

# CSE 291 I: Usability of Programming Languages ("Programmers Are People Too")

Michael Coblenz

# About Me

- Spent eight years as a software engineer at Apple
- But: no one knows how to design programming languages that are known to benefit people!
- Can we develop a science of PL design?
- Completed a PhD on PL usability at Carnegie Mellon (designed a new smart contract language; showed it benefits people)
- Postdoc at UMD

# Learning Goals

- After successfully completing this course you will be able to:
  - Apply qualitative and quantitative research methods to obtain insights about programming language design choices;
  - Critically analyze design questions in the context of languages and programming systems;
  - Read and interpret research papers in the area of usability of programming languages, and summarize major findings to date.

# Course Content

- Three categories of material
  - Methods
  - Results
  - Interesting languages
- Three categories of work
  - Reading assignments (w/ reading responses due)
  - Project: a small language study
  - Programming assignments (two)

# Grading

Assignment	Weight	Due Date
Project	35%	11/30
Programming	10%	
Reading responses	25%	Before lecture
Qualitative study assignment	10%	10/12
Methods Exam	20%	10/28 in class

# TA: Dylan Lukes

- PhD student in Programming Languages and Software Engineering

# Schedule

- No class next Monday
- Instead, reading is due Wednesday
- My office hours: Mondays 4-5 PM, CSE 3246
- Dylan's office hours: Thursdays 2-4 PM, CSE B215

# Resources

- Come to office hours!
- *Research Methods in Human-Computer Interaction* online and in library

# Collaboration

- Collaboration is encouraged! But *write your own answers*.
- Whiteboard discussion policy: discuss at a whiteboard, erase it, and write your own answers

# Technology Policy

- Screens are distracting to others [1] and decrease learning outcomes
- Laptops, tablets, and phones may not be used in class except for emergencies and for disability accommodations

[1] <https://blueprintcdn.com/wp-content/uploads/2016/05/Blueprint-Research-2016-Carter-Greenberg-and-Walker.pdf>

# TODAY

- What is usability?
- How might I measure usability?
- (Next time: more ways of assessing usability)

# WHAT IS USABILITY?

- "The extent to which a product can be used by *specified users* to achieve *specified goals* with *effectiveness, efficiency* and *satisfaction* in a *specified context of use*." (ISO 9241-11, Ergonomics of human-system interaction)

# USABILITY



# USABILITY

- Who are the users?
- What are the users trying to do?
- In what context are they doing it?
- How well do they do it?
- How much work does it take them to do it?
- Do they like it?

# WHO ARE THE USERS?

- Scope your target audience
  - Education
  - Experience
  - Knowledge
  - Skills
  - Work context

# PERSONAS

- Help you think like the user
- Build empathy

# CHARLIE



Charlie is in her late 20s to mid 30s. She has a Bachelor's degree but not necessarily in IT. She's a self-taught developer. Her coding is unconventional and she mixes genius lines with simple errors. She seeks to reinvent her software development career but the how is still unclear. Charlie has a family, which makes financial stability and work-life balance essential. She's new to the industry and thus looks for a company that offers a supportive, people-oriented environment, where she can learn and improve her skills.

# ROBIN



Robin is in his mid 20s and has completed his formal education, such as a Bachelor's degree in computer science. He is probably on his second or third job but has reached the ceiling in his current job, as in, he has learnt a lot and gained experience but would be keen on taking the next step to further his career. Even though he probably hasn't taken any steps to find a new job (applied), he is on the lookout for something challenging as well as purposeful. In his current role, he can be found working in a specialized programming area (front-end/back-end/mobile). On a personal level, he is probably in a relationship, he is also quite introverted and self-aware. He enjoys working on complicated tasks and really wants to be involved and feel a part of the company. He values transparency and is happy working with inspiring leaders. He's keen to know what is going on and where the company is headed. Salary isn't his top priority (as long as it is not too far below average). Instead, Robin appreciates non-financial rewards, especially those that make him feel valued for his work.

# MICROSOFT PERSONAS (CLARKE)

## **THE SYSTEMATIC DEVELOPER**

Writes code defensively. Does everything they can to protect their code from unstable and untrustworthy processes running in parallel with their code.

Develops a deep understanding of a technology before using it.

Prides themselves on building elegant solutions.

## **THE PRAGMATIC DEVELOPER**

Writes code methodically.

Develops a sufficient understanding of a technology to enable them to use it. Prides themselves on building robust applications.

## **THE OPPORTUNISTIC DEVELOPER**

Writes code in an exploratory fashion.

Develops a sufficient understanding of a technology to understand how it can solve a business problem.

Prides themselves on solving business problems.

# GOALS

- What does the user want to achieve?
- Start vague, and then drill down
- What does a C programmer want to achieve?
  - Fix a bug — what kind?
  - Implement something — what?
  - Understand code — for what purpose?

# CONTEXT

- Starting from scratch, maintaining a system, or legacy code?
- Large team, or lone developer?
- Beginning of project, or near shipping (risk-averseness)?
- One-off, or repeated task?

# HOW WELL DO THEY DO IT?

- Count and describe bugs/errors
- How readable is their code?
  - What does "readable" mean, anyway?
- What fraction of target population can do it?

# HOW HARD WAS IT?

- What fraction of participants succeeded?
- How long does it take?
- What obstacles did they encounter along the way?
- Other measures?
  - Galvanic skin response (emotional response)
  - EEG (cognitive load)

# DO THEY LIKE IT?

- Of course they do, if it's YOUR system!
- "We find that respondents are about 2.5x more likely to prefer a technological artifact they believe to be developed by the interviewer, even when the alternative is identical." [Dell et al., CHI 2012]
- "When the interviewer is a foreign researcher requiring a translator, the bias towards the interviewer's artifact increases to 5x."

# SO WHAT IF THEY LIKE IT?

- Maybe there are factors you didn't measure
- Adoption
- "Why not?"
- But maybe you don't care!

# VARIETIES OF USABILITY

- My "more usable" system might be better because:
- Learnability: it's easier/faster to learn
- Task performance: people finish tasks faster or more people finish tasks
- Audience: a new kind of person can do the task
- (Not an exhaustive list)

# YOUR TURN

- Identify a usability question YOU have about a COMMON PL.
  - With a partner.
  - Share afterward.

*"The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use."*

# RESEARCH QUESTIONS AND METHODS

- How can I understand this situation better? (*ethnography, contextual inquiry*)
- I think P is a problem. What fraction of people in a population want it fixed? (*survey*)
- I designed a tool. What challenges do people face when doing task X with my tool? (*usability study*)
- Can most people use my tool successfully to do task T? (*usability study + quantitative analysis*)
- Is my tool better than an existing one for task T? (*quantitative study: randomized controlled trial*)

# ARGUMENTS

- Which do you find convincing?
- What form *should* evidence take?

# READING ASSIGNMENT

- The article introduces the "language wars"
- Read it *critically*
- I don't agree with everything in there
- But it's key contextual information for the rest of the course

# NEXT TIME (REMINDER: NO CLASS MONDAY)

- Discuss the assigned paper
- How to run a usability study