Building User Interfaces

Course Introduction

Professor Bilge Mutlu

About Me

— Associate professor of computer science, psychology, and industrial engineering

— Background that bridges design and computer science

— Director of the *HCI Lab* and co-director of the *Collaborative Robotics Lab*

About Me[1]

— Associate professor of computer science, psychology, and industrial engineering

— Background that bridges design and computer science

— Director of the *HCI Lab* and co-director of the *Collaborative Robotics Lab*

[1] Image source

To Learn More

— Five-part video series on YouTube

— ~~HCI Lab~~ (People and Robots Lab) website

Instructional Team

TAs

Cole Nelson, Derek Manning, John Balis

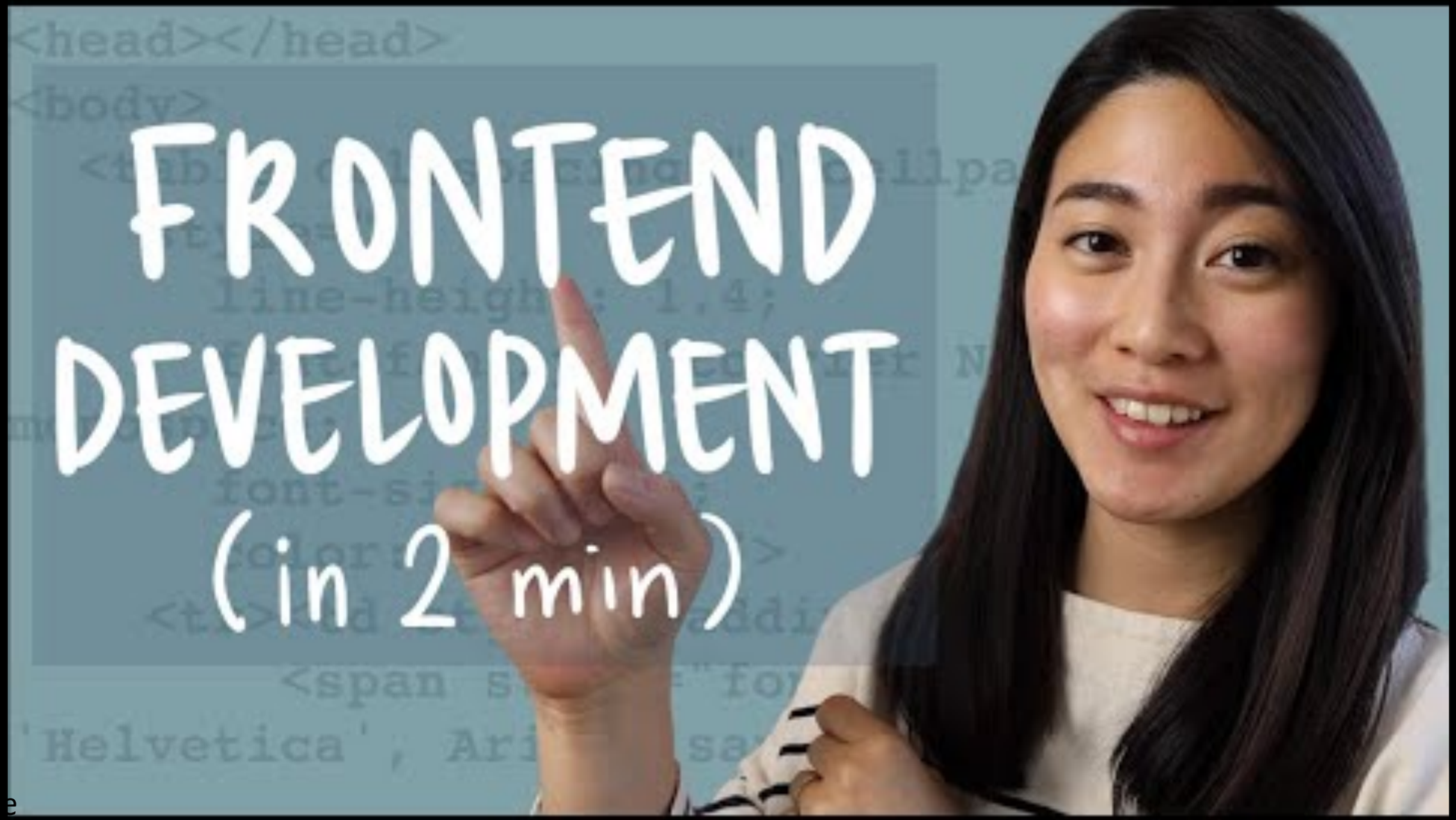
Peer Mentors

Sanjana Rao, Vera Wei

What is this course about?

UX Development

But, what is *UX development*?


FRONTEND
DEVELOPMENT
(in 2 min)

The realm of the UX developer exists somewhere between that of the traditional developer and the designer. We're not really designers, yet to be a good UX developer you certainly need to have an eye for design. In the same vein, we're not traditional developers but we certainly need to have development experience and expertise. Often this experience spans multiple technologies, languages, and platforms.

It falls on the UX developer to bridge the gap between design and technology. We need to be able to think and speak the language of designers. It's our job to help translate their vision to the development team in a way that they can understand and accept. This can be a critical piece of the puzzle in a project, especially if the design and the interactions behind it are complex.

Similarly, we need to speak on behalf the developers to help reign in the designers, at times. If they are coming up with concepts that will be extremely difficult or time consuming to implement, we can explain the limitations of the technology and the complexity involved in implementing their designs, and try to come up with an acceptable alternative.

— Tim R. Todish

UX Development ➜ software engineering + UX design.

What does a software engineer do?


which is devoured in a
large quantity.

Definition: A software engineer is a person who applies the *principles* of software engineering to the design, development, maintenance, testing, and evaluation of computer software.[5]

What are the principles of software engineering? Can you name one?

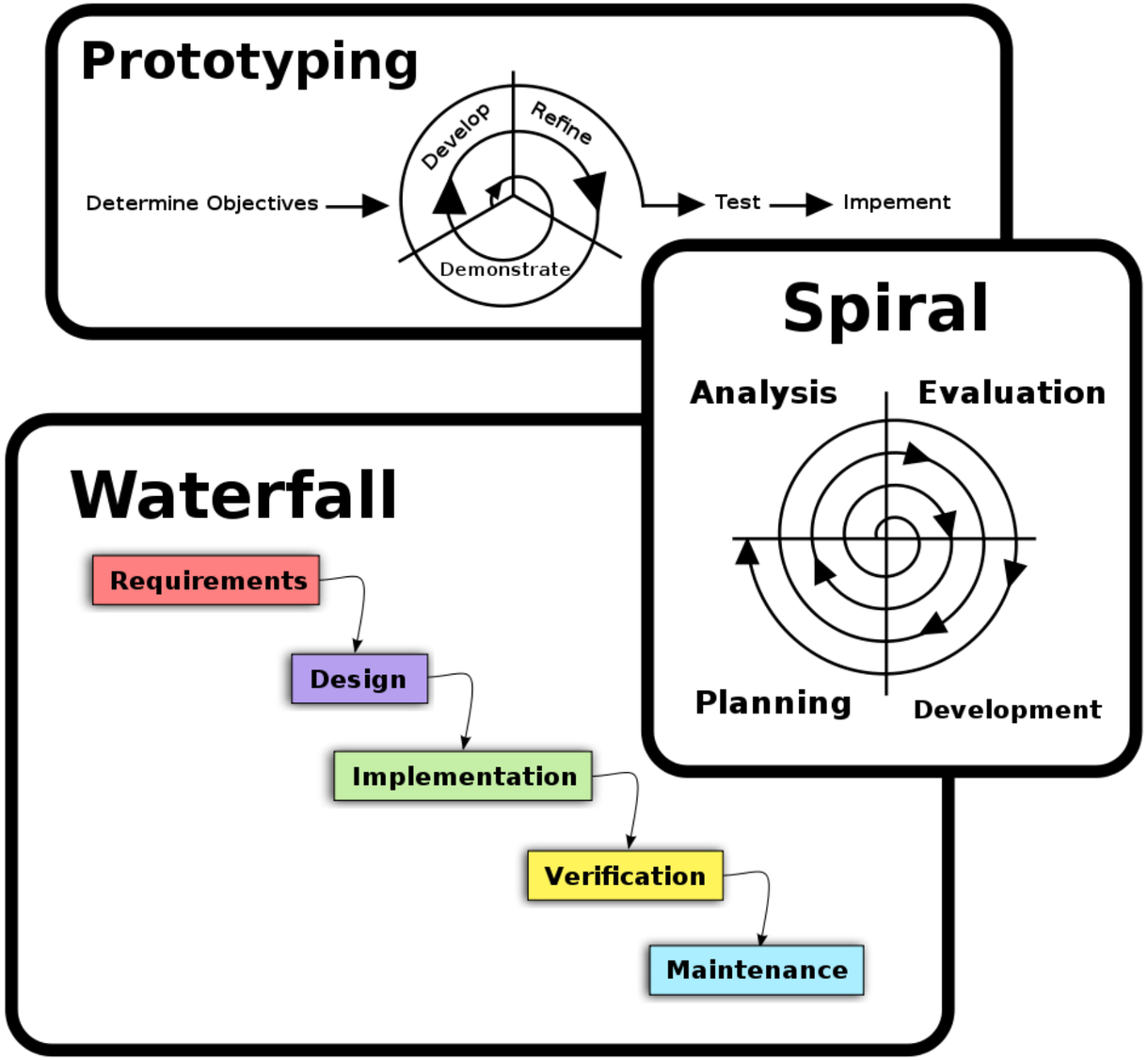
[5] Wikipedia: Software engineer

Pro Tip: Principles of software engineering include:

1. Separation of concerns
2. Modularity
3. Abstraction
4. Anticipation of change
5. Generality
6. Incremental development
7. Consistency

What *process* do software engineers follow? Can you name a step?

Software development process[6]



[6] Wikipedia: Software development process

What does a UX designer do?

Definitions: User experience (UX) design is the process design teams use to create products that provide meaningful and relevant experiences to users.

A *UX designer* is concerned with the entire process of acquiring and integrating a product, including aspects of branding, design, usability and function.[8]

What does this *process* involve? Can anyone name a step?

[8] Interaction Design Foundation

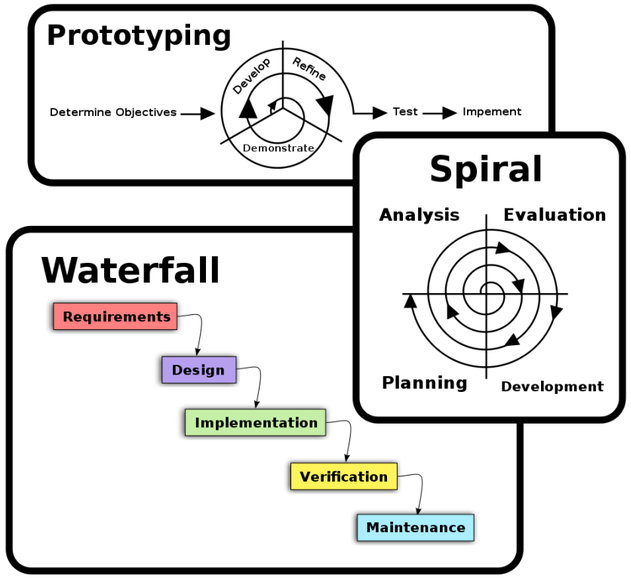
Pro Tip: UX design usually involves the steps:[9]

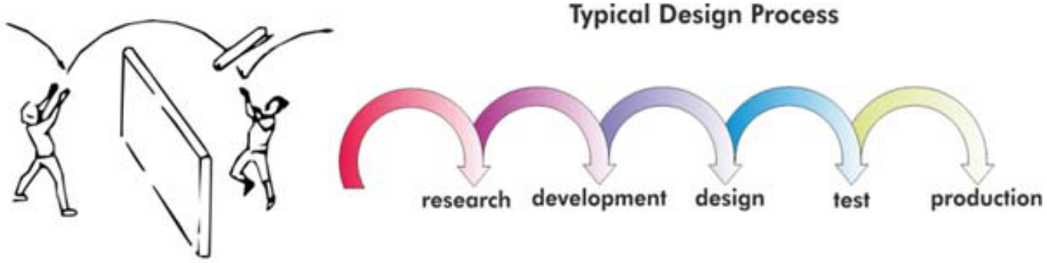
1. Empathize
2. Define
3. Ideate
4. Prototype
5. Test
6. Implement

Wait a minute! Some of these steps sound familiar!

[9] NN/g Design Thinking

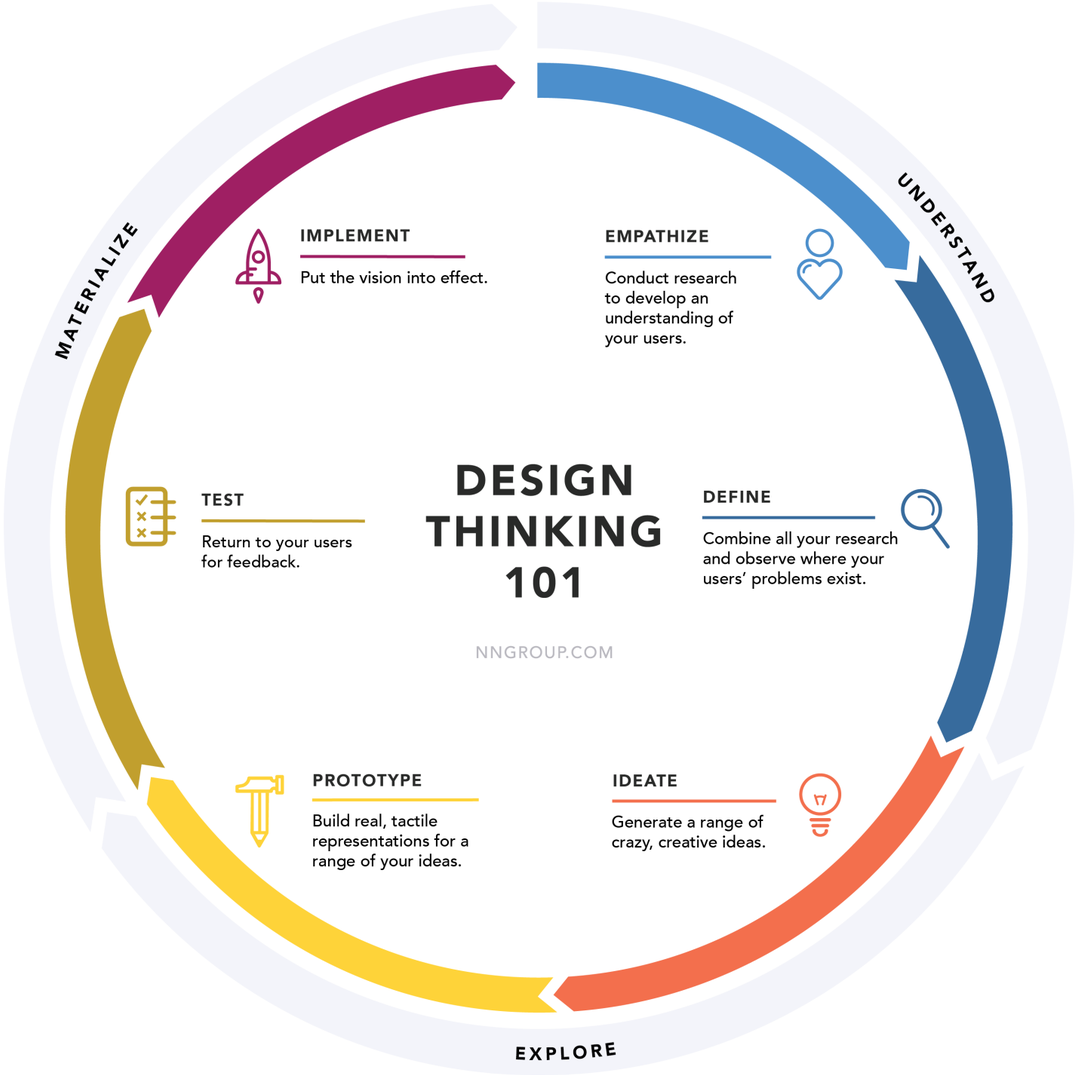
Design process sources[10]





[10] Reed & Bohemia, 2011, NN/g Design Thinking

A Process for UX Development



UX Development Trajectories

— One-person development team to build full-stack applications

— A developer who speaks the language of the designers

— A designer who can also build native prototypes

— A bridge/translator between designers and developers in large/complex organizations

[11] Videos: 1, 2, 3, 4

Ok, 👍, but what can I do as a UX developer?

Example 1: Clocks[12]

— Binary clock

— World clock

[12] React Examples

Example 2: Musical Instruments[13]

— Xylophone

— Electric guitar

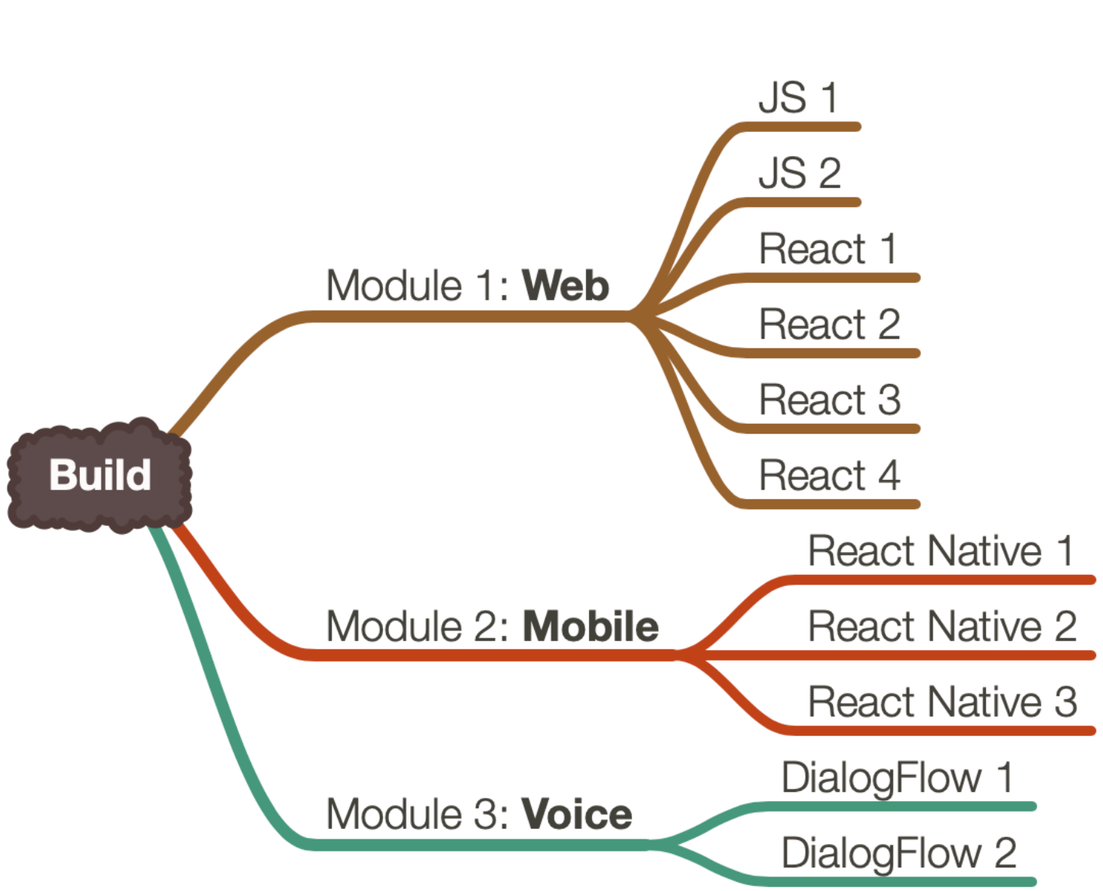
[13] CSS Tricks: Introduction to Web Audio API

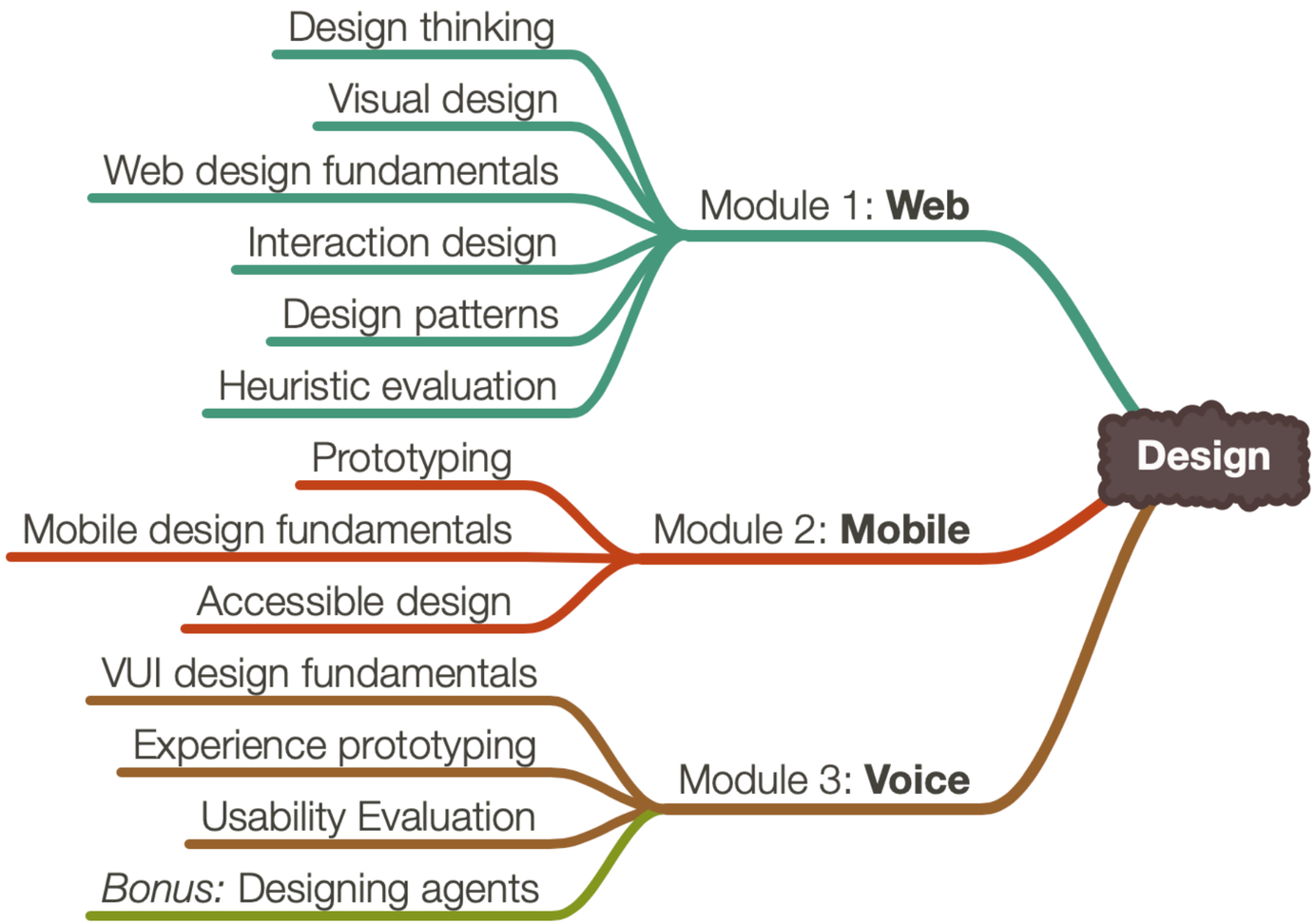
Questions about the Course Topic?

Course Mechanics and Logistics

14

[14] Image sources: left, right


Build
Module 1: Web
JS 1
JS 2
React 1
React 2
React 3
React 4
Module 2: Mobile
React Native 1
React Native 2
React Native 3
Module 3: Voice
DialogFlow 1
DialogFlow 2


Design
Module 1: Web
Design thinking
Visual design
Web design fundamentals
Interaction design
Design patterns
Heuristic evaluation
Module 2: Mobile
Prototyping
Mobile design fundamentals
Accessible design
Module 3: Voice
VUI design fundamentals
Experience prototyping
Usability Evaluation
Bonus: Designing agents

Build

We will learn the most popular implementation frameworks/libraries for each platform:

1. **Web** — *JavaScript, React*
2. **Mobile** — *React Native*
3. **Voice** — *DialogFlow*

Mostly introductory, but these will get you started.

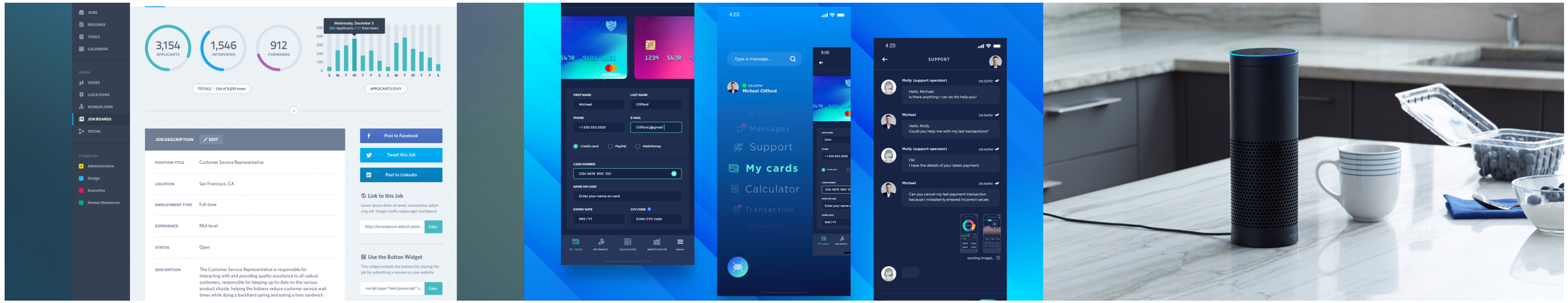
Design

We will learn design methods and techniques that will give you the best bang for the buck. Examples:

1. **Week 3:** How to empathize with your users using a *think-aloud*
2. **Week 7:** How to use existing design patterns to improve UX
3. **Week 10:** How to improve the accessibility of your designs

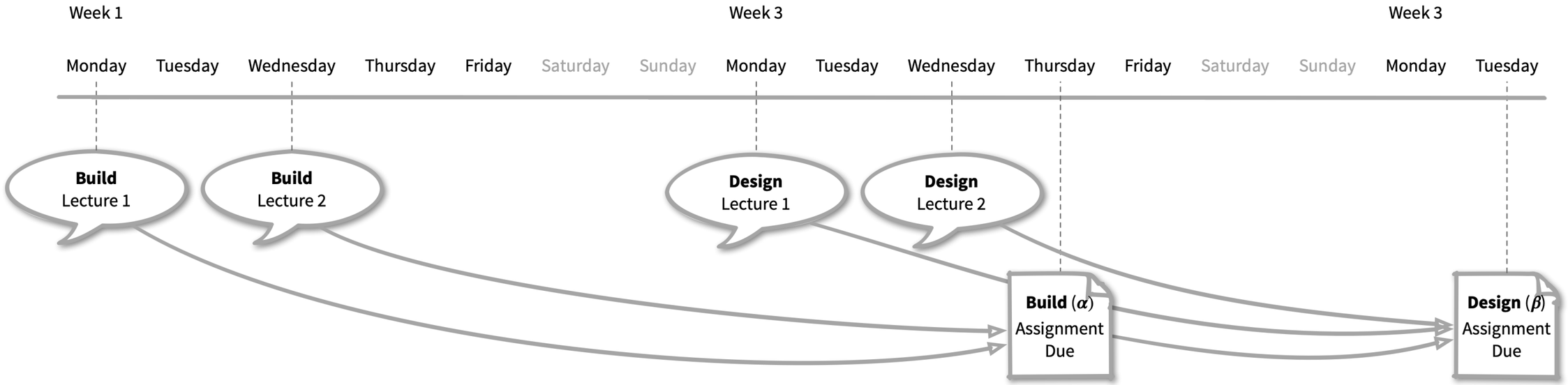
How they are put together

Across three modules—web, mobile, voice—you will work on individual projects where you will incrementally and iteratively design and build user interfaces.[15]



[15] *Image sources:* Left, center, right

The Anatomy of a Module



Participation

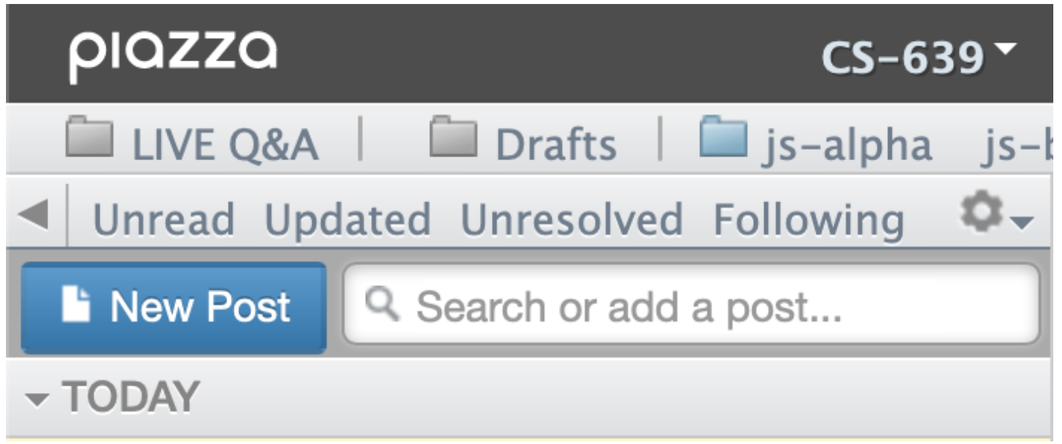
Three formats:

1. **In classroom:** MW 4-5:15 pm, Humanities 3650
2. **Online, sync:** through lecture live stream
3. **Online, async:** through recorded lectures (posted by 9 pm on day of lecture)[16]

[16] Those participating online, async will be asked to confirm their attendance format later this week.

Communication

- **Q&A during class:** Live Q&A on Piazza
- **Office hours:** Microsoft Teams video
- **Assignment questions:** Piazza
- **Personal questions:** Microsoft Teams chat



Assessment

1. Incremental module deliverables — α, β, γ builds
2. Two midterms (dates TBD, done by Thanksgiving)

   Alternative exam dates for synchronous/asynchronous
3. In-class quizzes:

   Synchronous: has to be completed in class

   Asynchronous: has to be completed within 12 hours

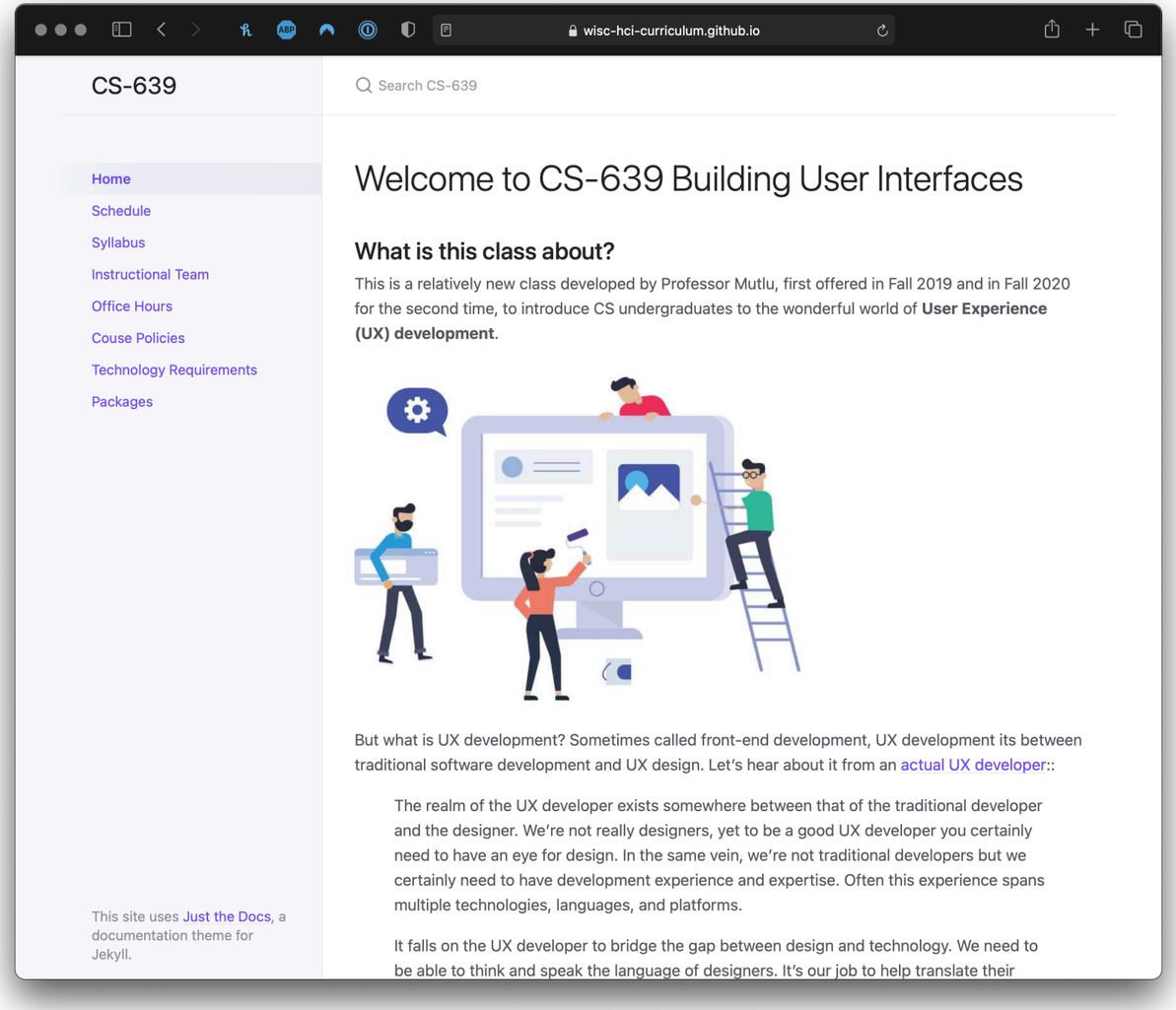
Grading

Assignment	Points
Weekly assignments	50
Midterm 1	20
Midterm 2	20
Quizzes	10
Total	**100**

Systems we will use

1. **Course website** for content (syllabus, lectures)
2. **GitHub Classrooms** + **Canvas** to share/ submit assignments
3. **Piazza** for Q&A
4. **Microsoft Teams** for office hours, personal questions[17]
5. **Canvas** for quizzes
6. **Zoom** for exam proctoring

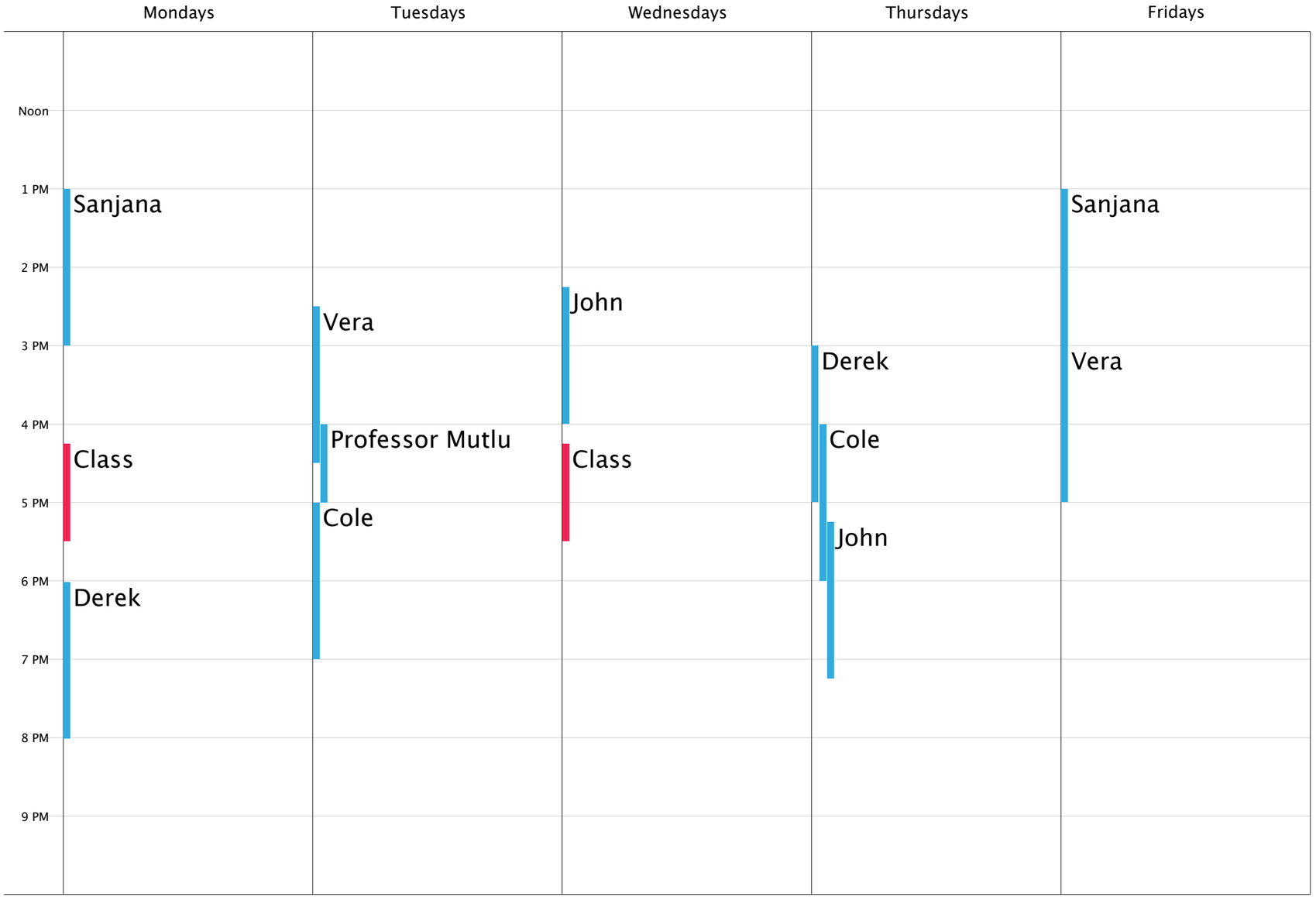
[17] Microsoft Teams Join Code: **b9ir4tq**. Click *Create and join teams* below your teams list and look for the *Join a team with a code* card.



Office hours[18]

You will have a lot of help through *clinics* offered through Microsoft Teams.

[18] Image source


Mondays
Tuesdays
Wednesdays
Thursdays
Fridays
Noon
1 PM
2 PM
3 PM
4 PM
5 PM
6 PM
7 PM
8 PM
9 PM
Sanjana
Class
Derek
Vera
Professor Mutlu
Cole
John
Class
Derek
Cole
John
Sanjana
Vera

Who to talk to about what?

— Grading questions; technical questions; assignment submission issues ➜ *John, Cole, Derek*

— Programming questions ➜ *John, Cole, Derek, Sanjana, Vera*

— Design questions ➜ *Vera*

— Personal questions; exceptions & emergencies; career, grad school, portfolio advice ➜ *Professor Mutlu*

Questions about Course Mechanics?

What's next?

— We'll hit the ground running on Monday with the first build lecture: *Javascript 1: An Introduction*

  Monday is Labor Day — recorded lecture will be posted Monday at 4 pm, watch + complete quiz by Wednesday 4 pm

— Review the course website and meet technology requirements

— Brush up on some basics: HTML + CSS + Git/GitHub