Exploration: GenderMag Heuristics

Introduction: What are Software Usability Heuristics?

Software usability heuristics are guidelines for making software user interface (UI) designs more usable. Software designers go through heuristics one-by-one when evaluating the usability of a UI. Software usability heuristics can also guide the creation of a new UI.

An advantage of software usability heuristics is they are an inexpensive way to make a UI more usable: they do not require testing with actual users.

What are the GenderMag Heuristics?

One set of software usability heuristics is the GenderMag Heuristics. Margaret Burnett, a leader in usability research, published this set of 9 "heuristics for avoiding gender-inclusiveness 'bugs' in software" in 2019.

The GenderMag Heuristics are used to find and fix gender bias bugs in software. They are based on research that has found that the ways people problem-solve in software tends to cluster by gender. However, many software features are inadvertently designed to work best for problem-solving styles favored mainly by men.

At the core of the GenderMag Heuristics are five *cognitive problem-solving facets*: a user's *motivations* for using software, their *information processing style*, their *computer self-efficacy*, their *tude towards risk*, and their *style of learning* new technology. The GenderMag Heuristics are guided by the GenderMag *personas*: The Abi persona represents how women tend to problem-solve in software, and the Tim persona represents how men tend to problem-solve in software. Abi and Tim are detailed in the images below.

Note: The blue text is customizable; the blue text shown is only an example. In addition, Abi and Tim may be any gender.

Abi (Abigail/Abishek)

- 28 Years Old
- Employed as an Accountant
- Lives in Cardiff, Wales

Abi has always liked music. When she is on her way to work in the morning, she listens to music that spans a wide variety of styles. But when she arrives at work, she turns it off, and begins her day by *scanning all her emails first to get an overall picture before answering any of them.* (This extra pass takes time but seems worth it.) Some nights she exercises or stretches, and sometimes she likes to play computer puzzle games like Sudoku

Background and Skills

Abi works as an accountant. She is comfortable with the technologies she uses regularly,but she just moved to this employer 1 week ago,and their software systems are new to her.

Abi says she's a "numbers person", but she has never taken any computer programming or IT systems classes. She likes Math and knows how to think with numbers. She writes and edits spreadsheet formulas in her work.

In her free time, she also enjoys working with numbers and logic. she especially likes working out puzzles and puzzle games, either on paper or on the computer.

Motivations and Attitudes

- *Motivations*: Abi uses technologies to accomplish her tasks. She learns new technologies if and when she needs to, but prefers to use methods she is already familiar and comfortable with, to keep her focus on the tasks she cares about.

- *Computer Self-Efficacy*: Abi has lower self confidence than her peers about doing unfamiliar computing tasks. If problems arise with her technology, she often blames herself for these problems. This affects whether and how she will persevere with a task if technology problems have arisen.

- *Attitude toward Risk*: Abi's life is a little complicated and she rarely has spare time. So she is risk averse about using unfamiliar technologies that might need her to spend extra time on them, even if the new features might be relevant. She instead performs tasks using familiar features, because they're more predictable about what she will get from them and how much time they will take.

Attitude to Technology

- *Information Processing Style*: Abi tends towards a comprehensive information processing style when she needs to gather more information. So, instead of acting upon the first option that seems promising, she gathers information comprehensively to try to form a complete understanding of the problem before trying to solve it. Thus, her style is "burst-y"; first she reads a lot, then she acts on it in a batch of activity.

- *Learning: by Process vs. by Tinkering* : When learning new technology, Abi leans toward process-oriented learning, e.g., tutorials, step-by-step processes, wizards, online how-to videos, etc. She doesn't particularly like learning by tinkering with software (i.e., just trying out new features or commands to see what they do), but when she does tinker, it has positive effects on her understanding of the software.

[1]Abi represents users with motivations/attitudes and information/learning styles similar to hers. For data on people similar to and different from Abi, see http://gendermag.org/Foundations.html

Tim (Timothy/Timara)

- 28 Years Old
- Employed as an Accountant
- Lives in Cardiff, Wales

Tim loves public transportation. He knows several routes to get there from home and he's always exploring ways to optimize his trips into the office. *Work starts with email, Which he answers one at a time, as soon as he reads them.* (Sometimes this backfires, if there is a second related message he hasn't read yet, but he doesn't mind sending a follow-up email.) Some nights he plays computer games with his online friends.

Background and Skills

Tim works as an accountant. He just moved to this employer 1 week ago,and their software systems are new to him. For Tim, technology is a source of fun, and he is always on the lookout for new computer software. He likes to make sure he has the latest version of all the software with all the new features.

Tim says he's a "numbers person" , but he has not taken any computer programming or IT classes. Tim likes Math and knows how to think in terms of numbers. He writes and edits spreadsheet formulas for his work.

He plays the latest video games, has the newest smart phone and hybrid car. He downloads and installs the latest software, and experiments with its settings. He is comfortable and confident with technology and he enjoys learning about it and using new technologies.

Motivations and Attitudes

- *Motivations*: Tim likes learning all the available functionality on all of his devices and computer systerms he uses, even when it may not be necessary to help him achieve his tasks. he sometimes finds himself exploring functions of one of his gadgets for so long that he loses sight of what he wanted to do with it to begin with.

- *Computer Self-Efficacy*: Tim has high confidence in his abilities with technology, and thinks he's better than the average person at learning about new features.If he can't fix the problem, he blames it on the software vendor it's not his fault if he can't get it to work.

- *Attitude toward Risk*: Tim doesn't mind talking risks using features of technology. that haven't been proven to work. When he is presented with challenges because he has tried a new way that doesn't work, it doesn't changes his attitudes toward technology.

Attitude to Technology

- *Information Processing Style*: Tim leans towards a selective information processing style or "depth first" approach. That is, he usually delves into the first promising option, pursues it, and if it doesn't work out he backs out and gathers a bit more information until he sees another option to try. Thus, his style is very incremental.

- *Learning: by Process vs. by Tinkering* : Whenever Tim uses new technology, he tries to construct his own understanding of how the software works internally. He likes tinkering and exploring the menu items and functions of the software in order to build that understanding. Sometimes he plays with features too much, losing focus on what he set out to do originally, but this helps him gain better understanding of the software.

[1]Tim represents users with motivations/attitudes and information/learning styles similar to his. For data on people similar to and different from Tim, see http://gendermag.org/Foundations.html

▶ summarize, the GenderMag Heuristics are a set of principles based on the GenderMag personas and cognitive problem-solving facets. They allow developers to quickly identify parts of a UI that potentially have gender bias bugs. Keep reading below to learn about each of the heuristics.

Heuristic #1 (of 9): Explain what <u>new</u> features do, and why they are useful

Abi uses software only as needed for their task. They prefer familiar features to keep focused on the task and may be wary of new features.

Tim likes using software to learn what new features can help them accomplish.

Allow Abi to quickly assess new features so they can choose whether to start using them. Allow Tim to quickly assess what a feature is for so they can move on to finding out what the rest of the software's features do.

▶ Heuristic #2 (of 9): Explain what <u>existing</u> features do, and why they are useful

Abi is risk-averse and so may avoid using features that have an unknown time cost and an unknown benefit.

Tim is risk-tolerant so may use features without knowing their cost or even what they do. Tim is also motivated to investigate new, cutting-edge features.

Allow Abi to determine whether existing features are familiar to them. Allow Tim to more efficiently determine which features are known and which are new and unique.

Heuristic #3 (of 9): Let people gather as much information as they want, and no more than they want

▶

Abi gathers and reads relevant information comprehensively before acting.

Tim likes to delve into the first option and pursue it, backtracking if need be.

Allow Abi to find the information they want, but don't force them to spend excessive time or effort gathering that information. Allow Tim to find correct information immediately, so that they don't go down the wrong path or get distracted from their task.

► Heuristic #4 (of 9): Keep familiar features available

Abi has low computer self-efficacy, so often takes the blame if a problem arises in the software, such as features having changed.

Tim has high computer self-efficacy, so often blames the software if a problem arises, such as features having changed.

To encourage Abi and Tim to continue using the software, allow them to interact with it in ways they expect.

▶ Heuristic #5 (of 9): Make undo/redo and backtracking available

Abi is risk-averse, so prefers not to take actions in software that might not be easy to reverse.

Tim is risk-tolerant, so is willing to take actions in software that might be incorrect and need to be reversed.

Provide undo/redo and backtracking to allow Abi to feel comfortable proceeding, and to allow Tim to recover from mistakes.

Heuristic #6 (of 9): Provide ways to try out different approaches

▶

Abi has low computer self-efficacy, so often takes the blame if a problem arises in the software. This can discourage Abi from persevering in the software.

Tim has high computer self-efficacy, so often blames the software if a problem arises but is willing to try numerous ways of addressing the problem.

Allow Abi to try a different approach when they feel unable to proceed with the current one. Allow Tim to try multiple ways of solving a problem.

Heuristic #7 (of 9): Communicate the amount of effort that will be required to use a feature

Abi is risk-averse, so may want to avoid features with high effort costs.

Tim is risk-tolerant, so may begin using features that require extra effort and time, and that are unrelated to the task at hand.

Allow Abi to decide whether or not a feature will require too much effort to use. Allow Tim to understand that a feature may take extra effort, and thus more time, to help them stay on track with their task.

▶ euristic #8 (of 9): Provide a path through the task

Abi is a process-oriented learner, so prefers to proceed through tasks step-by-step.

Tim learns by tinkering, so prefers not to be constrained by rigid, pre-determined processes.

Provide Abi a way to go through tasks using a clear process. Provide Tim a way to bypass step-by-step processes and tutorials if those are not required for learning the software.

▶ Heuristic #9 (of 9): Encourage mindful tinkering

Abi is a process-oriented learner so usually prefers not to tinker. Because of this, they might miss useful or important parts of the software.

Tim learns by tinkering, but sometimes tinkers addictively and gets distracted from their task.

Encourage Abi to tinker in ways that lead to them discovering task-relevant functionality. Encourage Tim not to over-tinker (e.g., by adding an extra click), so that they make fewer mistakes, have time to absorb important information, and stay on-task.

▶ Wrap Up

The GenderMag Heuristics are a set a 9 software usability heuristics for evaluating and improving the usability of UIs for different genders of users.

Quiz

What approach is Tim most likely to use when learning new software?

○ Click around the software to figure out how it works

○ Call a friend who has used the software before

○ Search YouTube for a instructional tutorial

✓ Check   ＞

↻ Reuse   <> Embed   H-P

References

Zoe Steine-Hanson, Claudia Hilderbrand, Lara Letaw, Jillian Emard, Christopher Perdriau, Christopher Mendez, Margaret Burnett, and Anita Sarma. 2019. *Fixing Inclusivity Bugs for Information Processing Styles and Learning Styles*. arXiv:1905.02813 [cs.HC].

▶