# Designing for the Public

SECT.02.2026

- Introduce User & Human Centered Design
- Discuss Project Milestone 1

# Design

Our focus today will be on *product design*, the development of a new solution to a problem.

Visual, interface, and system design will be important components of a finished product-- but not our primary focus for today.

We'll touch on *system design* later in this course.

# Design Philosophies

*How can we begin to approach designing a new product?*

## Tech-centric

This is what we *can* do, often what a new/emerging technology allows.

## User/Human Centered

Start with needs & understanding of *needs*, search for possible solutions from there.

## System/Organization Based

What is our existing structure designed to accomodate?

## Conway’s Law

*"Organizations which design systems (in the broad sense used here) are constrained to produce designs which are copies of the communication structures of these organizations."*

—Melvin E. Conway, How Do Committees Invent?

Left-unchecked, results in an example of *organization/system-centered design*. The designed solution will mirror the team structure.

https://martinfowler.com/bliki/ConwaysLaw.html

## User-centered Design

*"User-centered design (UCD) is an iterative design process in which designers and other stakeholders focus on the users and their needs in each phase of the design process. UCD calls for involving users throughout the design process via a variety of research and design techniques so as to create highly usable and accessible products for them."*

— Definition of user-centered design (UCD) by the Interaction Design Foundation via https://usabilitygeek.com/user-centered-design-introduction/

## Human-centered Design

*"We propose a radical change in design from experts designing for people to people designing for themselves. In the traditional approach, experts study, design, and implement solutions for the people of the world. Instead, we propose that we leverage the creativity within the communities of the world to solve their own problems: This is community-driven design, taking full advantage of the fact that it is the people in communities who best understand their problems and the impediments and affordances that impede and support change."*

https://jnd.org/community-based-human-centered-design/

# Civic Tech 🤍 Human Centered Design

## aka *"Build With, Not For"*

https://codeforamerica.org/news/what-does-it-mean-to-build-with-not-for/

A response to issues with & critiques of the early "tech for good" movement. Lots of well meaning technologists "air-dropping" solutions on organizations & communities.

- often solve wrong problem
- *"we've tried this before"*
- not equipped to make use of them (e.g. technology mismatch)

The **User Centered Design process** is built on user engagement at each phase, and benefits from rapid **iteration**.

Instead of a lengthy requirements-gathering phase that takes place in the days/weeks/months prior to beginning development, we'll re-evaluate repeatedly and respond to feedback.

Takes letting go of what you think you're going to build, and being guided by what you learn along the way.

# 1. Define "Context of Use"

*"How, when, and why will this tool be used? Who is using it?"*

Beyond just *audience*.

Identify the people who will use the product, how and why they will use it.

What aspects of our users might be relevant?

- **Demographics**: technically literate? common disabilities?
- **Occupation**: can take advantage of existing knowledge?
- **Relationship to Application**: volunteers or trained professionals?
- **Environment**: distractions? obstacles to adoption?

# 2. Gather Requirements

*What would success look like?*

## Tools

- Surveys
- Focus Groups
- Community Forums
- Ideation - build user stories

# 3. Design Solutions

*What could possibly work?*

## In first iterations:

- Paper Prototyping
- Figma
- HTML Wireframes

Don’t worry *much* about constraints at this phase. "If we had $1,000,000 we could X" often leads to good conversation & ideas.

**Low-fidelity** designs are easier for users to reject (a good thing!) and easier for us to accept rejection.

## In later iterations:

Build actual software, focusing on lessons learned from last iteration.

# 4. Evaluate

*How is this working so far?*

## In first iterations:

- Talk through mock-ups/paper prototypes with users.
- What's missing?

## In later iterations:

- Usability Testing
- User Feedback Sessions
- Recorded Interactions

## Post-launch:

- Continued user feedback sessions/testing
- Analytics

# Usability Testing

**Devs watching QA test the product**

Tim Mackey

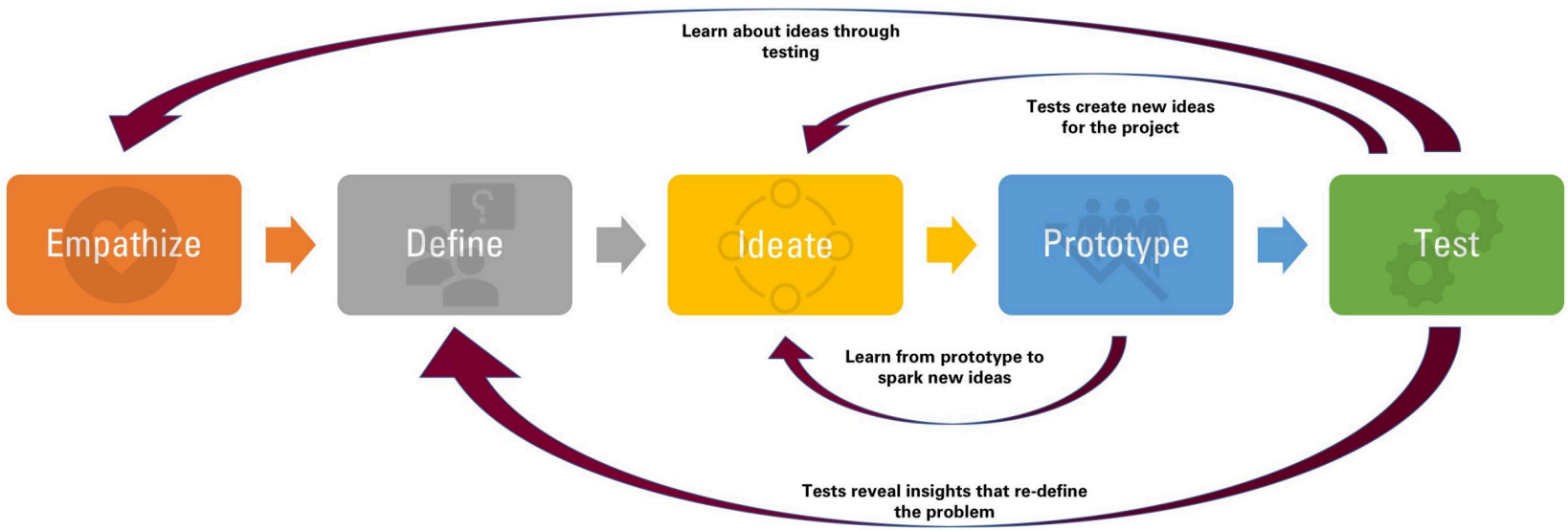
Watch on

# 5. Iterate

*"done or back to one"*

This process no longer ends when software is "released"-- release early & often.

# HUMAN-CENTERED DESIGN PROCESS




McMaster University | ENGINEERING

# Step 0: Empathize

Take time at the start to understand the problems of the community of interest.

- Wider range of stakeholders/needs evaluated: *how will this affect members of the community that are **not** using the application*?
- Ensure active participation of communities of interest.

https://digital.gov/guides/hcd/introduction/

# Distilled Principles

- Obtain diverse perspectives representing communities of interest.
- Design based on an understanding of the environment it will be used in.
- Users involved *throughout the whole process*, don’t assume you know what they will want or do.
- Process must be **iterative**.

# UCD/HCD Design Tools

- User Interviews
- Personas/Scenarios/Use Cases
- Mockups/Prototypes
- User Testing/Feedback

# User Interviews

*Define Audience & Gather Requirements*

## Ideal

Go to users where they are, observational interviews, flesh out personas.

## Compromise

Use friends, neighbors, colleagues. (Be sure to think about diverse backgrounds and needs!)

# Personas, Scenarios, Use Cases

*Further Define Audience & Refine Requirements*

Helps us identify key features, systems, necessary data, etc.

## Persona

*Who is this person? What unique challenges might exist?*

Burt is a 36 year old man with a college education. He works from home as a graphic designer. As the primary caregiver to an 8 year-old free time is a precious commodity. He feels strongly about education and environmental issues.

## Scenario

*For each persona, one or more scenarios as to why/how they would use the product.*

Burt would like to participate in community meetings, but finds it nearly impossible since they are always Wednesdays at 7pm. He is excited when the community organization announces they are going to allow submitting video feedback via the new NeighborApp.

## Use Case

*A story about how the user interacts with the application, from start to solution.*

The local commissioner has put out a request on NeighborApp for feedback on how the park department budget surplus could be used.

Burt receives a notification and is prompted to submit video testimony. He is given the opportunity to record a four minute video.

Burt's 2 minute video is submitted and reviewed by committee staff. During the user feedback portion of the live meeting, four proposals, including Burt's are shown to the live attendees.

# Iterative Design

*Early drafts benefit from speed over perfection.*

Typically not functional, paper prototypes, wireframes, or rapidly developed prototypes where interaction is important.

**Don’t spend too much time on prototypes!**

- Users might hate it and then you have the conundrum of throwing out your work or ignoring users. (And we rarely make the right choice faced with this dilemma!)
- Users are also more likely to give positive feedback to something that looks polished, even if it doesn’t fit their needs. *"I’d hate to make that nice woman throw out all her hard work, it seems like **someone** might like it."*

# Bazaar Model

Many of these ideas also surface in the "bazaar model", a *software development philosophy* that is hugely infuential in the Open Source community.

- "Every good work of software starts by scratching a developer’s personal itch.
- "Plan to throw one version away; you will, anyhow."
- "Release early. Release often. And listen to your customers."
- "Treating your users as co-developers is your least-hassle route to rapid code improvement and effective debugging."

# User Testing & Feedback

Evaluate, hopefully with real users.

Like interviews, in person usually best. Lots of nuance.

Screen recordings, analytics, etc. tell part of the story, but no substitute for talking to users "why did you do this?" Ensure testers align with personas. (or introduce new personas if needed!)

It can be painful, but you need to see them dismantle the printer in front of you.

# Milestone 1: Design

**Problem Statement** - Start with a problem that you'd like to solve. You may have ideas in mind on how to solve it, but hold those back for now.

**Audience Analysis** - Research & describe likely audience for your application and the need you are attempting to meet.

**Personas** - Write a few detailed personas representing different segments of your audience.

# Milestone 1: Design

https://github.com/uchicago-capp-30320/sect-2026/blob/main/milestones/m1.md

**Scenarios** - For each persona, write at least one (but consider more!) scenario where the person would engage with your application including their goals in doing so.

**Use Cases** - For each scenario, write at least one use case including the interaction steps a user would take, and how the application would solve the needs presented in their scenario.

**Wireframes** - Make a prototype suitable for walking some users through your ideas. You can use digital tools if you wish, but pen & paper is fine!

# Milestone 1: User Feedback Session

Sometime between Week 2 and 3 you will also need to show your prototype to a set of "users" and get their feedback.

**This can be submitted with Milestone 2, but earlier is better!**

- Actual Users (time permitting)
- Another Group/Peers
- Course Staff (book a meeting!)

# Next Time

*Design Exercise*

Please be here on time! (participation point #2)