Principles of GUI Design and Programming

Evaluation & Practice

Contents

- Evaluation
- Heuristics
- Practice of UI design
-

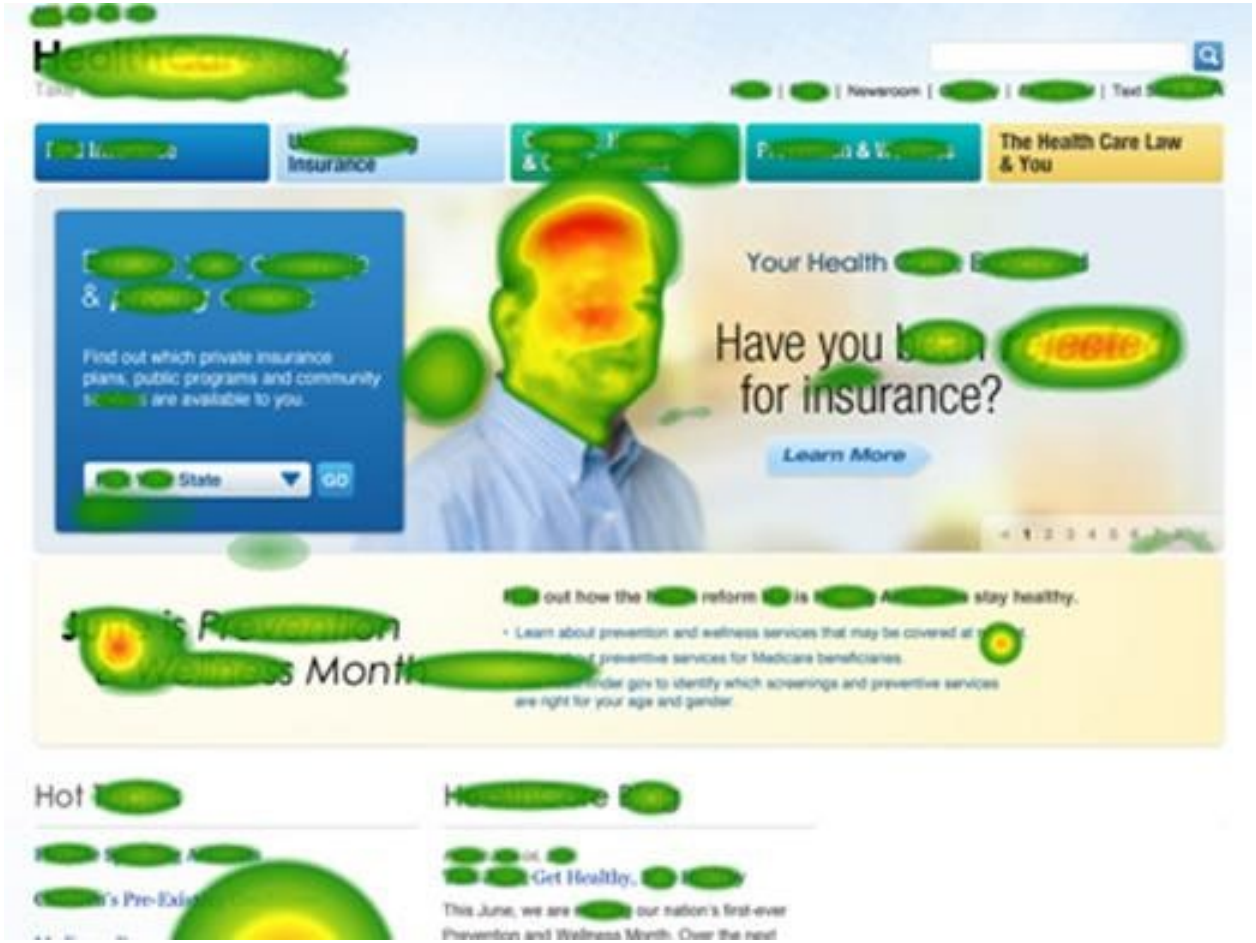
Evaluation

- Once you have built a UI you need to evaluate it
- This can be done
  - In a lab setting
  - In the field with beta testers
- You can measure
  - Time to complete a task
  - Number and type of errors
  - Use of help or manuals
  - Number of users making the same error
  - Number of users completing a task successfully

Eye tracking

- Can be used to
  - See the order in which users read screen
  - Measure how long they spend on parts of the screen
  - Determine what part of the page they missed
  - How they navigate the length of the page
  - How the size and placement of items on the page affects attention

Heat maps from eye tracking



* usability.gov

What???

**"it's the latest innovation in office safety.
When your computer crashes, an air bag is activated
so you won't bang your head in frustration."**

Heuristics

- Sometimes
  - You do not have time for usability testing
  - Users are not available for testing
  - It is too expensive to do usability testing

- An alternative is heuristic evaluation
  - Having experts determine if the interface conforms to a series of heuristics
  - The heuristics are similar to the guidelines used in the design of interfaces

Heuristics

- Visibility of system status
  - The system must keep the user informed
- Match between system and real-world
  - Right language, familiar terminology, natural and logical order
- Users control and freedom
  - Users can select order to do operations
  - Is forgiving by supporting undo and redo
- Consistency and standards
  - Similar operations are done similarly
  - Follows platform conventions

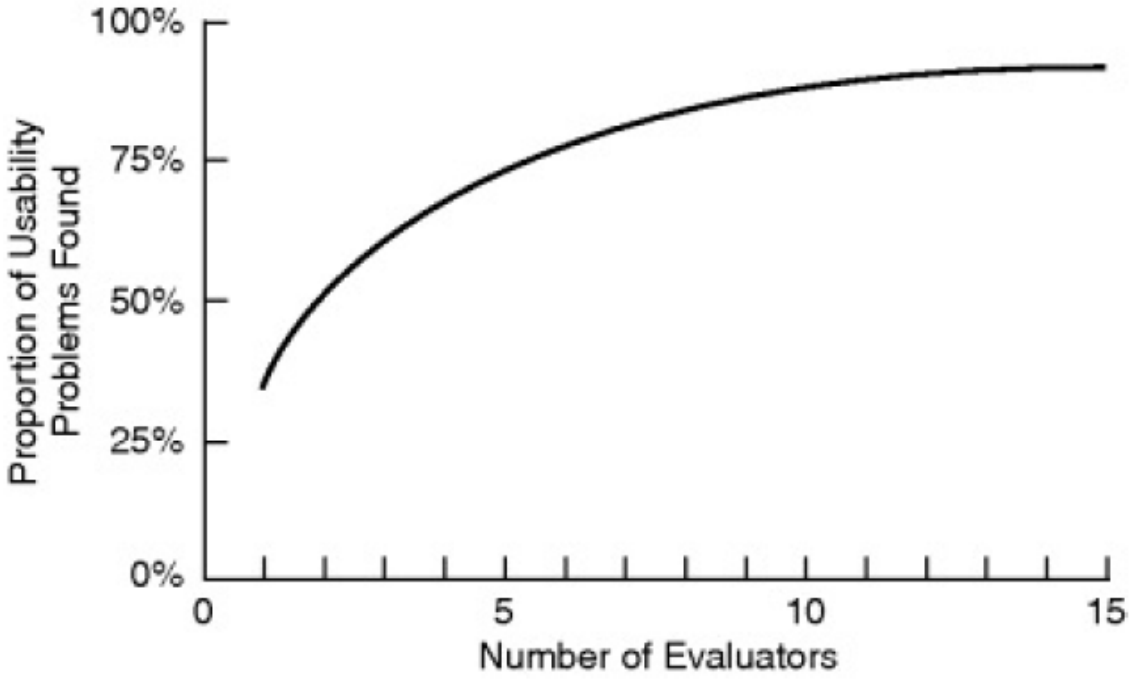
Heuristics

- Error prevention
  - Prevents the users from making errors rather than reporting errors that happened
- Recognition rather than recall
  - List of command to be selected
  - Remembers things for the user
  - Can copy things from one part of the interface to another
- Flexibility & efficiency
  - Has accelerators for advanced users
  - Laid out so as to maximize efficiency
  - Supports frequent and occasional users
  - Users can tailor frequent actions

Heuristics

- Aesthetics & minimalist design
  - Good looking interface
  - No irrelevant information or graphics
- Recognize and recover from errors
  - Meaningful error messages
  - Suggest how to fix the problem
- Help and documentation
  - Provides context sensitive help
  - Documentation is up to date and correct
  - Written in language the user can understand

Heuristics



More evaluators will find more problems with the interface

Cognitive walkthroughs

- Simulate a user's cognitive processes at every step of the human-computer dialogue
  - You identify the types of users you will have
  - Evaluators walk through the interface as each type of user, performing various tasks
  - At each step they ask
    - Is the correct action obvious
    - Does the user understand the next step to perform
    - Is all the information necessary to make a decision available
- The result is the identification of problems which need to be fixed.

Agile UX

- In agile software development
  - The requirements are not finalized until later in the project
  - Design and implementation begin before requirements are complete
  - Requirements can change mid-way through the project
- This creates problems for the UI designers since they do not know what to design at the start of the project.

Agile UX

- Create cross-disciplinary teams
  - UI designers
  - Developers
- Work together to
  - Understand the users and their needs
  - The technical capabilities of the system
- Decide
  - What user research to conduct and when
  - How to align UI design and development
  - What documentation to produce and when

Agile UX

"Enough storyboarding. Let's shoot something."

User research

- Agile work is done in short sprints
  - Conduct interviews to
    - Clarify requirements
    - Evaluate part of the design
- One way to deal with this is to conduct the research before the development starts so you have answers to most of the questions and just need to keep up to date as the development progresses.

Iterative design

- Since Agile development is iterative you can
  - Break UI development into a series of steps
  - Each step will go through a design-implement-evaluate cycle
  - Each part of the interface will be iteratively developed until it is felt to be complete
- Since the UI development is broken into parts
  - Different teams can develop parts in parallel
- Documentation is minimized since Agile development favours writing software over documentation