



Introduction to User Experience Design for Engineers

Presented by
Jeff LeBlanc



Presented By

Jeff LeBlanc

User Experience Team Lead @ ICS

Software developer for 20+ years

Certified Qt trainer since 2003

Adjunct faculty at WPI teaching HCI

Contact me: jeffl@ics.com



Two Questions

- Who has coded a User Interface because you like creating UIs?
- Who has coded a User Interface because you needed something for your back end code?

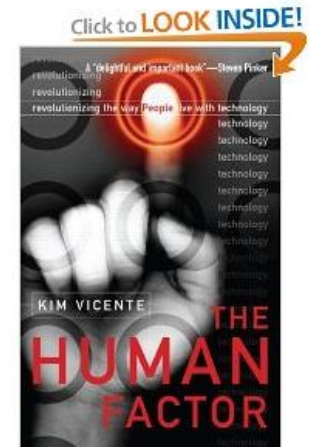
Why Do You Care About UX?

- Qt and QML are very powerful toolkits, but they focus on “how” and not “what”
- Software developers focus on “how”
- Users care about “what”

- What your app does is more important than how it does it (to the user)

Why Do You Care About UX?

- Bad design effects people
 - Adds stress
 - Can cause errors and cost lives
 - Can lose customers
- Good design effects people
 - Improves job performance and satisfaction
 - Can save lives
 - Makes people want to use your product



Why Do You Care About UX?

- A mobile app needs to capture the user's attention very quickly
 - 30 seconds or they move on
 - The user experience is what
 - makes someone keep using the app
 - makes someone recommend it to others

Focusing on UX

- Increase customer satisfaction leads to increased revenue
 - Wixon & Jones case study, 80% revenue increase by focusing on usability
- Reduce training costs
- Reduce support costs
 - McAfee reduced support 90% by focusing on UX (2005)

User Experience vs User Interface

- All software has a User Interface (UI) of some type, but how do you feel after using it?
- A positive User Experience (UX) means
 - You enjoyed using the system, or at least did not dislike it
 - You would use it again and recommend it to others

Can Engineers Focus on the UX?

- Software engineers think like engineers, not like end users or designers
 - Different mental models than target users
 - Different skill sets than UX/visual designers
- Often get an interface that only makes sense to other technical people
 - Techies like to figure things out, but not everyone else does

“Know Thy User” - Hansen

UX Design as an Engineer

- Things to keep in mind if you're doing the design and the implementation
 - Remember that you are not the target user
 - Work with a UX designer
 - Keep in mind UI guidelines
 - As easy to implement the UI “right” as it is “wrong”

UI Design Guidelines

8 Golden Rules for Interface Design

Ben Schneiderman – U. Maryland

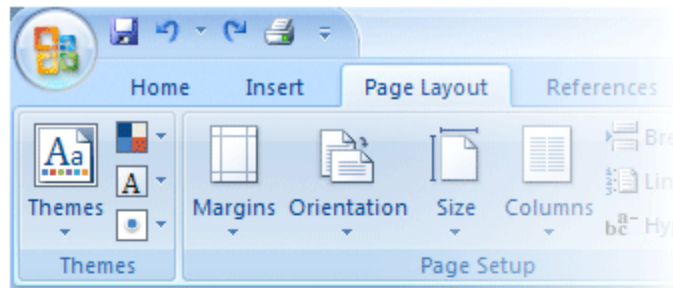
1. Strive for consistency
2. Cater to universal usability
3. Offer informative feedback
4. Design dialogs to yield closure
5. Prevent errors
6. Permit easy reversal of actions
7. Support internal locus of control
8. Reduce short term memory load

1. Strive for Consistency

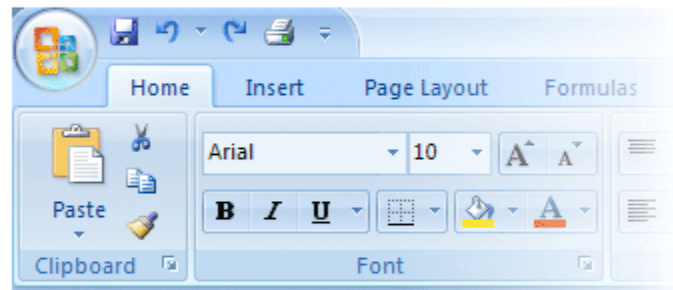
- Consistency can be
 - Within a product
 - Across products within a company
 - Across the software industry
 - Ctrl-C to copy
- Benefits of consistency
 - The user benefits from transfer of learning
 - Marketable as a “look and feel”
- Perhaps the most frequently violated or ignored rule...

Consistency

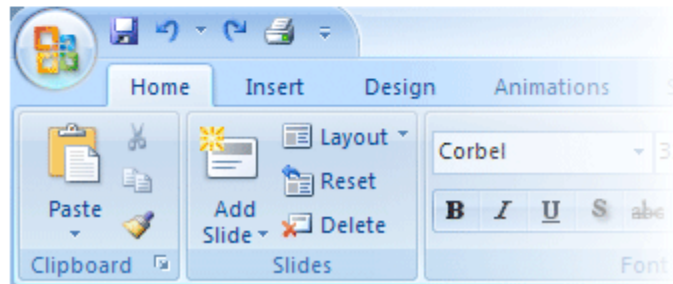
Microsoft Word



Microsoft Excel



Microsoft Powerpoint



2. Universal Usability

- Designing software to be usable by the widest range of reasonably possible users
- Many factors to consider
 - Physical abilities
 - Vision, coordination
 - Cultural differences
 - Colors, icons
 - Technical experience
 - Human perception

Universal Usability

- Anyone can have color vision problems under different circumstances



Tip: Use
Secondary
Encoding!

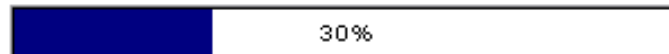
- “Cut.. the blue wire with the white stripe, not... the black wire with the yellow stripe...”

3. Offer Informative Feedback

- Every action should provide feedback that something happened, or is happening
- Controls should have descriptions
 - Describe what is possible or not possible

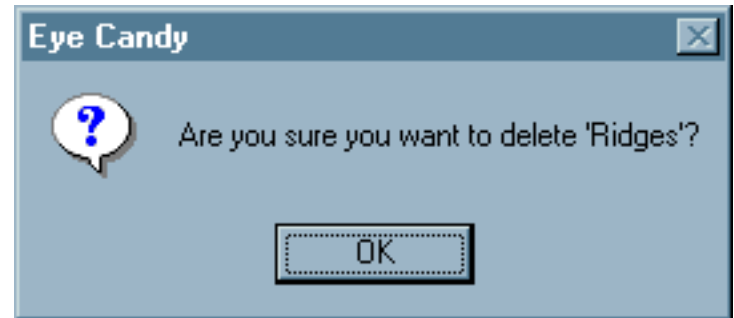
- Examples

- Progress bars
- Status messages
- Cursor changes
- Tooltips
- Animations



4. Design Dialogs to Yield Closure

- Users want to feel confident that a given task has
 - Obvious steps
 - Well-defined outcome
- A dialog box should
 - Appear for a purpose
 - Have logical steps
 - Give a good indication of the success or failure of the task

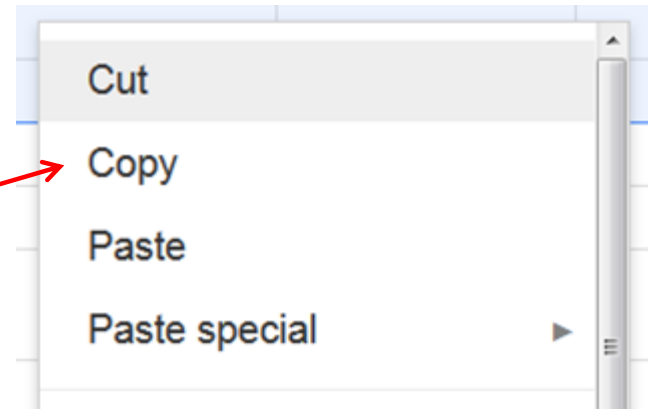


5. Prevent Errors

- Guide towards correct actions
 - Gray out inappropriate actions
 - Selection rather than freestyle typing
 - Automatic completion
 - Input validation
- Error prevention lessens the need for error messages, which are often poorly worded
- Make error messages specific, positive in tone, and constructive

Proper Choices

- Google Docs Context menu



???

Copying and pasting in Google Docs

These actions are unavailable via the Edit menu, but you can still use:

Ctrl+C

for copy

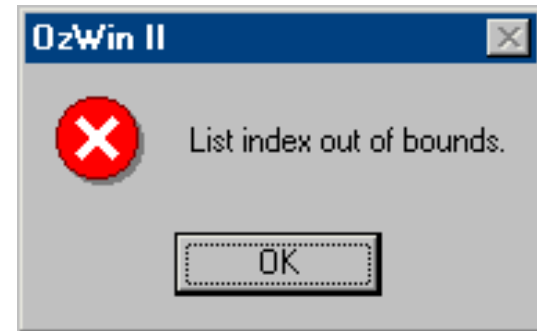
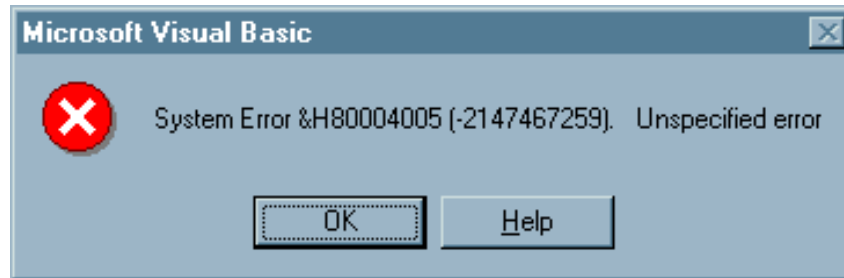
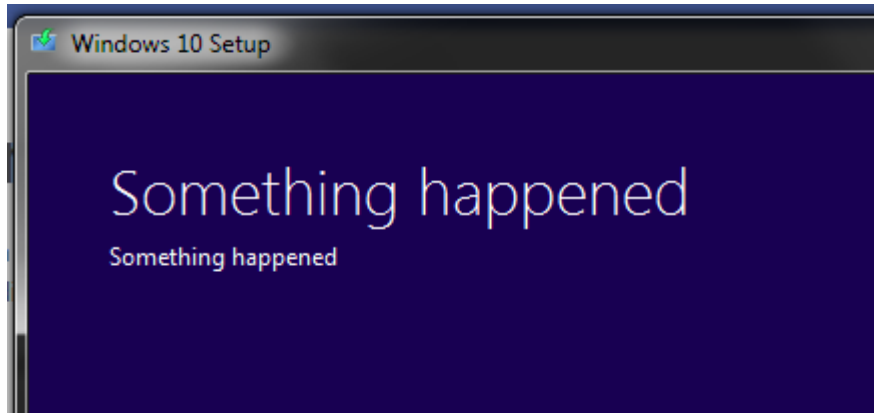
Ctrl+X

for cut

Ctrl+V

for paste

Unhelpful Error Dialogs



6. Provide Reversal of Actions

- Give the user some level of safety net
 - Undo
 - Easy “back to last screen”
 - Restore defaults
- Reduces anxiety, encourages experimentation
- Mechanism for reversal is almost always better than a confirmation dialog

7. Support Internal Locus of Control

- The user believes his/her actions completely control the application's behavior
 - The user is confident that certain actions will always provide certain results
- Providing flexibility for a user
 - Customization, personalization

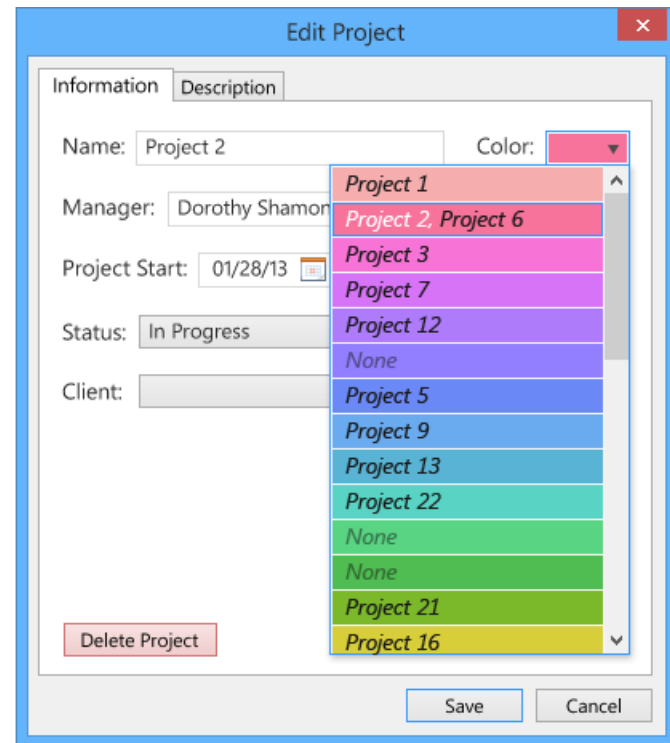
8. Reduce Short Term Memory Load

- Don't make the user have to remember things between steps / screens
- Short term and working memory are highly volatile
 - Disruptions cause loss of memory
 - Task switching, multi-tasking
 - The average person can only remember seven plus or minus two chunks of information at a time

Reduce Short Term Memory Load

From ICS's project.net:

- Current task is picking a color, but provides context of already used choices.



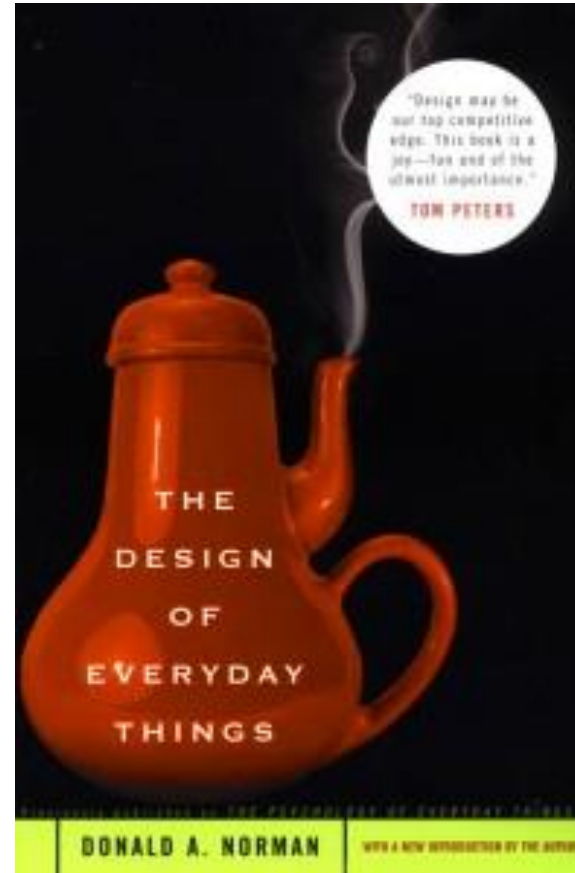
For More Information

Visit the ICS UXD Team web pages for more discussion

- <http://www.ics.com/uxd>
- <http://www.ics.com/blog/category/ux>

What's a Good Book on Design?

- *The Design of Everyday Things*, Donald Norman, 1988



To Recap

- The User Experience can be a competitive advantage
- UX Design uses a different skill set than software engineering
- Remember and apply
 - Know thy user
 - Be consistent
 - Make errors as impossible as possible
 - Design first, code later

Thanks For Attending

