



CS-311: The Software Enterprise

Prof. George Canea

School of Computer & Communication Sciences

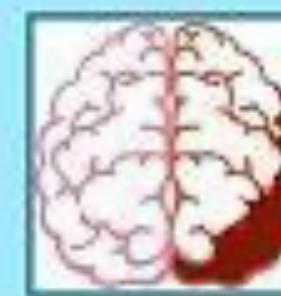




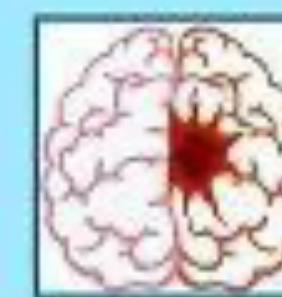
HUMAN BRAIN DISEASES



Multiple sclerosis



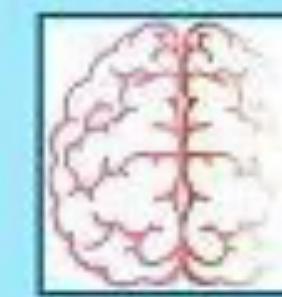
Cerebral contusion



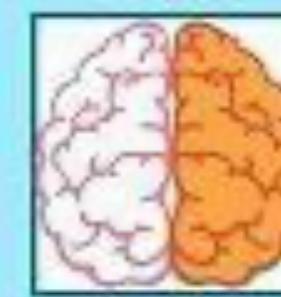
Brain Cancer



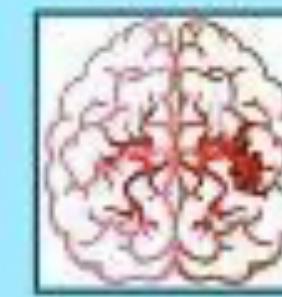
Alzheimer's disease



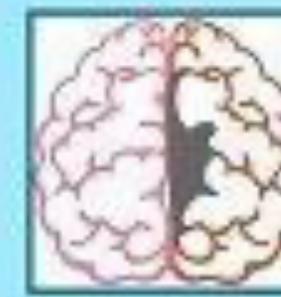
Dementia



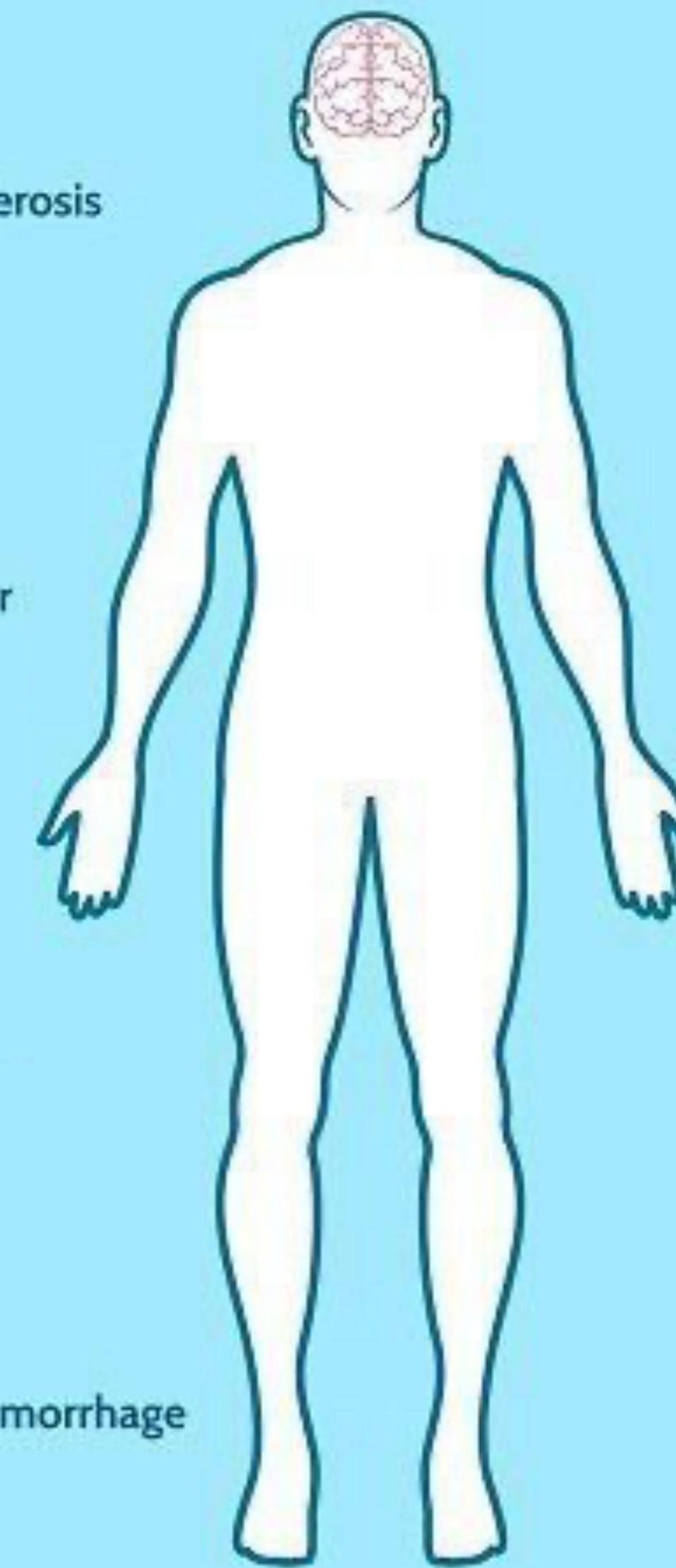
Encephalitis

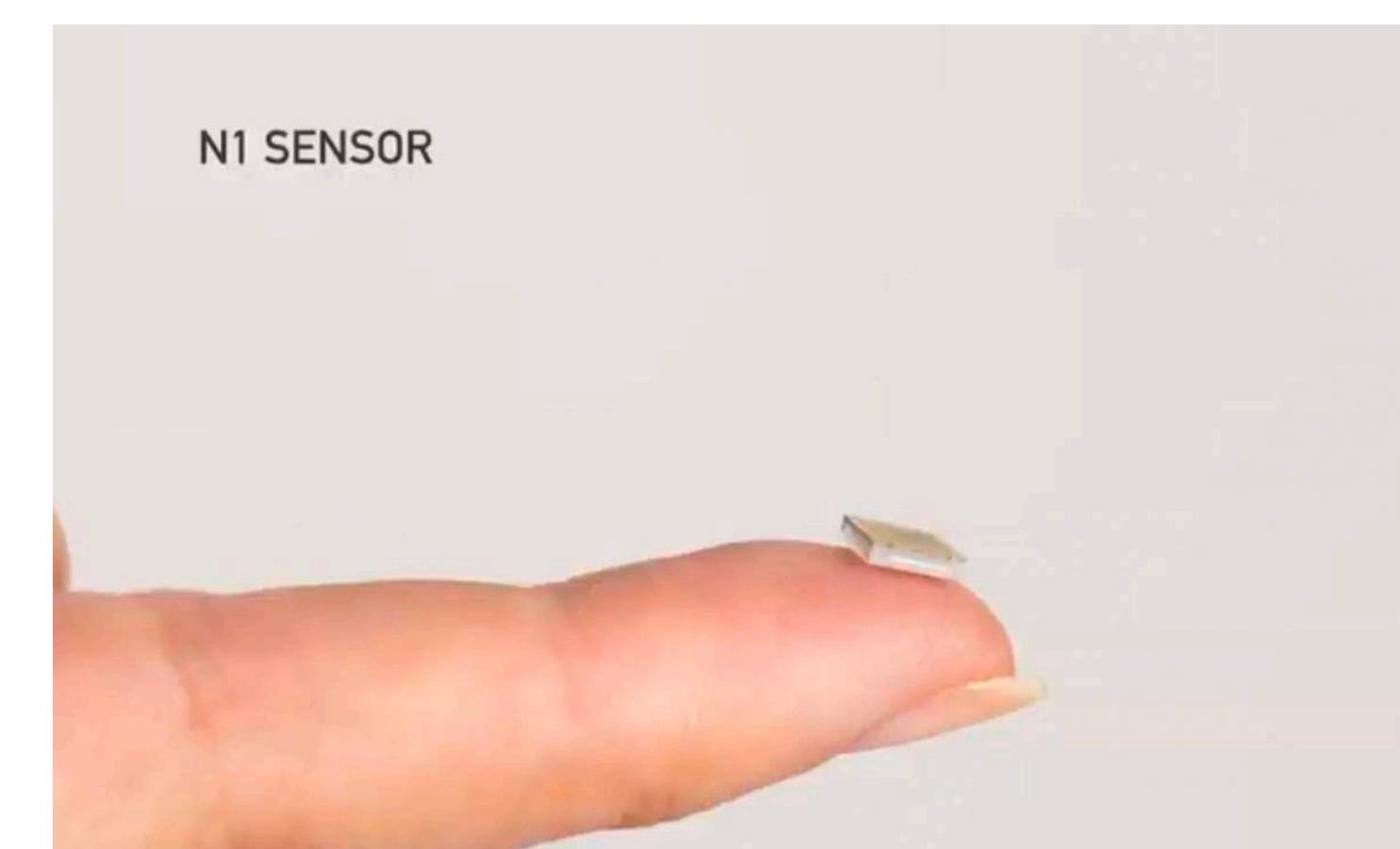
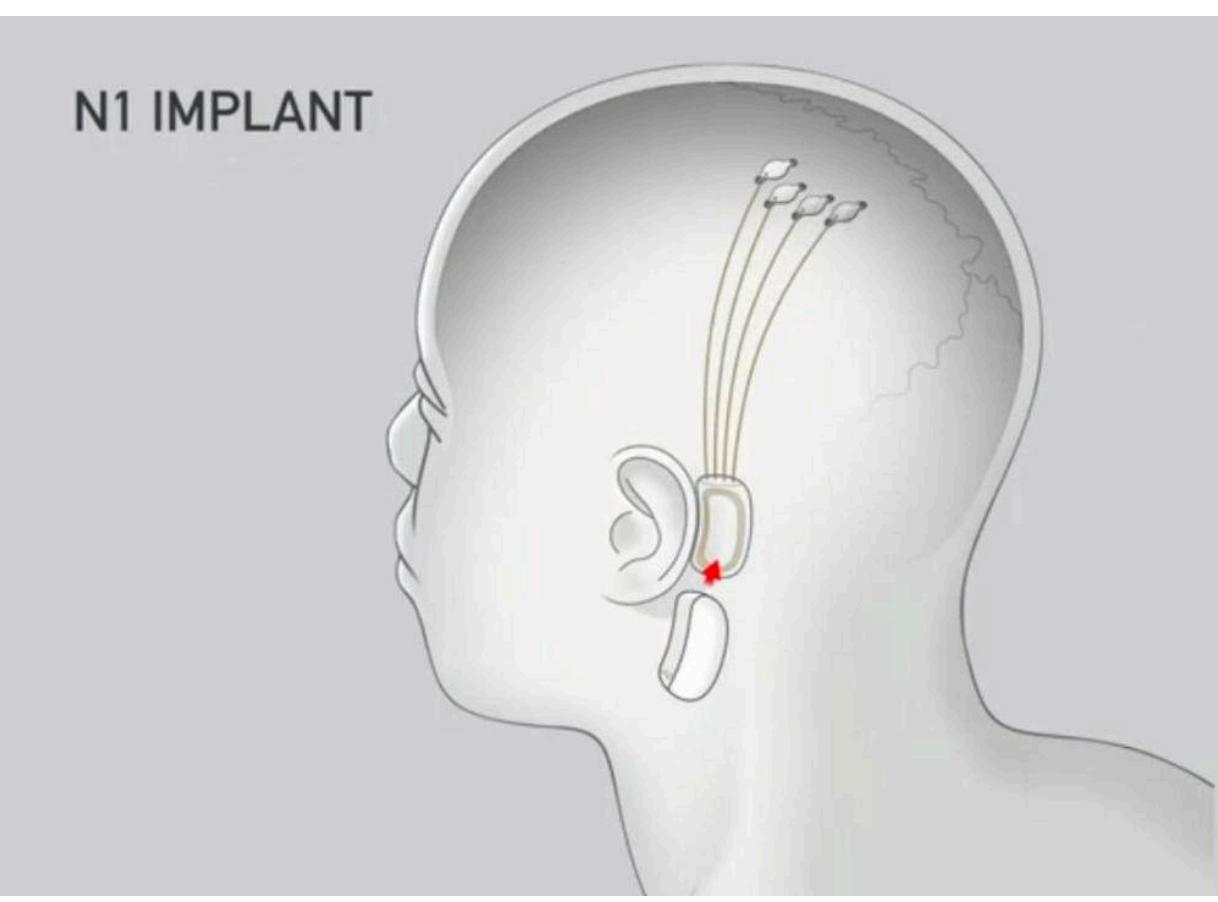
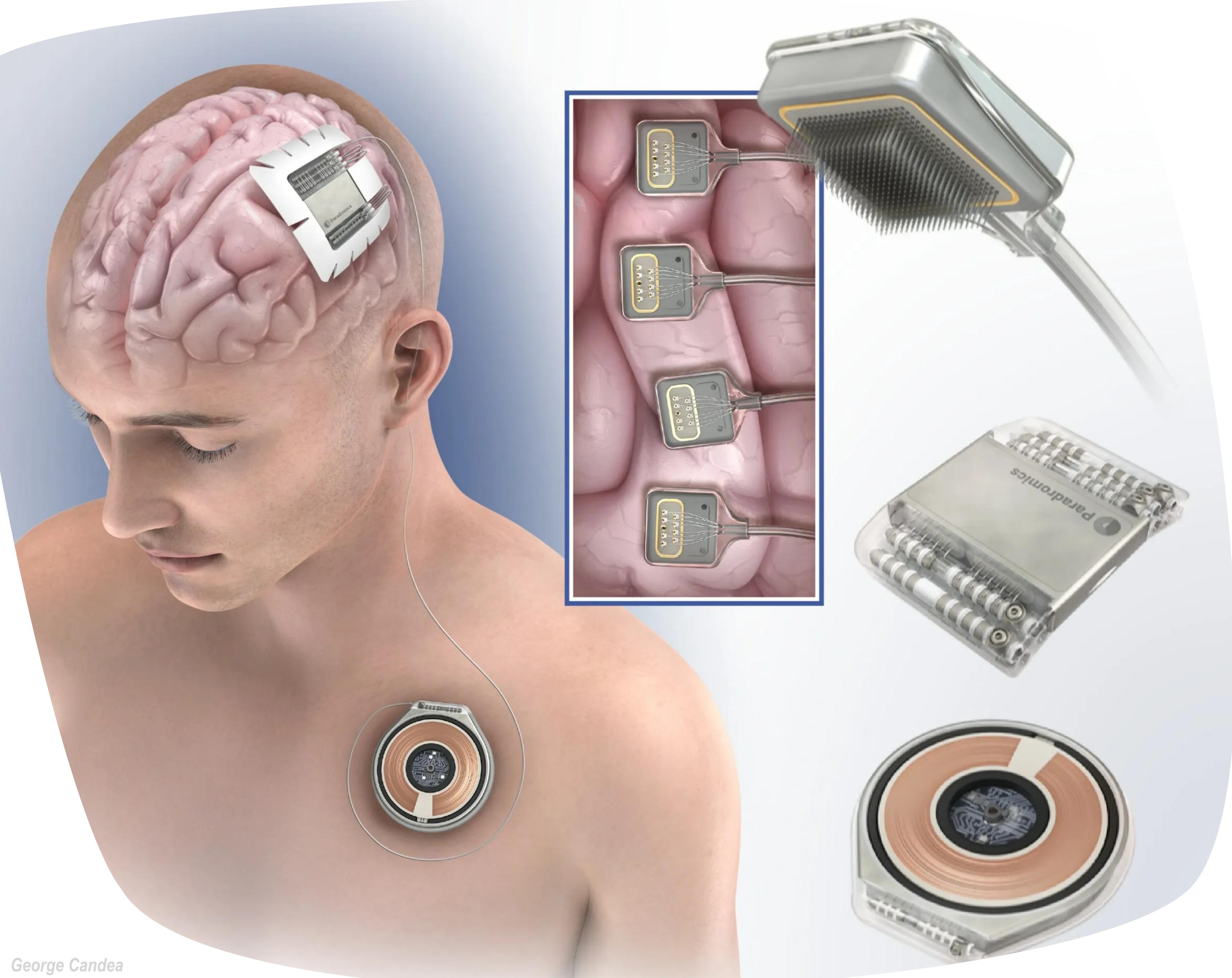


Cerebral Hemorrhage



Huntington's disease





Software makes life better...

- Voice assistants
- Deep learning
- Wearables
- Augmented reality (AR) and Virtual reality (VR)
- ...

Contributions to Percent Change in Real
Gross Domestic Product, 2021-2023

Software	0.8
Equipment	0.58
Structures	0.21

Source: Bureau of Economic Statistics

WSJ

AR Is Coming to the OR

Startups are designing augmented-reality systems to give doctors 'X-ray vision' into patients on the table. Medivis has a system cleared for planning surgeries, while Augmedics is already marketing its own for spinal procedures

By [Brian Gormley](#)

June 13, 2023 at 6:00 am ET | [WSJ PRO](#)

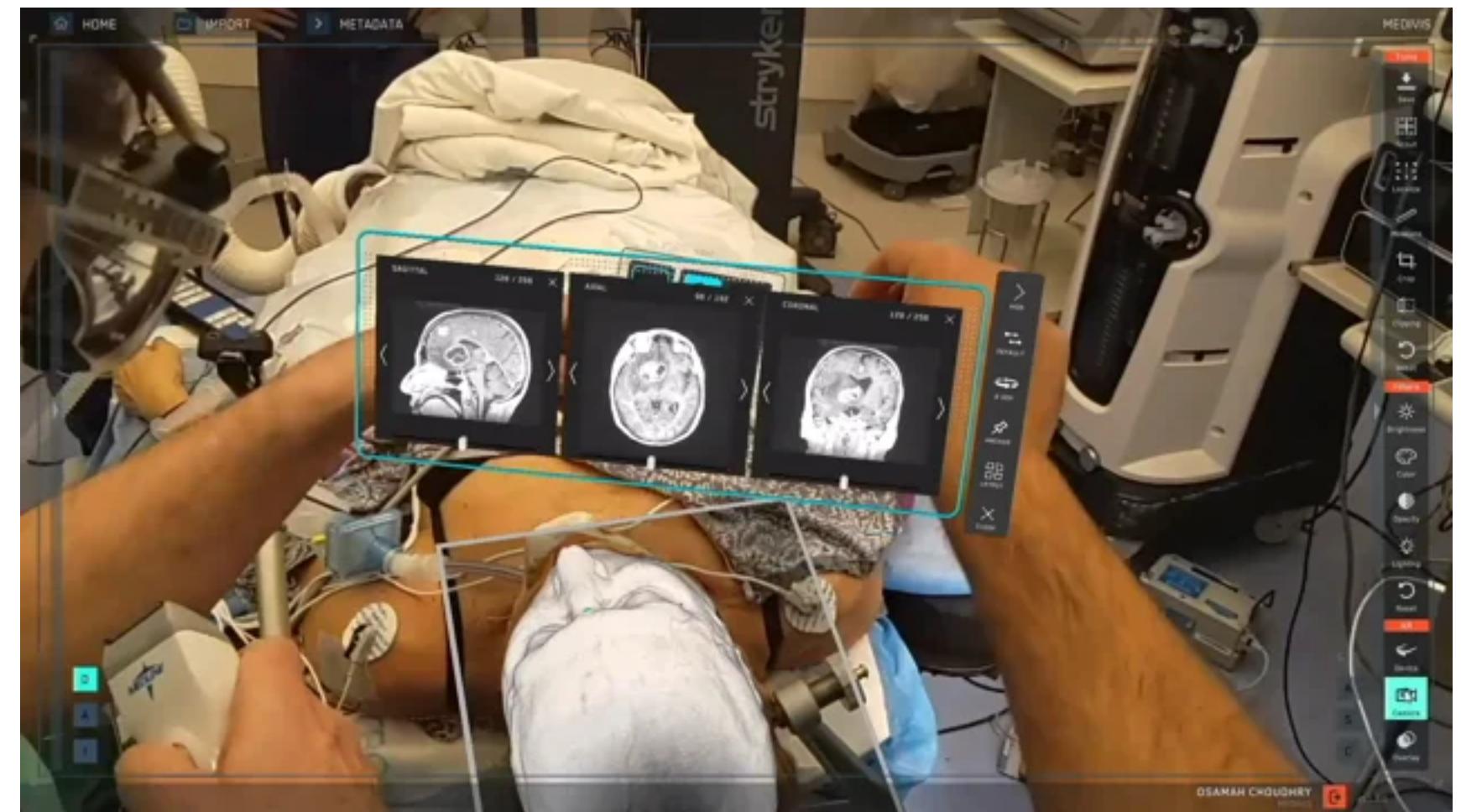
Advancements In AR Surgery: Transforming Healthcare And Saving Lives



Steven Cvetko Forbes Councils Member
Forbes Technology Council
[COUNCIL POST](#) | Membership (Fee-Based)



Oct 12, 2023, 09:00am EDT



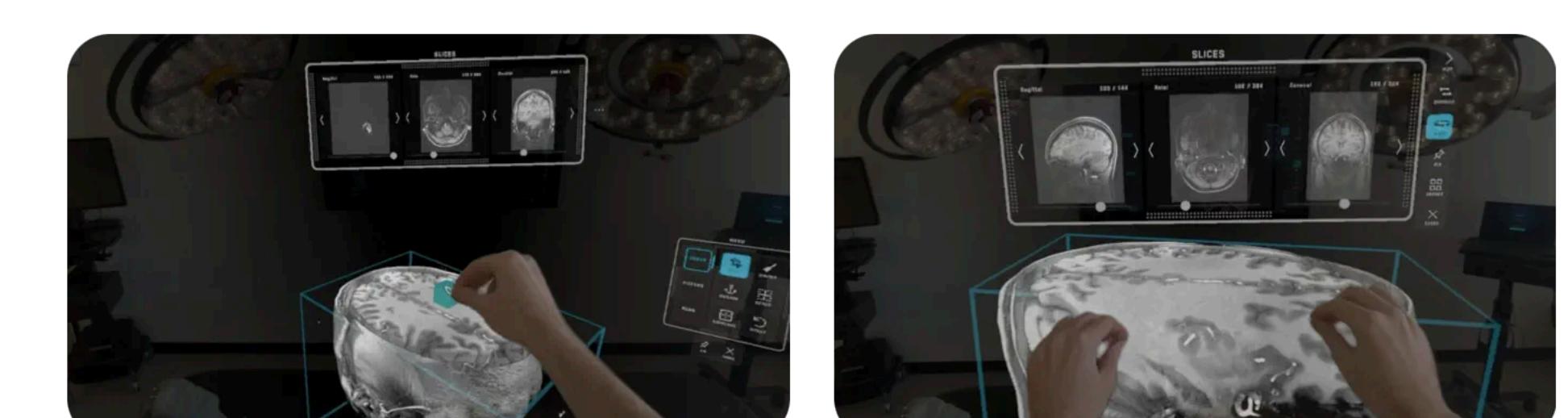
App Store Preview

SurgicalAR Vision (17+)
Medivis, Inc

Free



Screenshots

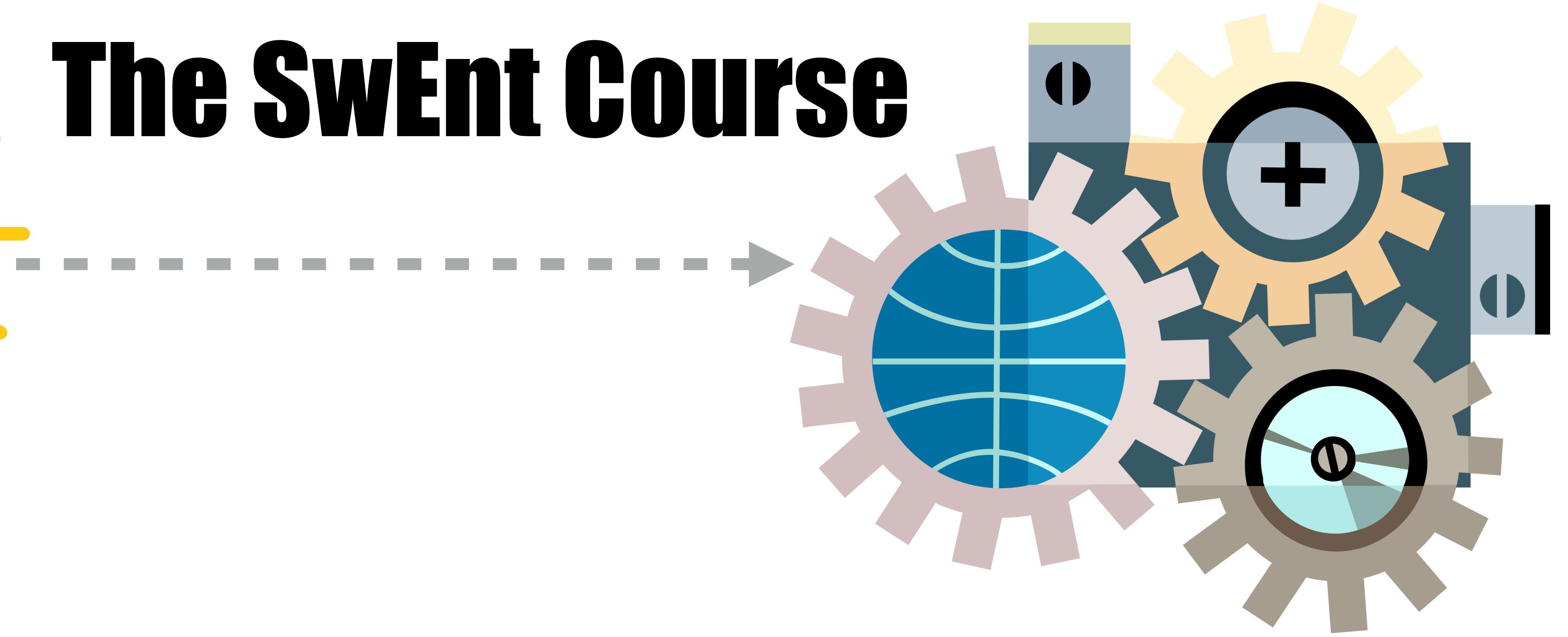


SurgicalAR uses AR, AI, and computer vision to enhance surgical precision, turning 2D medical imaging data (MRI/CT) scans into 3D, interactive visualization.

SurgicalAR Vision leverages spatial computing and artificial intelligence to improve surgical care. SurgicalAR Vision utilizes Apple Vision Pro for advanced visualization of medical imaging data and other healthcare data. It includes features for image review, image manipulation, measurement, and three-dimensional visualization (multi-planar reconstructions and volume rendering).



The SwEnt Course



Course Objectives

Get the skills to build effective software-based solutions in teams, either within an existing organization or as founders of your own organization

- How to write requirements and specifications
- Validation, testing, and debugging
- DevOps: Continuous integration, version control, project and issue tracking
- Behavior-driven and test-driven development
- Agile development processes
- Mobile-platform architecture
- Optionally...
 - How to scale to millions of users
 - Differentiation and value proposition / opportunity assessment
 - MVP and product roadmap
 - Business models

CS-311 (The Software Enterprise)

CS-202 (Computer Systems)
CS-214 (Software Construction)

CS-107 (Introduction to Programming)
CS-108 (Practice of Obj-Oriented Programming)

What to Expect

- SwEnt will take you out of your comfort zone
- Lots of work, highly variable (3x) across teams and students
- Prerequisites are crucial
- Semester schedule at a glance
 - *Weeks 1-3: lectures on software development 2x/week + bootcamp*
 - *Weeks 4-14: project + weekly meetings with coaching team*
 - *Week 14: Battle of the Apps*

Your Teaching Team

Prof. George Candea

- Co-founder of **aster data** (now )
 - more data. big insights. —
 - CTO, then Chief Scientist
- Co-founder of  **cyberhaven**
 - CEO, then Chief Scientist
- SW engineer/research   

- PhD  **Stanford**

- Bachelors and Master



**Massachusetts
Institute of
Technology**



Outline

- The Course
- Tools of the Trade
- Requirements (User Stories, Epics, ...)
- Your Journey in SwEnt
- Succeeding in SwEnt

Tools of the Trade

Writing Code

- Code Editors and IDEs
 - *Productivity, debugging, project management*
 - *Examples: Visual Studio Code, IntelliJ IDEA, Android Studio*
- Version Control Systems
 - *Collaboration, history, branching and merging*
 - *Examples: Git (and platforms like GitHub, GitLab, and Bitbucket)*
- Development Frameworks and Libraries
 - *Faster development, best practices, community support*
 - *Examples: React, Angular, Spring Boot, Android Jetpack – you will use this a lot !*

Getting Code Ready for Production

- Continuous Integration and Continuous Deployment (CI/CD)
 - *Automation: building, testing, and deployment with minimal manual intervention*
 - *Quality assurance: automated tests on every commit → catch bugs early*
 - *Frequent and reliable code deployments to production → faster release cycle*
 - *Examples: GitHub Actions, Jenkins, GitLab CI/CD*
- Code Quality/Security and Performance Monitoring
 - *Clean codebase, identify security vulns, insights into technical debt, real-time insights into app performance, system health, and user behavior — flag anomalies*
 - *Examples: Sonar Cloud, Prometheus, Grafana*

Deploying Code

- Containerization and Orchestration
 - *Consistency from dev to production (no "it works on my machine" complaints)*
 - *Efficiency by better utilizing hardware resources*
 - *Makes deployment simpler and more reliable*
 - *Examples: Docker, Kubernetes for automating deployment/scaling/management*
- Cloud Platforms and Services
 - *Managed: services for computing, databases, analytics, etc. run by specialists*
 - *Scalability: applications can scale elastically*
 - *Global performance and availability*
 - *Examples: Amazon AWS, Google Cloud Platform (GCP), Microsoft Azure*

Working in a Team

- Collaboration and Project Management
 - *Help team organize tasks, track progress, and manage project timelines*
 - *Communication: real-time communication and discussion, file sharing, ...*
 - *Integration with dev tools → bring together project management and dev workflows*
 - *Examples: Jira, Trello, Slack*

Tools: Recap

- Code Editors and IDEs
- Version Control Systems
- Development Frameworks and Libraries
- Continuous Integration and Continuous Deployment (CI/CD)
- Code Quality/Security and Performance Monitoring
- Containerization and Orchestration
- Cloud Platforms and Services
- Collaboration and Project Management

A New Tool: The AI Coding Assistant

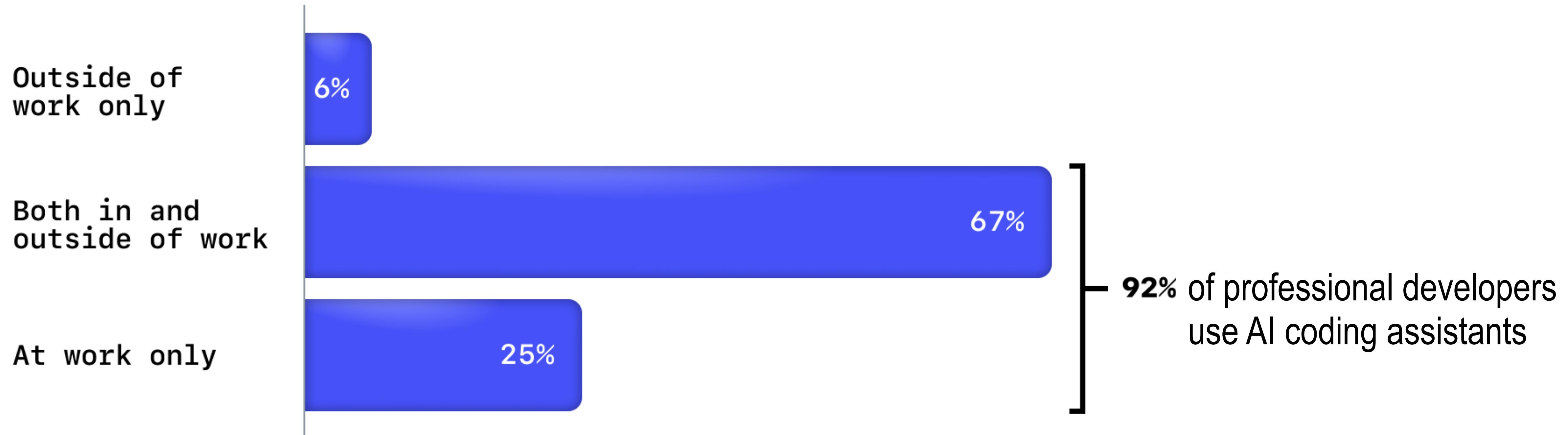


Created with Dall-E

https://d3eekel6mvl7.cloudfront.net/sites/default/files/styles/article_hero_image/public/6-hot-industries-for-software-engineering-careers.jpg?itok=z4yYM5Oz

Where AI coding tools are used

N=500



GitHub survey of 500 developers at companies with >1,000 employees (updated February 7, 2024)
<https://github.blog/news-insights/research/survey-reveals-ais-impact-on-the-developer-experience/>

AI Coding Assistants: What? How?

- AI coding assistant = tool that understands and generates code
- It's not magic !
 - *ML models get trained on vast amounts of code, docs, forums, and other resources*
 - *models learn patterns, coding styles, and best practices*
 - *use NLP to understand comments, documentation, and code semantics*
 - *analyze codebase context to make answers relevant and syntactically correct*
- Some AI coding assistants can learn from the interaction with the developers



AI Assistants: Brainstorming



<https://www.datacamp.com/blog/a-chat-with-chatgpt-on-the-method-behind-the-bot>

DA You

What are the most common needs for students in a university that could be translated into a digital product?

SpeedGPT

1. Study tools (e.g., flashcards, note-taking apps)
2. Time management (e.g., planners, calendars)
3. Online tutoring or peer help platforms
4. Textbook and literature resources
5. Career and internship finders
6. Mental health and wellness apps
7. Financial planning and budgeting tools
8. Housing and roommate finders
9. Campus navigation and event apps
10. Social networking platforms for university communities

DA

You

Develop number 6, give ideas for phone apps, functionalities. Make sure it has an accounts system and integrates a map feature

SpeedGPT

1. Mood Tracker & Wellness Journal

- **Accounts System:** Personalized profiles with privacy settings.
- **Functionalities:** Daily mood tracking, wellness activities suggestions, journaling with prompts, goal setting, and progress tracking.
- **Map Feature:** Locate nearby mental health resources, wellness centers, and support groups.

2. Mindfulness & Meditation Guide

- **Accounts System:** Customizable user experiences based on preferences and history.
- **Functionalities:** Guided meditations, mindfulness challenges, breathing exercises, personalized meditation paths.
- **Map Feature:** Find local meditation and yoga classes or retreats.

3. Stress Management Toolkit

- **Accounts System:** User profiles to track stress levels over time and recommend activities.
- **Functionalities:** Stress assessment quizzes, relaxation techniques, educational content on stress management, and community support forums.
- **Map Feature:** Integrates with map to suggest quiet, calming spaces on campus or nearby parks for relaxation.

4. Healthy Lifestyle Planner

- **Accounts System:** Profiles for tracking fitness, nutrition, and sleep patterns.
- **Functionalities:** Meal planning, workout routines, sleep quality tracking, water intake reminders, and healthy habit coaching.
- **Map Feature:** Shows healthy eateries, gyms, and sleep-friendly zones nearby.

5. Peer Support Network

- **Accounts System:** Allows for anonymous or profile-based participation in forums

AI Assistants: Prototyping



GitHub
Copilot

```
/*
 * User constructor. It can be constructed by passing
 * the user ID, or optionally a username.
 * @param int $id User ID
 * @param string $username Username (if this is not null,
 * ID will be ignored and username is used to fetch user)
 * @throws Exception If user not found
 */
public function __construct($id, $username = null) {
    global $db;

    $id = make_sql_safe($id);
    if ($username !== null) {
        $username = make_sql_safe($username);
        $sql = "SELECT *, UNIX_TIMESTAMP(created_time) AS created_time FROM users WHERE username = '$username'";
    } else {
        $sql = "SELECT *, UNIX_TIMESTAMP(created_time) AS created_time FROM users WHERE user_id = $id";
    }

    // Check if result, if not throw exception
    $result = mysqli_query($db, $sql);
    if (!$result OR !mysqli_num_rows($result) > 0) {
        throw new Exception("User not found");
    }

    $user = mysqli_fetch_assoc($result);
}
```

Developer

AI

AI Assistants: Prototyping



GitHub
Copilot

```
/*
 * User constructor. It can be constructed by passing
 * the user ID, or optionally a username.
 * @param int $id User ID
 * @param string $username Username (if this is not null,
 * ID will be ignored and username is used to fetch user)
 * @throws Exception If user not found
 */

public function __construct($id, $username = null) {
    global $db;

    $id = make_sql_safe($id);
    if ($username !== null) {
        $username = make_sql_safe($username);
        $sql = "SELECT *, UNIX_TIMESTAMP(created_time) AS created_time F
    } else {
        $sql = "SELECT *, UNIX_TIMESTAMP(created_time) AS created_time F
    }

    // Check if result, if not throw exception
    $result = mysqli_query($db, $sql);
    if (!$result OR !mysqli_num_rows($result) > 0) {
        throw new Exception("User not found");
    }

    $user = mysqli_fetch_assoc($result);
}
```

Developer

```
/*
 * Delete image from database and file system
 */
public function delete() {
    // Delete the file
    $path = "/img/uploads/file/original/".$this->img_orig;
    if(!(@unlink($path))) {
        throw new Exception("Could not delete ORIGINAL file at ".$path);
    }
}
```

Dev // Delete the thumb

```
$path = "/img/uploads/file/thumb/".$this->img_thumb;
```

AI

AI Assistants: Explain Code

You
What does this code do? My friend totally forgot to document what it does:
```  
list(\$x, \$y, \$type) = getimagesize(\$file\_path);  
  
if (\$x > \$a || \$y > \$b) {  
 \$ratio = \$x / \$y;  
 if (\$ratio > 1) {  
 \$x = \$a;  
 \$y = \$a / \$ratio;  
 } else {  
 \$x = \$b \* \$ratio;  
 \$y = \$b;  
 }  
}  
...  
```

SpeedGPT
This code resizes an image to fit within specified dimensions (`\$a` for width, `\$b` for height) without altering its aspect ratio. It first retrieves the image's original dimensions (`\$x` for width, `\$y` for height) and type using `getimagesize()`. If the image's width or height exceeds the specified dimensions, it calculates the new size to ensure the image fits within these constraints while maintaining its original aspect ratio.



Dev // This function will resize the image to fit the given dimensions
if (\$x > \$a || \$y > \$b) {
 \$ratio = \$x / \$y;
 if (\$ratio > 1) {
 \$x = \$a;
 \$y = \$a / \$ratio;
 } else {
 \$x = \$b * \$ratio;
 \$y = \$b;
 }
}



Dev // This function will resize the image to fit the given dimensions
// by keeping the aspect ratio. If the image is smaller than the
if (\$x > \$a || \$y > \$b) {
 \$ratio = \$x / \$y;
 if (\$ratio > 1) {
 \$x = \$a;
 \$y = \$a / \$ratio;
 } else {
 \$x = \$b * \$ratio;
 \$y = \$b;
 }
}

AI

AI Assistants: Debug / Refactor / Collaborate

- Debug and solve problems
 - *Can flag and suggest fixes for security bugs, performance issues, etc.*
- Refactor code
 - *= improving structure and quality of existing code without changing external behavior*
 - *AI can suggest opportunities to reduce redundancy, make code more efficient, readable and maintainable, more aligned with best practices, etc.*
- Enhance collaboration
 - *Can train a custom model on your code base, and enable it to (better) answer questions about the code for new team members, etc.*

Think of the AI coding assistant
as a smart summer intern



<https://tech.co/news/child-destined-computer-programmer-2016-11>

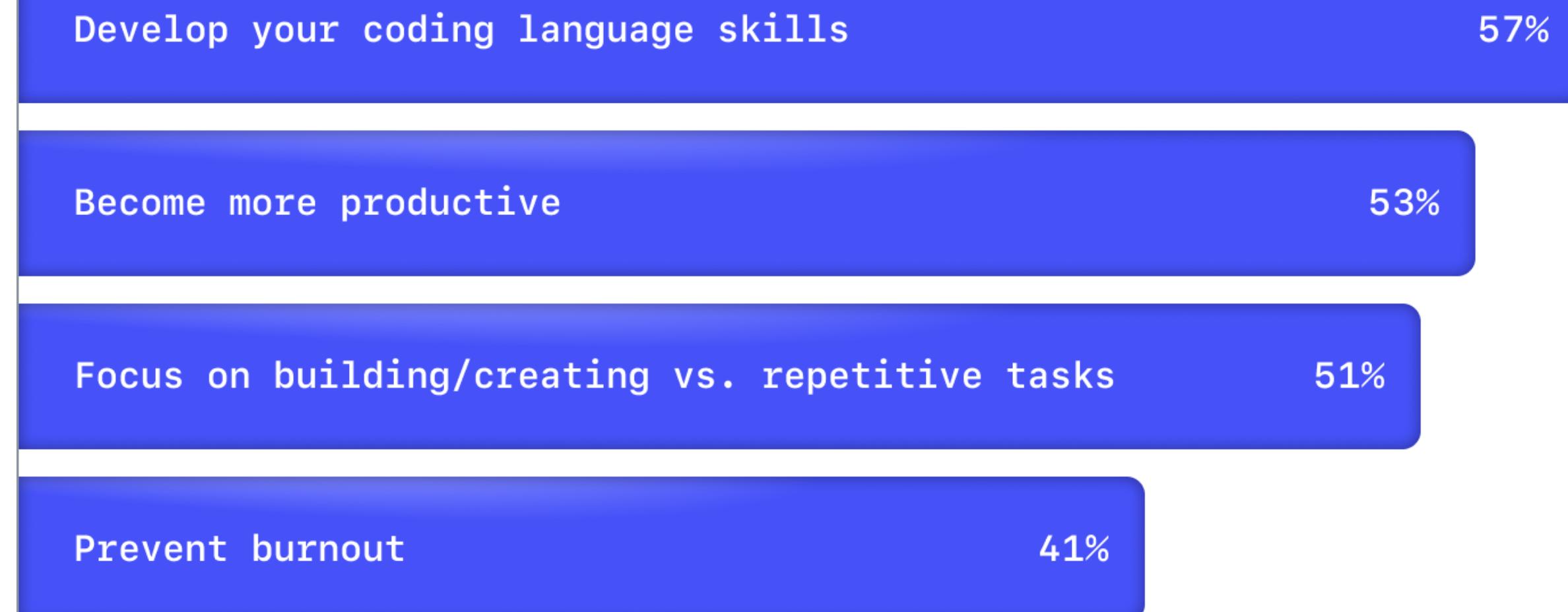
Example Workflow To Try Out

as of 2024 !

1. Articulate a high-level project description
2. Design the overall architecture
 - *classes/modules/packages, key methods, protocols and contracts, etc.*
 - **⚠ AI cannot design the architecture of a serious software product**
3. Give to an AI and ask for critique, alternatives, missing aspects, etc.
 - **⚠ Tell it that it is another AI that came up with the design**
4. Break implementation down into small, individual tasks → give a task to AI #1
 - **⚠ Context helps improve the AI's performance, but too much context confuses it**
5. Give the result + requirements to AI #2 and ask for a critique
6. Combine the suggestions, review / refactor / fix the code
7. Ask an AI for a code review, then address feedback
8. Ask AI to write tests... repeat steps 5 - 7... then go back to step 4 ... done when no more tasks left

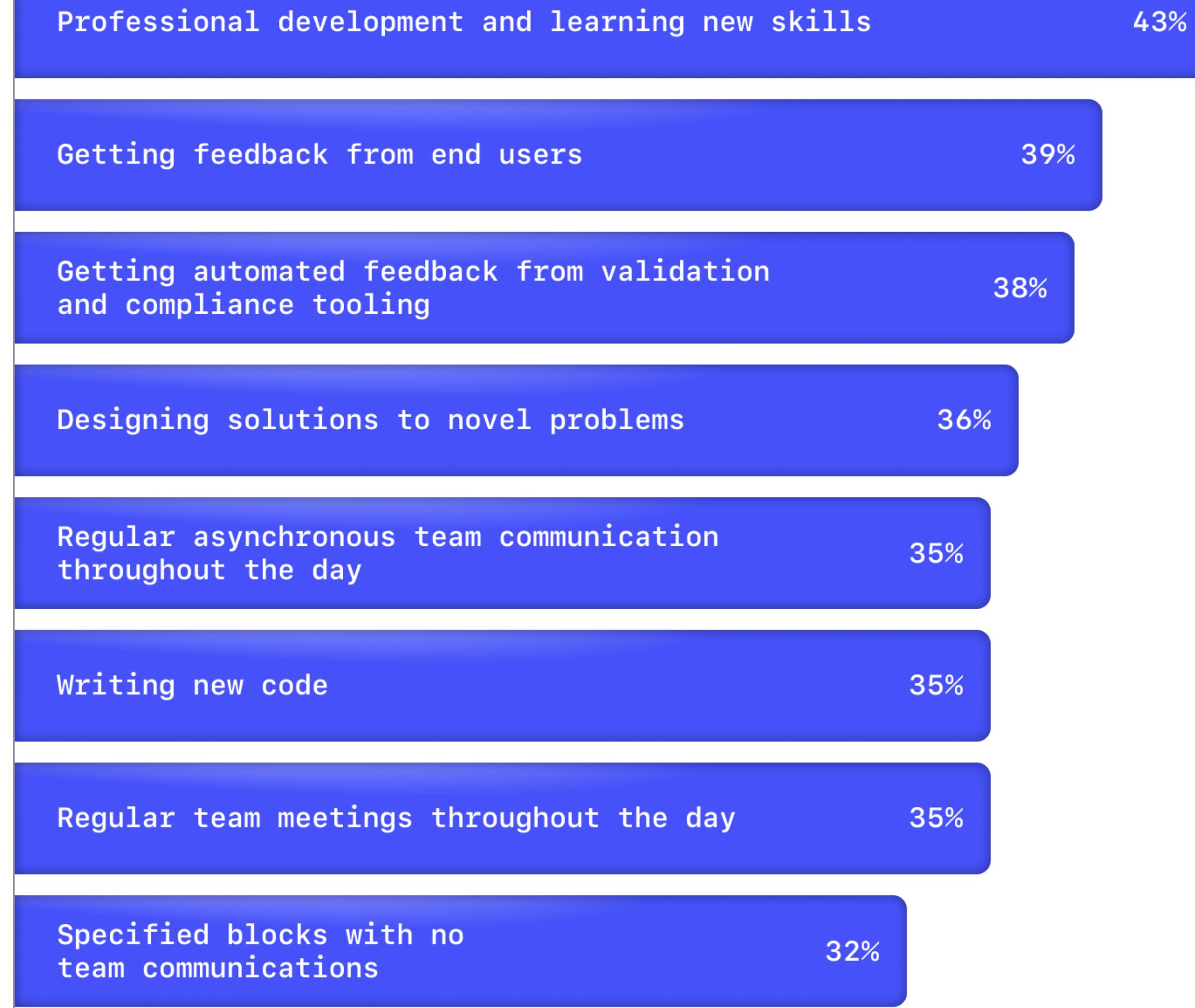
How AI coding tools can help with daily work

N=500



Things that positively impact workdays

Top responses shown, N=500



<https://github.blog/news-insights/research/survey-reveals-ais-impact-on-the-developer-experience/>

AI Assistants: Risks and Drawbacks

```
const input = document.querySelector('input');

// When pressing enter, the input value is logged
input.addEventListener('keypress', function(e) {
    if (e.keyCode === 13) {
        console.log(input.value);
    }
});
```

AI suggests using the keyCode property of the JS KeyboardEvent
This has been deprecated since 2015 !

Code-generating AI can introduce security vulnerabilities, study finds

Kyle Wiggers @kyle_l_wiggers 3:30 PM GMT+1 • December 28, 2022

```
<?php
// API file: Get the $_GET['username'] and perform a query to
// get the user's information
require_once('config.php');
```

```
if(isset($_GET['username'])){
    $username = $_GET['username'];
    $query = "SELECT * FROM users WHERE username = '$username'";
    $result = mysqli_query($conn, $query);
    $row = mysqli_fetch_assoc($result);
    echo json_encode($row);
}
else{
    echo "No username provided";
}
```

AI suggests code with a SQL injection vulnerability

AI Assistants: Risks and Drawbacks

- Over-reliance on AI → skill atrophy
 - *you must understand the underlying logic and not just accept suggestions blindly*
 - *ask yourself if you could've come up with the answer (at the cost of taking longer)*

AI Assistants: Risks and Drawbacks

- Over-reliance on AI → skill atrophy
 - *you must understand the underlying logic and not just accept suggestions blindly*
 - *ask yourself if you could've come up with the answer (at the cost of taking longer)*
- Homogenization of code
 - *widespread use of AI could lead to a homogenization of coding styles and solutions, stifling human creativity and innovation*
 - *you should use AI suggestions as starting points, not the final outcome*

AI Assistants: Ethical Problems



Tim Davis
@DocSparse

...

@github copilot, with "public code" blocked, emits large chunks of my copyrighted code, with no attribution, no LGPL license. For example, the simple prompt "sparse matrix transpose, cs_" produces my cs_transpose in CSparse. My code on left, github on right. Not OK.

The screenshot shows a Visual Studio Code interface with two code snippets side-by-side. On the left is a snippet of C code for sparse matrix transpose, labeled 'Untitled-1.cpp'. On the right is a snippet of C code for sparse matrix transpose, labeled 'GitHub Copilot X'. The GitHub Copilot code is identical to the code in Untitled-1.cpp, including variable names and comments. The GitHub Copilot interface shows a progress bar at the top indicating 'Synthesizing 10/10 solutions'.

```
Untitled-1.cpp
1 // sparse matrix transpose
2 cs_
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40

GitHub Copilot X
1 Synthesizing 10/10 solutions
2
3
4
5 Accept Solution
6 cs_transpose (const cs *A, int values)
7 {
8     int p, q, j, *Cp, *Ci, *w, *Ti, *Tj;
9     double *Cx, *Tx;
10    cs *T;
11    if (!CS_CSC (A) || !CS_TRIPLET (A)) return (NULL); // check inputs
12    T = cs_salloc (A->n, A->m, A->nzmax, values && A->x, 0);
13    w = cs_calloc (A->m, sizeof (int)); // get workspace
14    if (!T || !w) return (cs_done (T, NULL, w, 0)); // out of memory
15    Cp = T->p; Ci = T->i; Cx = T->x; Ti = A->i; Tj = A->j;
16    for (p = 0; p < A->nzmax; p++) w [Ti [p]]++; // column counts
17    cs_cumsum (Cp, w, A->m); // column pointers
18    for (j = 0; j < A->n; j++)
19    {
20        for (p = A->p [j]; p < A->p [j+1]; p++)
21        {
22            Ci [q = w [Ti [p]]++] = j; // place A(i,j) as entry C(j,i)
23            if (Cx) Cx [q] = Ax [p];
24        }
25    }
26    return (cs_done (T, NULL, w, 1)); // success; free w and return
27
28 // Language: cpp
29 // Compare this snippet from Untitled-1.cpp:
30 //
31 // cs *cs_transpose (const cs *A, int values)
32 // {
33 //     int p, q, j, *Cp, *Ci, *w, *Ti, *Tj;
34 //     double *Cx, *Tx;
35 //     cs *T;
36 //     if (!CS_CSC (A) || !CS_TRIPLET (A)) return (NULL); // check inputs
37 //     T = cs_salloc (A->n, A->m, A->nzmax, values && A->x, 0);
38 //     w =
39 // =====
```

Getting Started with AI Coding Assistants

- Play around with several different AI coding assistants
 - e.g., *Github Copilot, ChatGPT, Google Gemini, Visual Studio IntelliCode, DeepMind AlphaCode, TabNine, Amazon CodeWhisperer/CodeGuru, Meta's Code Llama, ...*
- GitHub Copilot is free to use for students, as are many of the above
- Follow the evolution of the technology
- Tutorials and online courses for learning
- Learn new use cases from others



GitHub Copilot is free to use for students, as are many of the above

Where team should focus most time if the team used AI coding tools

Top responses shown, N=500



<https://github.blog/news-insights/research/survey-reveals-ais-impact-on-the-developer-experience/>

Outline

- The Course
- Tools of the Trade
- Requirements (User Stories, Epics, ...)
- Your Journey in SwEnt
- Succeeding in SwEnt

Requirements



What is a requirement ?

- Functional vs. non-functional requirements

What is a requirement ?

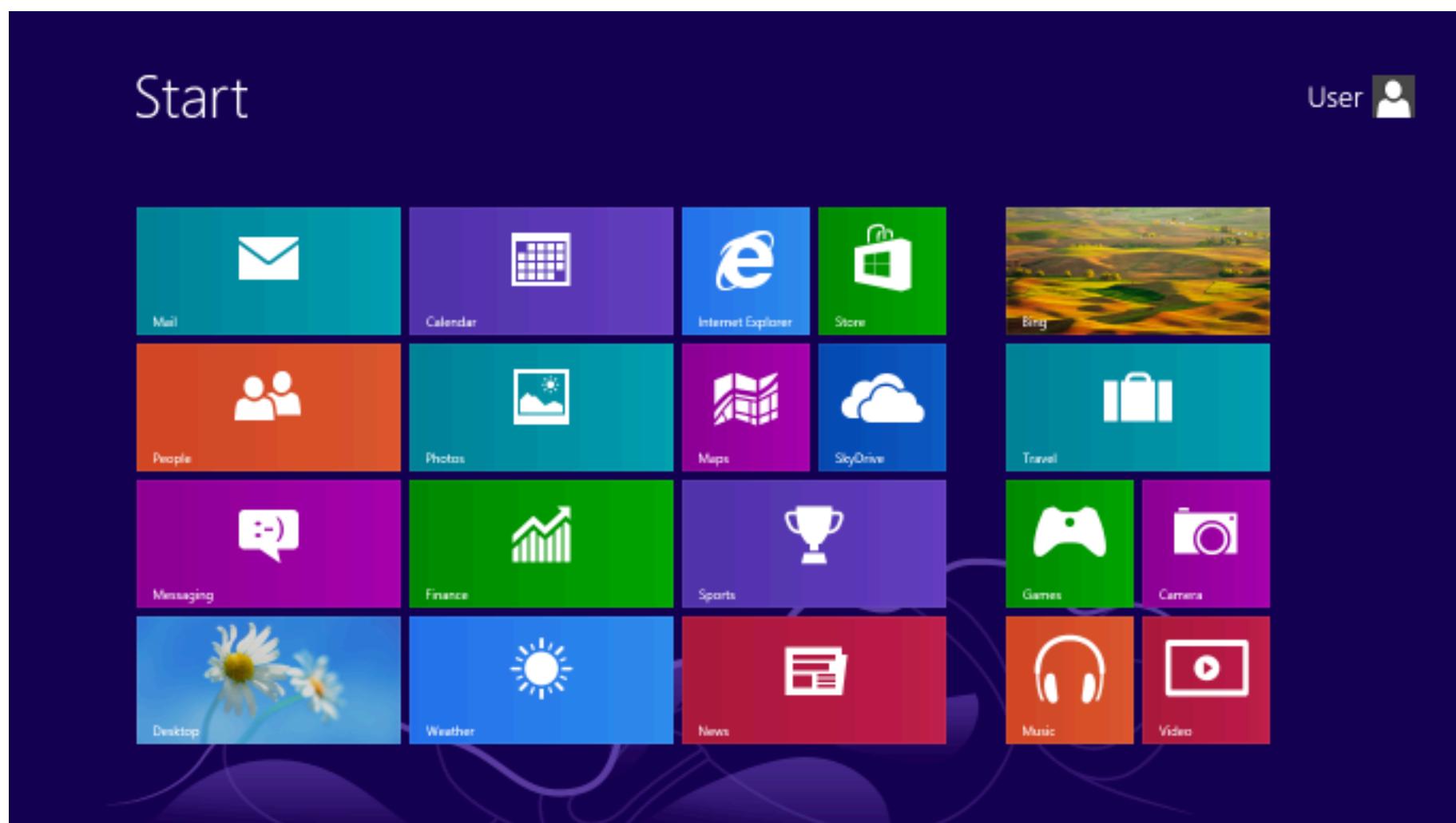


What is a requirement ?

- Functional vs. non-functional
 - *{features, functions}* vs. *{performance, security, reliability, maintainability, ...}*
- Requirement expresses a need, not an implementation
- Not listening to users (or the data) can be expensive

What is a requirement ?

- Functional vs. non-functional
 - *{features, functions}* vs. *{performance, security, reliability, maintainability, ...}*
- Requirement expresses a need, not an implementation
- Not listening to users (or the data) can be expensive



What is a requirement ?

- Functional vs. non-functional
 - *{features, functions}* vs. *{performance, security, reliability, maintainability, ...}*
- Requirement expresses a need, not an implementation
- Not listening to users (or the data) can be expensive
- User wants \neq user needs

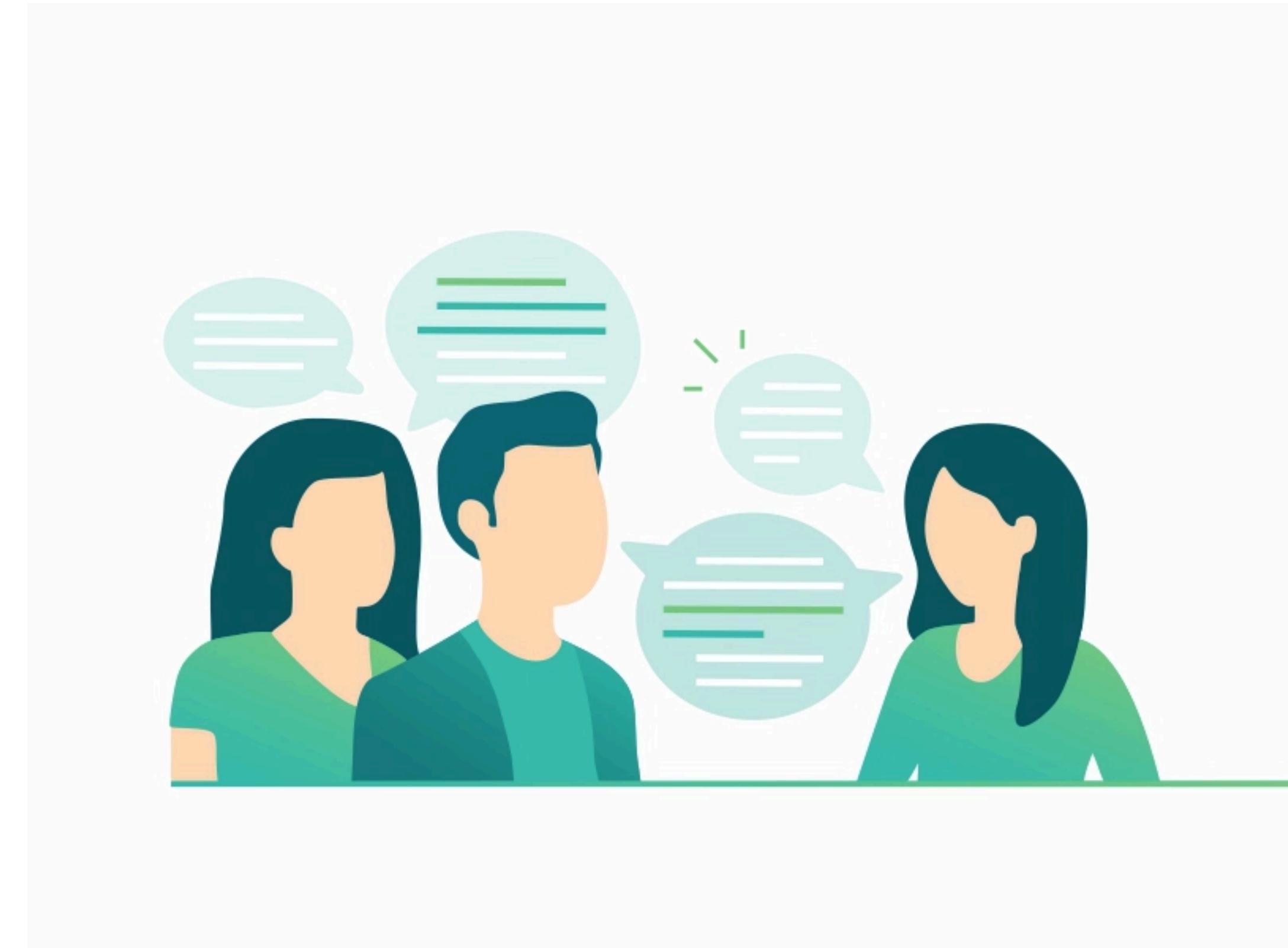
If I had asked people what they wanted, they would have said faster horses.

Henry Ford



How do we define requirements ?

- Most users are not precise and objective
- Elicit requirements by talking & listening to (potential) users



How do we define requirements ?

- Most users are not precise and objective
- Elicit requirements by talking & listening to (potential) users
- Group users in equivalence classes around *personas*



How do we define requirements ?

- Most users are not precise and objective
- Elicit requirements by talking & listening to (potential) users
- Group users in equivalence classes around personas
- Develop user stories

User stories

As a [role],
I want to [action],
so that [reason].

As a SwEnt student,
I want to learn about software requirements,
so that I can engineer better software.

User stories

As a [who is the primary stakeholder?],
I want to [what effect does she want the story to have?],
so that [what value will stakeholder derive from this effect?].

As a SwEng student,
I want to learn about software requirements,
so that I can engineer better software.

Good User Stories: INVEST

- **I**ndependent
 - *self-contained and able to be developed independently of other stories*
- **N**egotiable
 - *specify the problem not the solution, to give flexibility to the implementor*
- **V**aluable
 - *deliver visible value to the end user*
- **E**stimable
 - *small enough to estimate the effort required to complete it*
- **S**mall
 - *manageable and deliverable in a short time frame (in a "Sprint")*
- **T**estable
 - *implies clear acceptance criteria to confirm that it has been implemented correctly*

User Stories for PolyFood



- **#1: Simple order process**

As a user, I want the basic order process in the PolyFood app to take no more than six screen taps, so that I can complete my purchase easily.

- **#2: Filtering**

As a diner, I want advanced search capabilities for filtering restaurants and dishes by cuisine, price, dietary restrictions, and ratings, so that I can intuitively find food that matches my preferences.

- **#3: Intuitive dish comparison**

As a diner, I want to compare dishes easily, based on aspects like ingredients, nutritional information, and prices side-by-side, so that I can make informed decisions about my orders.

Validating & Prioritizing Requirements

Validating Requirements: Are They Right?

- Is your software really answering users' needs/wants ?
 - use prototypes to elicit feedback
 - validate "in the wild"
- Outcome of validation is not true/false

Prioritizing requirements

- Importance of features varies constantly
- Common prioritization schemes
 - $P0 / P1 / P2$
 - *Must-have, should-have, could-have, and won't-have*
- Prioritization is primarily driven by value to user
 - *but also takes into account technical feasibility, legality, cost, ...*



Epics

Epic

- = broad set of features that spans multiple user stories
- Captures larger, more complex user needs
- Enables a comprehensive understanding of a feature set
 - *once divided into actionable, specific user stories the big picture can get lost*
- Useful for organizing work
 - *ensures that the development process is aligned with user needs*
- User story
 - *becomes a specific, actionable item that contributes to the broader goal of the epic*

Epic for PolyFood on Mobile



- User stories #1 (Simple order process) + #2 (Filtering) + #3 (Intuitive dish comparison)
- **#4: Favorites and bookmarking**

As a PolyFood app user, I want to be able to bookmark my favorite dishes and restaurants or save them for later, so that I can quickly access them for future orders.

- **#5. Real-time availability alerts**

As a customer, I want to receive instant notifications from the PolyFood app when my favorite dishes or items from preferred restaurants become available, so that I don't miss out on limited-time offers or restocked popular items.

- **#6: Tailored recommendations**

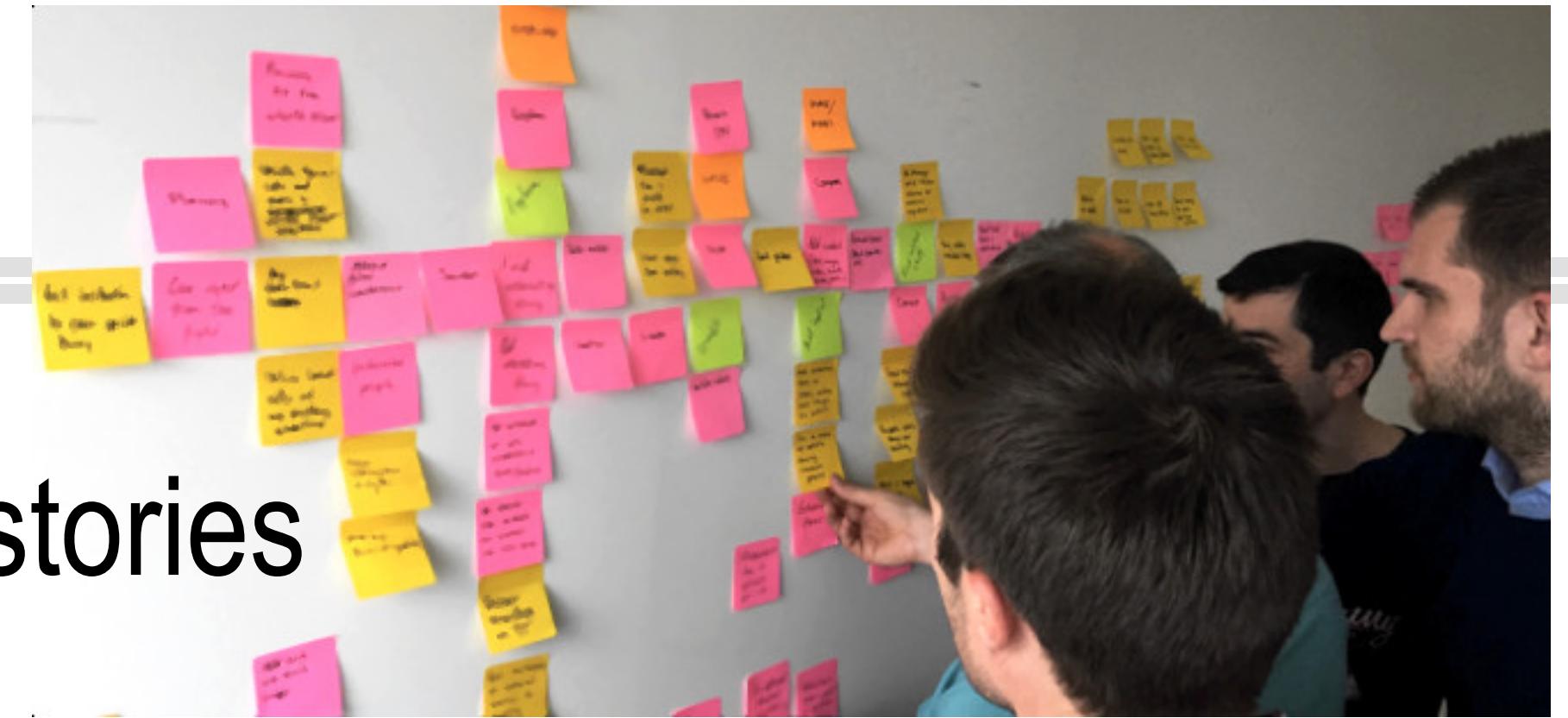
As a returning PolyFood user, I want to receive personalized recommendations for dishes and restaurants based on my order history and preferences, so I can discover and explore dishes I wouldn't otherwise eat.

- **#7: Robust review and rating system**

As a prospective diner, I want access to comprehensive reviews and ratings for restaurants and dishes within the PolyFood app, so that I can make choices based on other diners' experiences.

Story Mapping

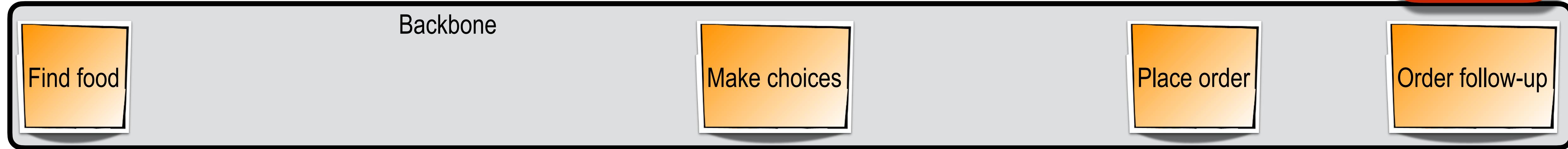
Story Mapping



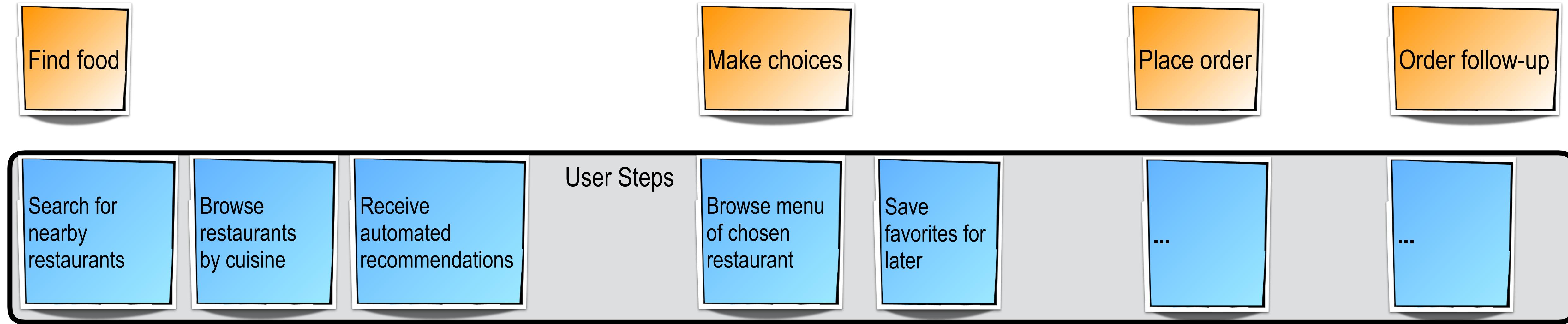
<https://www.linkedin.com/pulse/how-create-u>

- Visual exercise to organize and prioritize user stories
- Horizontal dimension
 - *Backbone = sequence of high-level user activities or tasks*
 - *Break down activities into the steps that users go through to complete the activities*
- Vertical dimension
 - *Details = user stories beneath each step, in order of priority*
- Story maps are for seeing the big picture, prioritizing work, finding gaps, ...

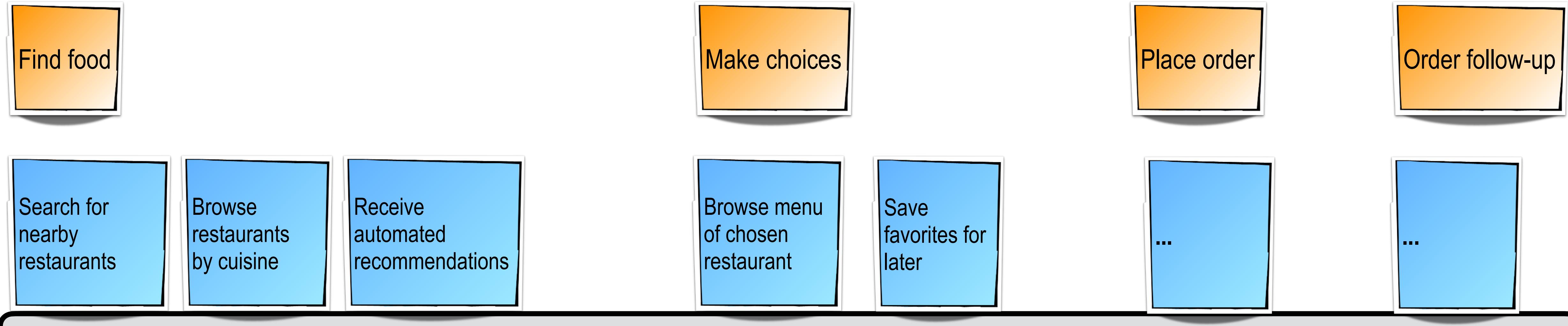
PolyFood: Example Story Map



PolyFood: Example Story Map



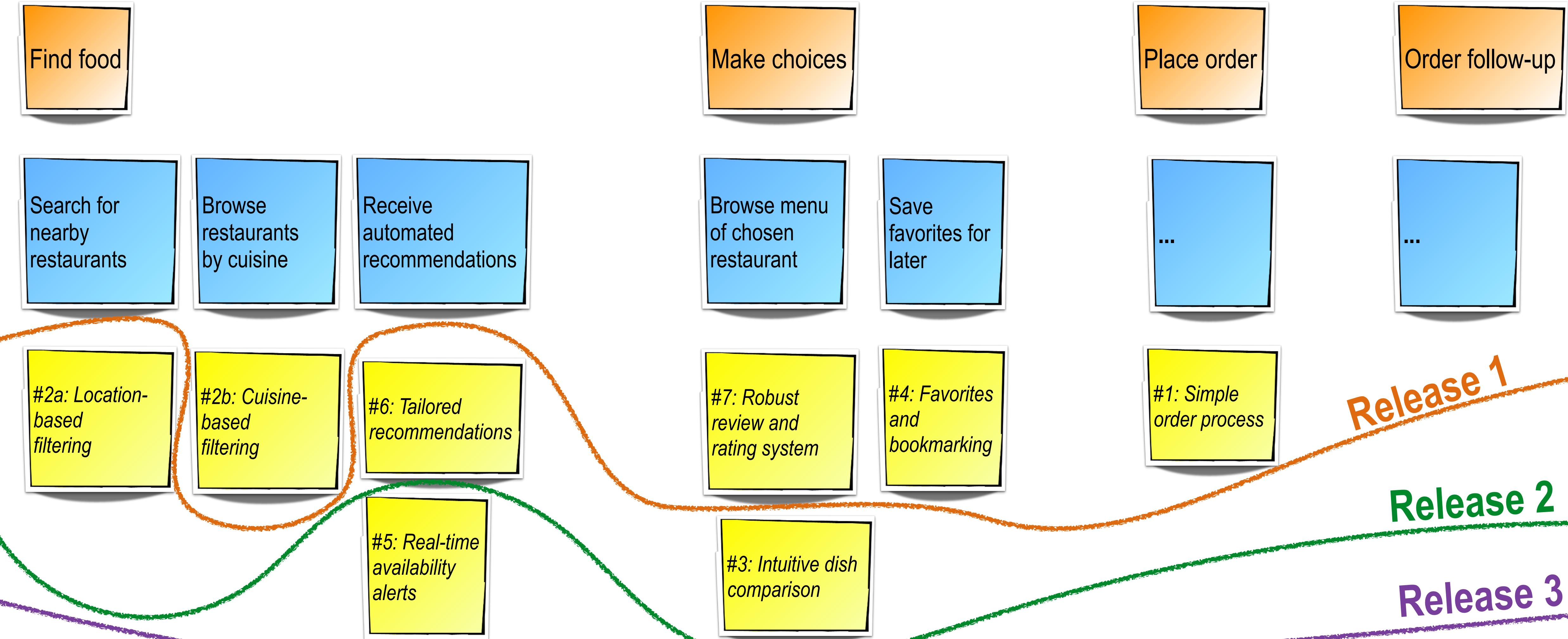
PolyFood: Example Story Map



Details

- #2a: Location-based filtering
- #2b: Cuisine-based filtering
- #6: Tailored recommendations
- #5: Real-time availability alerts
- #3: Intuitive dish comparison
- #4: Favorites and bookmarking
- #1: Simple order process

PolyFood: Example Story Map



Epics and Story Mapping

- Complementary approaches
- Decomposition
 - *use story mapping to break epics into user stories*
- Prioritization
 - *story map shows relative importance of the tasks => help decide which parts of an epic to implement first*
- Clarity
 - *story map adds context to epics by linking the user stories to the user journey*

Recipe for User-Centric Software

- Elicit user requirements
- Formulate user stories
 - *make sure they are Independent, Valuable, Small and Estimable, Testable*
- Validate and prioritize them
 - *all stakeholders must be involved*
- Organize user stories into epics
- Map user stories to user activities
 - *never lose the big picture*
 - *plan the work*

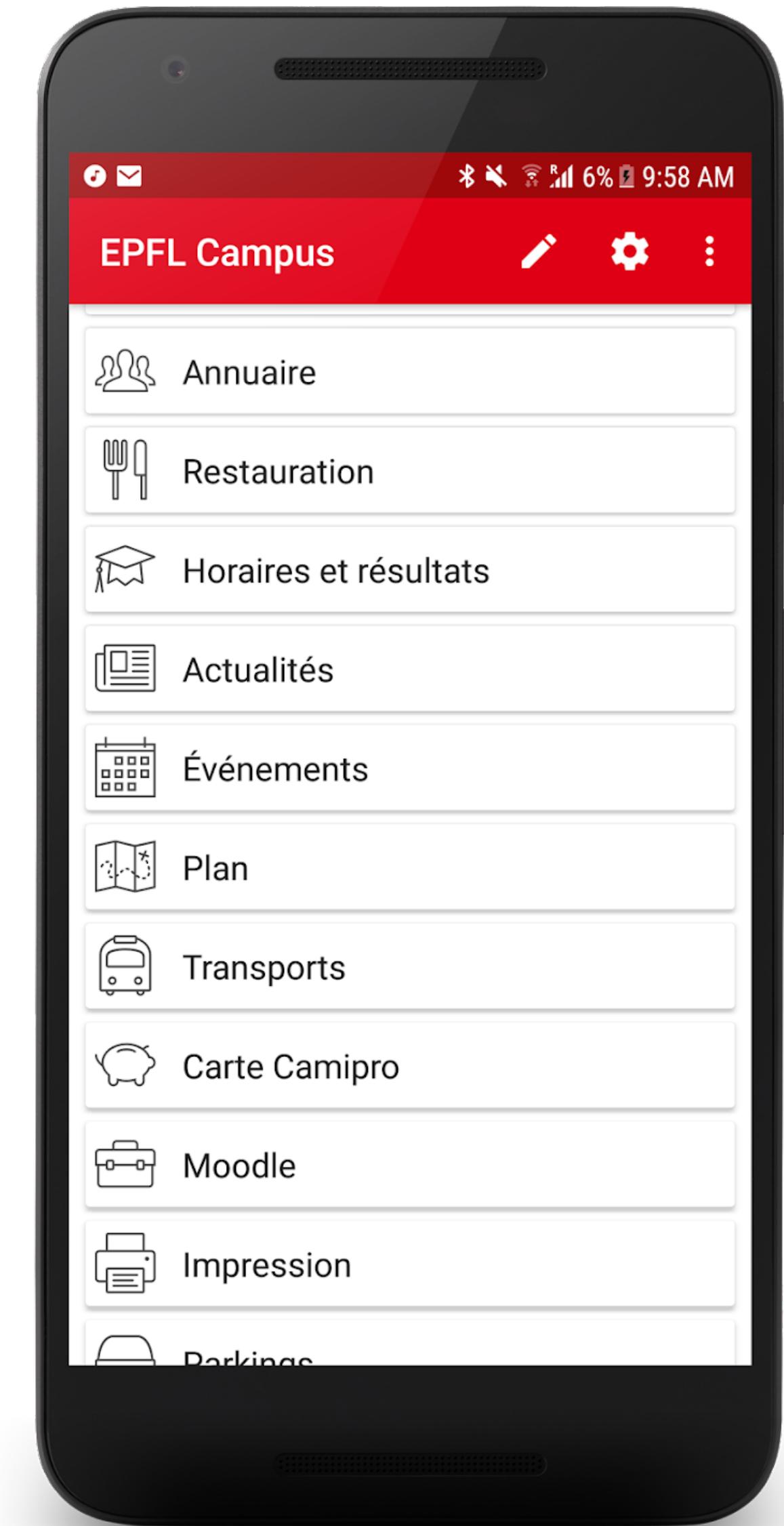
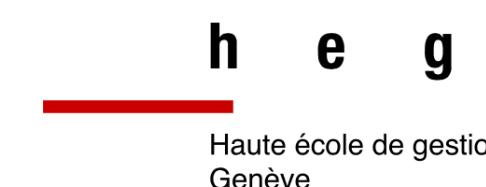
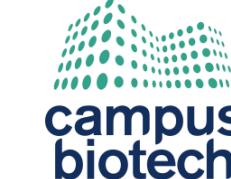
Outline

- The Course
- Tools of the Trade
- Requirements (User Stories, Epics, ...)
- Your Journey in SwEnt
- Succeeding in SwEnt

Your Journey in SwEnt

The App: Overview

- You pick your own app
- You build your own team (7 people)
- You pick a real problem that real people have
 - *then craft a solution using an app*



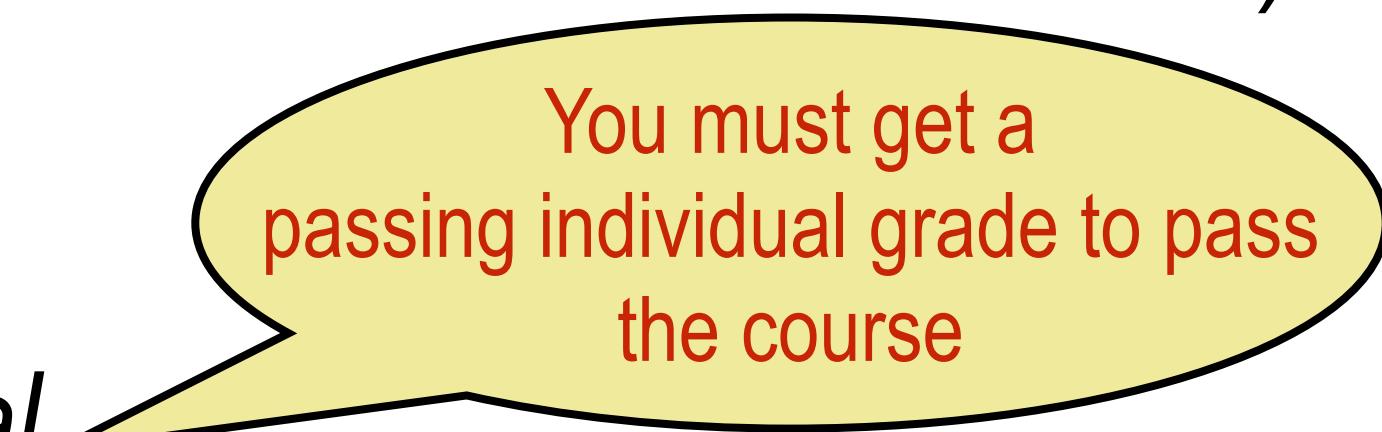


Fly2Find

partnered with
LYF Rescue Drone

The App: Overview

- You pick your own smartphone app + You build your own team
- Pick a real problem that real people have
 - *then craft a solution using an app*
- Send us ...
 - *team + app blurb (see README linked from Moodle for deadline)*
 - *we will give you feedback on app*
 - *re-submit final version (see README for deadline)*
- Course grade
 - *1/2 team + 1/2 individual*



You must get a
passing individual grade to pass
the course

The App: Design Constraints

- Basic
 - *Functionality: compelling tasks for clear use cases*
 - *Ease of use*
 - *Resilience: survive user errors, malice, external factors, ...*
 - *Solid test suite*
- Design
 - *Split app model using public cloud services (no back-end of your own!)*
 - *Support multiple users and authentication*
 - *Offline mode: usable without Internet connectivity*
 - *Use at least one phone sensor as a core feature*

Project Logistics

- Teams of exactly 7 students
 - 1-2 coaches/team – *meet in-person once per week*
 - *meet in-person with team one extra time/week*
 - *weekly feedback from coaches + 3 intermediate graded milestones*
- Battle of the Apps
 - *select projects will be invited to participate*
 - *pitch your app, demo it, and win awards (audience votes)*
- Your own repo for the project on GitHub incl. issue tracking
- Infrastructure: Moodle, GitHub, Ed Discussions, swent-staff@dslab.org

Bootcamp

- Essential !
 - *if you don't do the bootcamp you have no chance*
- Introduction to Android programming
- Set up skeleton for project
- Individual and graded (5% of course grade)
- Start today, immediately after lecture !

Submit your GitHub ID via Moodle
(see Preliminaries section)

Course grade = 45% Project_{indiv} + 50% Project_{team} + 5% Bootcamp

Outline

- The Course
- Tools of the Trade
- Requirements (User Stories, Epics, ...)
- Your Journey in SwEnt
- Succeeding in SwEnt

Succeeding in SwEnt

Succeeding in SwEnt: Basics

- CS-214 Software construction (BA3) or equivalent
 - CS-202 Computer systems (BA4) or equivalent
 - Proficient in C, Python, and object-oriented Java/Kotlin
-
- It will be impossible to pass SwEnt if you do not have the prerequisites

Succeeding in SwEnt: Your Attitude

- Work consistently every week, otherwise you fall behind and can't recover
- Be auto-didact, because lectures teach only a subset of what you need
 - *It takes a top developer 20x less effort to achieve what an average dev does*
- Be a problem solver — tech challenges inherent to building real sw
- Coordinate with your team
- Deal with your team (slacker, know-it-all, ...)
- Stay ethical in the face of deadlines and challenges

Succeeding in SwEnt: Wisdom and Recovery

- You will screw up ... and that's ok !
- To recover ...
 - *avoid foolish optimism*
 - *reduce scope of problem*
- Work smart, be agile
- Love what you do, be interested, don't brood
- Listen to the wisdom from past students



© Tatiana Adamenko/Caters

Social Contract

- Students
 - *Follow all lectures, do the bootcamp*
 - *Be a good team member and strive for your team's success*
 - *Be ambitious, a force for good, do not fall behind*
 - *The project is a lot of work*
- Staff
 - *Dedicated coaching team for each team of 7 students*
 - *Support you while you try to figure out (by yourself) a way forward*
- Promises
 - *Diligent students will have the tools to turn ideas into solutions and products*
 - *Slackers will fail*

Outline

- The Course
- Tools of the Trade
- Requirements (User Stories, Epics, ...)
- Your Journey in SwEnt
- Succeeding in SwEnt