Workshop
Evaluating UX without Users

Tobias Komischke, Ph.D.
Sr. Director of Design
Infragistics, Inc.
Tobias Komischke, Ph.D.

- Senior Director of Design at Infragistics
- Adjunct Professor at Rutgers State University of New Jersey
- Studied the stuff (Master’s & Ph.D.)
- 15+ years experience in research, development, consulting
- Published 40+ technical papers, gave 50+ public talks

"Think back ... which keys did you press?"
Global leader in user experience controls, components, and services

- Windows Forms
- ASP.NET
- WPF
- LightSwitch
- Sharepoint
- jQuery & HTML5
- Report Plus
- Share Plus
- Indigo Studio UX Prototyping
- iOS
- Android
- Windows Phone
- Design & Development Consulting
Workshop Agenda

1. Intro
   - The Process
   - The Challenge
   - Heuristic Evaluation
   25 minutes

2. Product Briefing
   5 minutes

3. Running the Heuristic Evaluation
   45 minutes

4. Debrief on Results
   45 minutes
Intro
The UX Process

- Plan the human-centered design process
- Understand and specify the context of use
- Evaluate designs against requirements
- System satisfies specified user & organisational requirements
- Specify the user and organisational requirements
- Produce design solutions

ISO 9241-210: Ergonomics of human-system interaction -- Part 210: Human-centred design for interactive systems
The Challenge

- Usability Testing is the single-best method for UX evaluation

- But what if you don’t have representative test users available?
  - You can test with colleagues
  - You can model the user-system interaction
  - You can use heuristic evaluation
Heuristic Evaluation

- Developed by Jakob Nielsen in the early 1990s
- Based on an empirical analysis of 249 usability problems
- The heuristics consist of ten design guidelines
- Other heuristics:
  - Shneiderman’s 8 Golden Rules
  - ISO 9241-110: 7 Dialogue Principles
Nielsen’s Heuristics

1. Visibility of system status
2. Match between system and real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, recover from errors
10. Help and documentation

http://www.useit.com/papers/heuristics/heuristic_list.html
1. **Visibility of system status.** The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

<table>
<thead>
<tr>
<th>Password</th>
<th>Feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6 characters or more (be tricky!)</td>
</tr>
<tr>
<td>●</td>
<td>Too short</td>
</tr>
<tr>
<td>••••</td>
<td>Too obvious</td>
</tr>
<tr>
<td>••••••</td>
<td>Weak</td>
</tr>
<tr>
<td>••••••••</td>
<td>Good</td>
</tr>
<tr>
<td>••••••••••</td>
<td>Strong</td>
</tr>
<tr>
<td>••••••••••••••</td>
<td>Very Strong</td>
</tr>
</tbody>
</table>
2. **Match between system and the real world.** The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. It should follow real-world conventions, making information appear in a natural and logical order.
3. **User control and freedom.** Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support for undo and redo.
4. **Consistency and standards.** Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
5. **Error prevention.** Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
6. Recognition rather than recall. Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
7. **Flexibility and efficiency of use.** Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
8. **Aesthetic and minimalist design.** Dialogs should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.
9. Help users recognize, diagnose, and recover from errors. Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
10. **Help and documentation.** Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.
Three Stages For Doing Heuristic Evaluation

1. Briefing session to tell experts what to do
2. Evaluation period of 1-2 hours in which:
   - Each expert works separately
   - Take one pass to get a feel for the product
   - Take a second pass to focus on specific features
3. Debriefing session in which experts work together to aggregate and prioritize problems
Severity Ratings

0 – not a usability problem

1 – cosmetic problem

2 – minor usability problem

3 – major usability problem; important to fix

4 – usability catastrophe; imperative to fix

Example

Heuristic 4 - Consistency [Severity 3]:

The interface used the string "Save" on the first screen for saving the user's file, but used the string "Write file" on the second screen. Users may be confused by this different terminology for the same function.

Nielsen & Mack, 1994
- This is a qualitative method – few evaluators work fine

- Curve of diminishing returns

http://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/
Advantages Of Heuristic Evaluation

- Few practical issues to consider, e.g., how long to run tests with participants?
- Few ethical issues to consider, e.g., need to have women running studies with women in for face-to-face methods because of culture rules separating sexes
- Heuristics have been shown to find a significant number of problems
- Heuristics cover a wide range of the types of problems that can occur

**Biggest advantage**
- Almost always cheaper than usability testing
There is no established way of running a heuristics study

No good definition of what defines an expert. Wide range of differences in amount of problems found by different experts

Heuristics do not cover domain specific issues

Domain specific experts may be required

Sufficient experts may be difficult to find

**Biggest disadvantage**

- Validity – evaluators are NOT the end users
- Important problems may get missed
- Oftentimes, many trivial problems are identified
Product Briefing
www.google.com/maps/place/bulgaria
Running the Heuristic Evaluation
Running The Heuristic Evaluation

- Take one pass to get a feel for the product  
  5 minutes
- Take a second pass to focus on specific features  
  40 minutes

Think about what typical user scenarios could be

Go through these scenarios using Google Maps

Map what you experience to the Heuristics
Debriefing
What issues did you find?

- Demonstrate the issue
- Explain why it’s an issue (positive or negative) – what heuristic(s) are affected?
- What severity rating is it?
- If it’s negative, do you have an idea how to improve on it?
thank you