

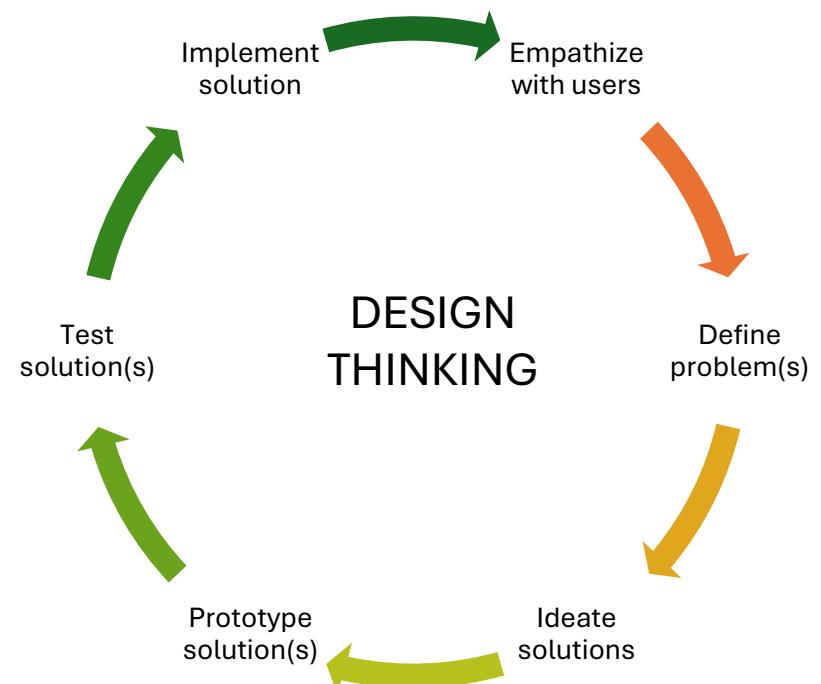
Lecture 13 – Evaluating prototypes

Logistics: Final grading scheme

- In-class assignments: 10%
- Quizzes: 15%
 - 2 more coming your way – one more coming up!
- Homework assignments: ~~20%~~ **10%**
- Mid-semester exam: 25%
- Final project: ~~30%~~ **40%**--the continuing assignment
 - Will include 4 milestones in total + a final presentation/report.

Last week

- Process of developing ML/AI systems still not as mature as developing software
- Closer to product design
 - Ill-defined
 - Similar to when computers were present, people were looking for automation opportunities.
 - Hard to estimate, visualize, validate



Also last week..

- How to:
 - Empathize with users
 - Define problems
 - Ideate solutions
 - Prototype solution(s)

Today: Evaluating prototypes

- Two kinds:
 - We don't know if we have built appropriate, useful things
 - Users know → so we ask users
 - Validation with users
 - We have gotten requirements right, but want to ensure usability
 - Ease of use, convenience, right flow at a micro-level, etc.
 - Usability tests

How do we validate with users?

- Build a prototype
- Show users
- Get feedback from them
 - Here is the system's flow
 - Is this correct, would this be useful, what are we missing, etc.
- How many users?
 - Usually, 1-5 (more if there are a lot of users).
 - Sometimes, users are just 1 or 2, in that case, them + their bosses / whoever substitutes for them when they are away.

What to do?

- Explain the purpose of the prototype
- Show how it works
- Tell the story in the users' language
 - Based on what you have “emphathized” with them
 - Usually, the same people that participate in the early “empathize” phase.
- Instead of:
 - ✖ Here is a web app that does predictive clustering of diseases and here are their labels and datapoints along with location.
 - ✓ You are a public health official, you log in and can immediately see what kinds of diseases are breaking out, and then you can click to see further details about where they are [ideally, this latter part be a map].

Additional things to do (or not)

- Do not defend yourself
- Listen to questions the users ask
 - If something that is obvious to you is not obvious to them, it is a usability issue, it needs fixing, not defending!
- If they need a feature / a specific vocabulary, give it to them – it is part of their domain/function/process
 - Technology is an enabler, people don't always need to change processes to fit into technology
- Do not interrupt
- Do not throw impossible data in your prototype – keep it as close to real as possible (can lead to confusion otherwise)

Usability

- The quality of being easy to learn, use and understand
- How effectively, efficiently, and satisfactorily can users achieve their goals?
- Partly done with preliminary prototypes
- Partly done on final product
- Series of changes → iterative → major to minor

How do we evaluate usability?

- Ideally, also with users
- Give the prototype to users
- Have them use it
- Observe them use it
- Every time they are stuck, have questions, are confused, seek help, do something surprising / inefficient, repetitive, etc.
- Ask: what were they expecting that I am unable to satisfy, why would they not do X instead of Y
 - Learnability is a common issue for non-tech savvy users.

Hard to always observe

- Have them use and periodically collect feedback
 - Go meet users, and ask how it is going
 - Elicit feedback, encouraging them to say what is not working
 - [Otherwise, people tend to be nice to you].
- Alternatives:
 - Telemetry, recording search/help pages, etc.
 - Ask to keep a diary of what worked / didn't work.
 - Ask to post on a spreadsheet, whatsapp group, etc.

A lot of issues can be caught beforehand

- Think like the user (be in their foot) – part of ability to empathize
- If you don't know what they would do – ask.
- Method – cognitive walkthrough
- Methods – heuristic evaluation
- Later..

In-class activity

- Go back to the prototype for the Pingala course suggestion
- Show it to someone who wasn't part of your team
- Ask for feedback on it
- Observe and make notes
- Homework – show it two students not part of the course
- Ask for their feedback on the prototype
- Validate usefulness and test usability
- Make notes, and turn in a list of usability issues