

# **IN4MATX 133: User Interface Software**

**Lecture 18:**  
**Wrap-Up**

Professor Daniel A. Epstein  
TA Eunkyung Jo  
TA Lucas de Melo Silva

# Today's goals

**By the end of today, you should be able to...**

- Describe how Jo and Lucas concepts from IN4MATX 133 in their research and practice
- Summarize what you learned in IN4MATX 133
- Describe the relevance of the topics to different disciplines in industry
- Fill out the course evaluation!

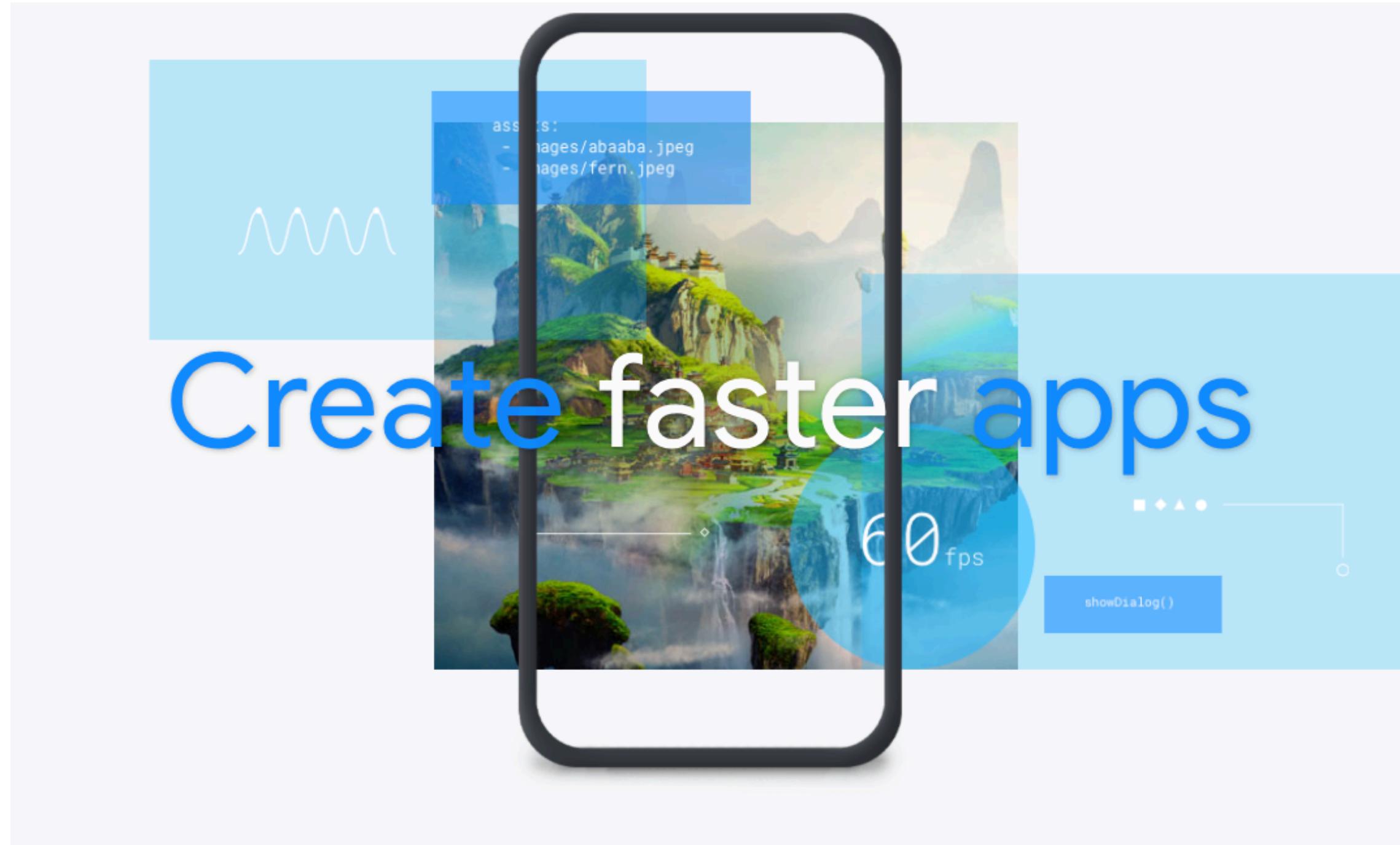
# **Putting 133 concepts to use**

- How do Jo and Lucas use 133 concepts in their research and practice?

# Reflecting on 133

## Technology changes quickly

# Technology changes quickly



Made by [Google](#)

Flutter is Google's UI toolkit for building beautiful, natively compiled applications for [mobile](#), [web](#), and [desktop](#) from a single codebase.

[Get started](#)

[Watch video](#)

<https://flutter.dev/>

# **Take away messages from the course**

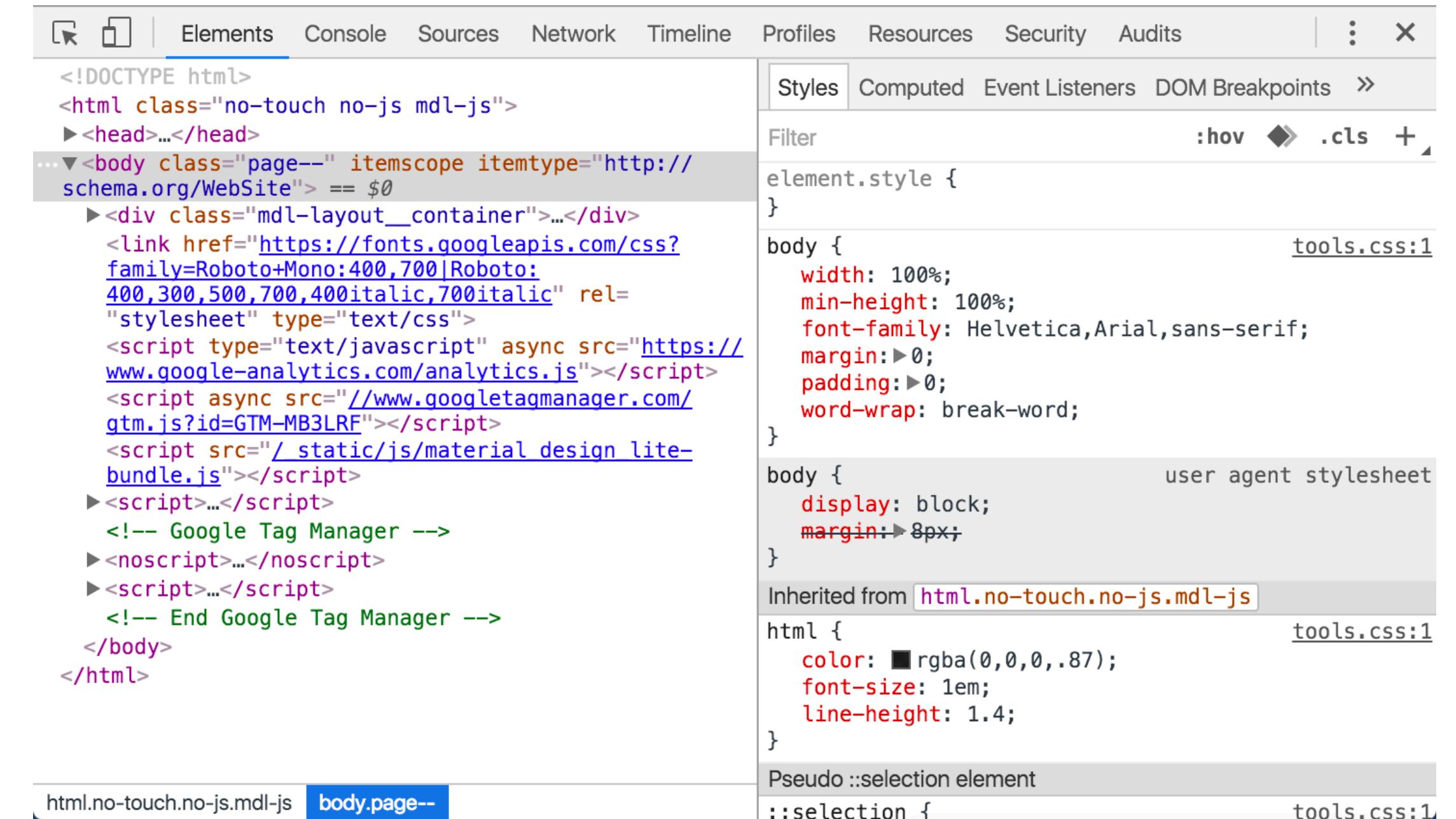
# Search before you build

- Do not reinvent the wheel!
- Use interfaces, algorithms, animations, etc. that have been created by other people



# Build by example

- Learn from others
- Read source code on webpages, GitHub, StackOverflow
- Use the element inspector in your browser to see someone's design or implementation



The screenshot shows the 'Elements' tab of a browser's developer tools. A specific body element is selected in the DOM tree, highlighted with a grey background. The right panel displays the CSS styles for this element, organized into several sections:

- Styles**: Shows the direct styles applied to the selected element, such as `element.style { }`.
- Computed**: Shows the final computed styles after all CSS rules are applied.
- Event Listeners**: Shows any event listeners attached to the element.
- DOM Breakpoints**: Allows setting breakpoints in the DOM tree.

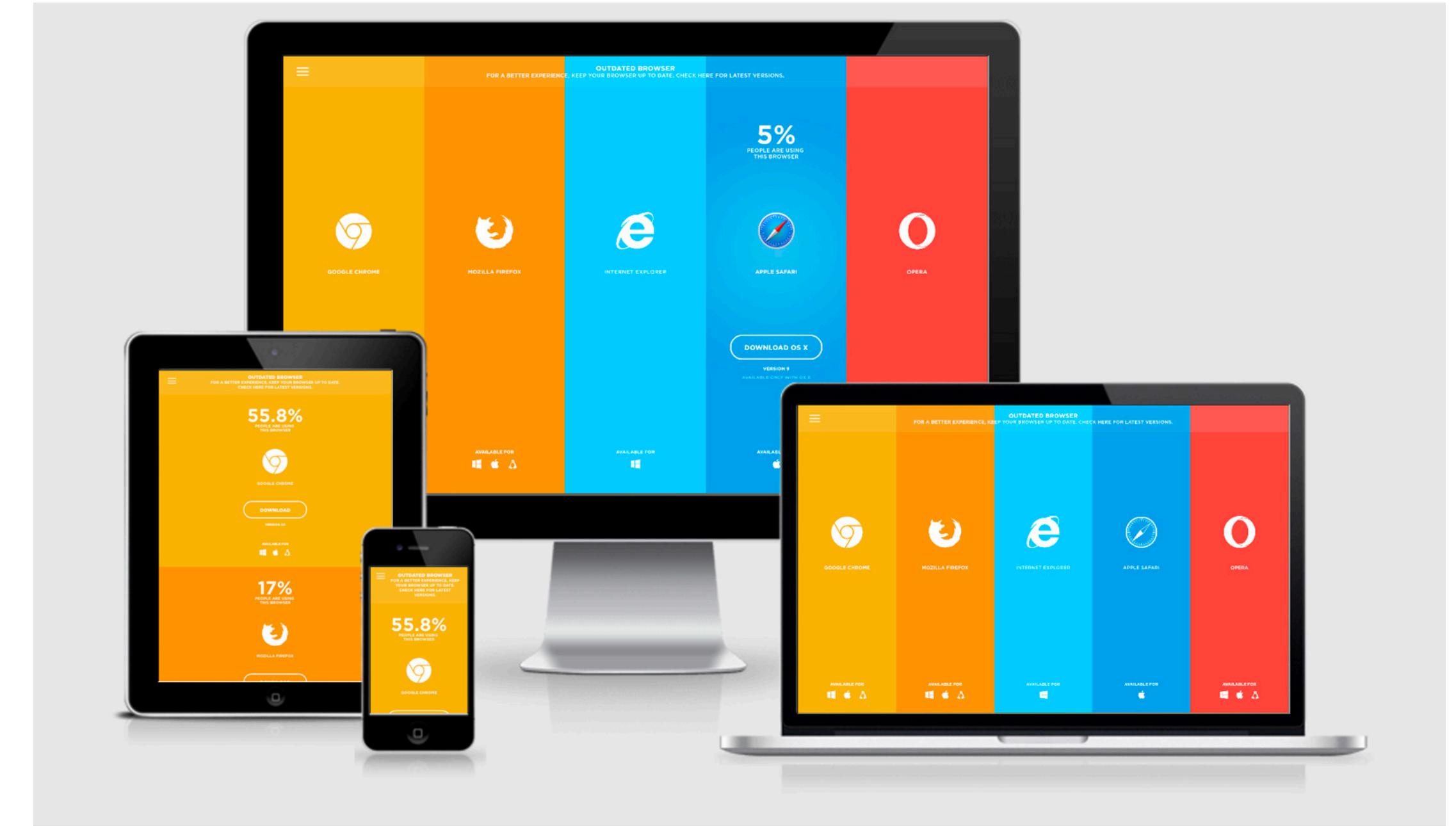
The main pane shows the following HTML structure:

```
<!DOCTYPE html>
<html class="no-touch no-js mdl-js">
  <head>...</head>
  ...<body class="page-- itemscope itemtype="http://schema.org/WebSite"> == $0
    <div class="mdl-layout__container">...</div>
    <link href="https://fonts.googleapis.com/css?family=Roboto+Mono:400,700|Roboto:400,300,500,700,400italic,700italic" rel="stylesheet" type="text/css">
    <script type="text/javascript" async src="https://www.google-analytics.com/analytics.js"></script>
    <script async src="//www.googletagmanager.com/gtm.js?id=GTM-MB3LRF"></script>
    <script src="/static/js/material_design_lite_bundle.js"></script>
    <script>...</script>
    <!-- Google Tag Manager -->
    <noscript>...</noscript>
    <script>...</script>
    <!-- End Google Tag Manager -->
  </body>
</html>
```

Below the styles section, there are additional panels for **Inherited from** (showing styles from the `html.no-touch.no-js.mdl-js` selector) and **Pseudo ::selection element** (showing styles for the `::selection` pseudo-class).

# Build for accessibility

- Keep in mind who you are designing for!
- Make sure your app works for:
  - All users
  - All browsers
  - All devices



# Build with caution

- Use version control!
- Test while you build
- Iteratively refine and debug



**GitHub**

# Build on a solid foundation

- A new framework will come out next year
  - Or next month or next week
- But some fundamental principles unite them all
  - Separating interface from data and interaction, for example

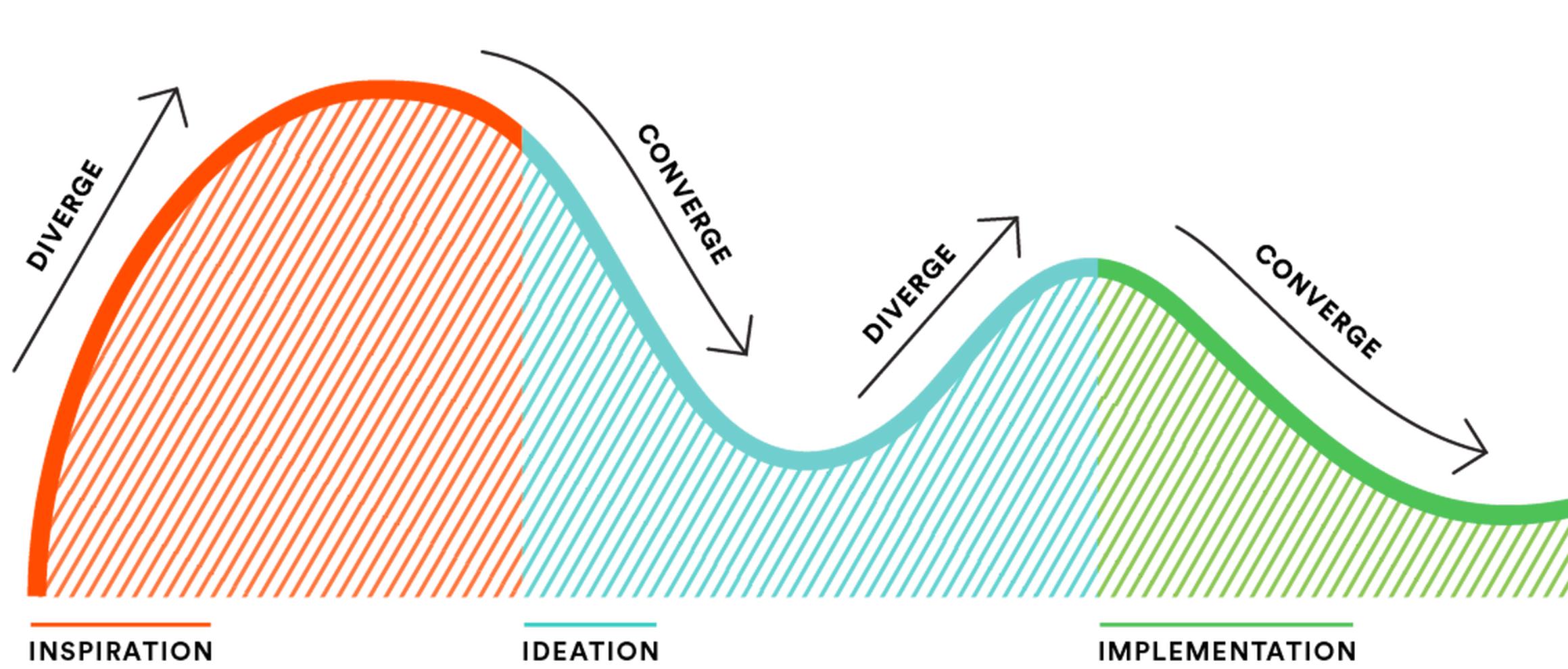


# Take away messages

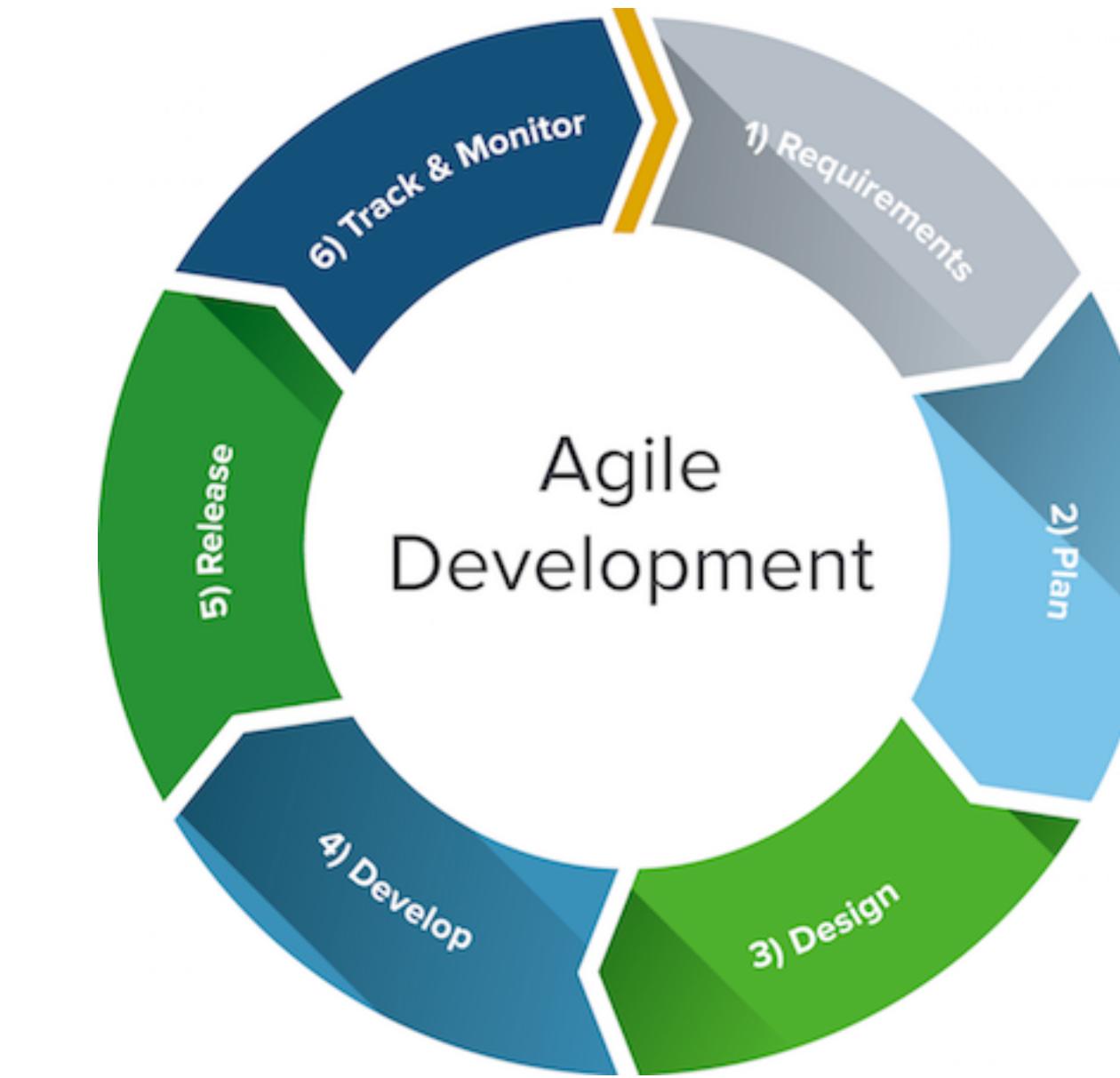
- Search before you build
- Build by example
- Build for accessibility
- Build with caution
- Build on a solid foundation

# **Applying this course in practice**

# Product design process

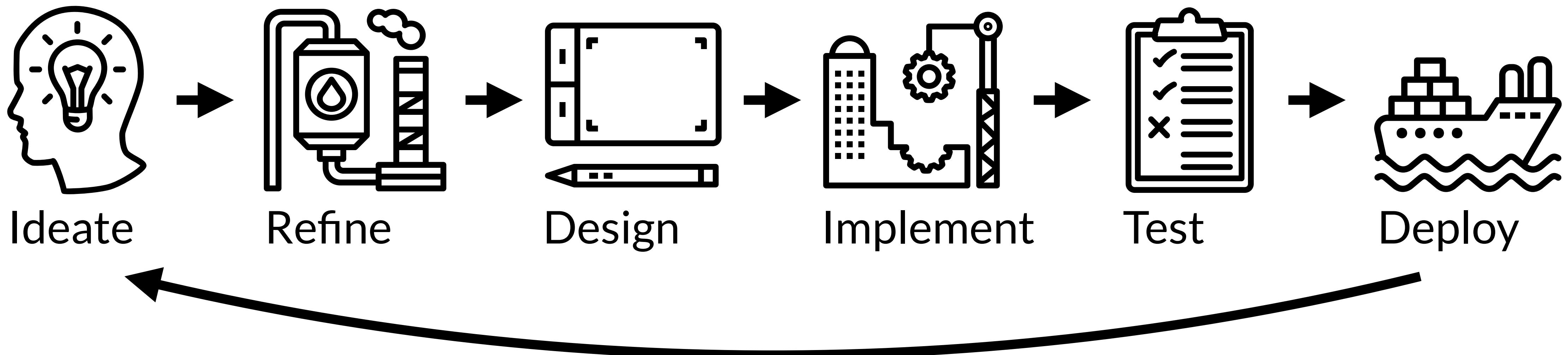


Human-Centered Design, IDEO



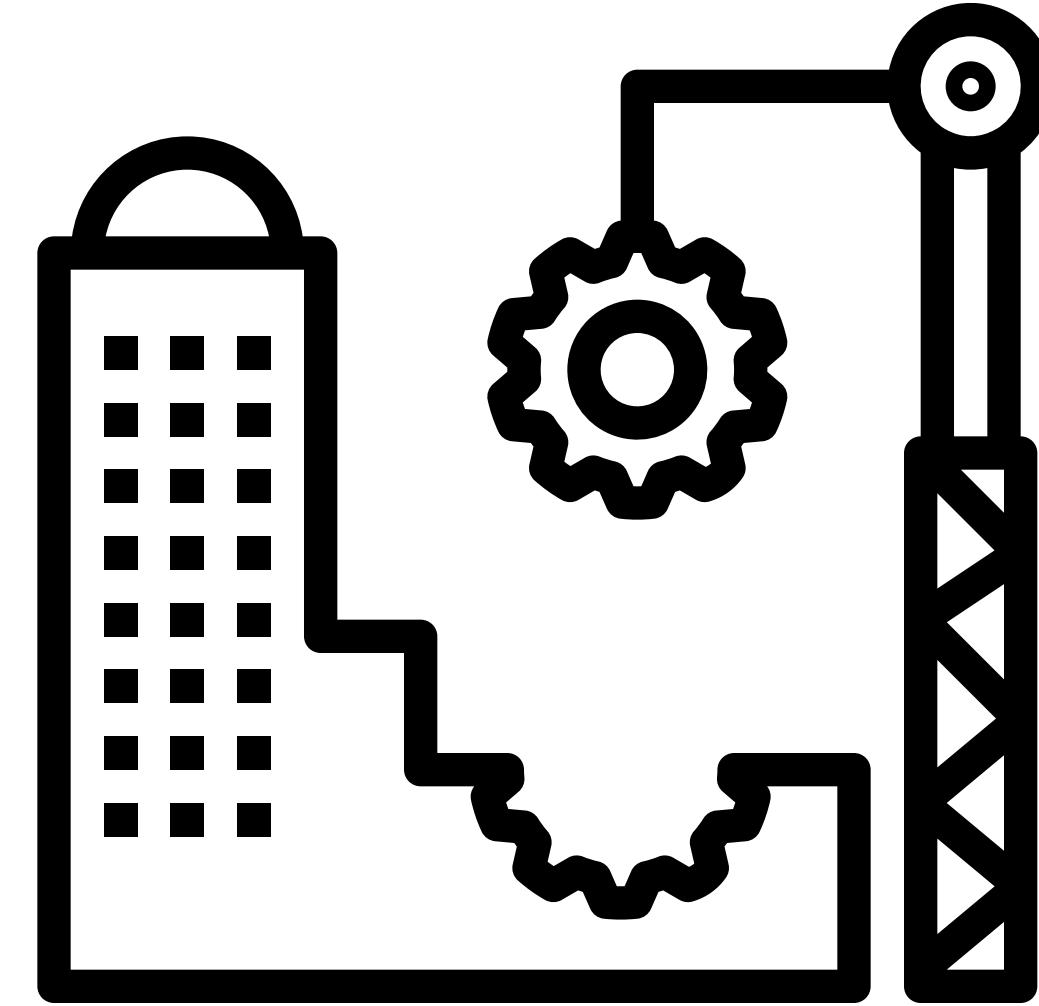
Agile Development, Agile Manifesto

# Product design process, simplified



# User interface implementation

- Has the power to turn ideas into reality
- Often dictates design decisions and timelines, for better or for worse
- Either you will be implementing, or you will need to communicate with your colleagues who are



**What job might you get  
when you graduate?**

# Some job options

- User experience designer
- User experience researcher
- Front-end software engineer
- Back-end software engineer
- Academic researcher (graduate student)
- Software consultant
- Something unrelated to technology
- ... others?

# If you're going into UX, you can now...

- Follow principles of web, mobile, AR design
  - Responsive design! Error prevention! Give clear instruction!
- Be conversational in web and mobile programming
  - Be able to understand what tasks are easy and what are hard
  - And understand when a developer is BSing you about how long something will take
- Style a webpage
  - Use CSS and SASS to change a design and even add animations

# Front-end software engineering...

- Build a webpage in plain HTML
  - Make it responsive with Bootstrap
- Use a framework to build a richer application
  - Angular for a web frontend
  - Ionic for a mobile frontend
- Style a webpage
  - Use CSS and SASS to change a design and even add animations

# Back-end software engineering...

- Build a web server
  - Allow it to respond to requests from a front-end interface
  - Allow it to make requests to APIs made by other developers
- Follow authentication and authorization protocols
  - Enable users to sign on
- Use a database
  - Data can persist between sessions

# Academic research...

- Explain some key problems in a couple of areas
  - Ubiquitous computing
  - Human performance
  - Mixed reality design
  - Smartphone systems security
  - Wearable computing
  - Augmented and virtual reality

# Software consultancy...

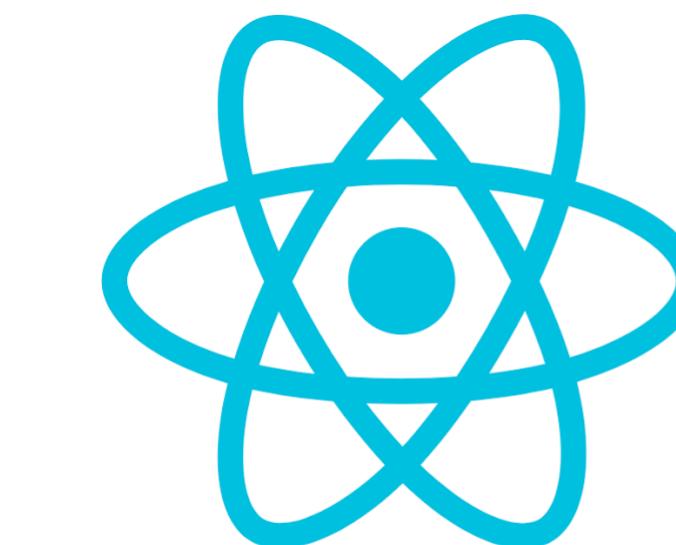
- Process and analyze data
  - Retrieve it from an API
  - Parse and process it to answer your question
- Visualize data
  - Use an appropriate tool for the task

# Something unrelated...

- Make a portfolio to show off your skills
  - Selling yourself is key
- Judge new devices and apps that come along
  - Is this solving a real problem?
  - Is this well designed?

# What is interface implementation today?

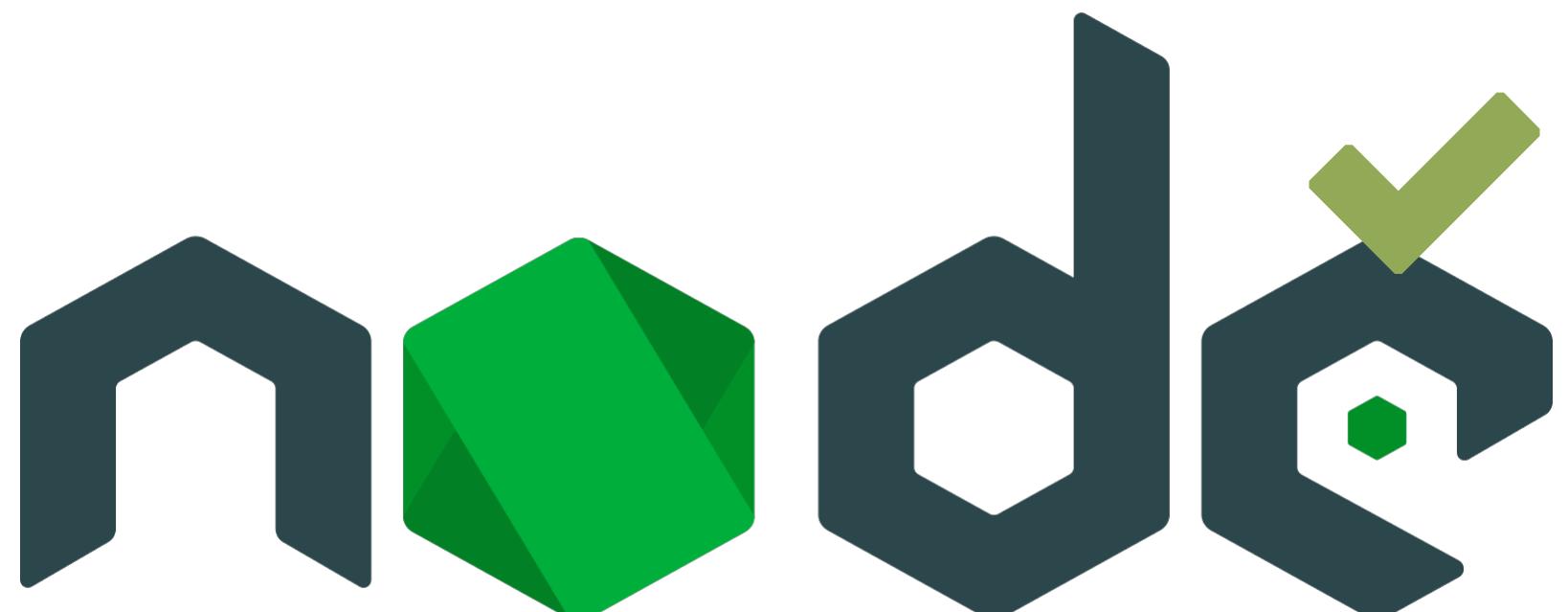
Often HTML, CSS, and JavaScript



React JS



Bootstrap



Vue.js



ionic

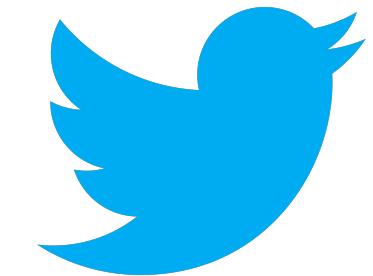
ember

# Assignments

- A1: Personal web portfolio



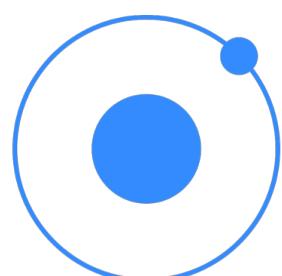
- A2: Programming on the web



- A3: Web frameworks

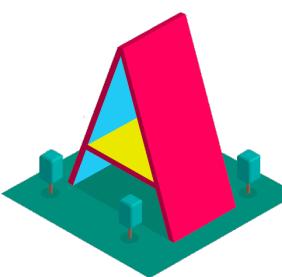


- A4: Mobile development



ionic

- A5: Beyond Web & Mobile



# Other skills

- Git and GitHub
- Package management in npm
- SASS
- Visualization in Vega-Lite

# Question



What framework or language did you find the most challenging to pick up?

- A HTML and CSS
- B JavaScript and TypeScript
- C Angular
- D Ionic
- E Platforms beyond web & mobile

# Question



What framework or language did you find the most rewarding to learn to use?

- A HTML and CSS
- B JavaScript and TypeScript
- C Angular
- D Ionic
- E Platforms beyond web & mobile

# Congratulations!

- We said this class would be challenging
- You have risen to the challenge and worked hard (and still are)
- You have created impressive work as a result

**It's been an honor  
to be able to teach you.**

**I look forward to seeing  
what you do next!**

# Today's goals

By the end of today, you should be able to...

- Describe how Jo and Lucas concepts from IN4MATX 133 in their research and practice
- Summarize what you learned in IN4MATX 133
- Describe the relevance of the topics to different disciplines in industry
- Fill out the course evaluation!

# **IN4MATX 133: User Interface Software**

**Lecture 18:**  
**Wrap-Up**

Professor Daniel A. Epstein  
TA Eunkyung Jo  
TA Lucas de Melo Silva