and visible lineups) during peak time.
- What if one person a day decides to get gas elsewhere because she saw a long line? 32 cent loss, PER DAY!
- $0.32 \times 365 = $116 a year. Well, maybe this example isn't so bad.
Now that's some money!

- 10 extra seconds for a novice customer. (Repeat customers have it all figured out and remember what to do).
- Average time to gas-up goes from 180 to 184 seconds across all customers.
- Average profit per customer fill-up is 32 cents.
- Extra time at pump leads to longer wait times (and visible lineups) during peak time.
- What if one person a day decides to get gas elsewhere because she saw a long line? 32 cent loss, PER DAY!
- What if you have 14,321 gas stations nationwide? 0.32 * 14,321 = $4,583 a day, or $1,672,692 dollars a year!
Another example:

- “Suzie, can you bring up the discount page on my computer and give Guitar Hero a $30 discount for the holiday rush?”
- “How do I do that?”
- “select PKU GH13122.....navigate to the discount field, and subtract 29.99”
- “Ok, got it! The stock boys will handle the price change overnight, lets head out.”
Are discounts negative or positive? (-29.99 or 29.99?)
Case Study: New Dream Networks, L.L.C.

• Four CS undergraduates started the company in 1997 with “no capital apart from a single Pentium 100 web server...using shared bandwidth on a T1 line that a friend gave us at no cost.”

• You may have heard of them as www.DreamHost.com

• The company is employee owned, and in 2009 their corporate website claimed to have 70 employees and over 1,500 servers which host over 800,000 websites.
A note about DreamHost's transparency...

- Meet Josh Jones and Dalis Kashuba, two remaining co-founders (out of four)

- As you may have figured from the founders' photos, DreamHost isn't like most corporations.
- They have a blog, and when something goes wrong with their servers they notify customers via the blog.
- The true story that follows is based upon three of their blog posts (written by Josh) in January 2008.
A bit of history: December 29th-30th 2007

- Dreamhost started to replace some of their internal “control” servers with better hardware.
- Josh: “These are the machines that run all of our 'behind-the-scenes' services; things from adding a user to registering a domain to configuring [webservers] to rebilling customers.”
- Josh was on a vacation, but when he got back he noticed that the daily credit card payments seemed a bit low.
- In an effort to fix this error, he re-ran the biller program.
About the DreamHost Biller program

- The DreamHost biller program is a piece of custom in-house software that looks in the dreamhost database to see what services customers have.
- If a customer has a service that has not yet been billed, it will bill the customer for the service. (And record the fact that the customer was billed.)
- If the customer is set up with a credit or debit card on file, the biller will automatically charge their card.
- Other parts of the system also send email notifications of pending re-billing events to customers, giving them a chance to cancel the service before being billed.
About the DreamHost Biller program

- The biller program had been rock solid for almost a decade. As Josh puts it:

  “I knew this was safe, because after 10 years, the one thing you DO get perfect is your billing system. Our biller is pretty bug-free and robust at this point, because we’d be broke and eating bugs if it weren’t.

  In fact, it’s so robust you can just run it on any day you want, and it’s safe. It won’t double-charge people and it’ll even automatically find any missing charges and catch everything up to the day you said.”
Josh re-ran the biller program on the last week of 2007, and it worked just fine.

It caught several missed charges, and applied them.

He didn't have time to look into WHY the charges had been missed initially, and went home for the weekend.
Monday, January 14th, 2008 everything was just fine...

- Josh comes in on Monday and tries to figure out why credit card payments are still low: “So I looked at the logs for some of the biller services, and I noticed they were only failing on the machines that had been recently upgraded!”
- Problem involved upgrading 32 bit servers to 64 bit servers and a credit card processing PHP module that only worked on 32 bit servers.
- Josh installed a new module that worked on 64 bit servers, and tested it.
- Everything is now working!
Late Monday night, Josh realizes that:
“when I re-ran those biller services last week, they must not have fixed everybody then either! It’s just that by running it again I randomly got different people being charged on the working controllers who had been assigned an upgraded (and therefore broken) one before.”

So why not run it one more time, now that all the controller computers are working?
“Sure, it should be no problem!”
Late Monday: Josh re-runs the biller for the final time

Josh re-runs the billing program for all days this year to make sure that everybody who was supposed to be charged had been charged.

Then, because he couldn't remember when exactly the had replaced the controller computers, he re-ran the biller for the end of last year too:

“I probably should have just stopped there. But then I thought better. I thought to myself, 'When did we start upgrading these controllers anyway?'

I couldn’t remember. But, since the biller is super-safe and robust anyway, I went ahead and ran it for 2008-12-31, 2008-12-30, 2008-12-29, 2008-12-28, 2008-12-27, 2008-12-26, and 2008-12-25, just for the hell of it.”
Notice anything?

- Don't feel bad if you didn't, neither did Josh.
Graph of DreamHost charge receipts...
Notice anything?

- Don't feel bad if you didn't, neither did Josh.
- Those last few days were supposed to be the end of 2007.
- “That super-robust and stable biller did what it was programmed to do, it ran as though today was December 31st, 2008!”
- (Today was actually Monday January 14th, 2008)
The biller program had worked hard overnight!

As told by Josh:
“[the biller] saw a whole lot of accounts (essentially all of them) who for some unknown, mysterious reason hadn’t been charged at all for eleven and a half months!

So off it went, busily through the night, 'fixing' everything up for 'today', December 31st, 2008.”

Other software also ran which cut off accounts of those customers who were more than a month overdue.
Keeping the biller program flexible

- Some programmer probably thought to themselves, “Well, we might want to bill for a future date, what's the harm in keeping it flexible?”
Some programmer probably thought to themselves, “Well, we might want to bill for a future date, what's the harm in keeping it flexible?”

$2,100,000.00

Over two million dollars in erroneous charges to credit/debit cards.

Plus just a small amount of customer dissatisfaction.
They were actually lucky....

- The biller program actually tried to bill

$9,600,000.00

but due to a bug with the new credit card software on the 64 bit servers only the first charge per second was passed through to their credit card merchant account, resulting in “only” two million dollars of erroneous charges:

$ 2,100,000.00
Keeping the biller program flexible

- Josh said “NO! We will NEVER want to rebill as though today were a day that hasn’t happened yet!”

- And: “When designing a program, you’ve got to make some tough decisions .. and when you really can’t decide if this is something your users will need someday, err on the side of leaving it out.”
Repercussions...

- 10,000+ (angry) customer service emails on Tuesday.
- Thousands more over the next several weeks.
- Loss of customers.
- Loss of “Blue Sky” or “good will”.
- Monetary costs for charging and then refunding $ 2.1 Million (Credit card processing fees, typically 1-3%).
- Reimbursing customers for overdraft and NSF charges from their banks (when debit cards overcharged or credit cards pushed past their limit).
- Customer service representative's overtime pay.
Um, Whoops.

January 15, 2008 on 9:52 am | In Helpful, Insider View, Musings by Josh Jones | 667 Comments

Hello... how's your morning going?

I hope it's been a little better than mine.

We had a teensy weensy little billing error last night... my first clue something was up when I saw this morning's daily billing report (so far): $7,500,000.

It turns out due to my excessively fat fingers, nearly every one of our customers has been seriously over-billed in the last 12 hours.

I bet when you read this part of the last newsletter:

4. Now Official

...
Customers value their money!

- When you “steal” money from their credit (or worse, debit) card, they get mad, very mad.
- Especially if the unexpected charge pushes their credit card account over their credit limit, or empties their debit card (checking) account!
- They (mostly) understand that mistakes can happen, but expect a certain level of gravity when discussing such a major mistake.
“First, I just want to apologize for the regular-style blog post about it yesterday. Hopefully this will be the (picture, bold, and italics-free) blog post many of you would have liked to have seen yesterday.”

and:

“P.S. I apologize for that joke about the triple billing in the newsletter thing too, but you have to admit, it was kind of ironic that I actually did screw up billing less than a week later.”
Other things that Josh learned:

Monday: Although charging a credit card is instantaneous, refunding really does take 3-4-5-6-or-more business days to process.

Tuesday: You can erroneously credit an expired credit card. The money does leave your merchant account.

Wednesday: You can credit a canceled credit card. The money does leave your merchant account.

Thursday: You can credit a debit card tied to a checking account that has been closed for months. The money does leave your merchant account.

Friday: If you charge somebody with an international credit card and then refund their money, by the time the money gets back on, the dollar will have weakened!
How this could have been prevented

- Safety interlock – At a minimum, prompt the user for confirmation if the billing date is in the future, or even flatly disallow future billing behavior. (The DreamHost biller now flatly disallows future billing.)

- Other checks implemented by DreamHost (after the fact):
  - System will not charge a customer more than 150% of their previous largest charge.
  - If the biller finds more than 3 “missed” charges, it quits billing and notifies the (human) financial team.
  - If total number of charges processed in a day is more than double the average for that calendar day, the biller stops and notifies the financial team.
Overall DreamHost did a good job

- Made a serious mistake had had negative consequences for their customers due to a human error (magnified by lack of confirmation / sanity checking of the biller software)
- Accepted responsibility, apologized, and made reparations.
- Communicated openly with their customers about the mistake.
- Took steps to ensure that that particular class of mistakes would not be repeated.
Takeaway points

• “...when you really can’t decide if this [feature] is something your users will need someday, err on the side of leaving it out.”

• Don't always assume the user knows what they are doing.