

IN4MATX 133: User Interface Software

Lecture 20:
Beyond Web & Mobile
and Wrap-Up

Professor Daniel A. Epstein
TA Lucas de Melo Silva
TA Jong Ho Lee

Today's goals

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

- Summarize what you learned in IN4MATX 133
- Describe the relevance of the topics to different disciplines in industry
- Fill out the course evaluation!

What is a wearable computer?

A MUCH More Diversified Market Than Investors Realize



CREDIT SUISSE

What is a wearable computer?

- A computer on the body that is:
 - Always on
 - Always accessible
 - Always connected
- Other actions:
 - It augments user actions
 - Is aware of the user and their surroundings

Rhodes, B.J. 1997. The wearable remembrance agent: a system for augmented memory.
Personal Technologies, 1(4), 2018-224.

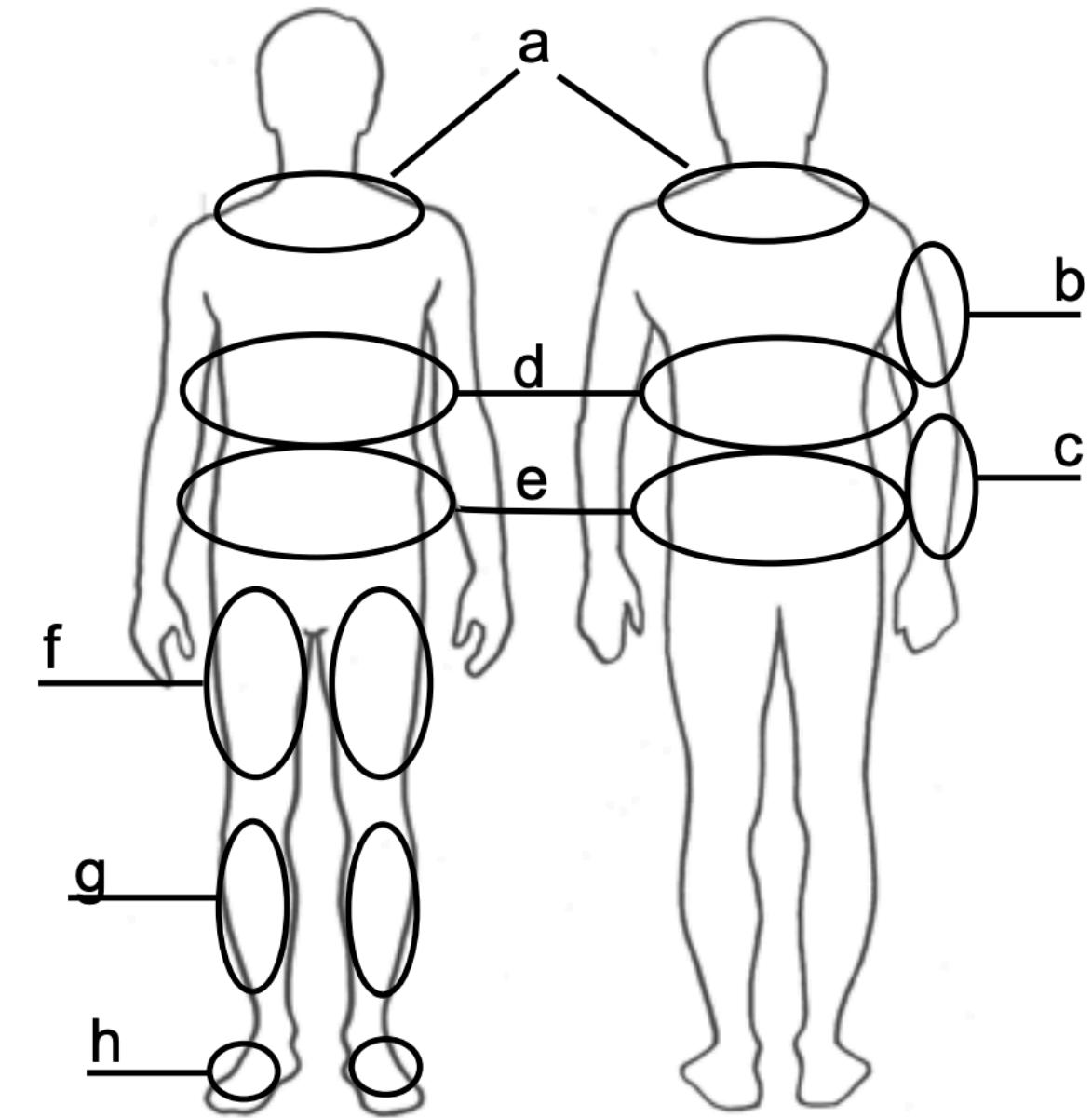
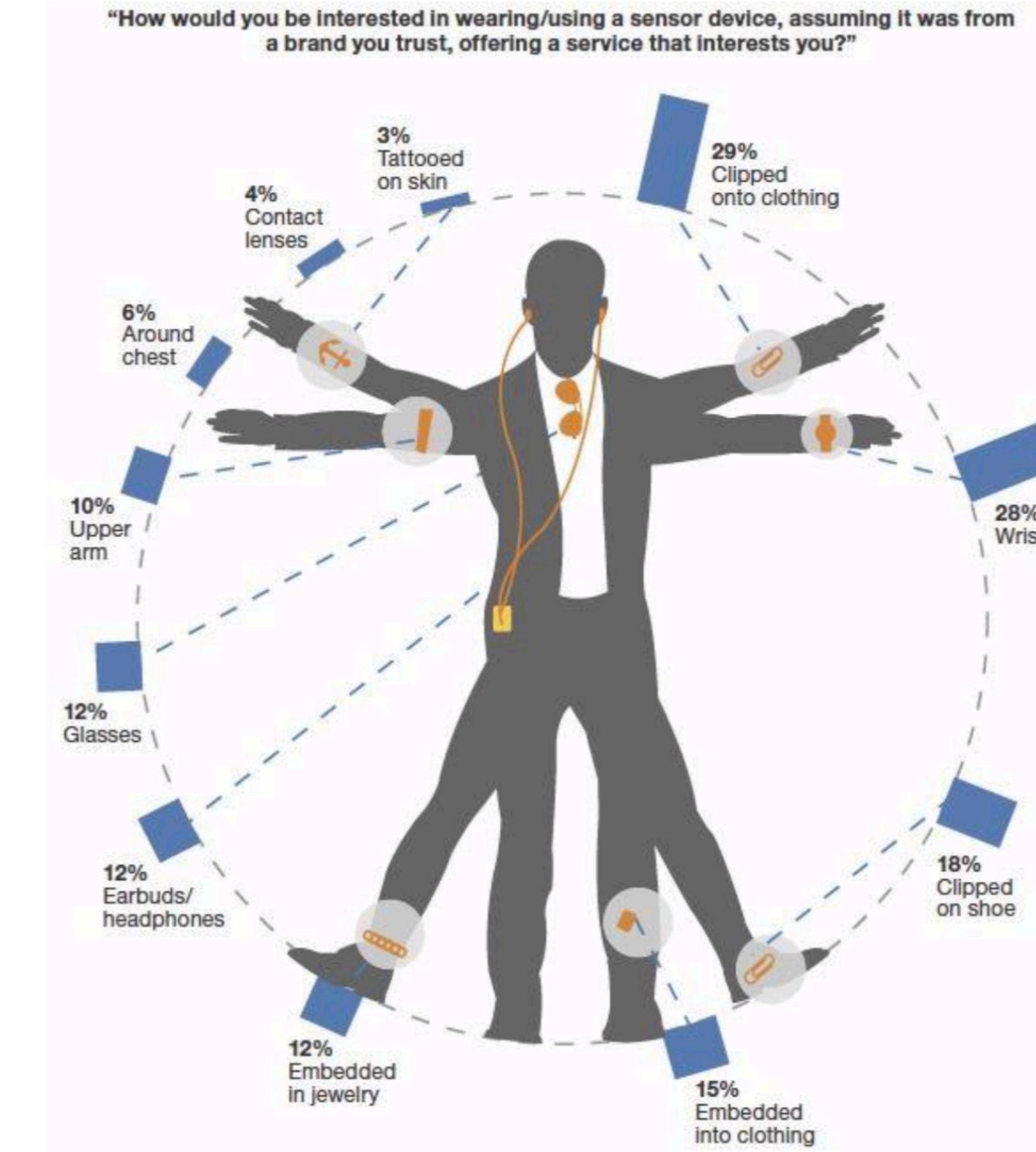


Fig. 1, The general areas we have found to be the most unobtrusive for wearable objects are: (a) collar area, (b) rear of the upper arm, (c) forearm, (d) rear, side, and front ribcage, (e) waist and hips, (f) thigh, (g) shin, and (h) top of the foot.

Gemperle, F. Kasabach, C., Stivoric, J. Bauer, M. Martin, R. Design for Wearability. ISWC 1998.



North American Technographics
Consumer Technology Survey, 2013.

Body and head-mounted wearables

MIT Wearable Computing (1996)



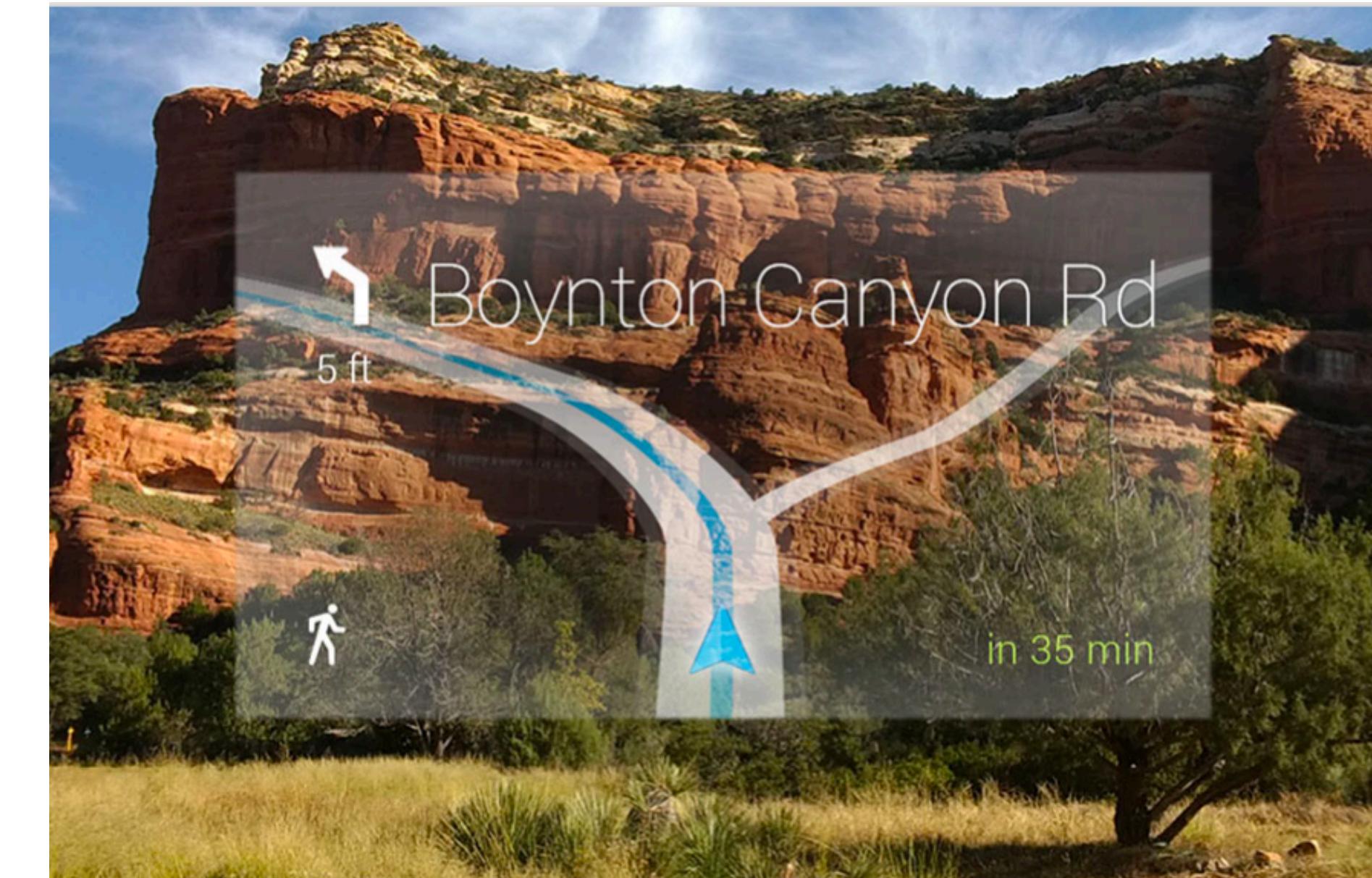
Google glass

- Commercial smart glasses, released in 2013
 - Technology lead by Thad Starner, part of the MIT group
- Front-facing camera, rear-facing display
- Gyroscope/accelerometer/magnometer
- Natural language input capabilities



<https://www.x.company/glass/>

Google glass



<https://www.x.company/glass/>

Google glass

- Privacy and safety concerns prevented take-off in the consumer space
- Lives on in enterprise spaces
 - New version released in 2017
 - Used in manufacturing, healthcare



<https://www.x.company/glass/>

Wrist-worn wearables

Fitbit (2011)

- One of the first commercially successful digital pedometers
- Early versions were hip-worn, now almost exclusively wrist-worn
- Current models are “fitness-first” smartwatches
 - Activity prominently included on the home screen
- Acquired by Google in 2019



Pebble (2013)

- Arguably the first commercially successful smartwatch
 - Two of the most funded Kickstarter projects ever
- E-ink display led to high battery life (a week vs. a day)
- Paired with a phone via Bluetooth
 - Could retrieve email, control music, receive notifications, etc.
- Acquired by Fitbit in 2016



Apple Watch (2015)

- From the onset, it was intended to be a “second screen” companion to iOS devices
- Original versions could do almost nothing without pairing to an iOS device
- Apps add a secondary component to an existing iOS app



Design recommendations for (wrist-worn) wearables

One visual thought per screen

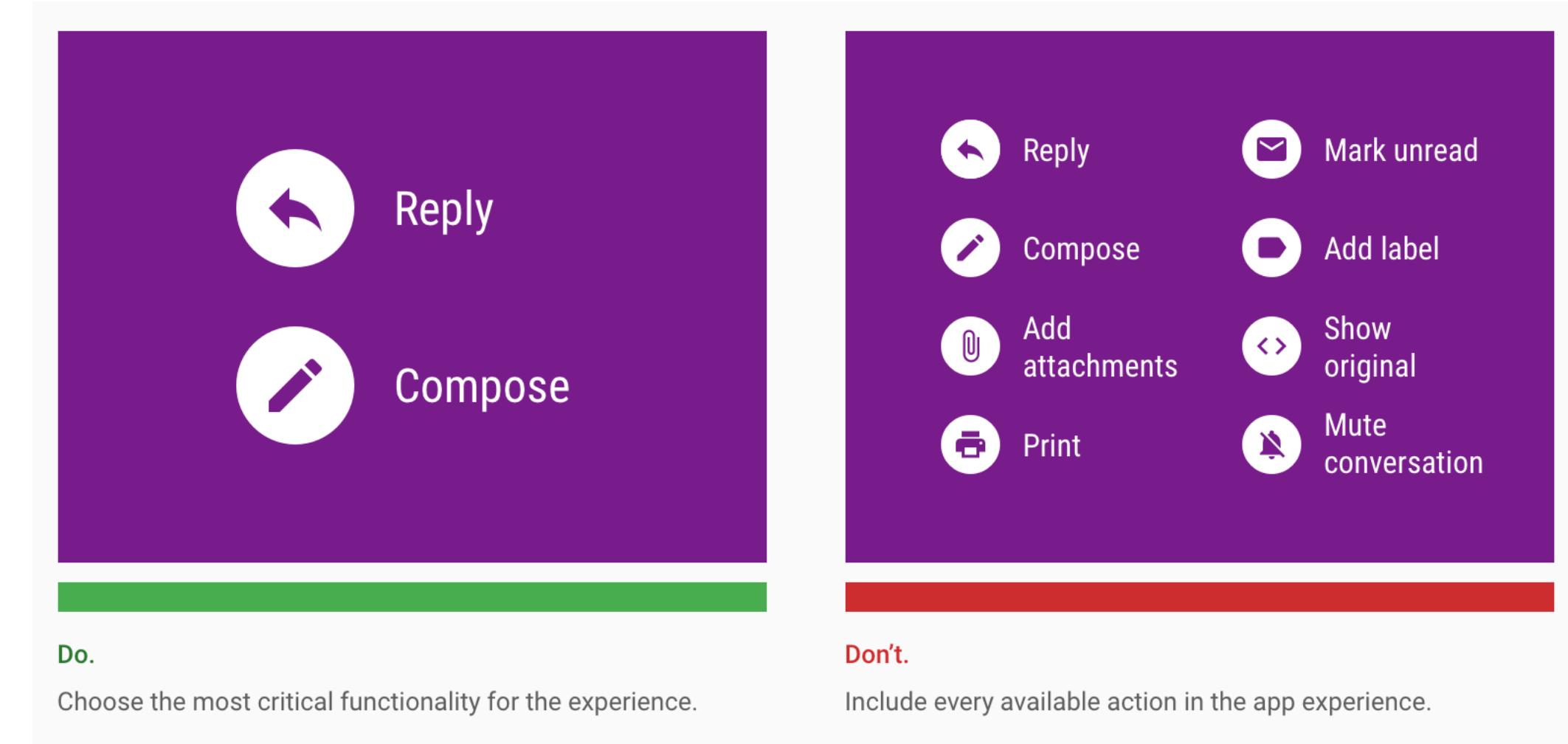
- Real estate is extremely valuable on watches
- Shrinking a mobile or desktop app will create a bad experience
- Keep words and interactions to a minimum



<https://mayvendev.com/blog/10-tips-for-designing-for-wearables-and-watches>

Reduce input options

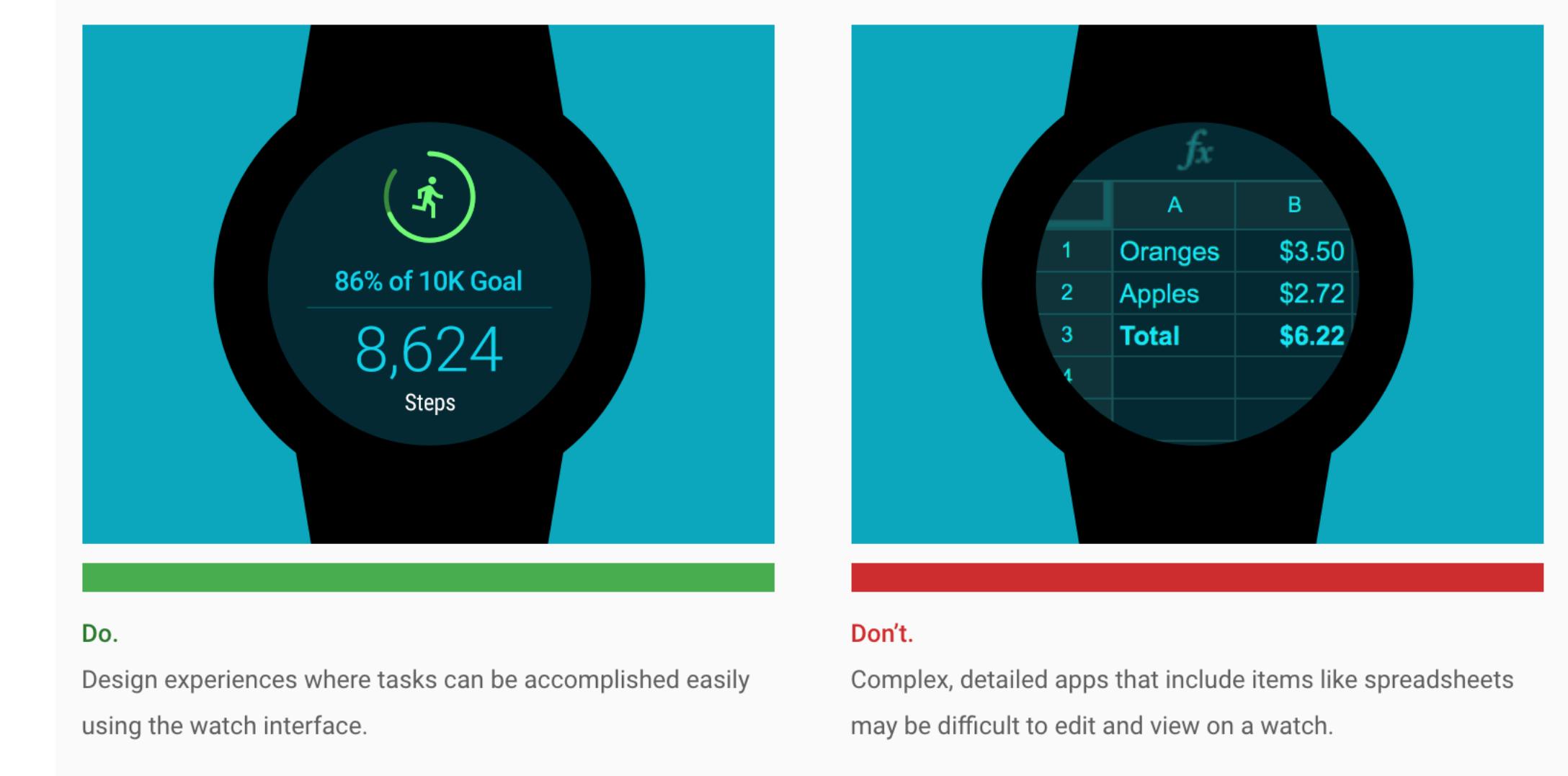
- Have only a couple of buttons per screen
- This may mean a watch app has fewer features than a mobile app
 - That's okay!
 - The watch augments the experience
- Consider voice input when longer instruction is needed



<https://designguidelines.withgoogle.com/wearos/wear-os-by-google/designing-for-watches.html>

Some apps don't need a watch interface

- For some apps, a watch app may not add to the experience
- Focus on use cases which make sense
 - Quick input
 - Glanceable feedback



<https://designguidelines.withgoogle.com/wearos/wear-os-by-google/designing-for-watches.html>

Question



These are all Apple Watch apps.
Which have a compelling use case?

Pick whichever choice is closest to what you believe.

- A None of them have a compelling use case
- B Dark Sky
- C Dark Sky, RunKeeper, Fandango
- D Dark Sky, RunKeeper, New York Times, OneNote
- E All of them have a compelling use case

New York Times (news)
Dark Sky (weather)
Fandango (movie purchase)
RunKeeper (fitness)
OneNote (notetaking)

Question



These are all Apple Watch apps.
Which have a compelling use case?

Pick whichever choice is closest to what you believe.

- A None of them have a compelling use case
- B Dark Sky
- C Dark Sky, RunKeeper, Fandango
- D Dark Sky, RunKeeper, New York Times, OneNote
- E All of them have a compelling use case

New York Times (news)
Dark Sky (weather)
Fandango (movie purchase)
RunKeeper (fitness)
OneNote (notetaking)

Open to interpretation,
I think NYT and Fandango
are a stretch but there are
potential uses

Microsoft apps for Apple Watch



Powerpoint remote



Skype messaging



OneNote voice memos

Are these compelling use cases?

<https://www.theverge.com/2015/4/28/8508445/microsoft-apple-watch-apps-hands-on>

Questions to consider

- Would a watch app add anything to my mobile app?
 - Is there timely information the app needs to provide?
 - Can it be shown in a very small format?
 - Are there simple controls to the app that would be added to a watch?
- Do I have the resources/time to do this?
 - Currently limited market impact, but growing
- What type of interaction do you want the user to have?

Implementing watch apps

- Requires native development, as far as I know
 - WatchKit for iOS, Wear OS for Android
- Requires a companion iOS or Android app for building/deploying, though may be able to run as a standalone
- However, you can develop a hybrid mobile app and connect it to a native watch app

<https://developer.android.com/training/wearables/apps>

<https://developer.apple.com/documentation/watchkit>

What is augmented reality?
What is virtual reality?
How do they differ?

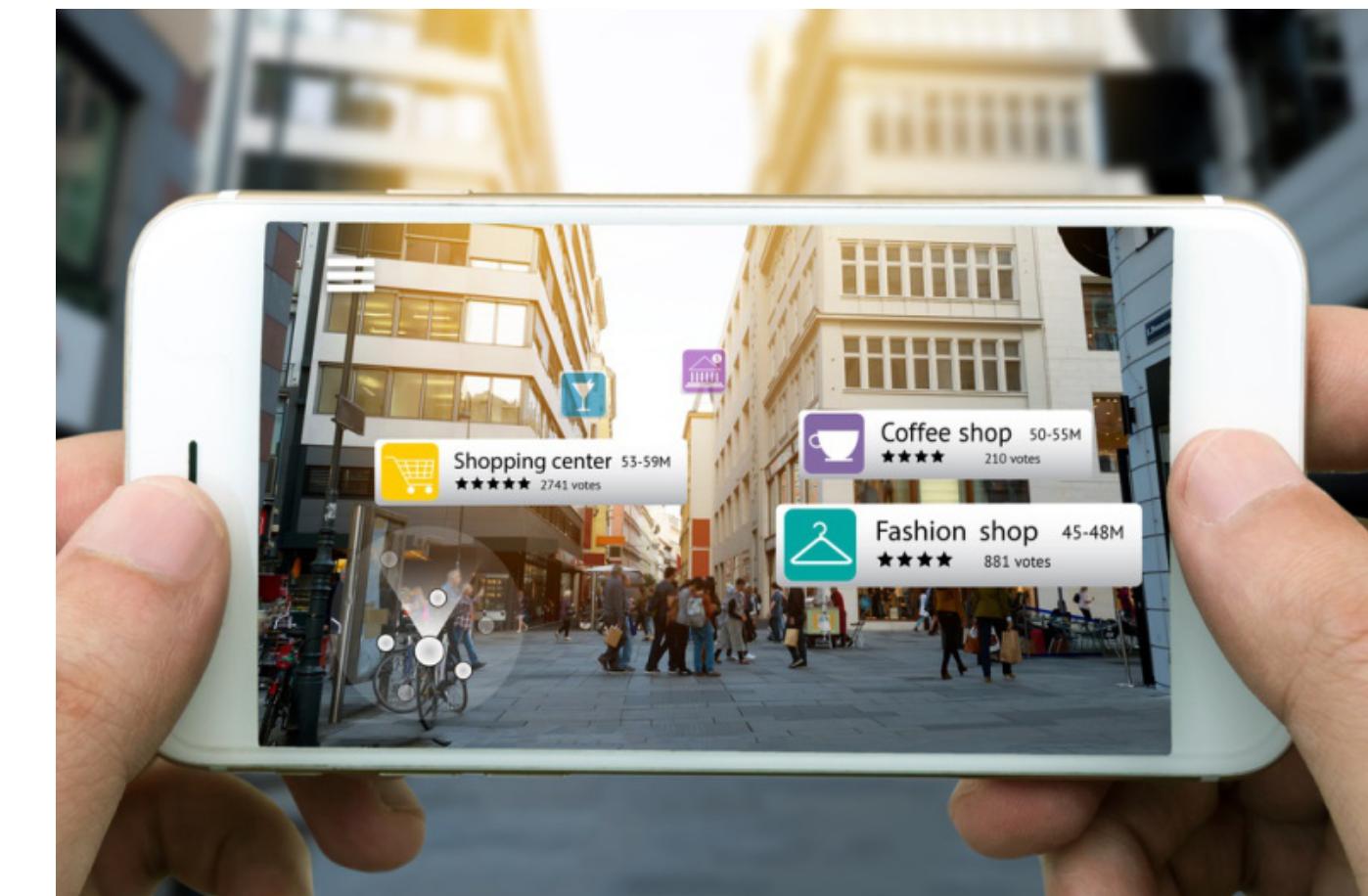
Augmented reality vs. Virtual reality

- Virtual reality aims to transport a person to an entirely new place by blocking out the real world as much as possible
 - Ideally a full sensory experience



Augmented reality vs. Virtual reality

- Augmented reality layers digital enhancements on top of a view of the real world
 - Digital content is interactive
 - Digital content is viewable in 3D (or projected 3D)



Mixed reality (hybrid reality)

- Physical objects in the real world can be interacted with in the digital world
 - Kind of a hybrid of AR and VR
- By comparison, VR seeks to immerse a person in a completely artificial environment
- While AR just overlays objects in the real world

Virtual, Augmented, and Mixed reality

- Virtual reality
 - HTC Vive, Oculus rift
- Augmented reality
 - Apple ARKit, SnapChat lenses, Pokemon Go
- Mixed reality
 - Microsoft HoloLens

CAVE (1992)



<https://www.youtube.com/watch?v=-Sf6bJjwSCE>

Carolina Cruz-Neira, Daniel J. Sandin, Thomas A. DeFanti, Robert V. Kenyon, and John C. Hart.
The CAVE: audio visual experience automatic virtual environment. *Communications of the ACM* 35, 6 (June 1992)

CAVE (1992)

- Viewing headset
(though it used projection walls)
- Sensors to detect a person's position and orientation in the space
- Audio feedback
- Motion controller



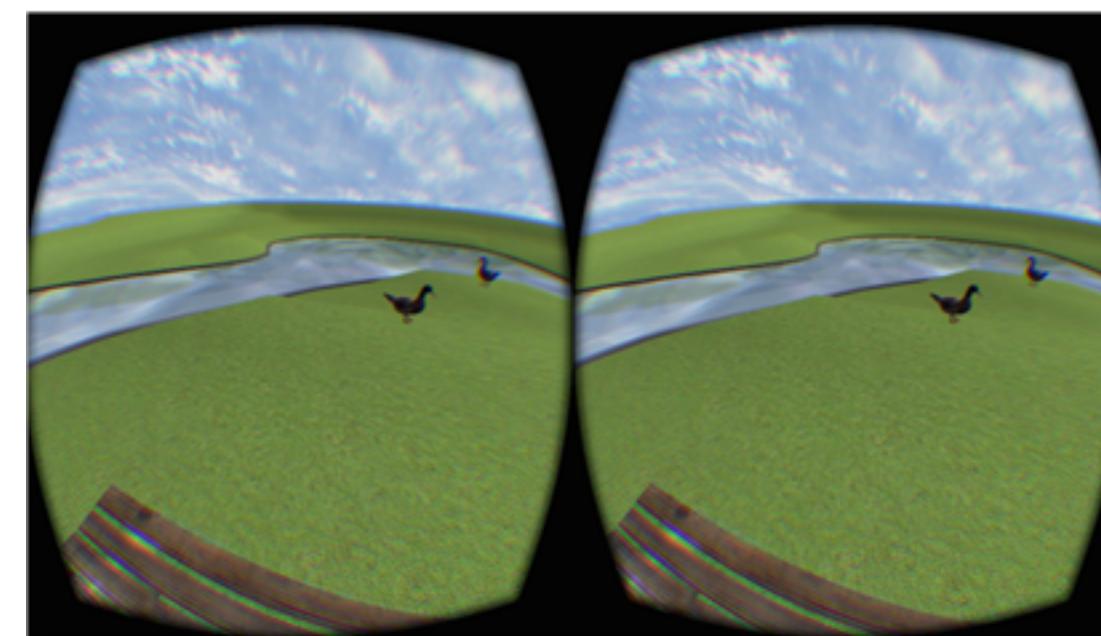
CAVE (1992)

- Necessitated a specialized setup
 - Large room
 - Four projection walls
 - Expensive wearable displays
- Technology could never be adopted by consumers
 - Who has that kind of space to devote to VR, never mind money?



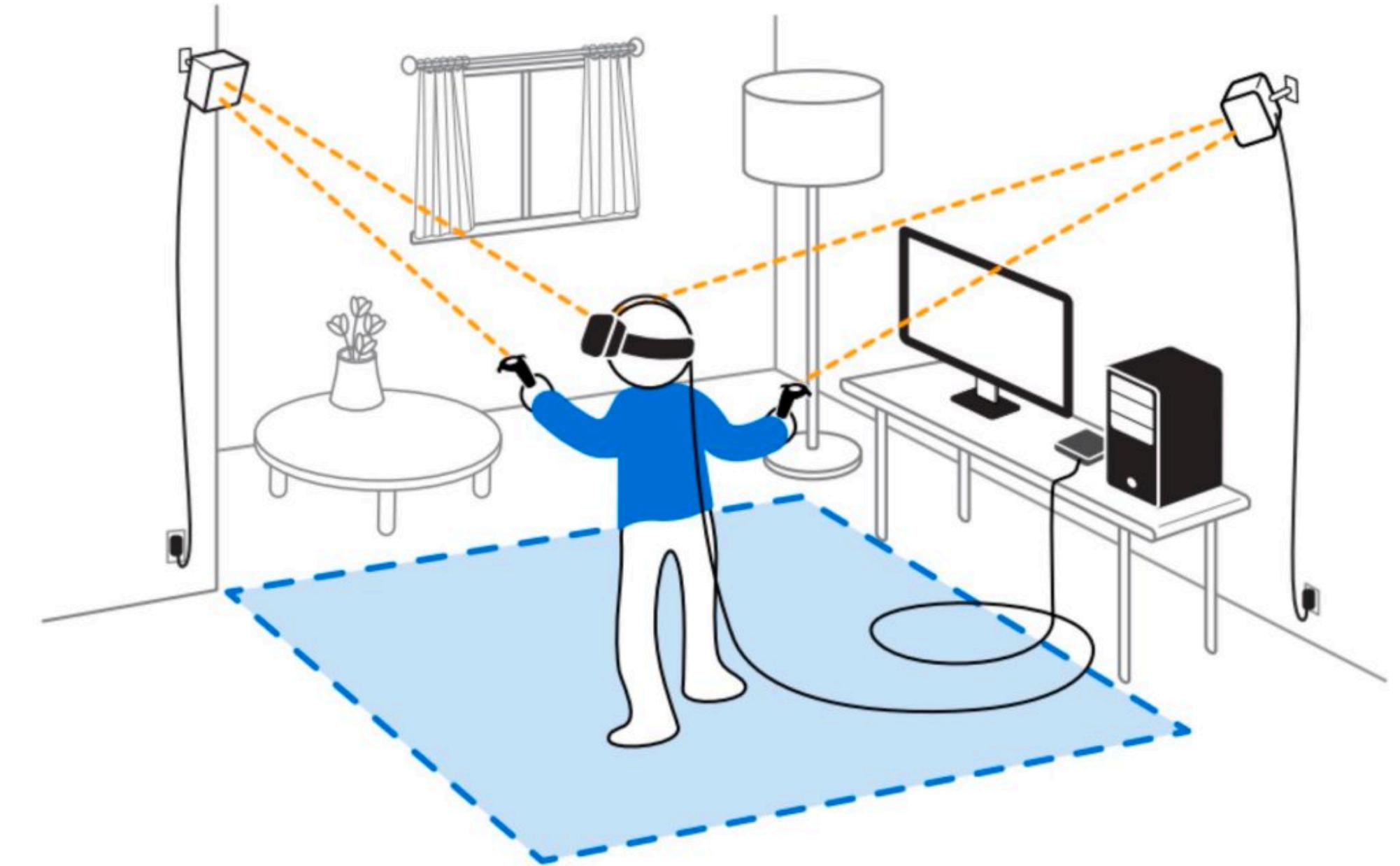
Google Cardboard (2014)

- Explicitly *not* a fully immersive experience
- Uses a phone to generate a screen for each eye
- Uses accelerometer and gyroscope for positioning



Oculus Rift & HTC Vive (2016)

- Clear out a play space
- Position sensors around the environment
- Motion controls
- An overlay headset



Microsoft HoloLens (2016)



Microsoft HoloLens (2016)

- Mixed reality system
 - Like AR, adds layers to the real world
 - Some physical objects can be interacted with
- Focused on commercial uses
 - Pricing: \$3000/unit
 - Though there are other videos which demonstrate entertainment applications



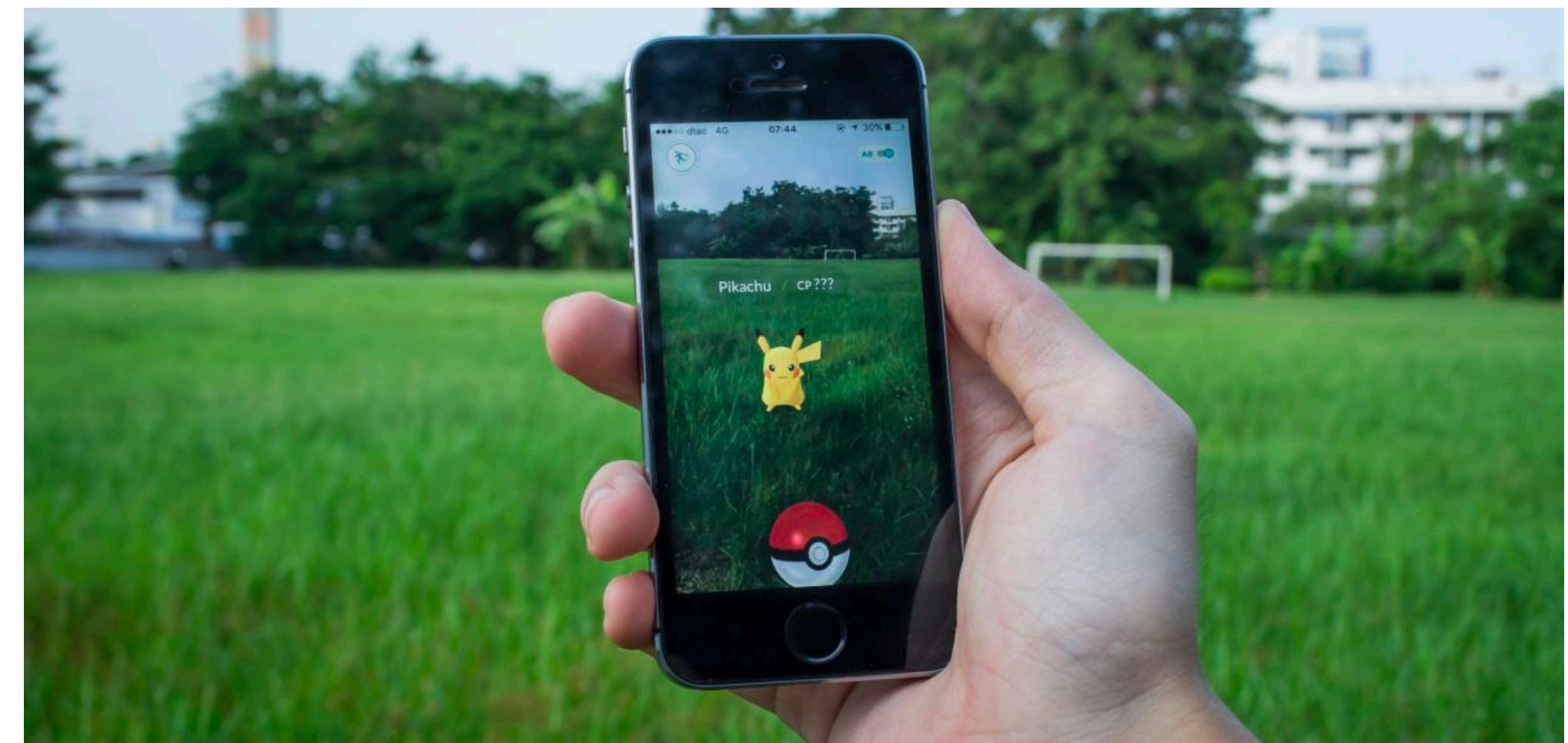
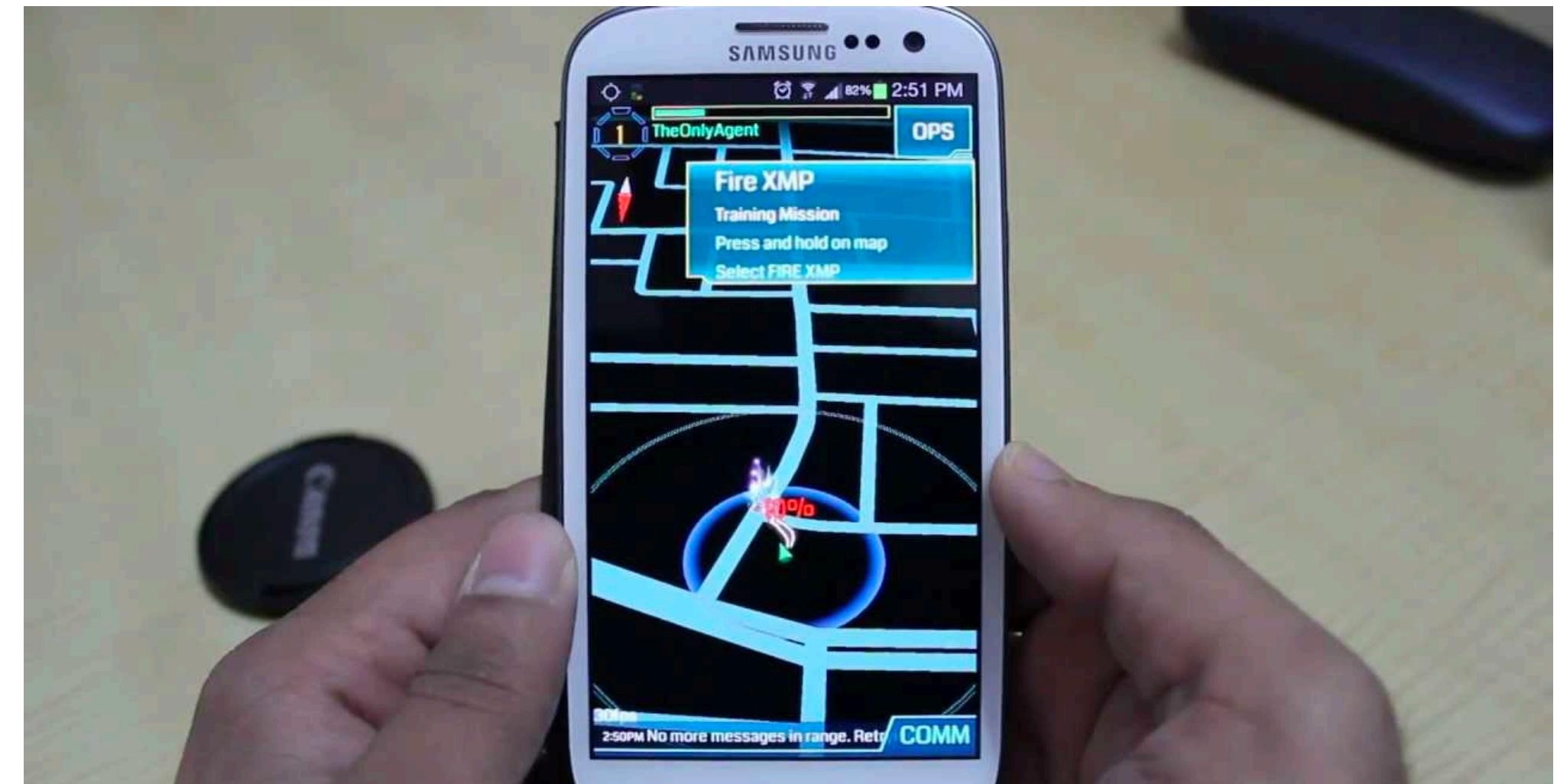
Snapchat (2011)

- A key component is augmenting the environment in fun ways
 - Mostly known for facial filters (I think? You tell me)
- A “camera company”, not a “social media company”
 - Core product is currently the social network



Ingress (2012) and Pokemon Go (2016)

- Popular games with millions of users (even today)
- AR aims to augment the real world
 - A layer over the camera is just one sense (vision)
 - Location-based PokeStops, Gyms, etc. add others
 - Location-based gameplay leads to social interactions between players



Question

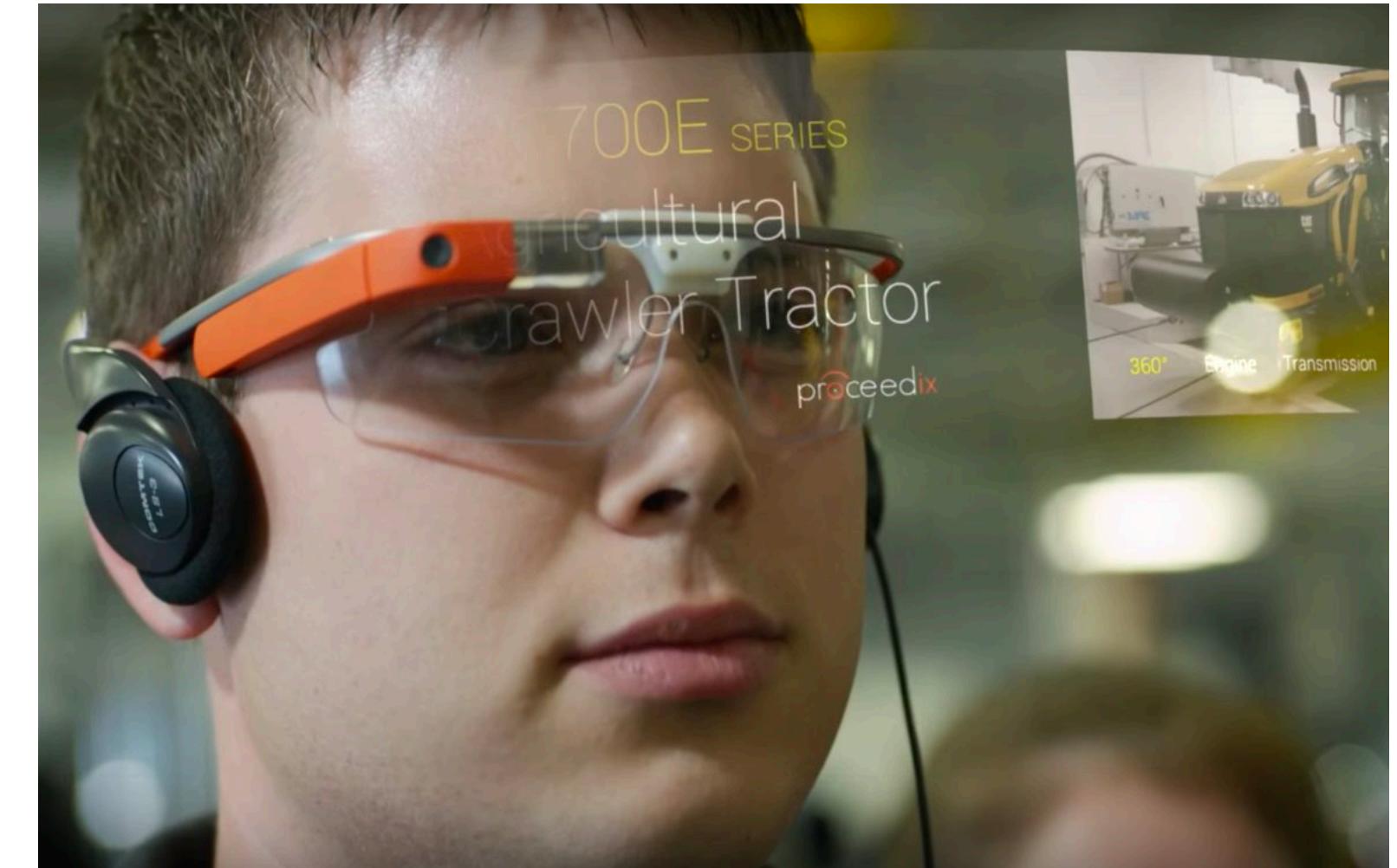


Augmented or virtual reality?

- A Both are AR
- B Cardboard is AR, Glass is VR
- C Both are VR
- D Cardboard is VR, Glass is AR
- E Both are neither AR nor VR



Google Cardboard



Google Glass

Question

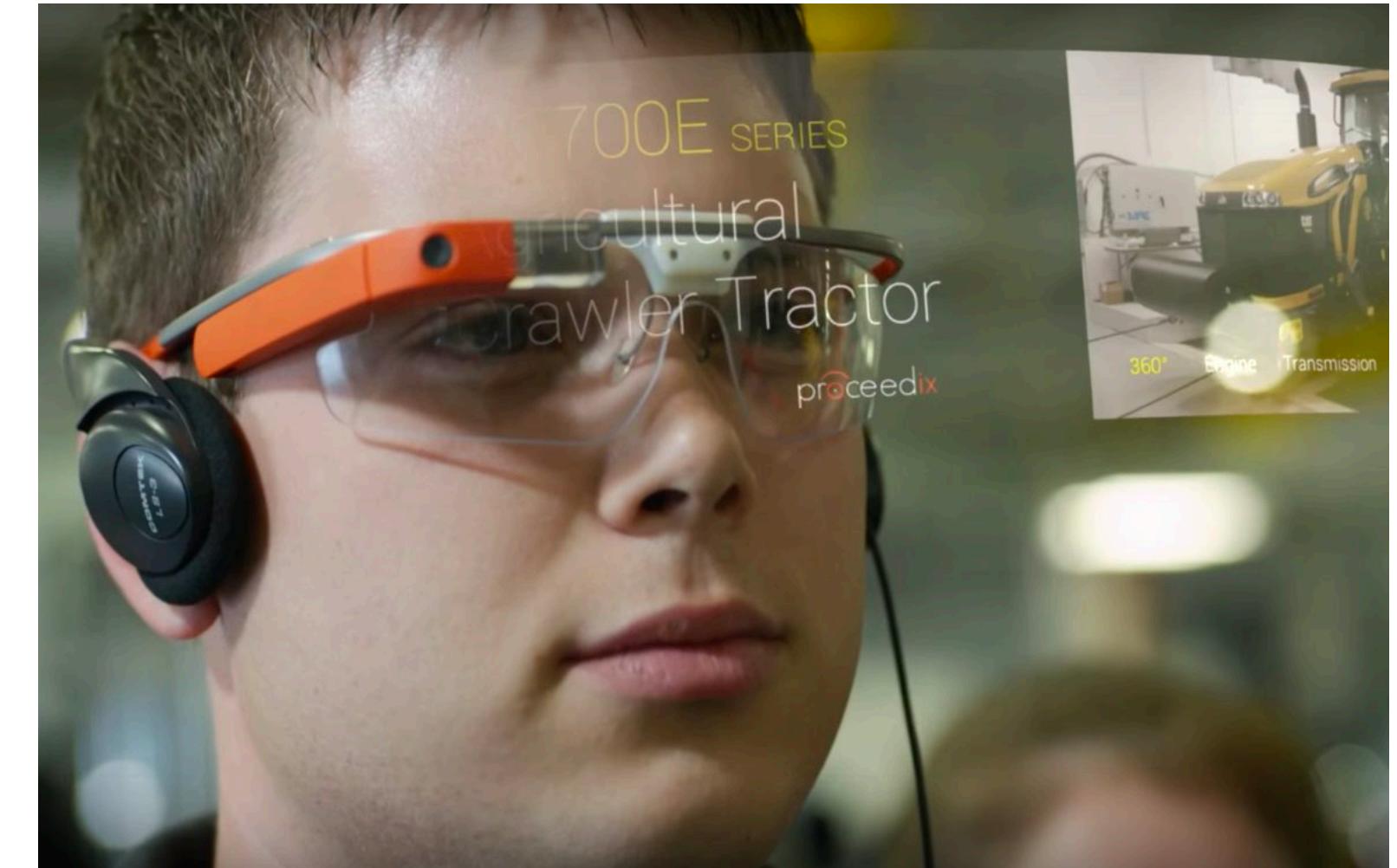


Augmented or virtual reality?

- A Both are AR
- B Cardboard is AR, Glass is VR
- C Both are VR
- D Cardboard is VR, Glass is AR
- E Both are neither AR nor VR



Google Cardboard



Google Glass

VR applications

- Games
- Flight simulators
- Sophisticated testing (car prototypes, etc.)
- Training (military, medical, industry, etc.)
- Recreation (nature hikes, exploring)

VR challenges remain consistent

- Creating a truly immersive experience is difficult
 - Need to replicate touch, sight, hearing... smell and taste?
- Most people do not have a physical space which can be taken over by a virtual environment
- Requires instrumenting the body, the environment, or both
 - Heavy, bulky, expensive, etc.
- Most practical uses are for enterprises rather than consumers

AR and Mixed reality avoid some problems

- The experience does not need to be fully immersive
- The physical space is literally part of the environment, and maybe even part of the experience
- Halfway-decent AR is pretty cheap computationally
 - It can run on your phone!
 - Other sensors like Gyroscope, etc. add to the experience
- Low cost makes AR practical for entertainment, etc.
 - Mixed reality is still very expensive

Implementing Augmented and Virtual Reality

VR Implementation

- Many common 3D development environments can be used
 - VR changes the rendering and the input
- Unity VR (and AR)
 - <https://unity3d.com/learn/tutorials/s/xr>
- Unreal VR (and AR)
 - <https://www.unrealengine.com/en-US/vr>
- Google cardboard
 - <https://developers.google.com/vr/develop/unity/get-started-android>

AR Implementation

Hybrid Development

- React 360
 - <https://github.com/facebook/react-360>
- Argon JS
 - <https://github.com/argonjs/argon>
- AFrame
 - <https://aframe.io/>

AR Implementation

Native Development

- Google ARCore
 - <https://developers.google.com/ar/>
- Apple ARKit
 - <https://developer.apple.com/arkit/>

A few AR design recommendations

Label surfaces during set-up phase

- Highlight surfaces where objects can be placed
 - No visual cues can create confusion when a person goes to place an object
- Differentiate multiple surfaces
 - Update surfaces as a person moves their phone around



<https://designguidelines.withgoogle.com/ar-design/>

Provide guidance on interactions

- There gestures and interactions in AR are not standardized
 - Standards will come as the platform matures
- Use a combination of visual cues and text explanations to guide interactions
 - Dismiss hints when they have been performed successfully



<https://designguidelines.withgoogle.com/ar-design/>

Anticipate space limitations

- Do not require a user to walk backward or make quick sweeping motions
- Give a clear idea of the amount of space needed for an interaction
- Avoid putting large objects close to the user on the screen
 - They will instinctively back up

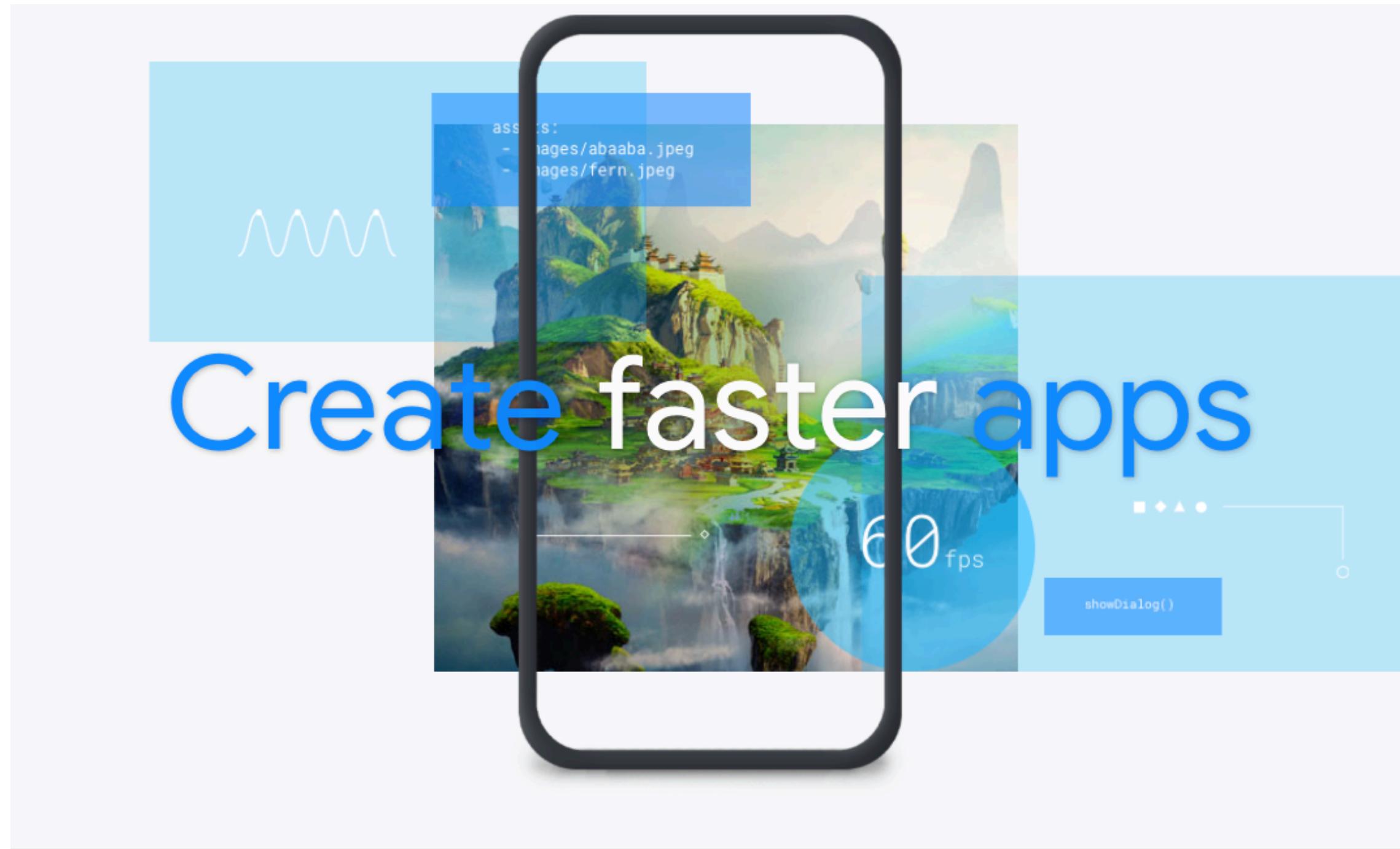


<https://designguidelines.withgoogle.com/ar-design/>

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.

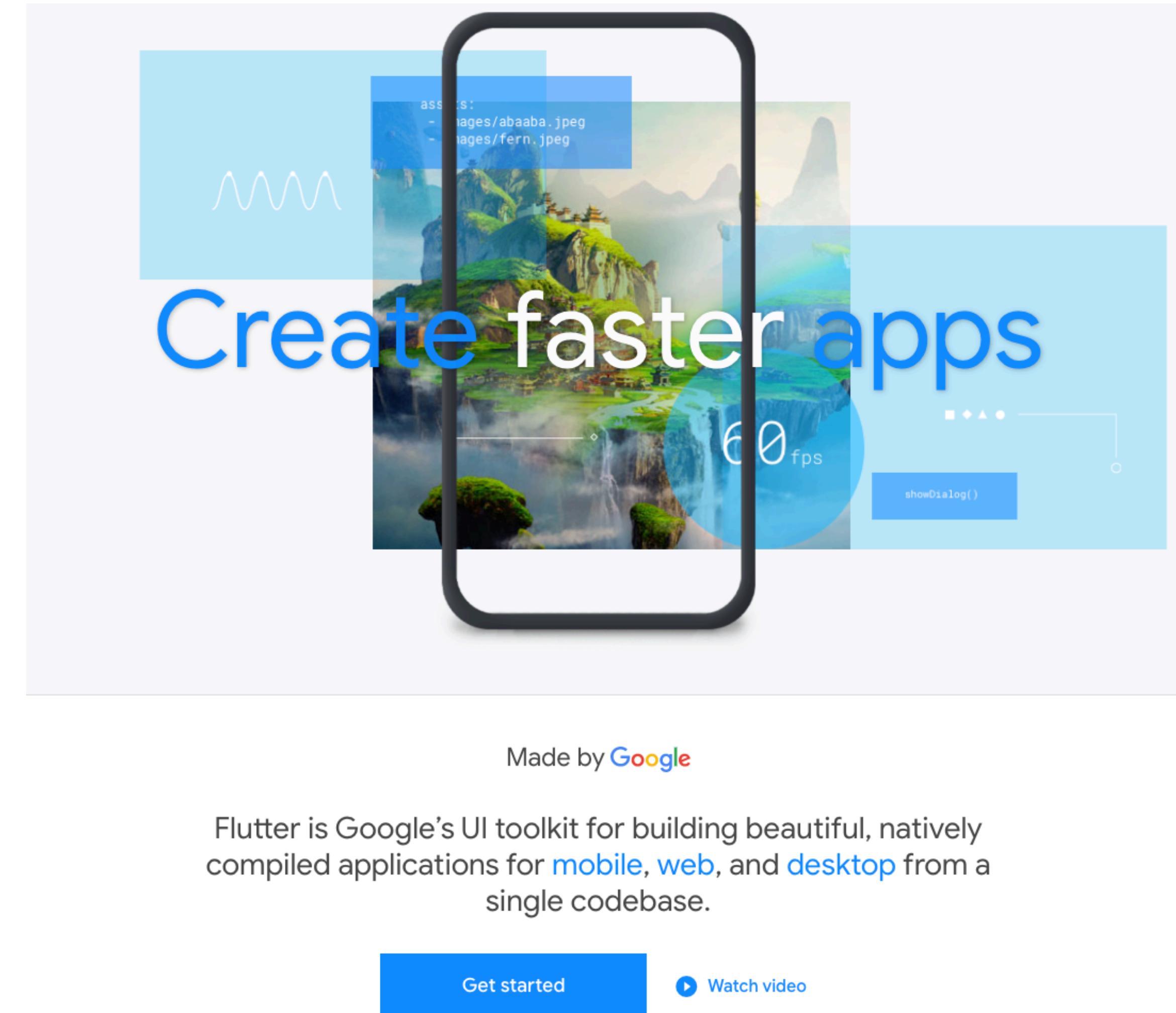
[Get started](#)

[Watch video](#)

<https://flutter.dev/>

Technology changes quickly

- A hybrid framework for building Android and iOS apps
 - Goal: higher performance
 - Written in Dart, an object-oriented language Google has been pushing
 - Downside: new language...
 - Includes libraries for some native resources (Camera/photos)
 - May be more reliable than Ionic
- <https://flutter.dev/>



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](#)

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

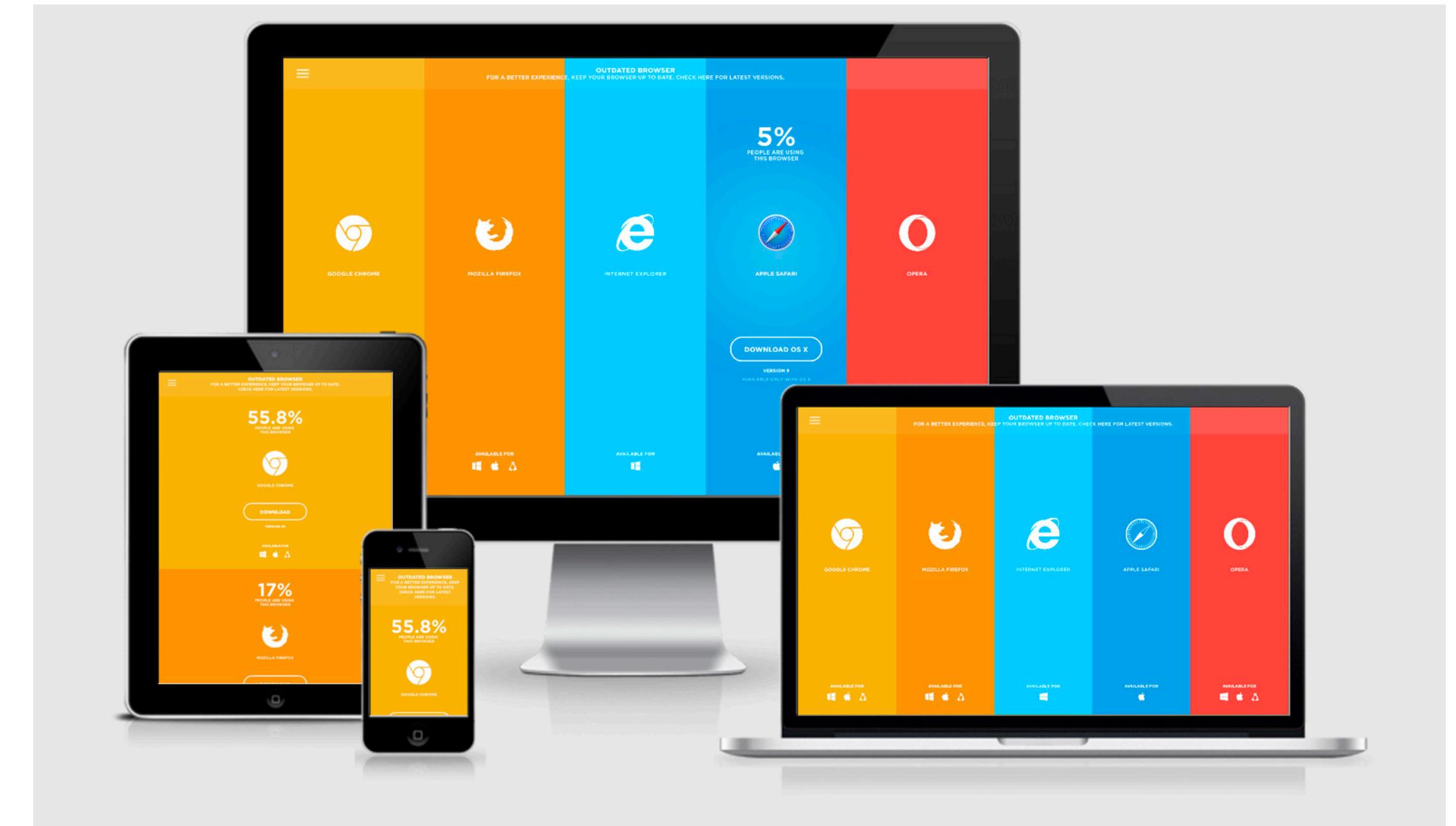
The screenshot shows the 'Elements' tab of a browser's developer tools. A specific `<body>` element is selected in the DOM tree. The right panel displays the CSS styles applied to this element, categorized into 'element.style', 'user agent stylesheet', and 'Inherited from'. The 'element.style' section includes rules like `width: 100%; min-height: 100%; font-family: Helvetica, Arial, sans-serif; margin: 0; padding: 0; word-wrap: break-word;`. The 'user agent stylesheet' section includes `display: block; margin: 8px;`. The 'Inherited from' section includes `color: #rgba(0,0,0,.87); font-size: 1em; line-height: 1.4;`.

```
<!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>
```

html.no-touch.no-js.mdl-js body.page--

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

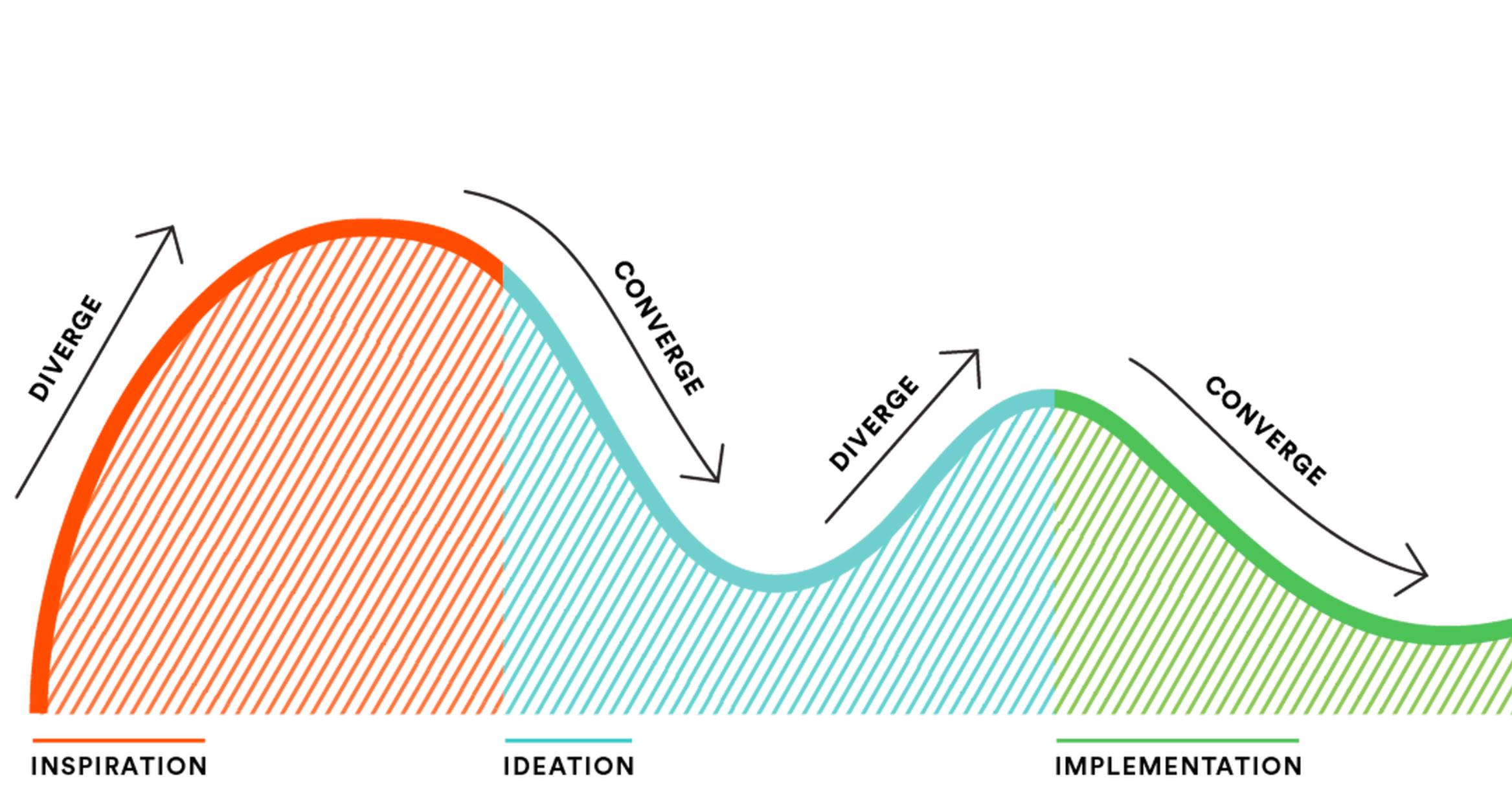


Take away messages

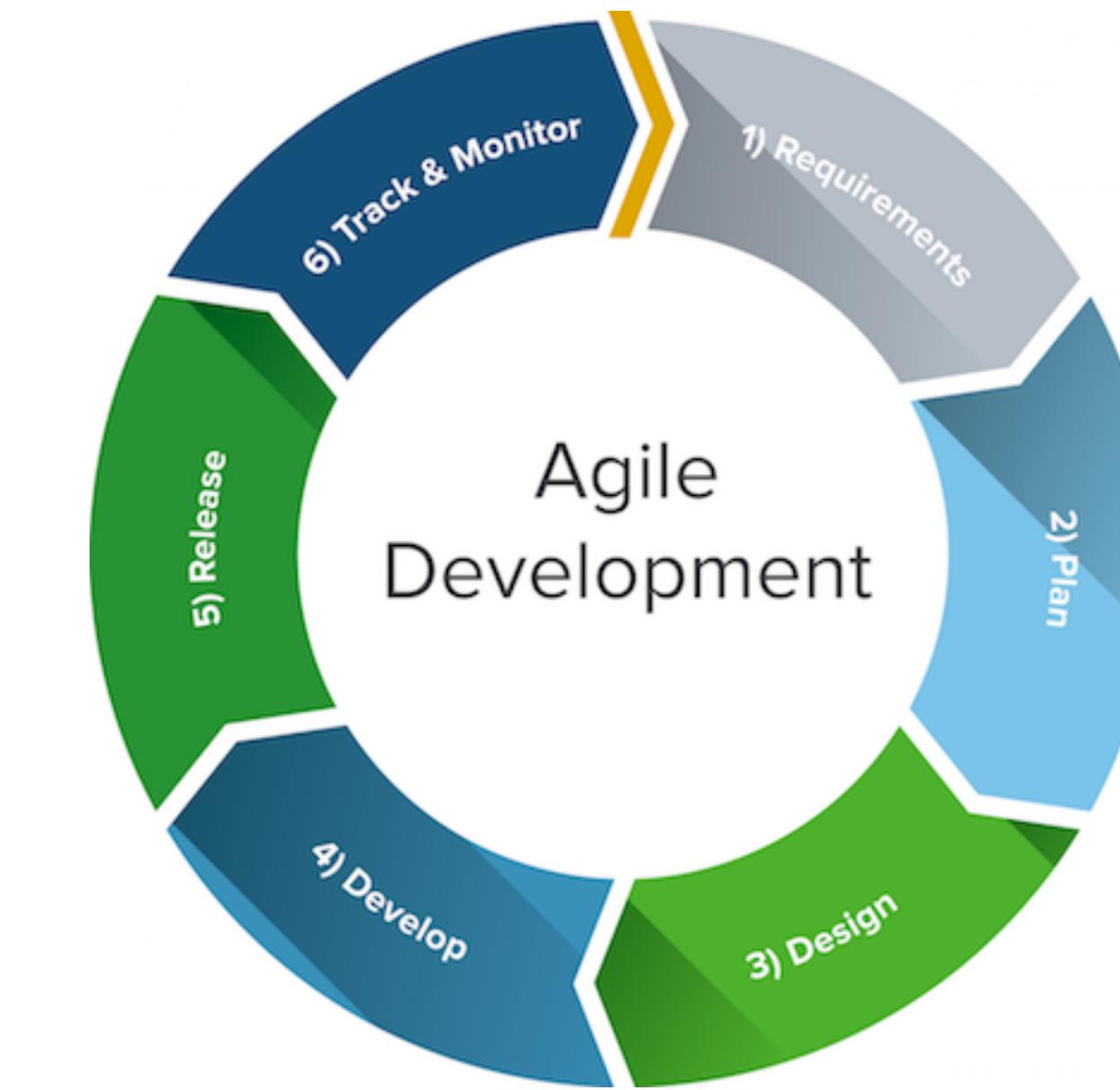
- Search before you build
- Build by example
- Build for accessibility
- Build with caution
- Build on a solid foundation

Applying this course in practice

Product design process

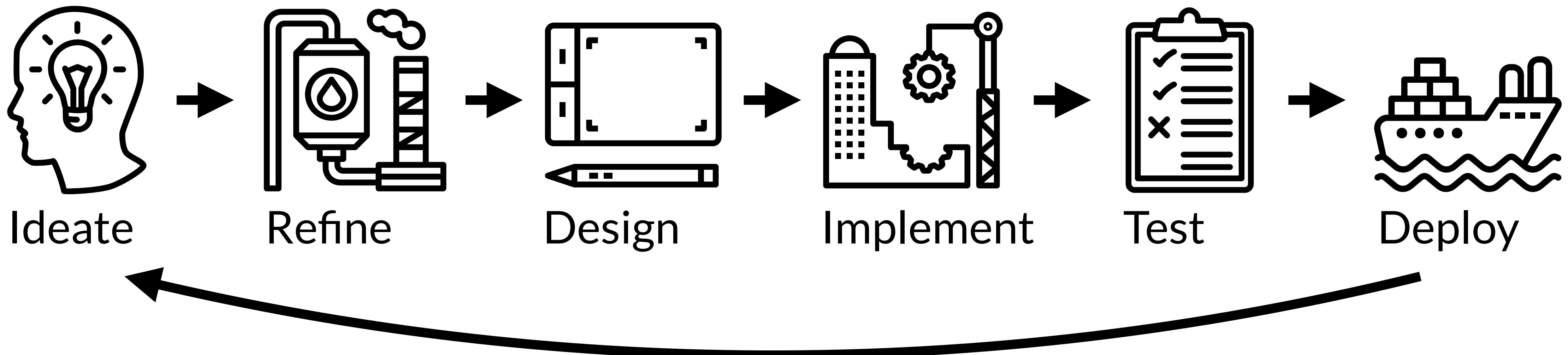


Human-Centered Design, IDEO



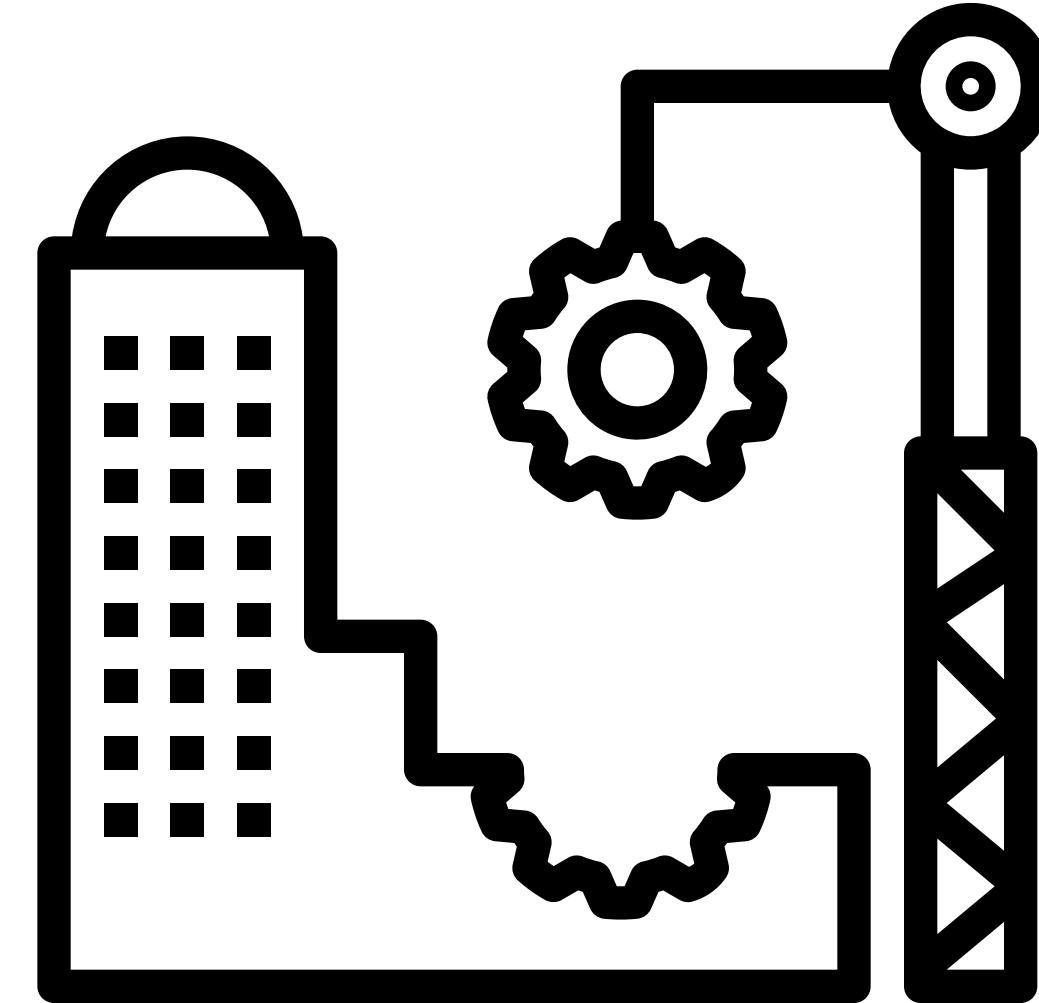
Agile Development, Agile Manifesto

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!
- Achieve conversational knowledge of 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...

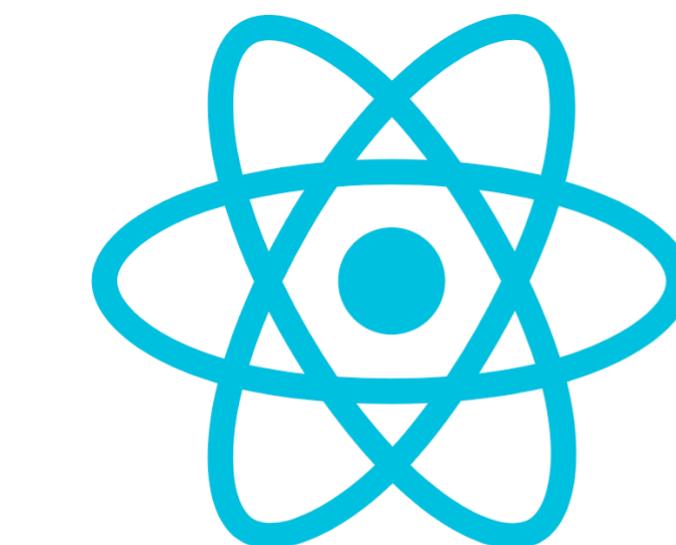
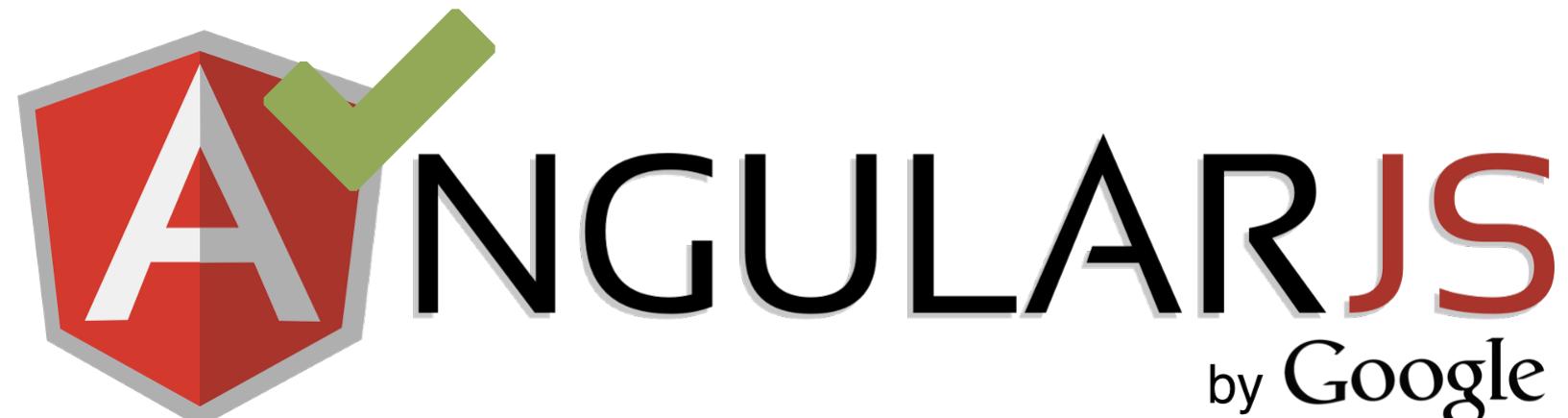
- Process and analyze data
 - Retrieve it from an API
 - Parse and process it to answer your question
- Visualize data
 - Use an appropriate tool for the task

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?

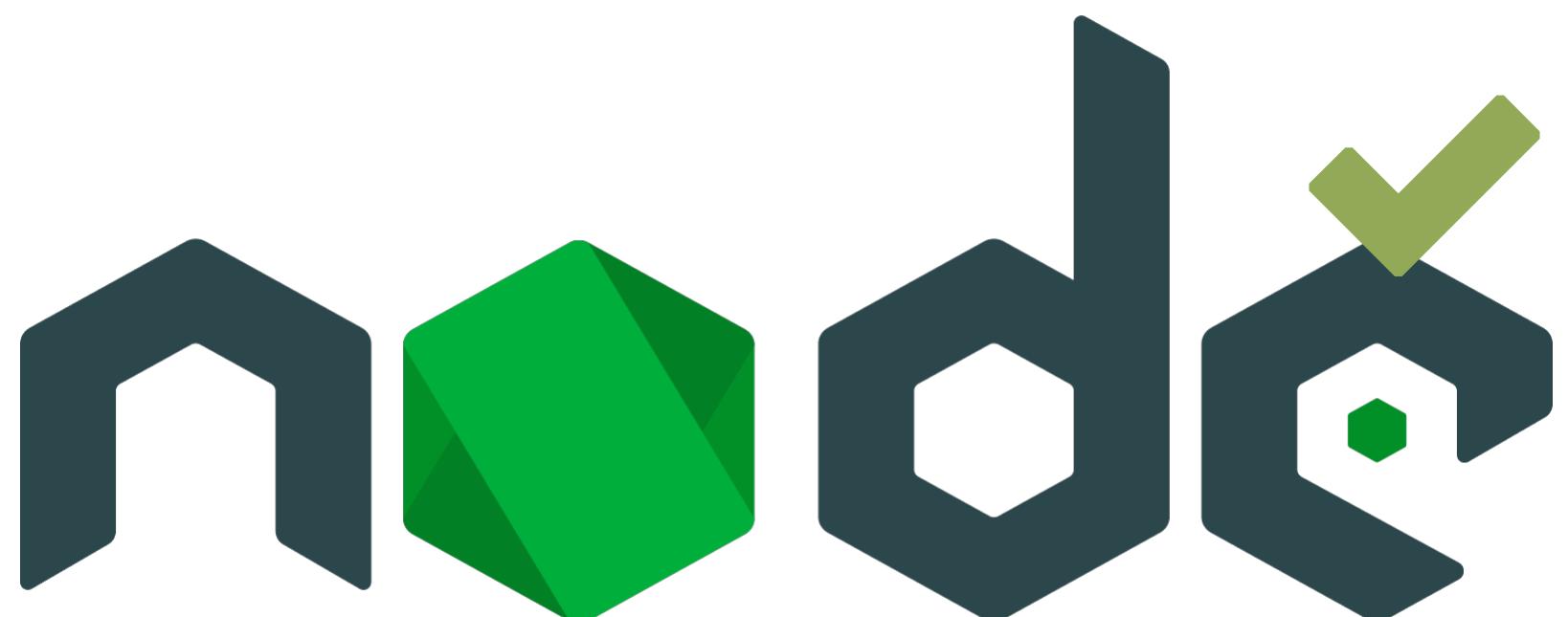
Often HTML, CSS, and JavaScript



React JS



Bootstrap



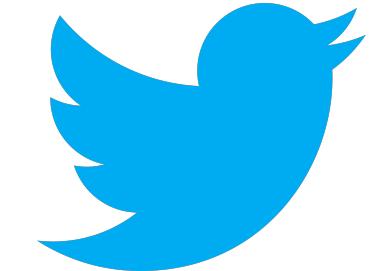
ember

Assignments

- A1: Personal web portfolio



- A2: Programming on the web



- A3: Web frameworks



- A4: Mobile development



Other skills

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

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...

- 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 20:
Beyond Web & Mobile
and Wrap-Up

Professor Daniel A. Epstein
TA Lucas de Melo Silva
TA Jong Ho Lee