Usability Heuristics

COSC 480: User-Centered Design
Madeline E. Smith
November 28, 2016

COSC Tea Tomorrow

Contributing to Open Source Projects

Michael Chavinda '17

An essential part of code literacy is the ability to read and understand large code bases and subsequently contribute to them. This sort of experience is difficult to get at undergraduate level. Contributing open source is a great way to improve your programming skills, keep abreast with changes in technology, and connect with a wider community of developers. This talk will offer a condensed guide of how to get started with open source contributions and some personal advice on how to be a productive programmer in general.

Looking Ahead

- A5: Individual Prototype due Mon 11/14
- Thanksgiving Break (no class Mon 11/21 – Fri 11/25)
- T7: User Testing due Wed 11/30 (PR Fri 12/2)
- T8: Hi-Fi Prototype due Mon 12/5 (PR Wed 12/7)
- T9: Final Presentation on Wed 12/7 (PR Fri 12/9)
- A6: Final Reflection due Mon 12/12

T7: User Testing

- Optional: Revise System Concept Statement
- Optional: Revise Site Map
- Optional: Revise Low-Fi Prototype
- Prepare for user testing
- Run user tests
- Organize observations
- List design changes
- Reflection & Contributions

Design Heuristics

Design Heuristics

- Design heuristics are broad rules of thumb
- Heuristic evaluation of user interface design developed by Jakob Nielsen in the early 1990's
- The original set of heuristics was derived from an empirical analysis of 249 usability problems

Nielsen's Heuristics

1. **Visibility of system status**
   The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

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. 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 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.

Nielsen's Heuristics

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.

Nielsen's Heuristics

8. **Aesthetic and minimalist design**
   Dialogues 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.

Wizard of Oz Prototype

Google NYC Trip Report

shoutkey.com/now

submit responses
before class Wednesday

Colgate @ Grace Hopper 2016

NOVEMBER 8, 2016

Six members of the Colgate University computer science department re
participate in the 2016 Grace Hopper Celebration of Women in Computi
Zoila Rodriguez '18, Stephanie Tortora '17, and Bria Vicenti '17 and pro
and Madeline E. Smith were among nearly 15,000 attendees at the con