Software Engineering

ECE444 (Fall2021)

Shurui Zhou

Assistant Professor

The Edward S. Rogers Sr. Department
of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Shurui Zhou [pronunciation: Shoo-ray Joe]

Carnegie Mellon University

2014 - 2020 Ph.D.
School of Computer Science
Institute for Software Research

UNIVERSITY OF TORONTO

2020 Fall – Assistant Professor

FORCOLAB

Research Interests

- Software Engineering (SE)
- SE for AI
- AI for SE
- Collaborative Software Development
- Open Source

https://www.eecg.utoronto.ca/~shuruiz/
shuruiz@ece.utoronto.ca

First of all:

You are not alone!
We are undertaking this new experience together.

This is not normal. We understand.

- Expect:
  - Feeling overwhelmed
  - Many additional sources of stress
  - Hard time dealing with *everything*...

Talk to us about accommodations of any kind

Simulating NORMAL in-class Experience

- Discussions and interactions are important. We'll have regular in-class discussions and exercises
- I may call on you
- Contact me for accommodations!

Active Lecture

- Case study driven
- Discussion highly encouraged
- Contribute own experience
- Regular active in-class exercises
- In-class presentation
- Discussions over definitions

Agenda for Today

- Introduction of the course
- Introduction of Software Engineering
- Process and Team

Lecture Logistics during a Pandemic

- Live lecture on Zoom, recording for student who cannot attend in-person lecture
- Recording for PRA lab sessions

COVID-19: IN-CLASS GUIDELINES

GENERAL MEASURES

Community Check-In

- **Self-screen** – How are you feeling? Remember: do not come to campus if you have a fever, sore throat, cough, difficulty breathing, runny nose, or feeling unwell. For more information: https://www.utoronto.ca/utogether/ucheck.
- **Vaccination** - The University of Toronto **will require** that all those intending to be present on our campuses be fully vaccinated against COVID-19, in accordance with all applicable laws and regulations. For more information: https://www.utoronto.ca/utogether/vaccines
- **Hygiene** – Wash hands regularly, avoid touching face, sneeze or cough into your arm, no shared surfaces or tools
- **Masks** – Wear a mask while in-class unless given permission to remove it by the instructor or have an accommodation/exemption. For more information, please refer to the Policy on Face Masks: https://governingcouncil.utoronto.ca/secretariat/policies/face-masks-policy and the Joint Provostial Guidelines on Face Masks: https://www.provost.utoronto.ca/planning-policy/joint-provostial-and-human-resources-guideline-on-facemasks-at-the-university-of-toronto/
- **Traffic Flow** – Follow the designated traffic flow for entering, moving through, and exiting room
- **Furniture Placement** – Use designated furniture, do not move furniture without permission

PREVENTION AND PRECAUTIONS

Please remember to:

Stay home if you are ill.

Avoid touching your eyes, nose and mouth.

Cough or sneeze into a tissue and immediately dispose of it in the garbage and wash your hands afterwards.

Wash your hands often with soap and water for at least 15 seconds. If soap and water are not available, use a hand sanitizer with at least 60% alcohol.

Avoid prolonged personal contact, such as touching or shaking hands. Consider greeting others with a nod, wave or bow.

If you don't have a tissue, sneeze or cough into the bend of your arm.

Clean and disinfect frequently touched objects and surfaces.

REMEMBER...

What we can do to keep us all safe.

WHAT TO DO IF UNWELL?

Non- Urgent:

1. Report your sickness to your course instructor
2. Go home
3. Email U of T's Occupational Health Nurse (ehs.occhealth@utoronto.ca) who will conduct assessment and contact tracing, and will provide further direction

NOTE: *The University has suspended the need for a doctor's note or medical certificate for absences if experiencing COVID-19 symptoms.*

UNIVERSITY OF TORONTO

Mask policy

- **Instructors** have to keep masks on while teaching. You can remove your mask to sip water, i.e., remove it for short short periods, but not the whole class. **For long lectures and other types of teaching where wearing a mask may be arduous or impede teaching**, EHS will work with divisions to process requests for exceptions in a manner that is consistent and rapid.

https://www.provost.utoronto.ca/planning-policy/joint-provostial-and-human-resources-guideline-on-facemasks-at-the-university-of-toronto/

Mask policy

- **Students**: For some **specific components of the class** and based on the instructor's recommendation, student(s) can temporarily remove his/her/their mask(s)
- **Student refusing to wear masks**: Our expectation is that the vast majority of students will wear masks as required (covering nose, mouth and chin without gaps). **Instructors can ask the student refusing to wear a mask to leave the classroom.**

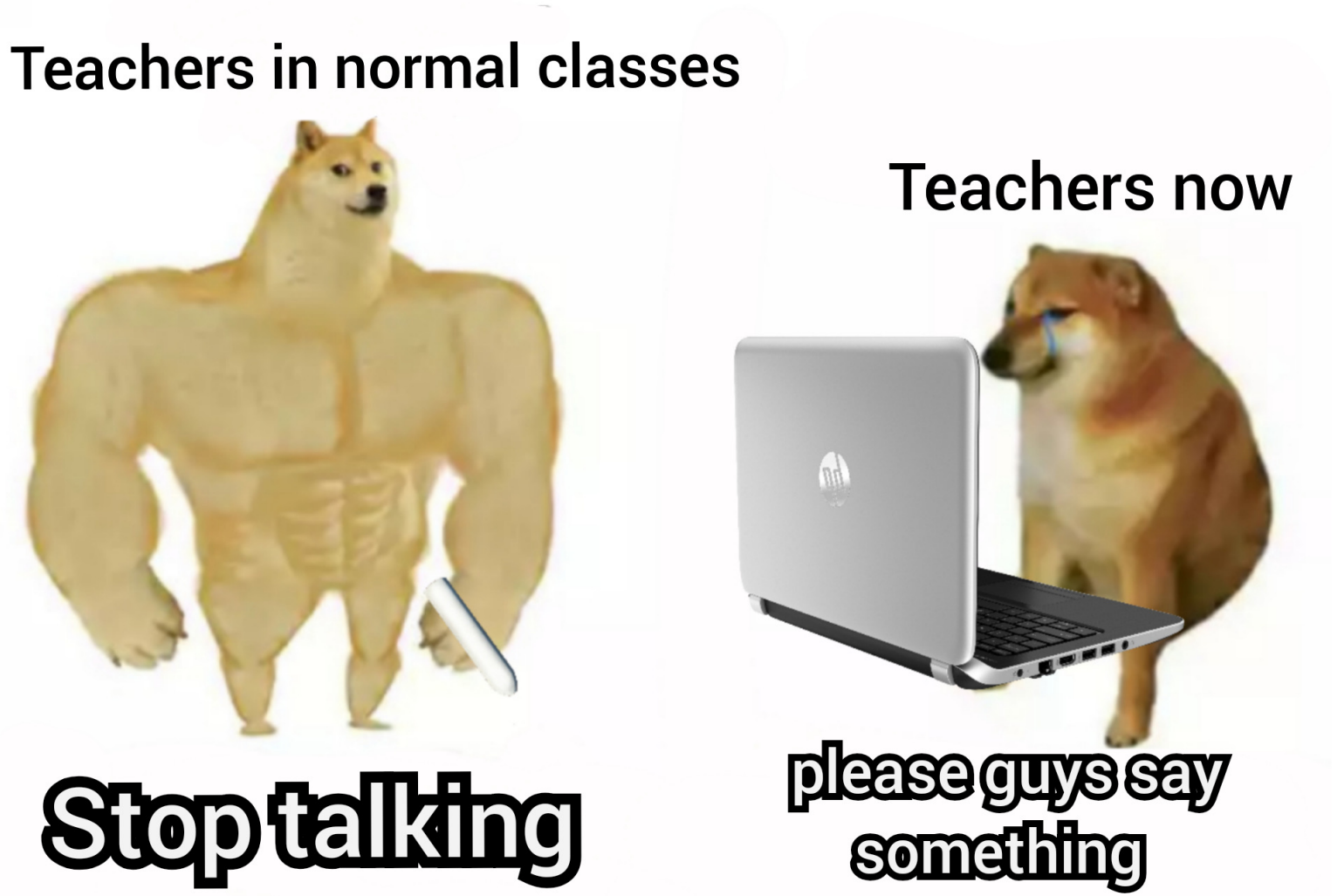
https://www.provost.utoronto.ca/planning-policy/joint-provostial-and-human-resources-guideline-on-facemasks-at-the-university-of-toronto/

Mask policy

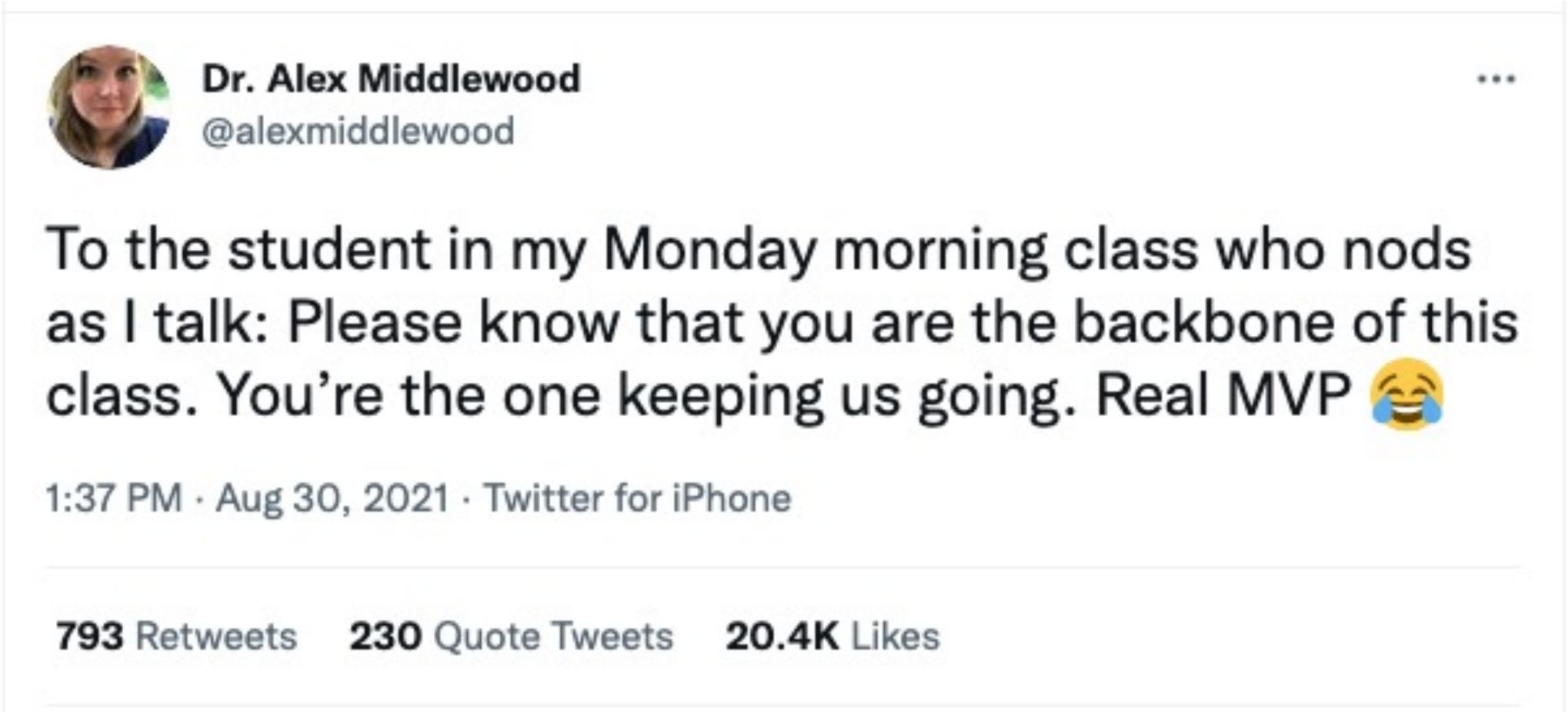
- **Students with a medical exception to wearing masks**: Some students may have a medical exception to wearing masks. If a student states they cannot wear a mask for disability related reasons, they should be directed to the undergrad office as soon as possible. If there is a disability related accommodation pertaining to mask wearing, Accessibility Advisors will connect with instructors directly. We will also be following up with U of T's environmental health and safety office as needed.

https://www.provost.utoronto.ca/planning-policy/joint-provostial-and-human-resources-guideline-on-facemasks-at-the-university-of-toronto/

2020


Teachers in normal classes
Stop talking
Teachers now
please guys say something

2021



Dr. Alex Middlewood
@alexmiddlewood

To the student in my Monday morning class who nods as I talk: Please know that you are the backbone of this class. You're the one keeping us going. Real MVP 😂

1:37 PM · Aug 30, 2021 · Twitter for iPhone

793 Retweets **230** Quote Tweets **20.4K** Likes

Any Questions?

Learning Goals

- Learn how software is developed in a systematic way
- Learn by doing - 2 main group projects
  - Web application development
  - Open source excursion
- Learn the state-of-the-art research topics in software engineering
  - Reading papers
  - Case studies

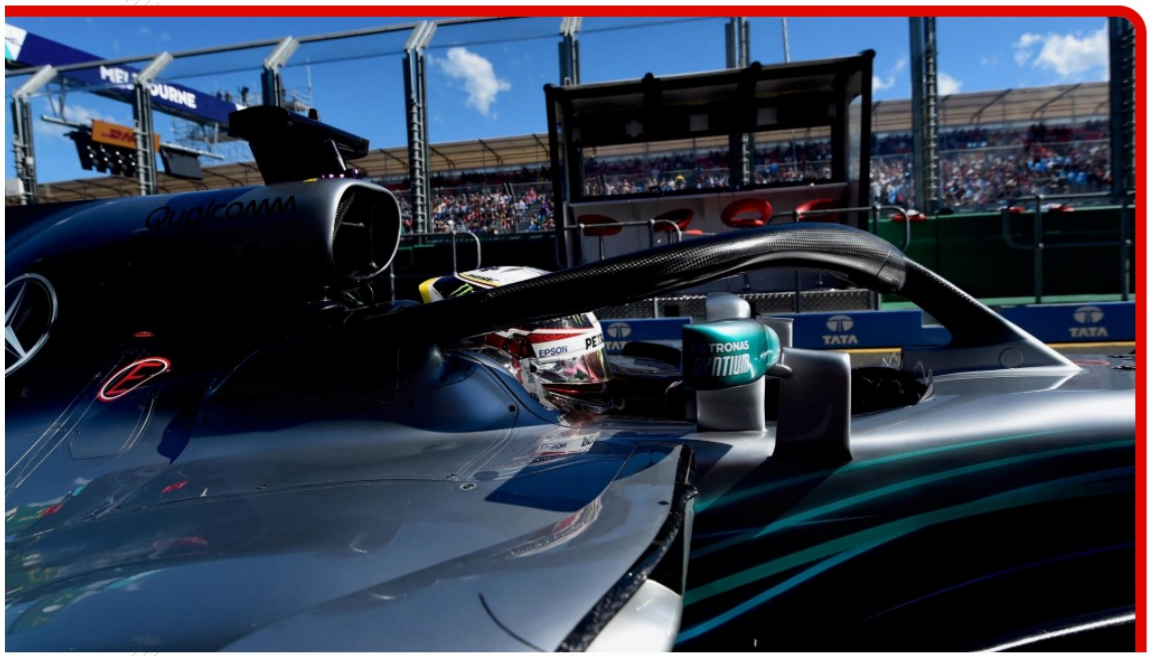
Software is everywhere

https://www.youtube.com/watch?v=-dYiH5AXUCU

Software glitch cost Hamilton victory - Mercedes

25 March 2018

MERCEDES | AUSTRALIA | HAMILTON



The software the team has used for five years to simulate such scenarios had generated the incorrect figures, consigning Hamilton to a second-place finish behind Vettel's Ferrari.

"Lewis did nothing wrong - it was down to a software bug or an algorithm that was simply wrong"

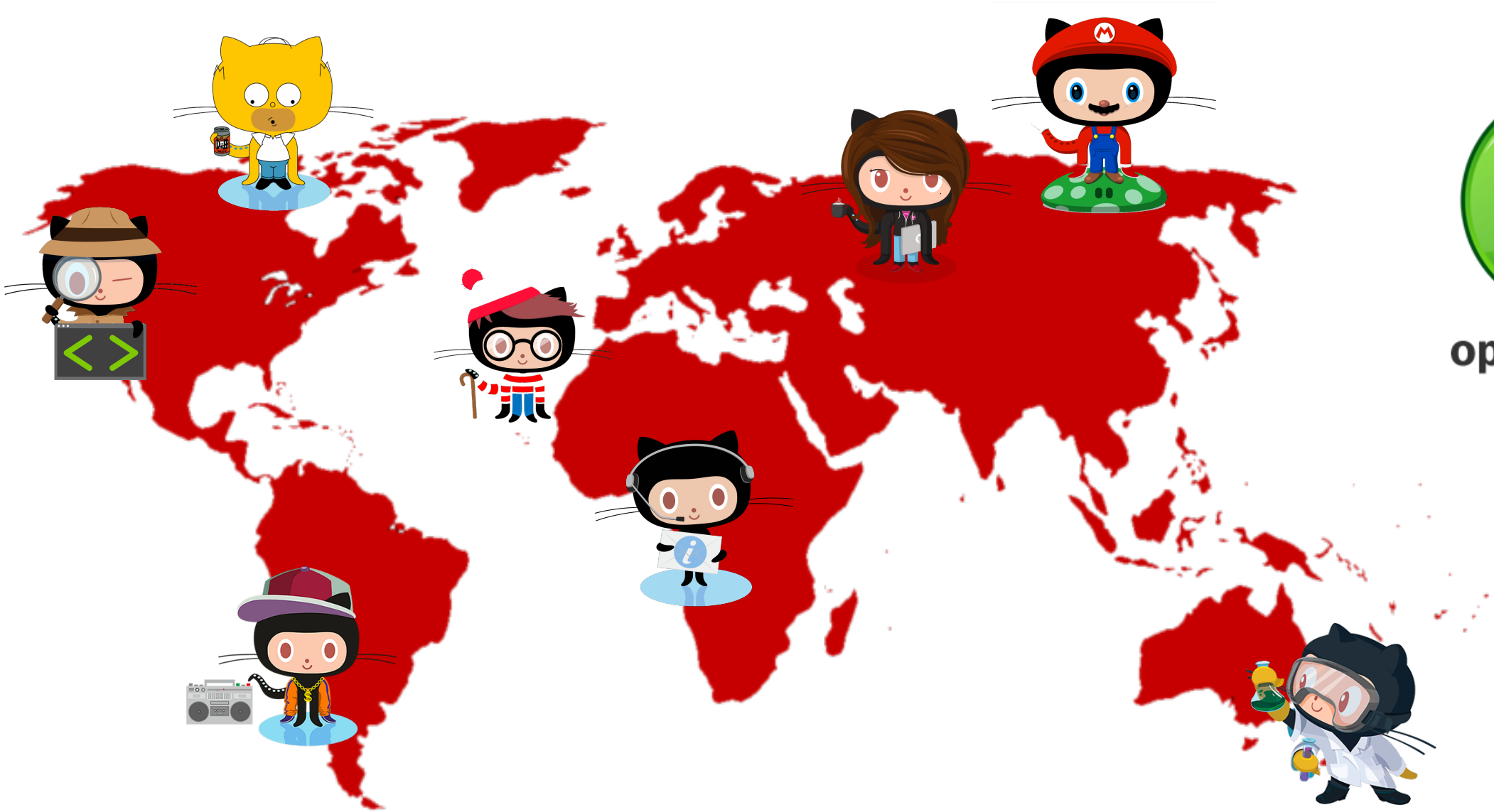
Toto Wolff

https://www.formula1.com/en/latest/article.software-glitch-cost-hamilton-victory-mercedes.6VzyCYpEpauaIYsOWYCqYS.html#:~:text=A%20software%20glitch.,season%2Dopening%20race%20in%20Australia.&text=The%20world%20champion%20immediately%20asked,time%20Mercedes%20had%20given%20him.

https://www.youtube.com/watch?v=n3-ZQqU8m08&t=10s

Globally Distributed Software Development

Globally Distributed Software Development



Intro of the class

Significant redesign (course structure/homework)

ECE444 (Fall2019-UofT)

+ 17-313 Software Engineering (CMU)

+ 17-214 Principles of Software Construction (CMU)

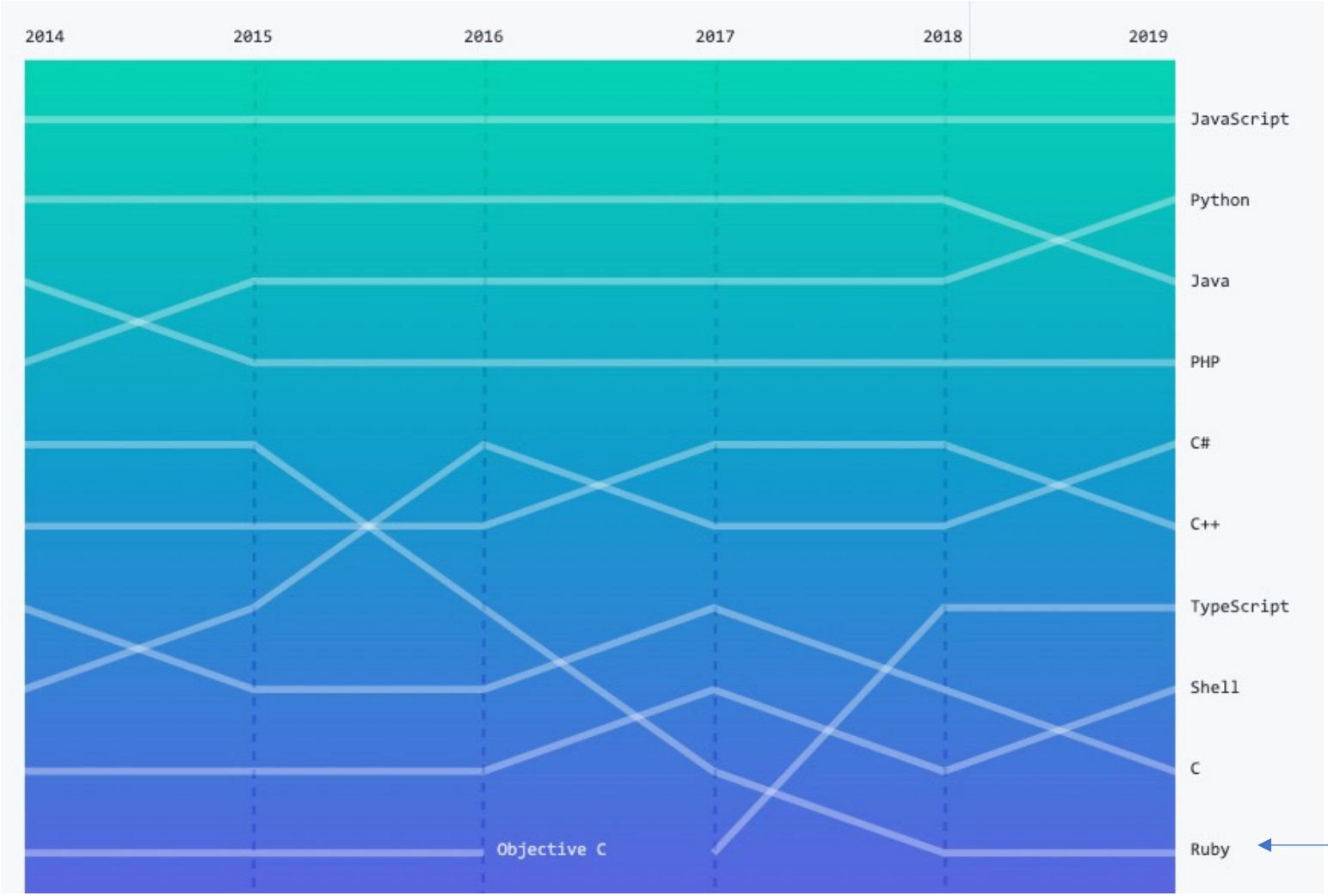
+ 17-652 Requirement Engineering (CMU)

See ECE444 (2020F) at
https://www.eecg.utoronto.ca/~shuruiz/teaching/ECE444-2020F/

ECE444
2019 vs 2020→

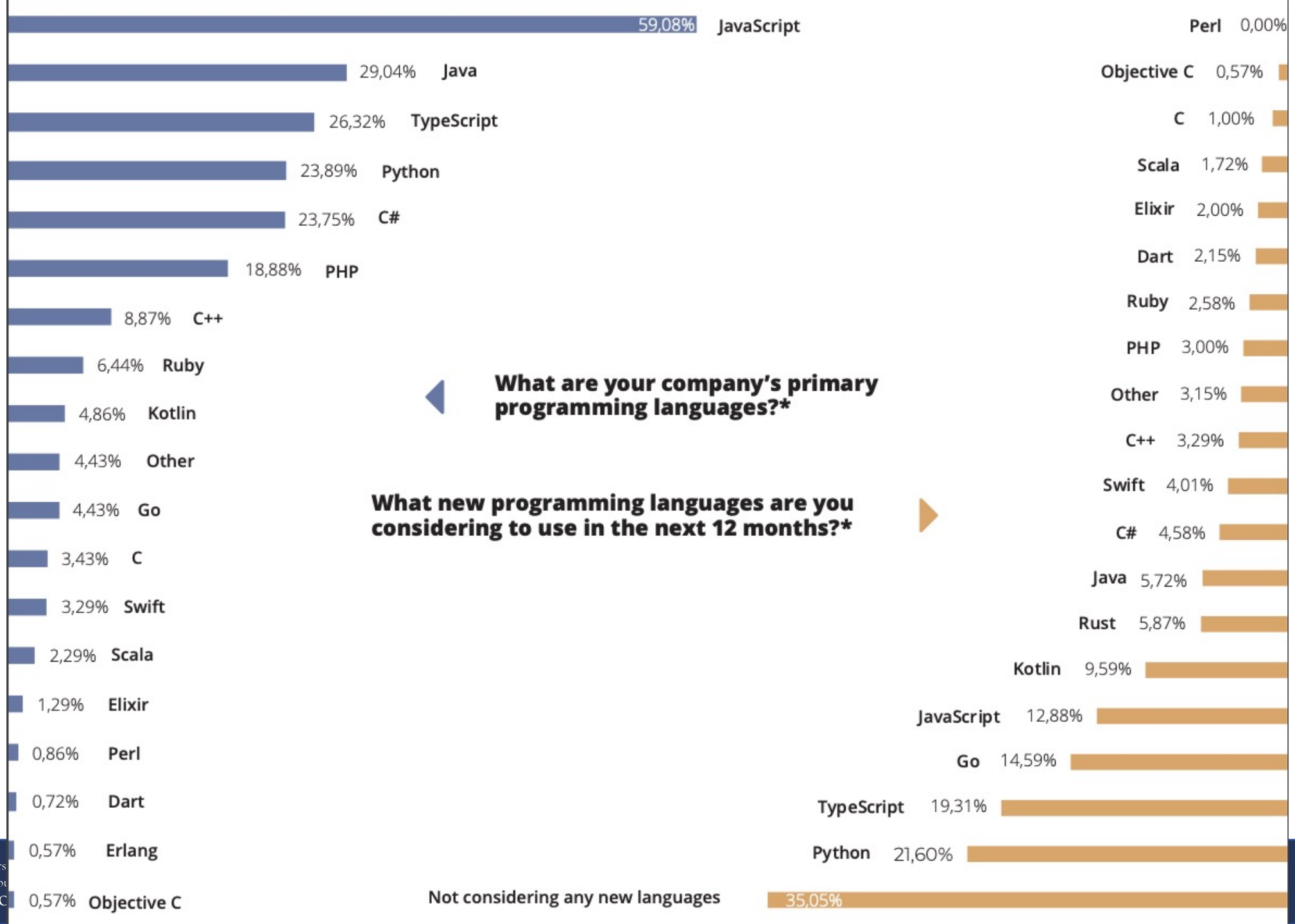
https://octoverse.github.com/

Top Languages

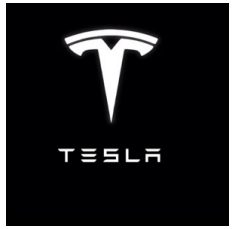


Python

Ruby


What are your company's primary programming languages?*
59,08% JavaScript
29,04% Java
26,32% TypeScript
23,89% Python
23,75% C#
18,88% PHP
8,87% C++
6,44% Ruby
4,86% Kotlin
4,43% Other
4,43% Go
3,43% C
3,29% Swift
2,29% Scala
1,29% Elixir
0,86% Perl
0,72% Dart
0,57% Erlang
0,57% Objective C
What new programming languages are you considering to use in the next 12 months?*
Perl 0,00%
Objective C 0,57%
C 1,00%
Scala 1,72%
Elixir 2,00%
Dart 2,15%
Ruby 2,58%
PHP 3,00%
Other 3,15%
C++ 3,29%
Swift 4,01%
C# 4,58%
Java 5,72%
Rust 5,87%
Kotlin 9,59%
JavaScript 12,88%
Go 14,59%
TypeScript 19,31%
Python 21,60%
Not considering any new languages
35,05%

Companies using **Ruby on Rails**



https://www.ideamotive.co/blog/40-best-ruby-on-rails-companies-websites

Companies using **Python**

Instagram

https://realpython.com/world-class-companies-using-python/

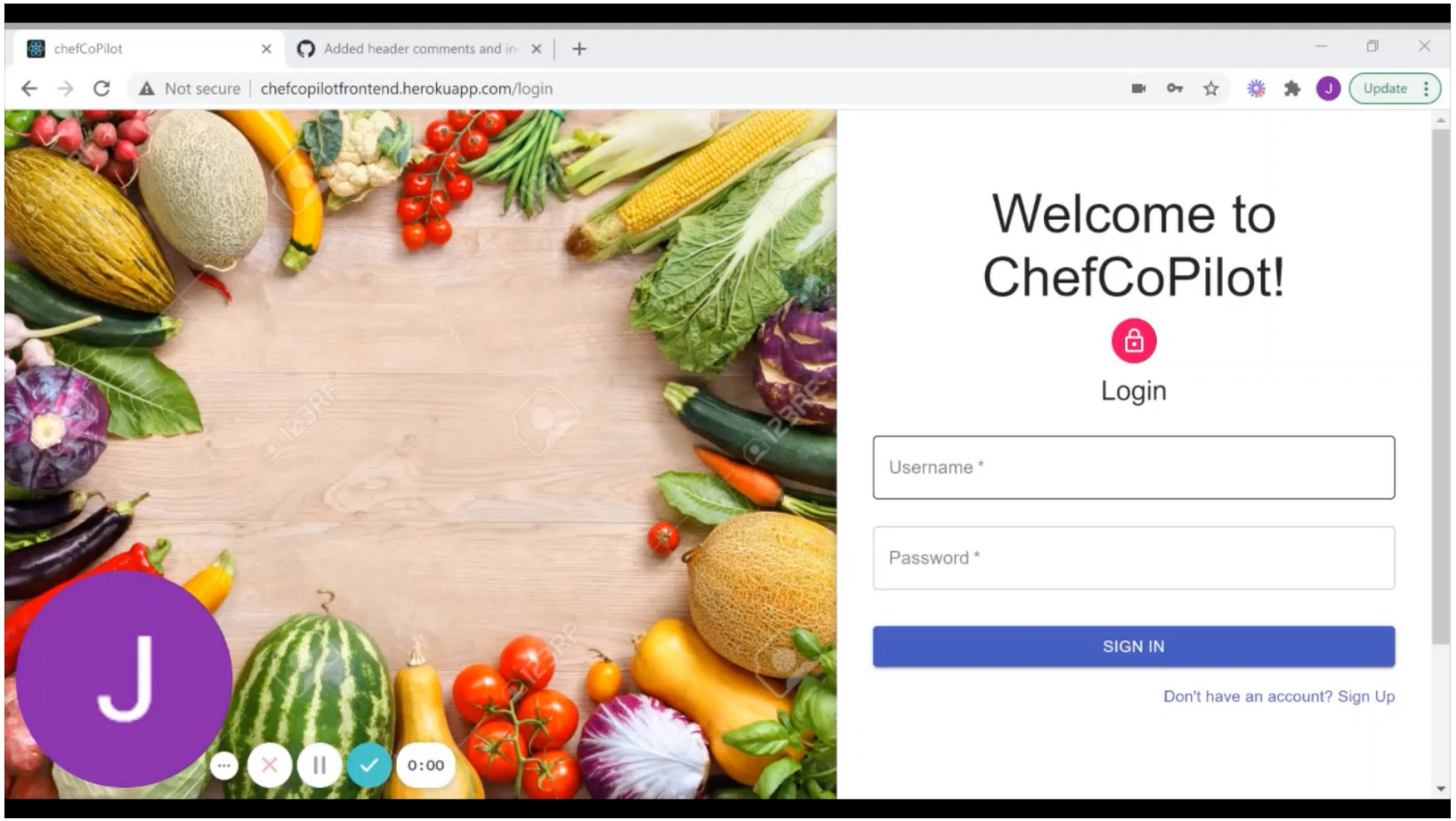
Point of Comparison	Ruby on Rails	Django
Language	Ruby	Python
Known for	Rapid development	Dynamic applications
Main benefit	Powerful RubyGems	AI & ML apps
Syntax	Flexible coding	One Obvious way
Popularity	Vibrant community	Academic teaching
Unique features	COC, DRY	Data science
Pros	• Easy migration • Quick development • Diverse tools • Automated testing • Active community	• Scalable apps • Highly configurable • REST API • MVC programming • High compatibility
Cons	• Tricky API creation • Low flexibility • Poor runtime • Poor documentation	• No multiple requests • Based on ORM • Standalone • Too tight knit

Project 1 (Web application Design) - Fall 2020

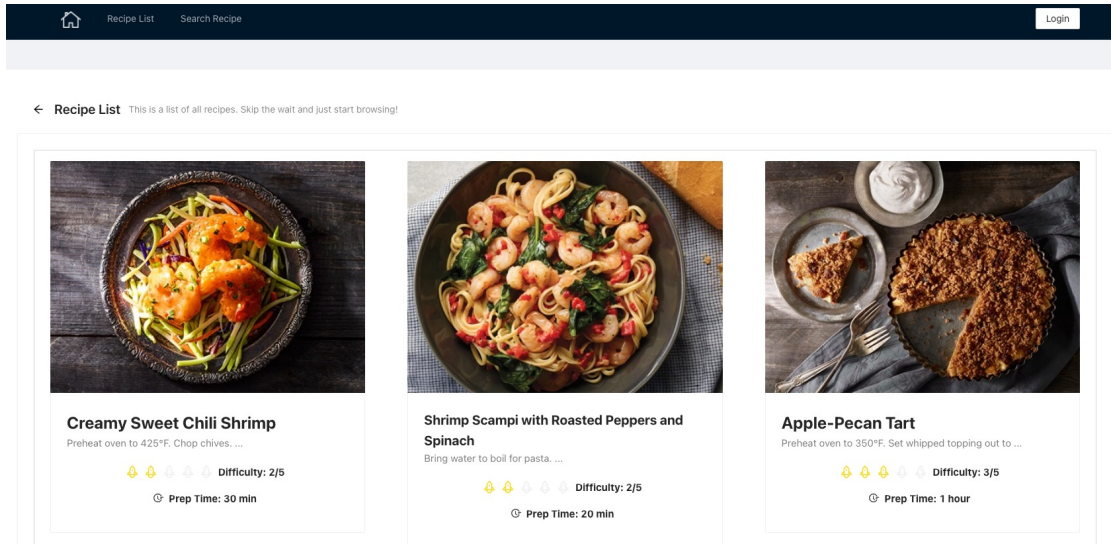
Milestone	Title	Marks	Due Date	Description
0	Meet your team & Proposing a project idea	2%	9/16	Submit 1-2 paragraphs (max 1/2 page) suggesting an online service worth having for group design and implementation in this class. Your submission will be evaluated based on: originality, understandability, feasibility.

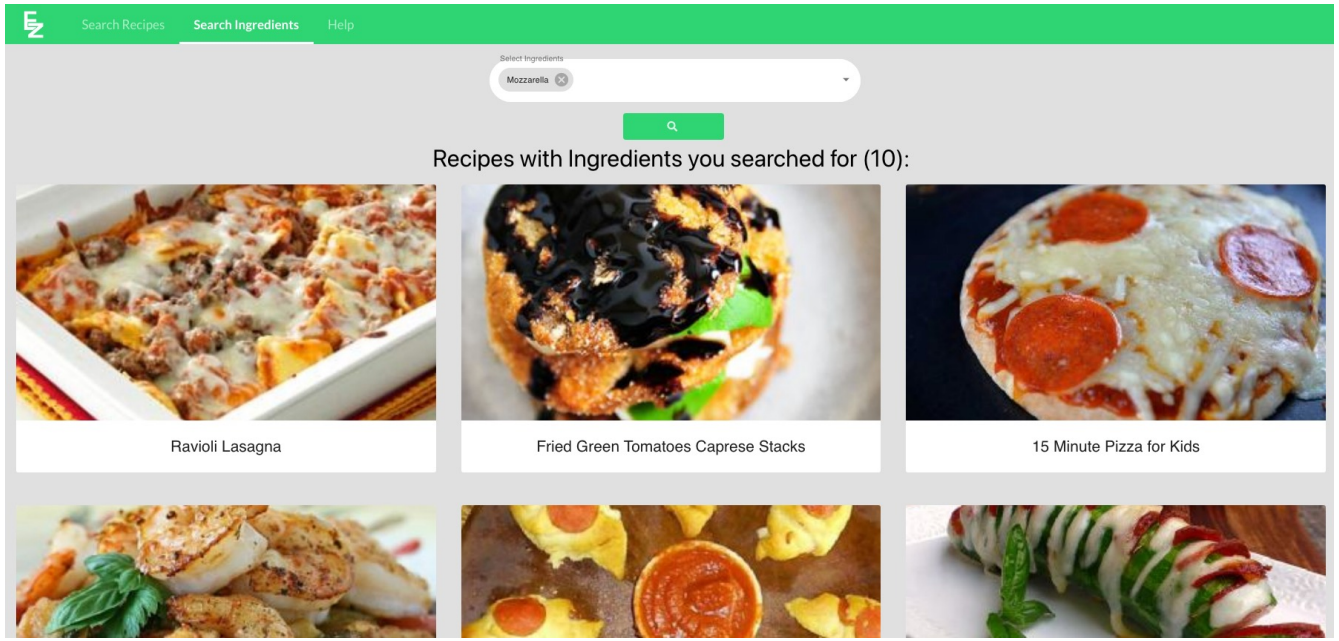
Web application Design -- Showcase (Fall 2020)

Web application Design -- Showcase (Fall 2020)



Web application Design -- Showcase (Fall 2020)





UNIVERSITY OF TORONTO | Engineering

Search

Centre for Analytics and Artificial Intelligence Engineering

About | Faculty Affiliates | Research | Education | Partnerships

The hub for analytics and artificial intelligence at UofT Engineering

Research

- CARTE Seed: Funding for high-impact multi-disciplinary research
- CARTE Match: Connecting faculty with Analytics/AI/ML projects with students
- In-house Analytics/AI/ML research support
- Research drop-in clinics
- Enhancing Equity, Diversity, and Inclusion (EDI) in Analytics/AI/ML research

Education

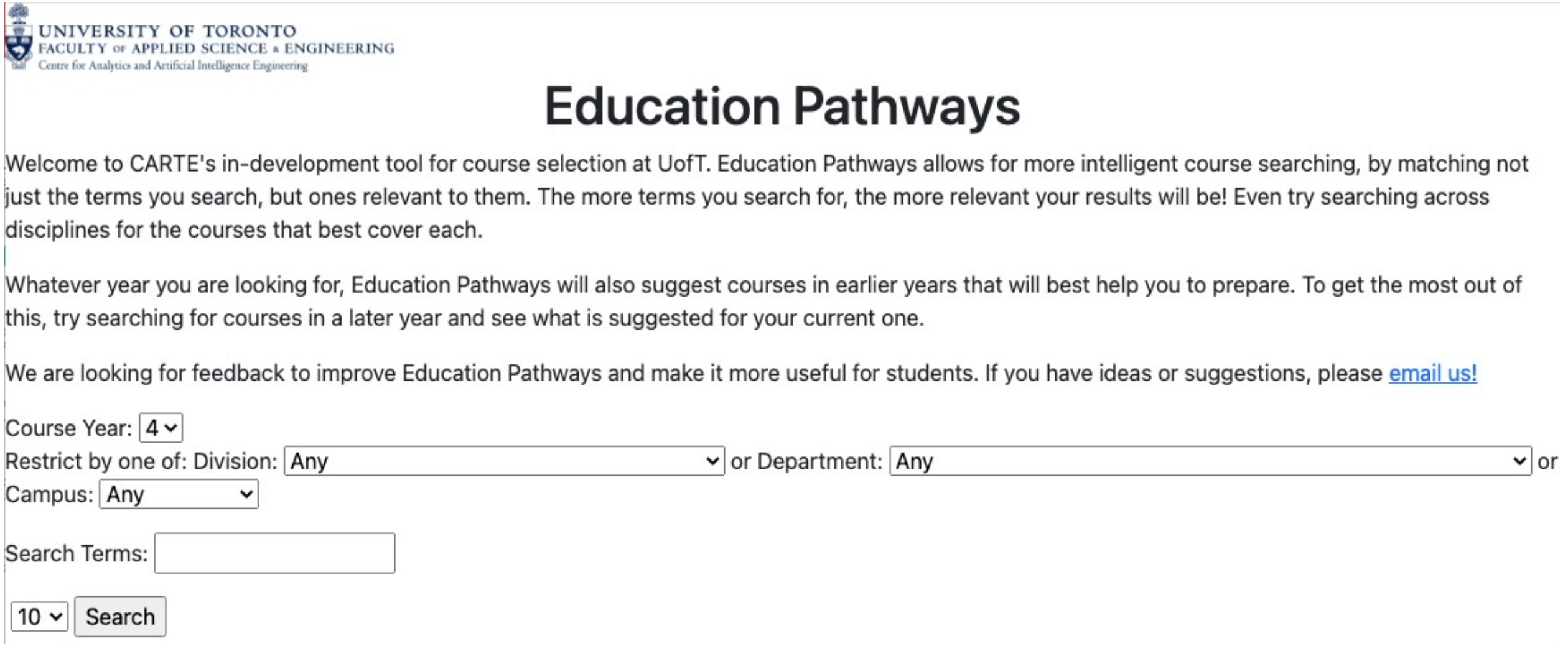
- ML Bootcamp for faculty
- Pathway of courses across the university for students
- Analytics/AI/ML student community of practice
- Information repository on computing resources
- MITACS Accelerate internship opportunities in Analytics/AI/ML

Professor Chandra Veer Singh uses high throughput simulations, machine learning, and available experimental data to enable design of high entropy alloys for structural applications.

Learn more | Subscribe to our newsletter

Media Mentions

https://educationpathways.herokuapp.com/



Magellan (online course selection tool)

Click here for ECE201's 2020- 2021 Magellan slides

Magellan is a software tool that has been developed in-house in order to help you plan and verify both your program and CEAB (Canadian Engineering Accreditation Board) requirements. It was developed in order to make it easy to verify the CEAB requirements as it automatically calculates the academic units (AU's) when you are building your study plans for 3rd and 4th year. In addition, it will confirm the program requirements at a glance.

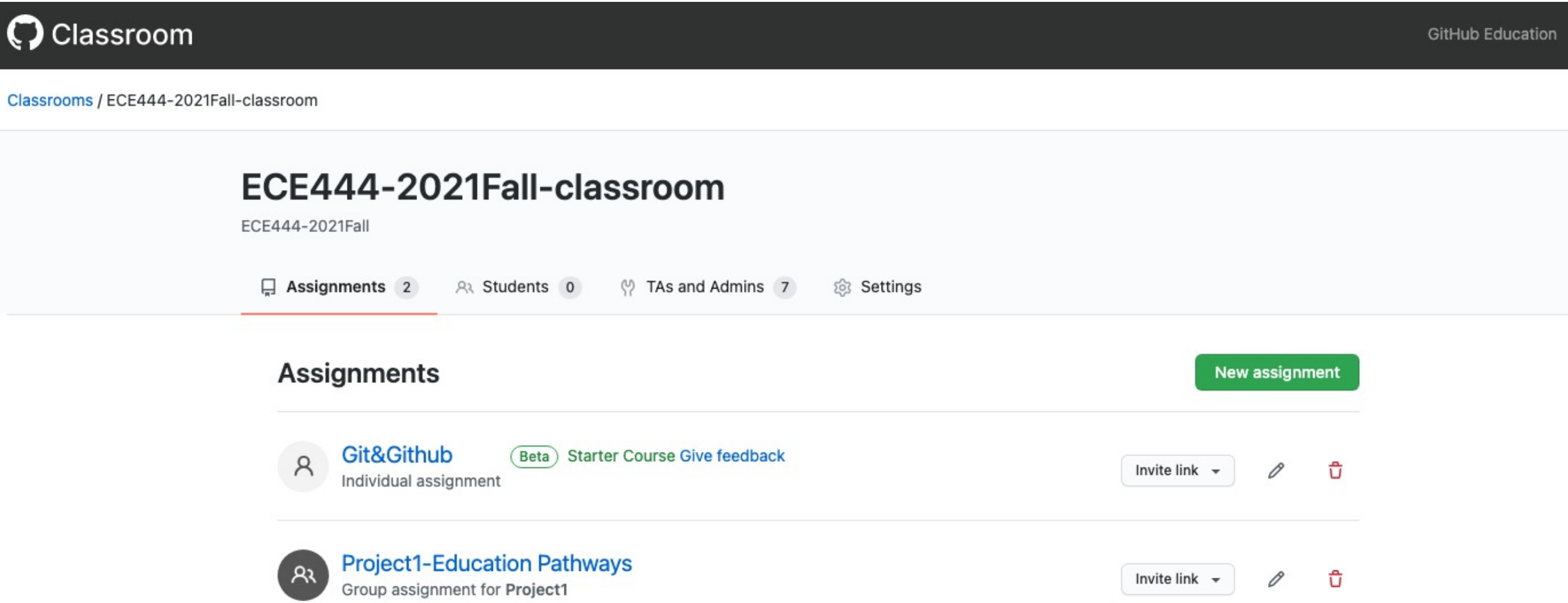
MAGELLAN PRE-REGISTRATION (for *2021-2022 courses)* **ENDS:** January 26, 2021 at 11:59pm EST

Main profiles will then be *LOCKED until July, exact date TBC.* for preregistration planning.
ECE courses will be uploaded to ACORN timetables for students with **valid** Magellan profiles.
Magellan: http://magellan.ece.toronto.edu *(log in with UTORID and password) /* Magellan Manual

Main Profiles will be UNLOCKED in July*
***CHANGES** made to main profiles once UNLOCKED in July*
*are **NOT** included in the preregistration process and thus not uploaded to ACORN.*

****Click HERE for DETAILED pre-registration information ****

For students returning from PEY Co-op in Fall 2021:
You must manually shift any courses you may have listed in 2020-2021 to 2021-2022 as courses do not automatically roll over. Courses listed in the current academic year will overwritten with 'PEY500'.


Classroom
GitHub Education
Classrooms / ECE444-2021Fall-classroom
ECE444-2021Fall-classroom
ECE444-2021Fall
Assignments 2
Students 0
TAs and Admins 7
Settings
Assignments
New assignment
Git&Github
Beta
Starter Course Give feedback
Individual assignment
Invite link
Project1-Education Pathways
Group assignment for Project1
Invite link

0 Open ✓ 11 Closed Author Label Projects Milest

Update README.md (missing period)
#11 by UTkzhang was merged on Nov 17, 2020 • Approved

Adjusted readme and shifted things to Wiki
#10 by shadow-blade-X was merged on Nov 17, 2020 • Approved

tab fixes on string response returns bug
#9 by UTkzhang was merged on Nov 13, 2020 • Approved

Documentation update
#8 by UTkzhang was merged on Nov 13, 2020 • Approved

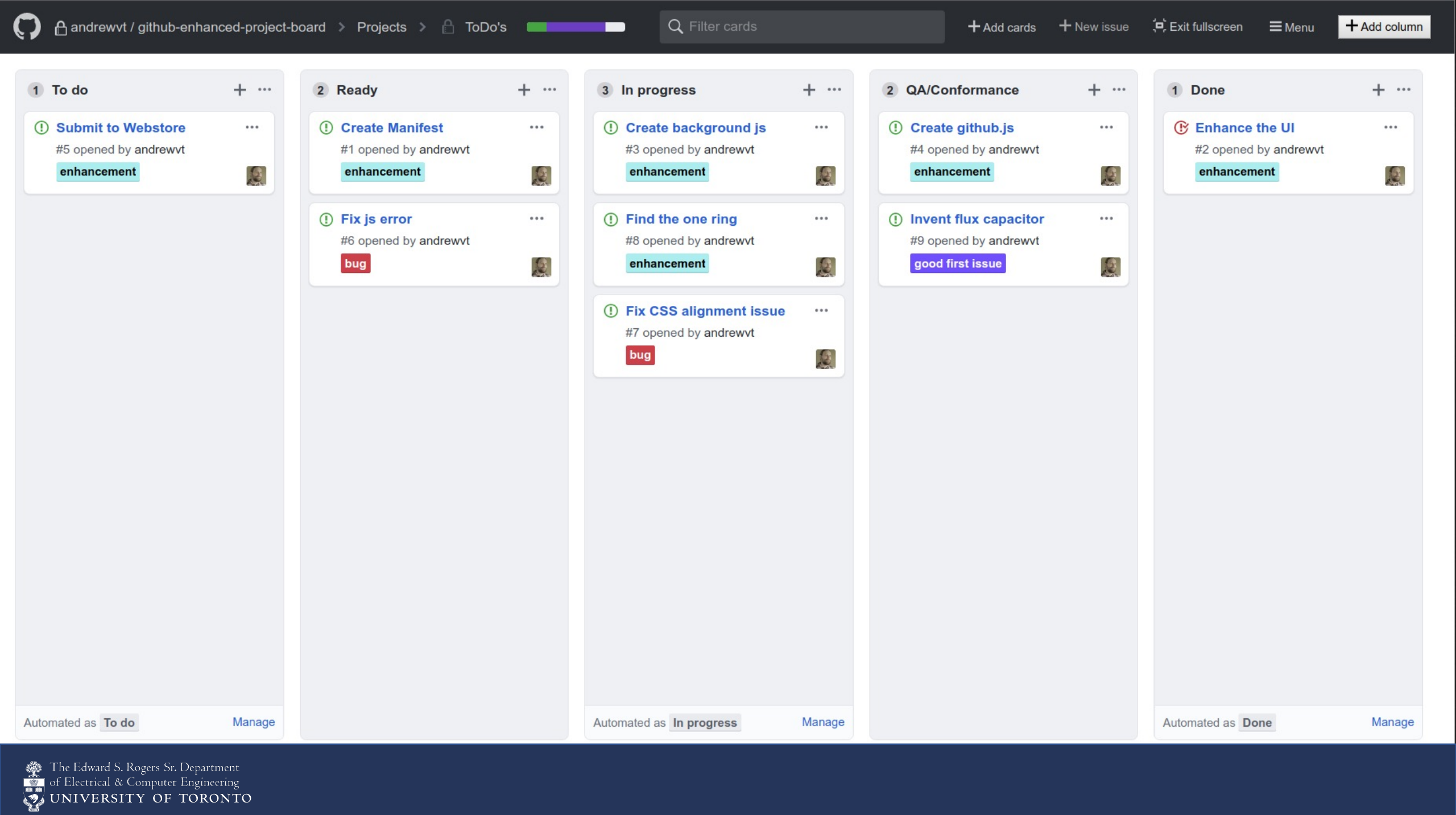
Documentation update
#7 by UTkzhang was merged on Nov 13, 2020 • Approved

patched bad recipe update issue bug
#6 by UTkzhang was merged on Nov 13, 2020 • Approved

Test added for all API
#5 by shadow-blade-X was merged on Nov 13, 2020 • Approved

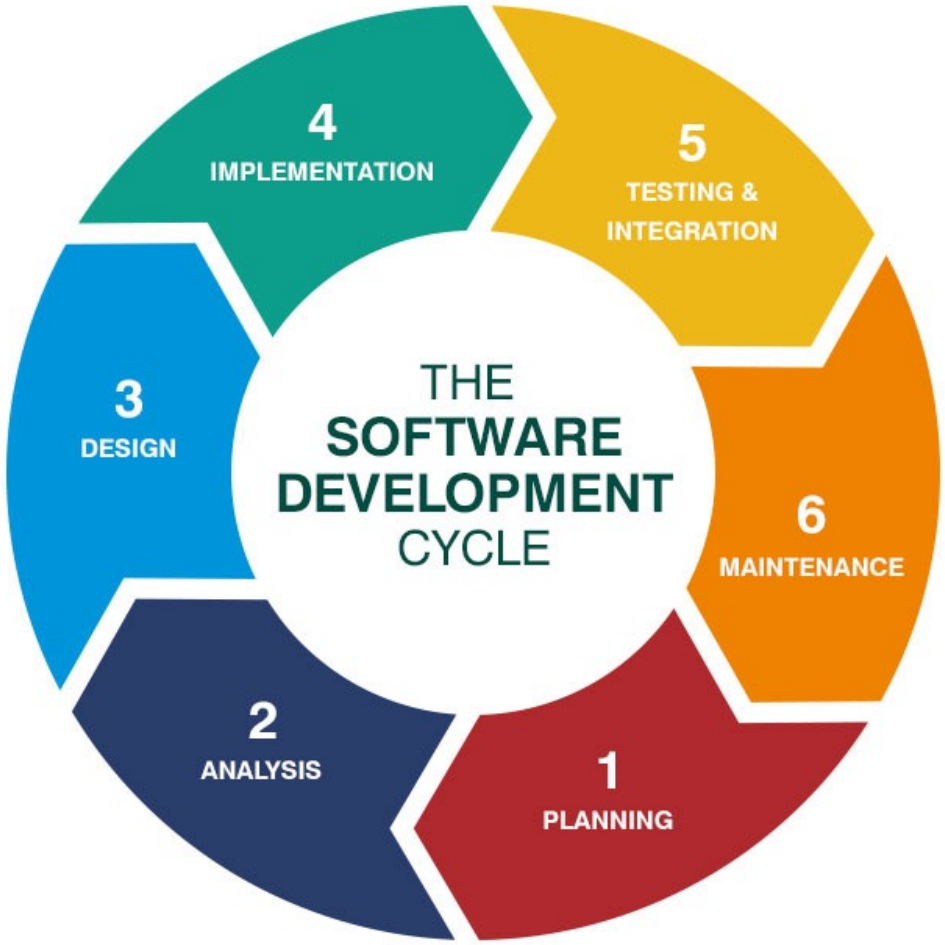
Logic fix
#4 by shadow-blade-X was merged on Nov 13, 2020 • Approved

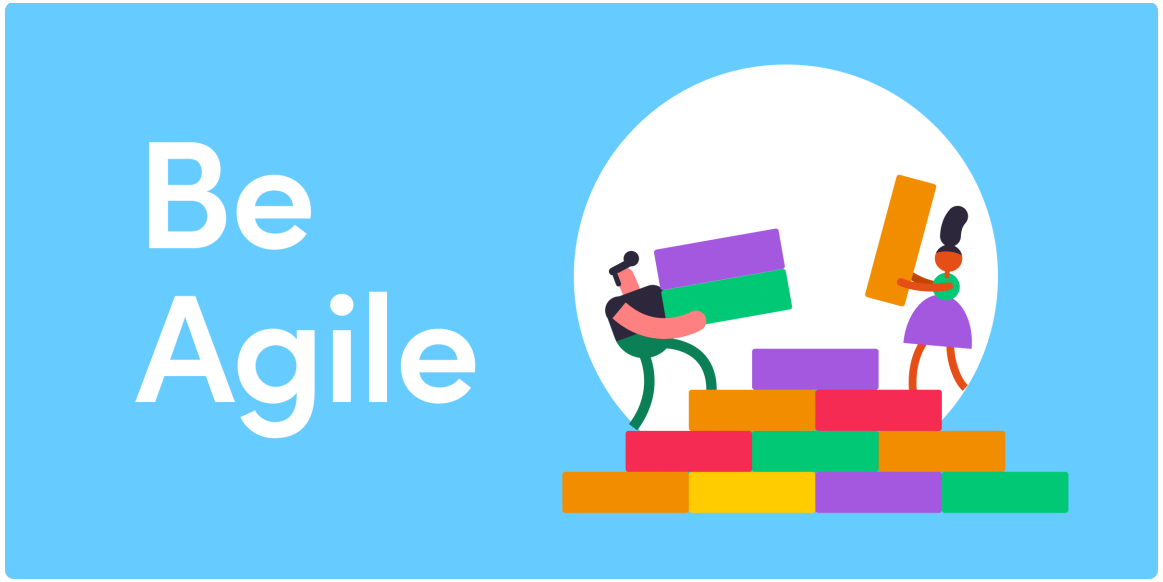
Add comments, Fix filter by ingredient and add test for inventory and shopping list
#3 by shadow-blade-X was merged on Nov 13, 2020 • Approved


andrewvt / github-enhanced-project-board
Projects
ToDo's
Filter cards
Add cards
New issue
Exit fullscreen
Menu
Add column
1 To do
Submit to Webstore
#5 opened by andrewvt
enhancement
2 Ready
Create Manifest
#1 opened by andrewvt
enhancement
Fix js error
#6 opened by andrewvt
bug
3 In progress
Create background js
#3 opened by andrewvt
enhancement
Find the one ring
#8 opened by andrewvt
enhancement
Fix CSS alignment issue
#7 opened by andrewvt
bug
2 QA/Conformance
Create github.js
#4 opened by andrewvt
enhancement
Invent flux capacitor
#9 opened by andrewvt
good first issue
1 Done
Enhance the UI
#2 opened by andrewvt
enhancement
Automated as To do
Manage
Automated as In progress
Manage
Automated as Done
Manage

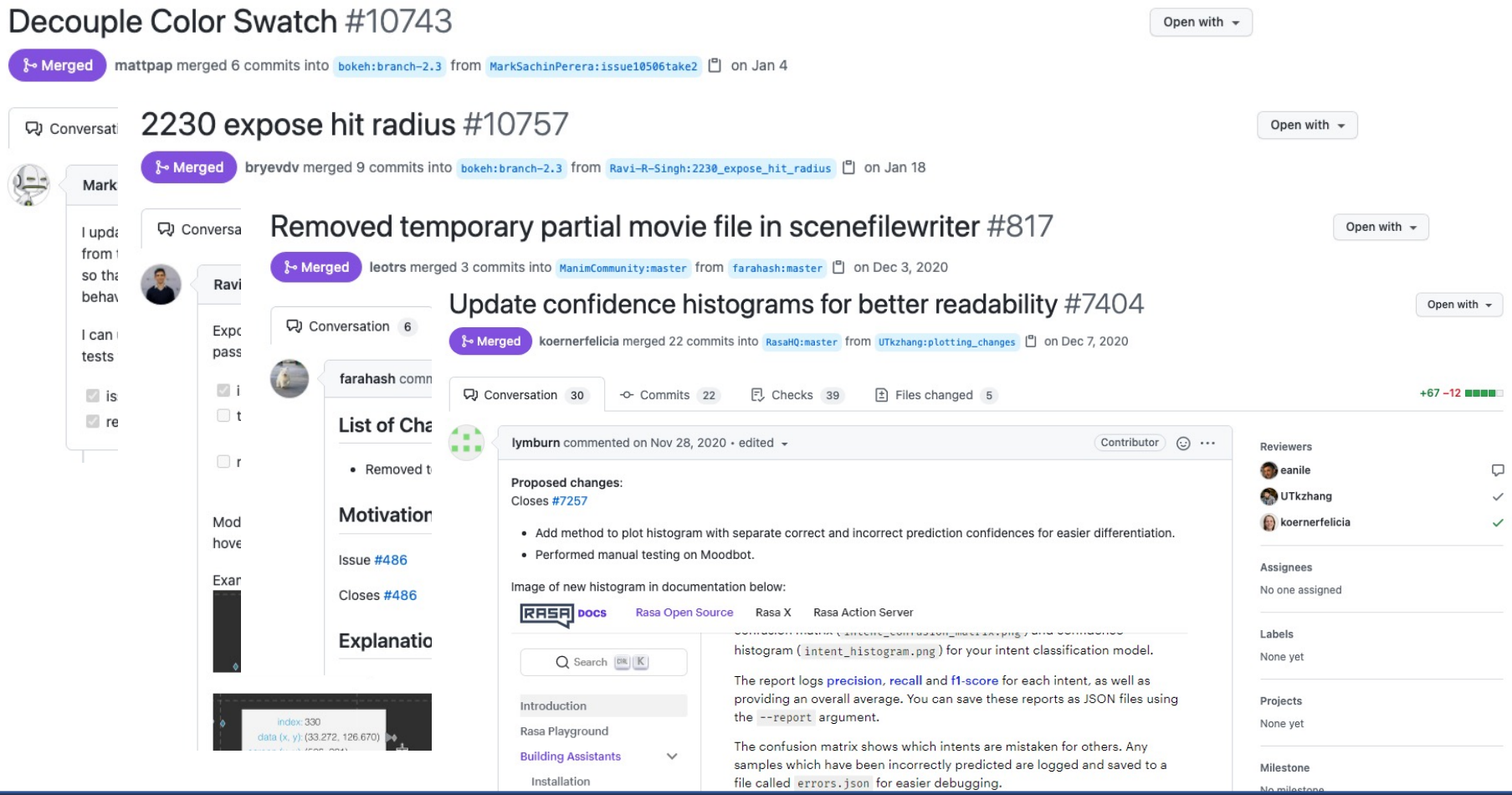

VOTE!
VOTE!
VOTE!
VOTE!
VOTE!
VOTE!
VOTE!
VOTE!
VOTE!

Software Development Lifecycle

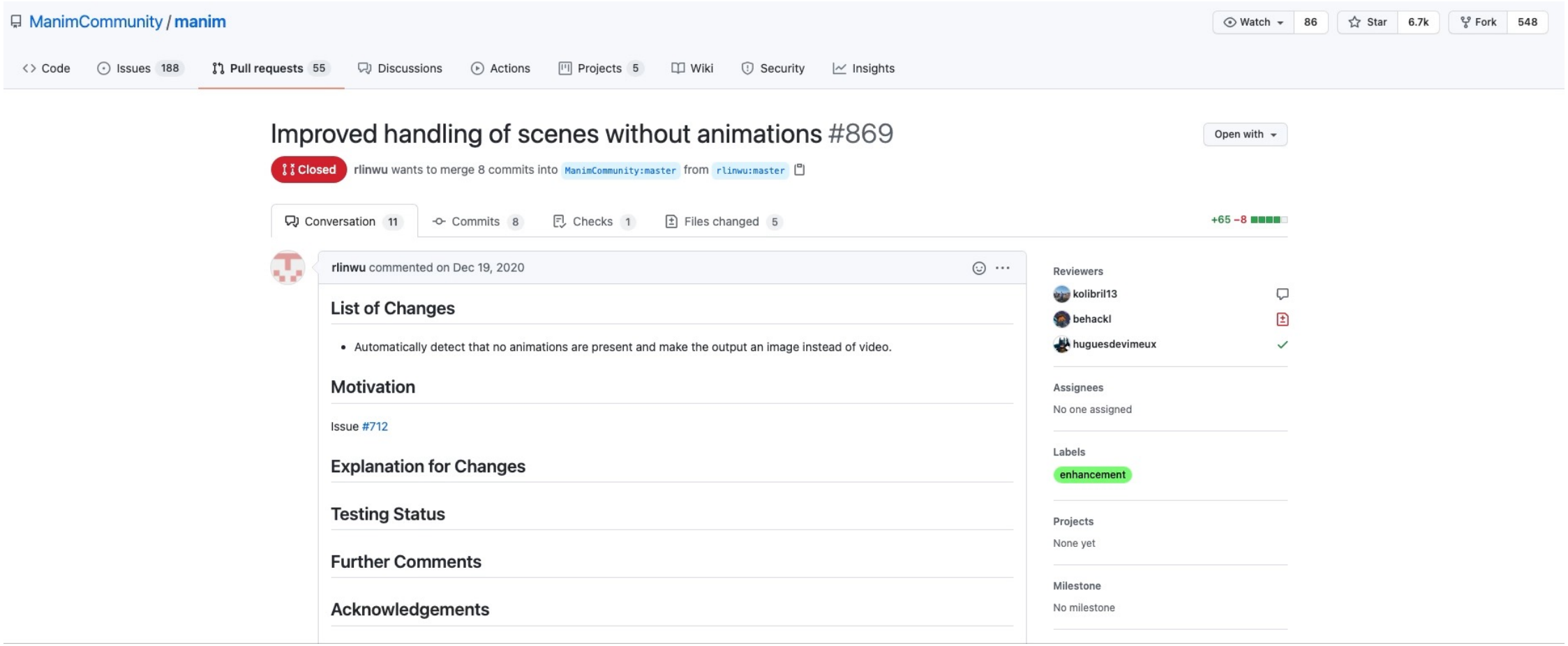




Open-Source Excursion—Showcase (Fall 2020)



Open-Source Excursion—Showcase (Fall 2020)



ECE444-First Class Survey (Fall2021)

Please complete this survey, to be used only for the purposes of our course to help the facilitators understand your expectations and needs.

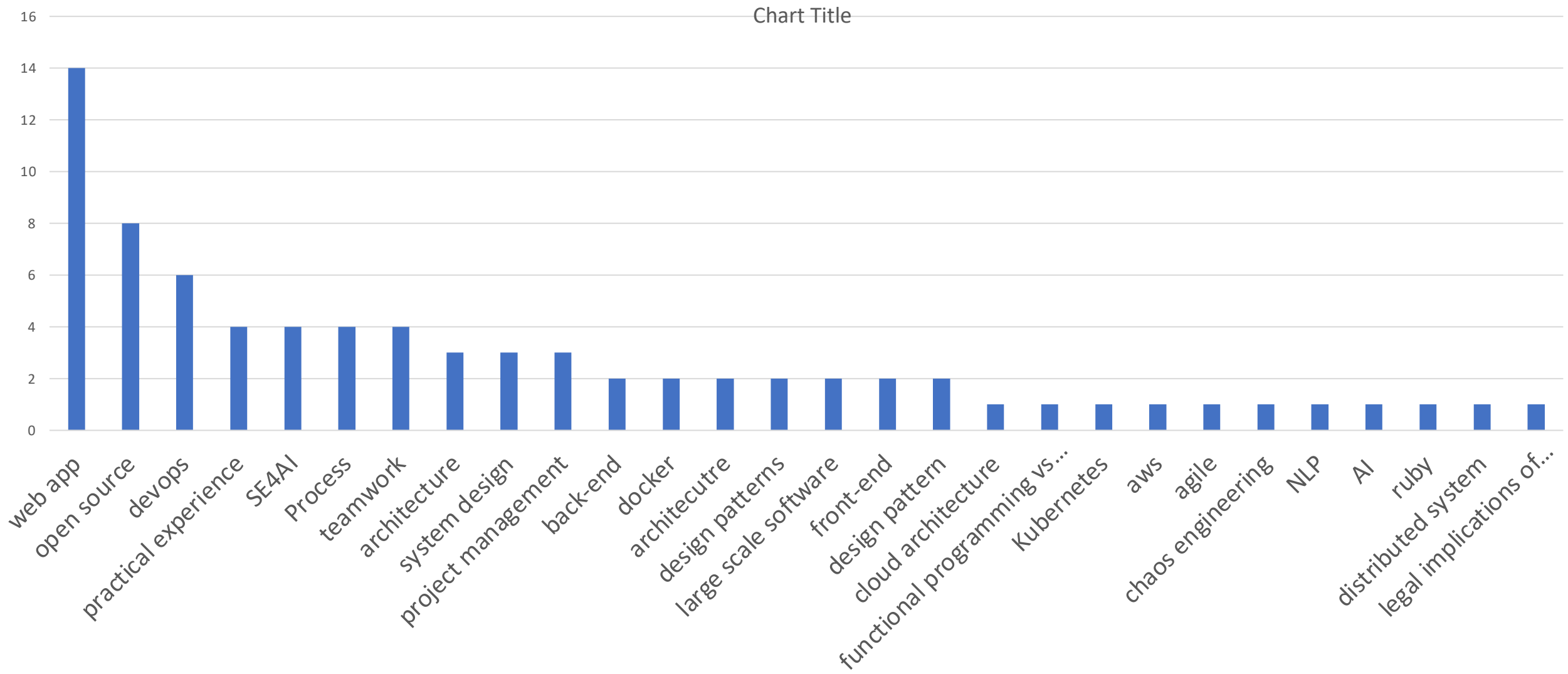
The goals of this survey are:

- Forming groups based on your background and experience.
- Shaping the courses based on
    * your background knowledge
    * your interests
- Identifying experience/interest

We wish you all the best during this uncertain time. Please don't hesitate to reach out if you have any questions. We may also reach out to you based on your responses to this form.

*Note: This form is adapted from The Faculty of Applied Science & Engineering (U of T), Annelise Heinz (University of Oregon), Gray Garmon (University of Texas) & Katie Krummeck (Educational Designer). Some language adapted from Kimberly Rogers (Dartmouth) and Danya Glabau (NYU, Tandon School of Engineering).

(Fall 2020) What do you want to learn?



Guest Lectures

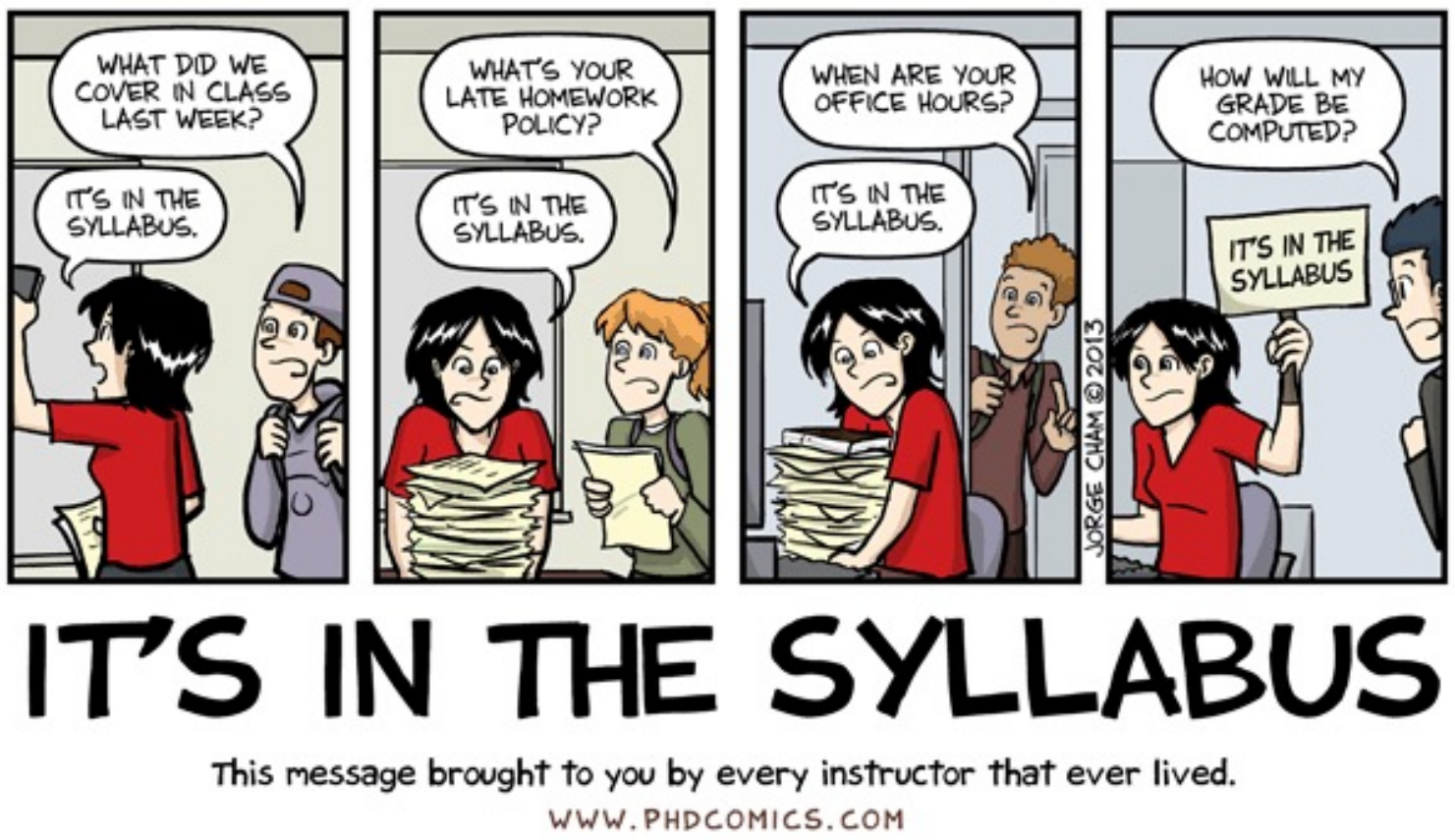
Any Questions?

Syllabus and course mechanics

- https://shuiblue.github.io/UofT-ECE444/
- Tools
  - Quercus: Assignment distribution, hand-in, and grades
  - Git, GitHub, GitHub Classroom: code management
  - Piazza: Discussion board, Q&A

piazza ECE 444H1 F LEC0101

Please post your questions on Piazza, not to TA's personal email



Logistics – Lectures & PRAs

Lectures: Thursday 12:00-15:00 EST
PRA-1: Friday 12:00-15:00
PRA-2: Wednesday 09:00-12:00
PRA-3: Thursday 09:00-12:00

Assignment 1 – fill in the surveys

- To help us to tailor class and form teams

First Class Survey

Publish | Edit | ⋮

There are two surveys:

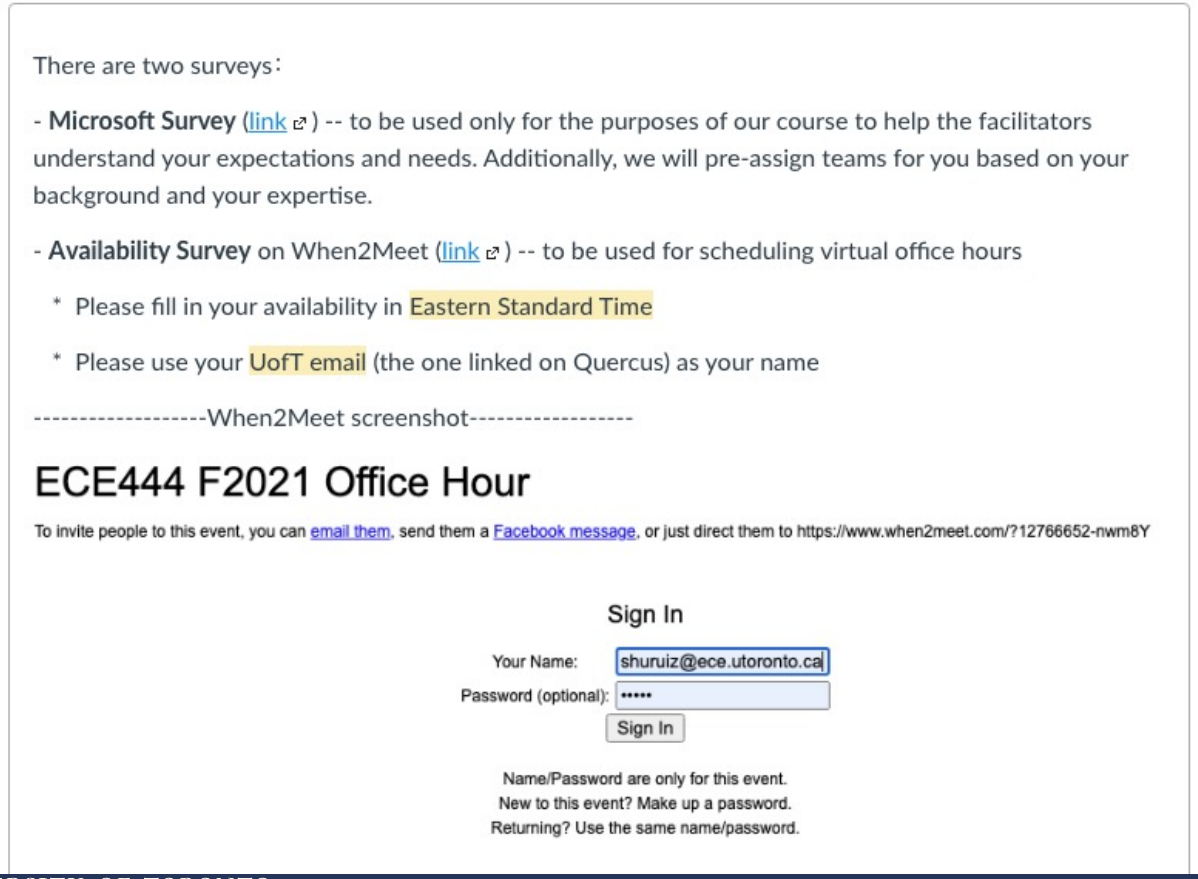
- **Microsoft Survey** (link) -- to be used only for the purposes of our course to help the facilitators understand your expectations and needs. Additionally, we will pre-assign teams for you based on your background and your expertise.

- **Availability Survey** on When2Meet (link) -- to be used for scheduling virtual office hours

  * Please fill in your availability in Eastern Standard Time

  * Please use your UofT email (the one linked on Quercus) as your name

----------When2Meet screenshot----------



Forming Teams

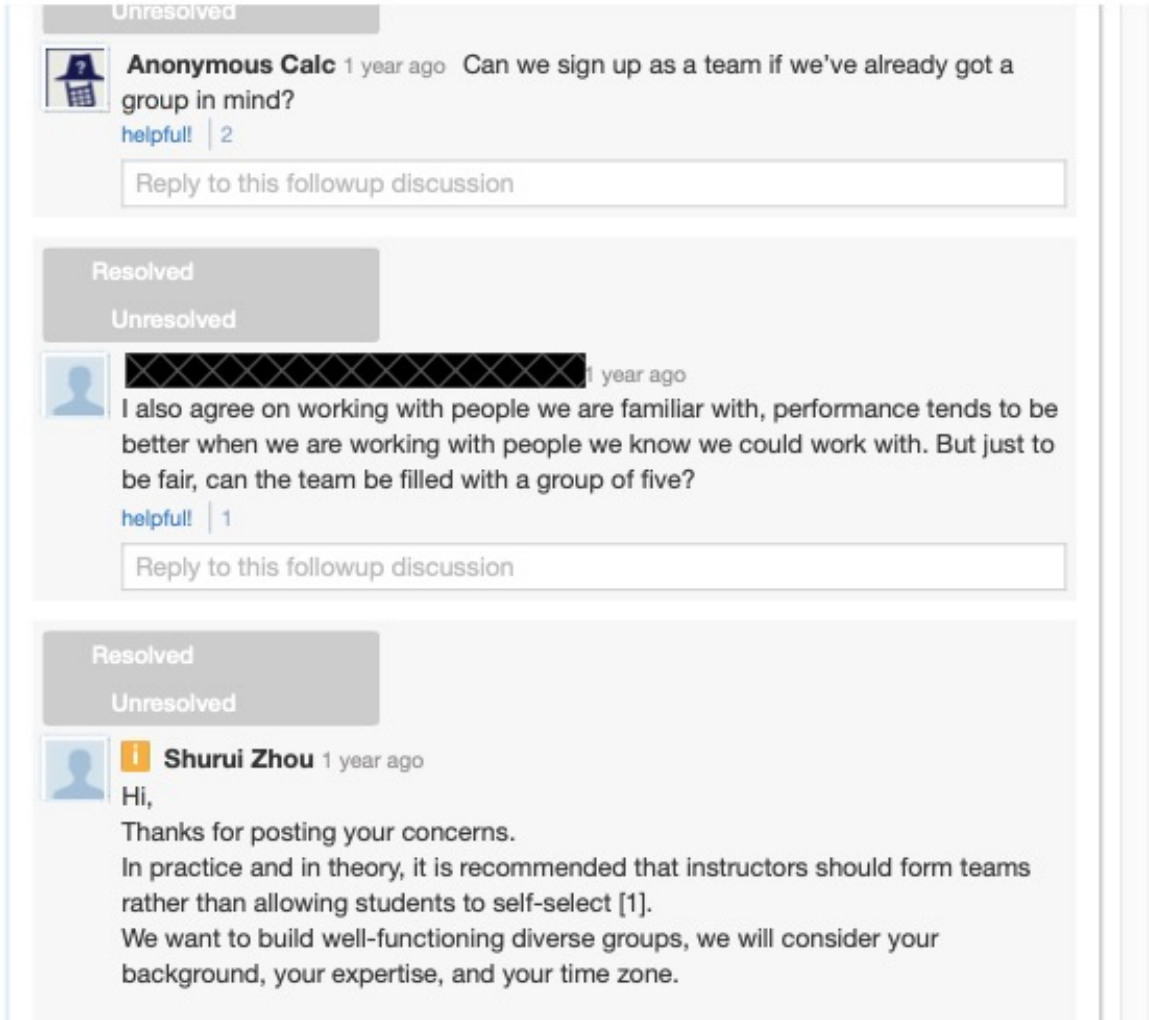
We will form teams rather than allowing students self-select.

- 4-5 students per team
- Criteria: Form teams whose members are diverse in ability levels [1].
- Gaining experience on generating and comparing alternative solutions and resolving conflicts
- We will send out the list of formed teams before Monday (9/13)

[1] Oakley, Barbara, et al. "Turning student groups into effective teams." *Journal of student centered learning* 2.1 (2004): 9-34.

Concerning about pre-assigned teams?

- Check out the Q&A from last year on Quercus.



Common interview conflict resolution questions

Tell me about a time you had a conflict at work.

How do you deal with conflict with a coworker?

Describe a time you disagreed with a manager or supervisor and how you handled the situation.

Tell me about a time you disagreed with a company policy or rule and how you handled the situation.

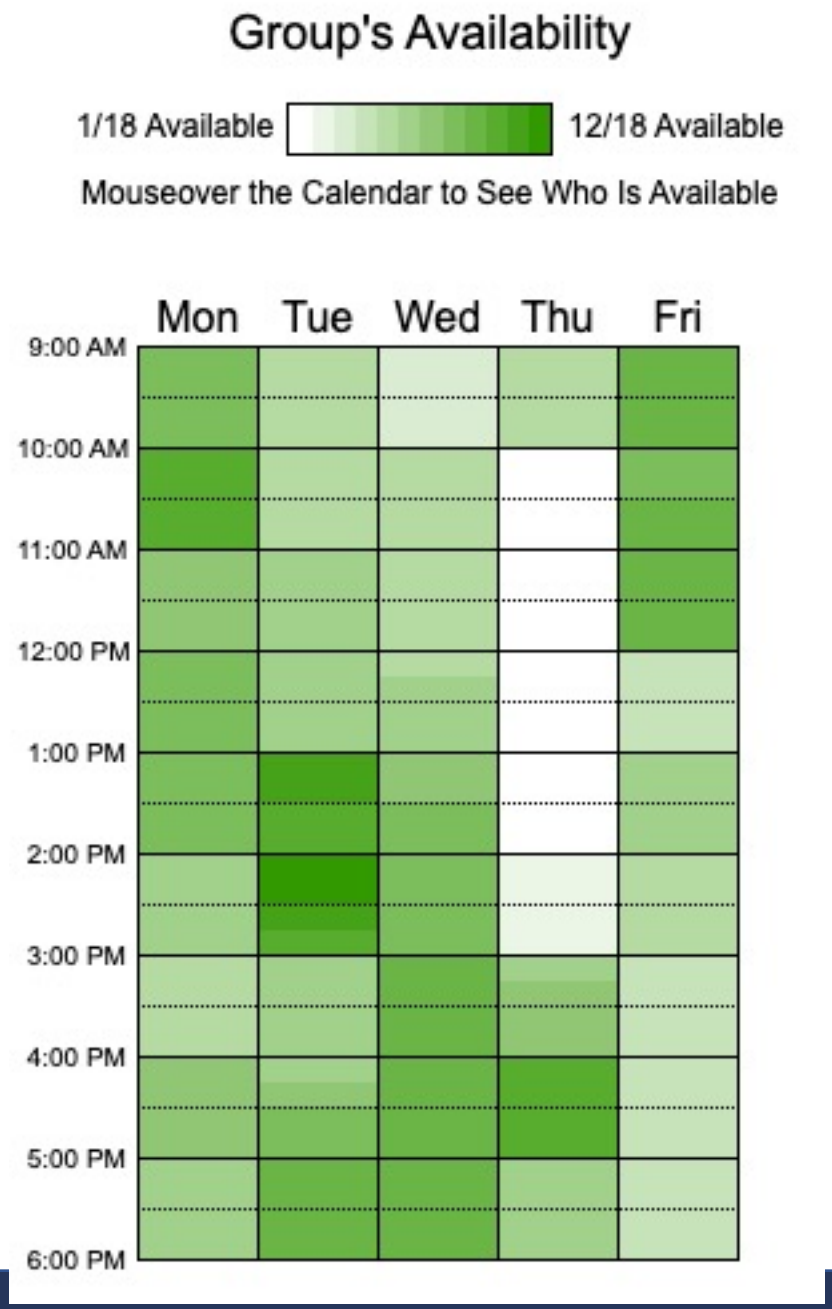
How do you approach diversity in the workplace?

https://ca.indeed.com/career-advice/interviewing/conflict-resolution-interview-questions

Teaching Assistants

Keerthi Nelaturu	keerthi.nelaturu@mail.utoronto.ca
Kunal Dewan	kunal.dewan@mail.utoronto.ca
Enmeng Liu	enmeng.liu@mail.utoronto.ca
Jiayi Sun	jiayisaria.sun@mail.utoronto.ca
Imtihan Ahmed	imtihan.ahmed@mail.utoronto.ca
Martiya Zare Jahromi	martiya.zare@mail.utoronto.ca

Logistics -- Office Hours



Reading and Quizzes

- Reading assignments for some lectures
  - Preparing in-class discussions
  - Background material, case descriptions, possibly also podcast, video, wikipedia

- Short and easy online quizzes on readings, due by start of lecture

Evaluation (under review)

- Web application development (50%)
- Contribute to an open source project (30%)
- Participation in reading quizzes and lab tasks (20%)

Participation

Both quality and quantity are important, quality more than quantity

Professionalism

- Being a professional means you should work well with others
- The best professionals are those who make those around them better
- If you feel someone is not treating you or someone else in a professional manner, you have two options:
  - If you feel you have the standing to do so, speak up!
  - Reach out to the course staff, and we will meet with you privately to discuss it, as well as preserve your anonymity

Academic Honesty

- See web page
- In a nutshell: do not copy, do not lie, do not share or publicly release your solutions
- In group work, be honest about contributions of team members, do not cover for others
- If you feel overwhelmed or stressed, please come and talk to us (see syllabus for other support opportunities)

Peer evaluation for every milestone

On Team Citizenship

The rating refers to team citizenship, not the amount of expertise that a person brought to the team. For example, it is perfectly fine if the work was not balanced equally or one team member took easier or less technical tasks if the team together agreed to it.

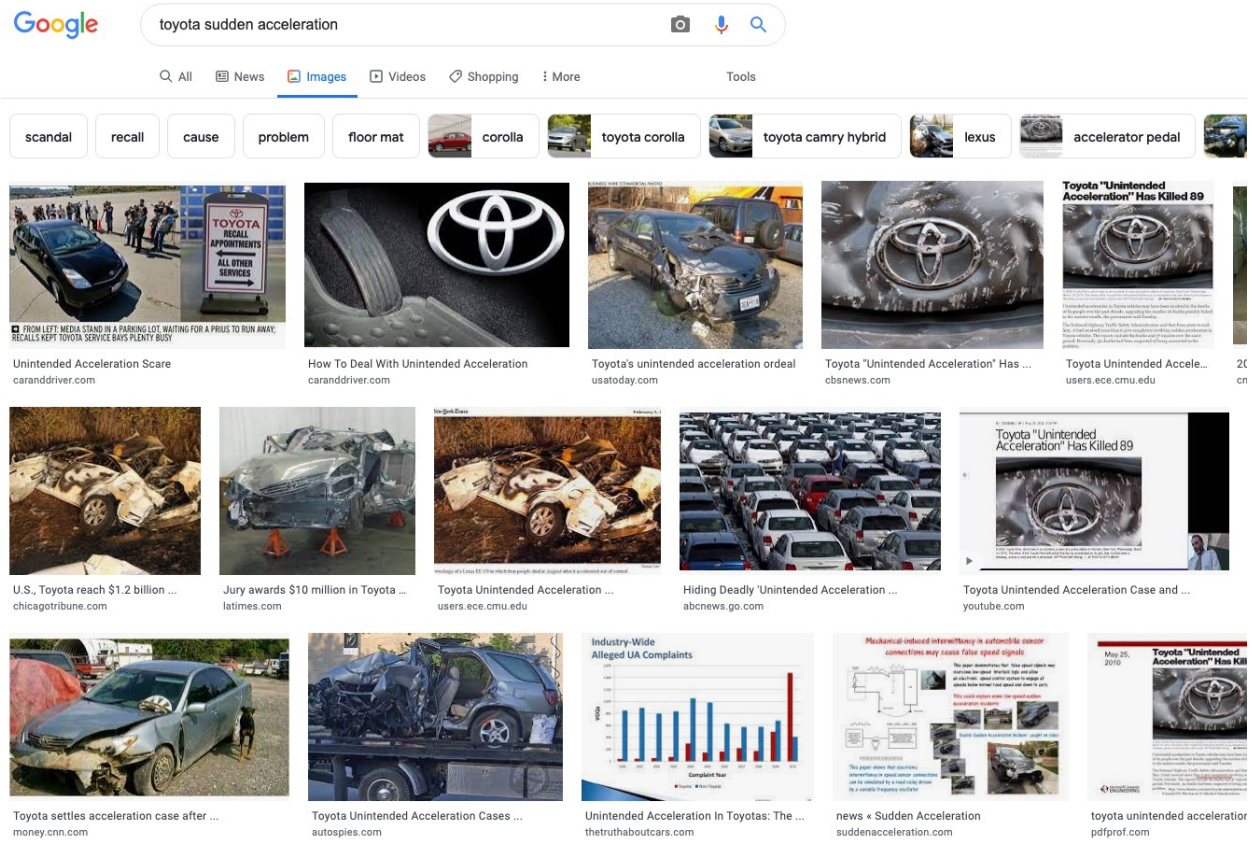
Here are a couple of questions that can guide your evaluation of team citizenship:

- Has the student attended team meetings?
- Has the student made a serious effort at assigned work before the team meetings?
- Has the student notified the team if they would not be able to attend a meeting or fulfill a responsibility?
- Does the student attempt to make contributions in group meetings?
- Does the student listen to their teammates' ideas and opinions respectfully and give them careful consideration?
- Does the student cooperate with the group effort?

Agenda for Today

- Introduction of the course
- Introduction of Software Engineering
- Process and Team

A bad code, a bug could cost more than the victory



Perhaps 89 deaths, hundreds of serious injury lawsuits

- $1.6B class action settlement
- Jury found system defective – Toyota “acted in reckless disregard”
- Many of issues were SW, but also a HW problem

A bad code, a bug could cost more than the victory

EE Times

DESIGNLINES | AUTOMOTIVE DESIGNLINE

Toyota Case: Single Bit Flip That Killed

By Junko Yoshida 10.25.2013 0

During the trial, embedded systems experts who reviewed Toyota's electronic throttle source code testified that they found Toyota's source code defective, and that it contains bugs — including bugs that can cause unintended acceleration.

"We did a few things that NASA apparently did not have time to do," Barr said. For one thing, by looking within the real-time operating system, the experts identified "unprotected critical variables." They obtained and reviewed the source code for the "sub-CPU," and they "uncovered gaps and defects in the throttle fail safes."

The experts demonstrated that "the defects we found were linked to unintended acceleration through vehicle testing," Barr said. "We also obtained and reviewed the source code for the black box and found that it can record false information about the driver's actions in the final seconds before a crash."

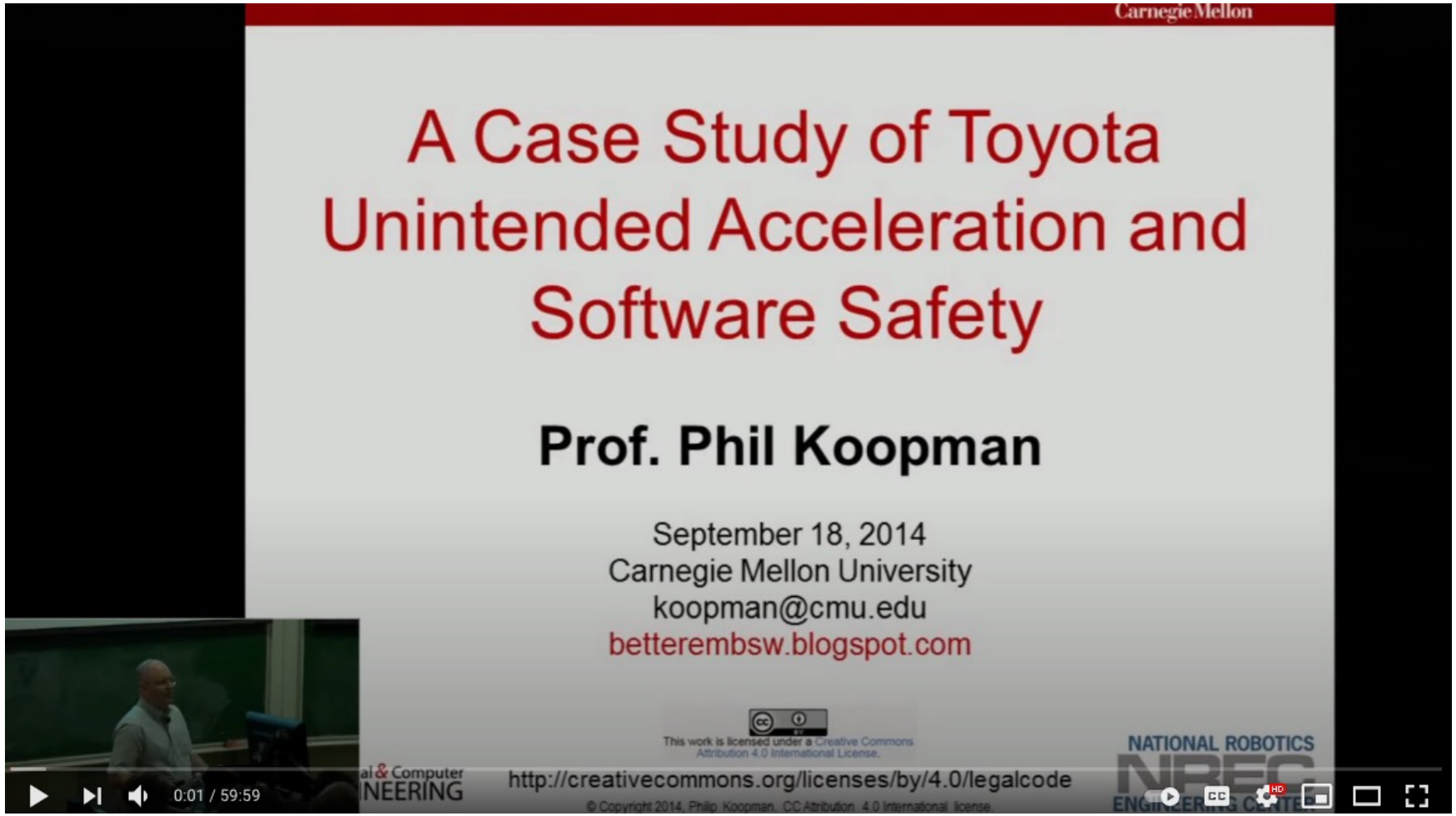
Stack overflow and software bugs led to memory corruption, he said. And it turns out that the crux of the issue was these memory corruptions, which acted "like ricocheting bullets."

When asked if the whole case for unintended acceleration could be pinned on the task X death, Barr replied, "The task X death in combination with other task deaths." There are

Barr also said more than half the dozens of tasks' deaths studied by the experts in their experiments "were not detected by any fail-safe."

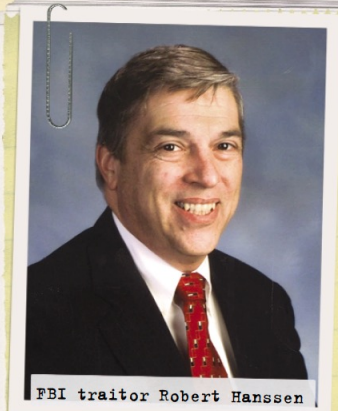
https://www.eetimes.com/toyota-case-single-bit-flip-that-killed/

Supplementary material



https://www.youtube.com/watch?v=DKHa7rxkvK8

SPECIAL REPORT



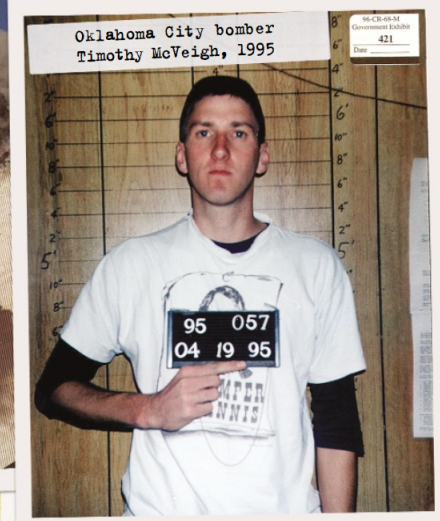
FBI traitor Robert Hanssen

In the early 1990s, Russian mobsters partnered with Italian Mafia families in Newark, N.J., to skim millions of dollars in federal and New Jersey state gasoline and diesel taxes. Special Agent Larry Depew set up an undercover sting operation under the direction of Robert J. Chiaradio, a supervisor at the Federal Bureau of Investigation's Washington, D.C., headquarters.

Who Killed the Virtual Case File?

By Harry Goldstein

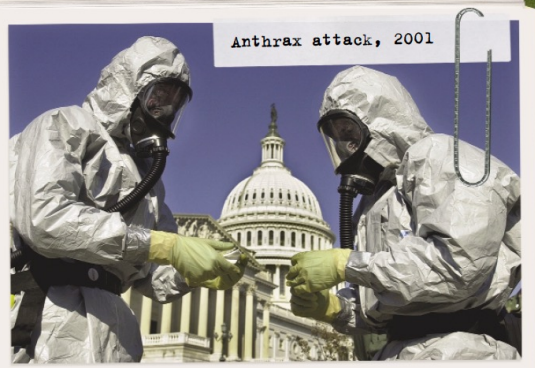
9/11



Oklahoma City bomber Timothy McVeigh, 1995

CLOCKWISE FROM TOP: FBI/AP PHOTO; AP PHOTO; CHAO SOI CHEONG/AP PHOTO

D.C. sniper John Allen Muhammad, 2002



Anthrax attack, 2001

FROM LEFT: STEVE HELBER/AP PHOTO; KENNETH LAMBERT/AP PHOTO

How the FBI blew more than $100 million on case-management software it will never use

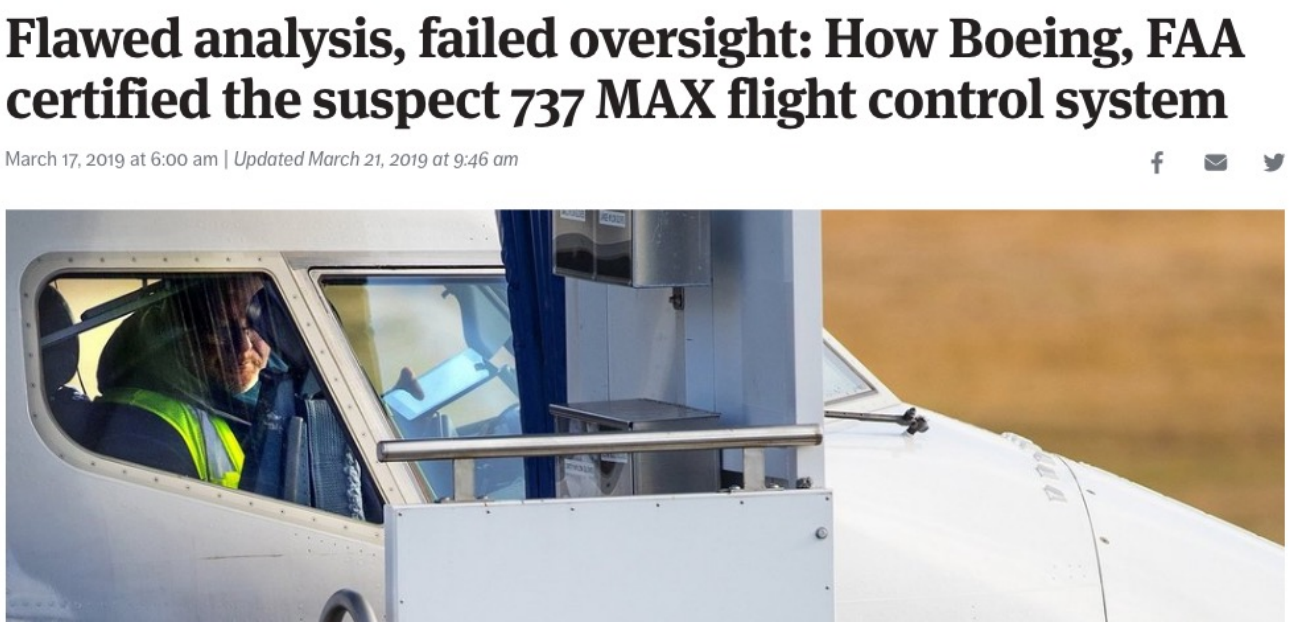
Depew collected reams of evidence from wiretaps, interviews, and financial transactions over the course of two and a half years. Unfortunately, the FBI couldn't provide him with a database program that would help organize the information, so Depew wrote one himself. He used it to trace relationships between telephone calls, meetings, surveillance, and interviews, but he could not import information from other investigations that might shed light on his own. So it wasn't until Depew mentioned the name of a suspect to a colleague that he obtained a briefcase that his friend had been holding since 1989.

"When I opened it up, it was a treasure trove of information about who's involved in the conspiracy, including the Gambino family, the Genovese family, and the Russian components. It listed percentages of who got what, when people were supposed to pay, the number of gallons. It became a central piece of evidence," Depew recalled during an interview at the FBI's New Jersey Regional Computer Forensic Laboratory, in Hamilton, where he is the director. "Had I not just picked up the phone and called that agent, I never would have gotten it."

A decade later, Depew's need to share information combined with his do-it-yourself database skills and connection to his old supervisor, Chiaradio, would land him a job managing his first IT project—the FBI's Virtual Case File.

Depew's appointment to the FBI's VCF team was an auspicious start to what would become the most highly publicized software failure in history. The VCF was supposed to automate the FBI's paper-based work environment, allow agents and intelligence analysts to share vital investigative information, and replace the obsolete Automated Case Support (ACS) system. Instead, the FBI claims, the VCF's contractor, Science Applications International Corp. (SAIC), in San Diego, delivered 700 000 lines of code so bug-ridden and functionally off target that this past April, the bureau had to scrap the US $170 million project, including $105 million worth of unusable code. However, various government and independent reports show that the FBI—lacking IT management and technical expertise—shares the blame for the project's failure.

In a devastating 81-page audit, released in 2005, Glenn A. Fine, the U.S. Department of Justice's inspector general, described eight factors that contributed to the VCF's



Flawed analysis, failed oversight: How Boeing, FAA certified the suspect 737 MAX flight control system

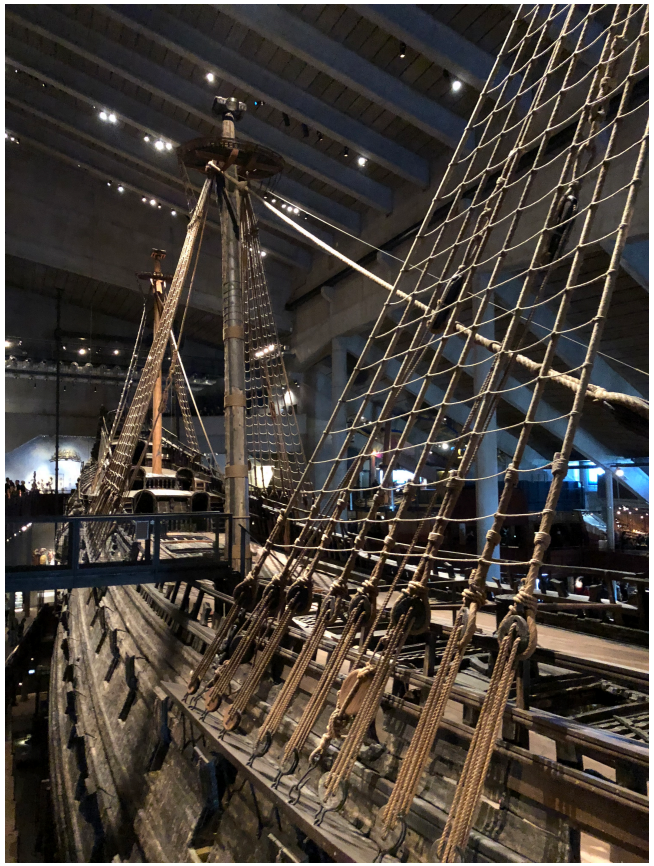
March 17, 2019 at 6:00 am | *Updated March 21, 2019 at 9:46 am*

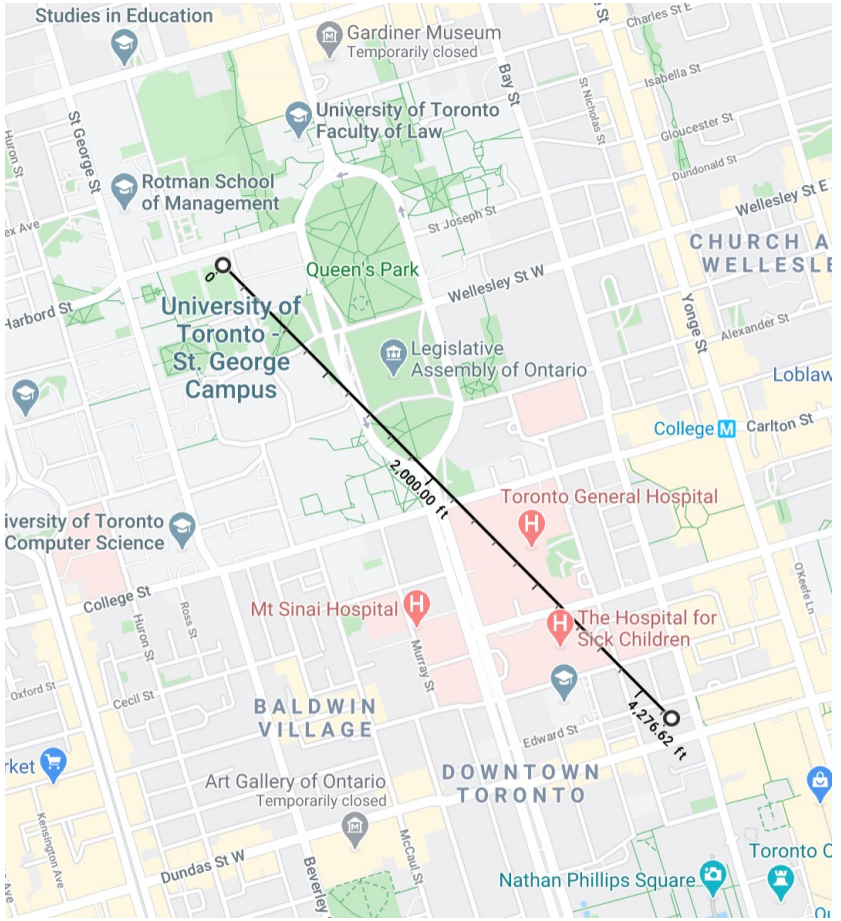
- One pilot said it was “unconscionable that a manufacturer, the FAA (Federal Aviation Administration), and the airlines would have pilots flying an airplane without adequately training, or even providing **available resources and sufficient documentation** to understand the highly complex systems that differentiate this aircraft from prior models”

https://www.theverge.com/2019/5/2/18518176/boeing-737-max-crash-problems-human-error-mcas-faa

https://www.seattletimes.com/business/boeing-aerospace/failed-certification-faa-missed-safety-issues-in-the-737-max-system-implicated-in-the-lion-air-crash/

A failure of project management -- Swedish Vasa warship





Why This 17th-Century Warship Was a Disastrous Failure

https://www.youtube.com/watch?v=1a0PIhMpfLU

Vasa syndrome

From Wikipedia, the free encyclopedia

Vasa syndrome is a term used in both management and marketing circles referring to problems in communication and management affecting projects, sometimes causing them to fail. Its basis lies with the Swedish 17th-century warship *Vasa*, a ship that sank on its maiden voyage because it was too unstable.

The disaster of the *Vasa* has been interpreted by management experts to have been caused by problems with communication, goal setting, and adaptability. The sinking of *Vasa* has also been used as an example for business managers on how to learn from previous mistakes.[1]

What happened? (Vasa Sinking)

- Changing shipbuilding orders
- No specifications for modified keel
- Shifting armaments requirements

Requirements

- Shipwright's death

Teams

- No way to calculate stability, stiffness, or sailing characteristics

Metrics

- Failed pre-launch stability tests

QA

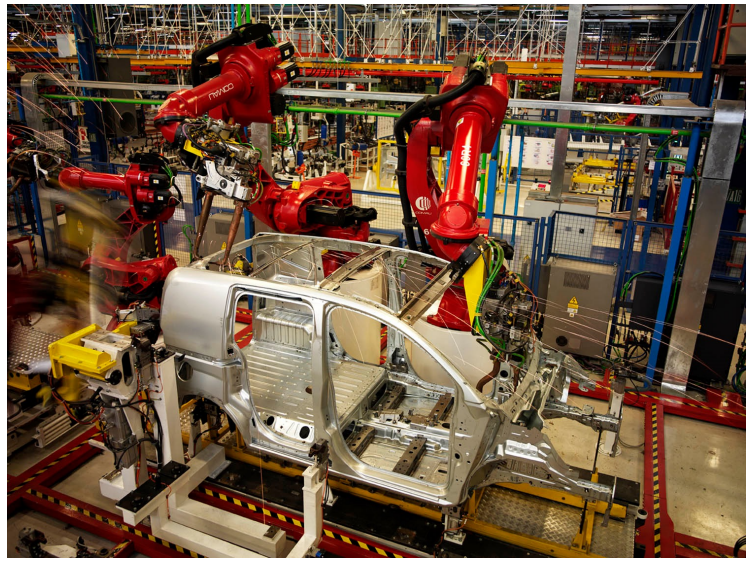
Software Engineering

What is **engineering**? And how is it different from

hacking/programming?

Producing a car/bridge

- Estimable costs and risks
- Expected results
- High quality
- Separation between plan and production
- Simulation before construction
- Quality assurance through measurement
- Potential for automation



Software Engineering?

*„The Establishment and use of sound **engineering principles** in order to obtain **economical** software that is **reliable** and works **efficiently** on **real** machines.”*

[Bauer 1975, S. 524]

"**Software engineering** is the branch of computer science that creates practical, cost-effective solutions to computing and information processing problems, preferentially by applying scientific knowledge, developing software systems in the service of mankind.

Software engineering entails making **decisions** under constraints of limited time, knowledge, and resources. […]

Engineering quality resides in engineering judgment. […]

Quality of the software product depends on the engineer's faithfulness to the engineered artifact. […]

Engineering requires reconciling conflicting constraints. […]

Engineering skills improve as a result of careful systematic reflection on experience. […]

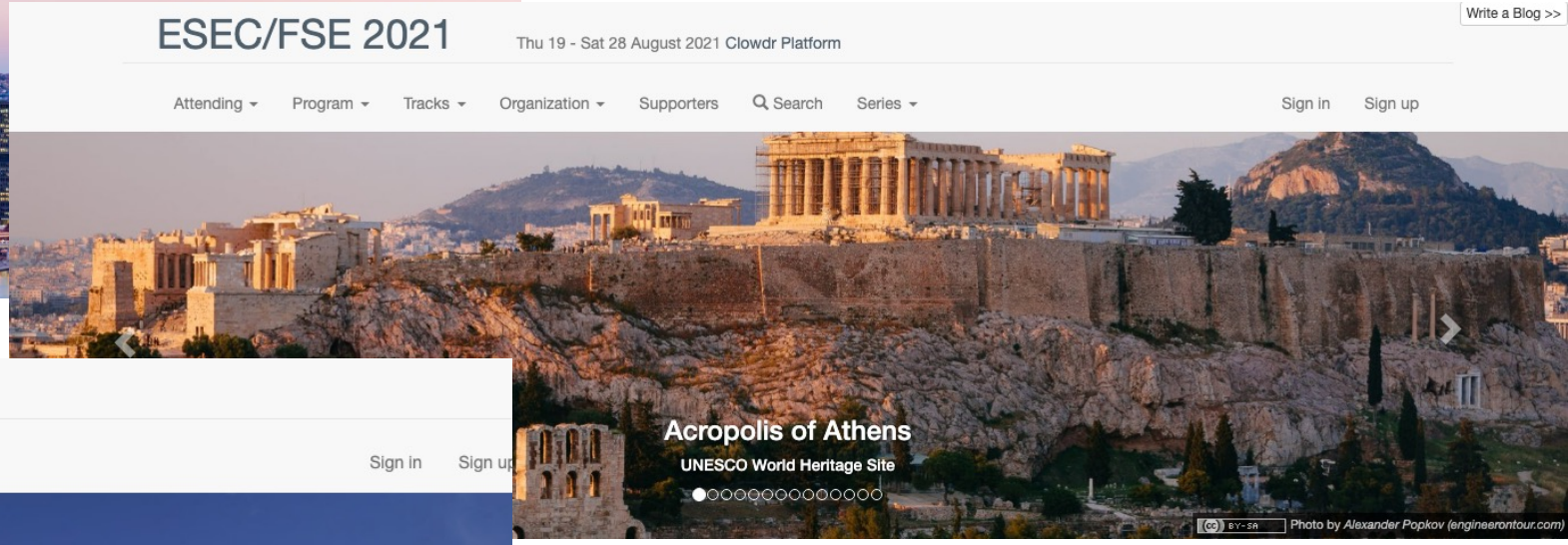
Costs and time constraints matter, not just capability. […]

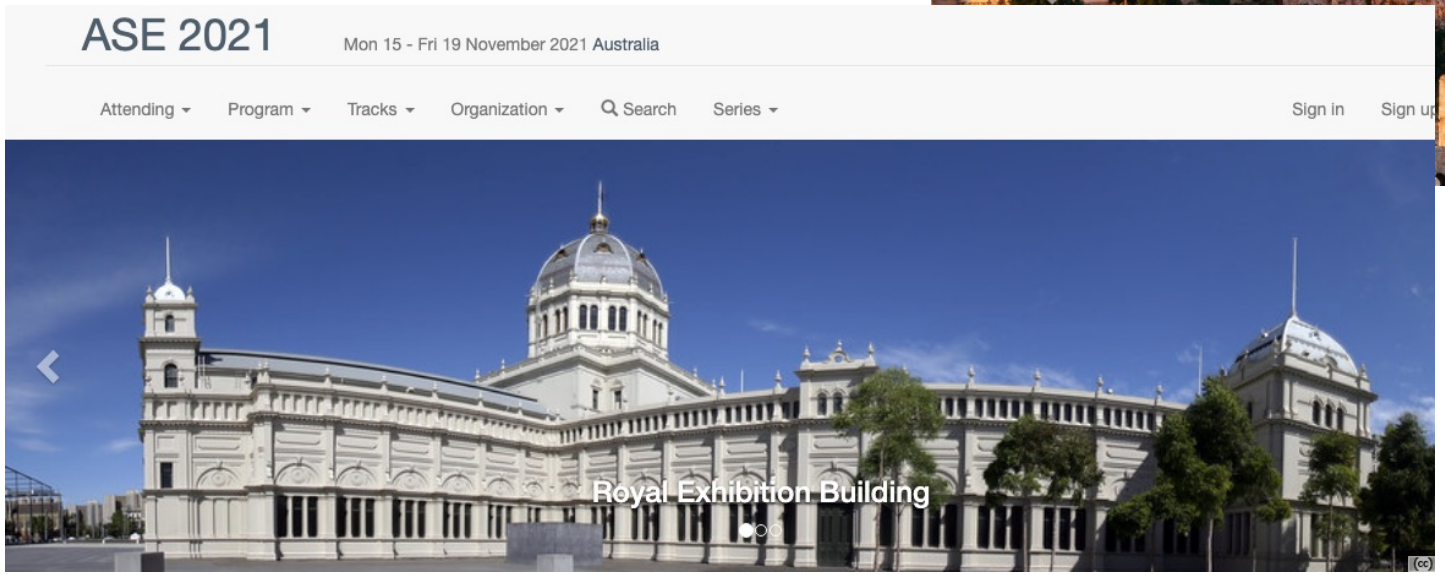
Software Engineering for the 21st Century: A basis for rethinking the curriculum Manifesto, CMU-ISRI-05-108

1968 NATO Conference on Software Engineering

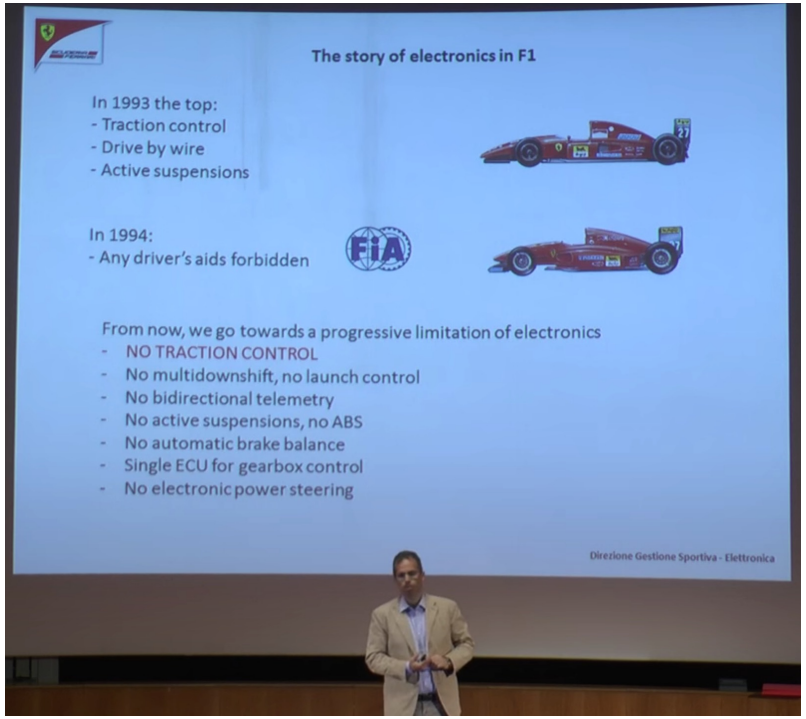
- international experts on computer software who agreed on defining best practices for software grounded in the application of engineering.


Write a Blog >>
May 21–29, 2022, Pittsburgh, PA, USA
44th International Conference on Software Engineering (ICSE 2022)
Tracks
Organization
Sponsorship
Search
Series
Sign in
Sign up
Pittsburgh, USA
Welcome to ICSE 2022 in Pittsburgh!


Write a Blog >>
ESEC/FSE 2021
Thu 19 - Sat 28 August 2021 Clowdr Platform
Attending
Program
Tracks
Organization
Supporters
Search
Series
Sign in
Sign up
Acropolis of Athens
UNESCO World Heritage Site
Photo by Alexander Popkov (engineerontour.com)


ASE 2021
Mon 15 - Fri 19 November 2021 Australia
Attending
Program
Tracks
Organization
Search
Series
Sign in
Royal Exhibition Building

International Conference in Software Engineering



ICSE 2015 'Software Engineering in Ferrari F1'

ICSE 2018 'The Language as a Software Engineer' (Margaret Hamilton)

Agenda for Today

- Introduction of the course
- Introduction of Software Engineering
- Process and Team

Introduction to Process

ECE444 Software Engineering (Fall 2021)

The Edward S. Rogers Sr. Department
of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Learning Goals

- Recognize the Importance of process
- Understand the difficulty of measuring progress
- Use milestones for planning and progress measurement

2013

- 2M people working on 300K software projects in the US
- 1/3 - 2/3 exceed schedule and budget targets before delivery
- Of the most expensive software projects, about half will eventually be canceled for being out of control.

https://ptgmedia.pearsoncmg.com/images/9781572316218/samplepages/9781572316218.pdf

Software projects succeed or fail based on how carefully they are planned and how deliberately they are executed

Process

How to develop software?

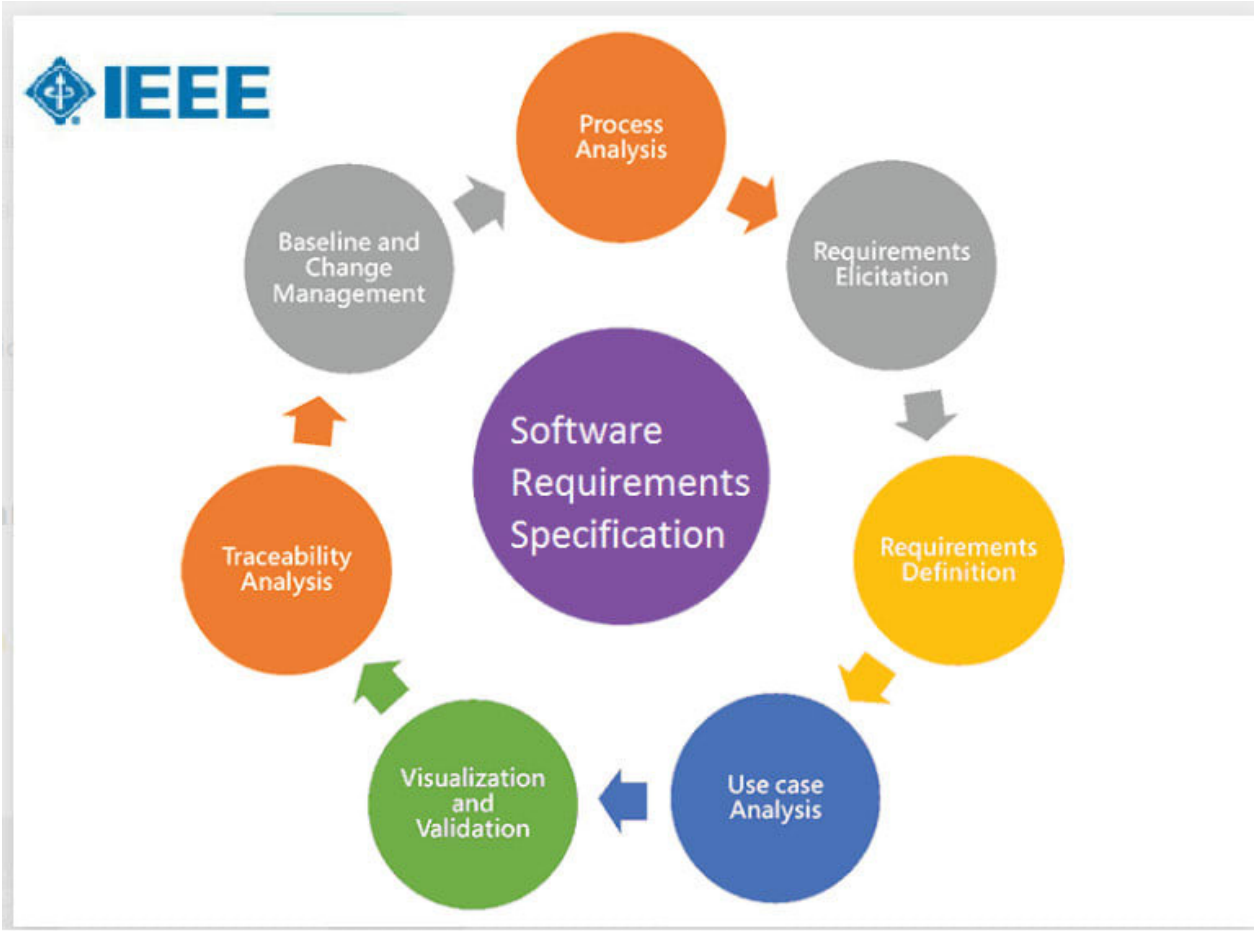
1. Discuss the software that needs to be written
2. Write some code
3. Test the code to identify the defects
4. Debug to find causes of defects
5. Fix the defects
6. If not done, return to step 1

Software Process

The set of activities and associated results that produce a software product

Example of Process Decisions

- Writing down all requirements

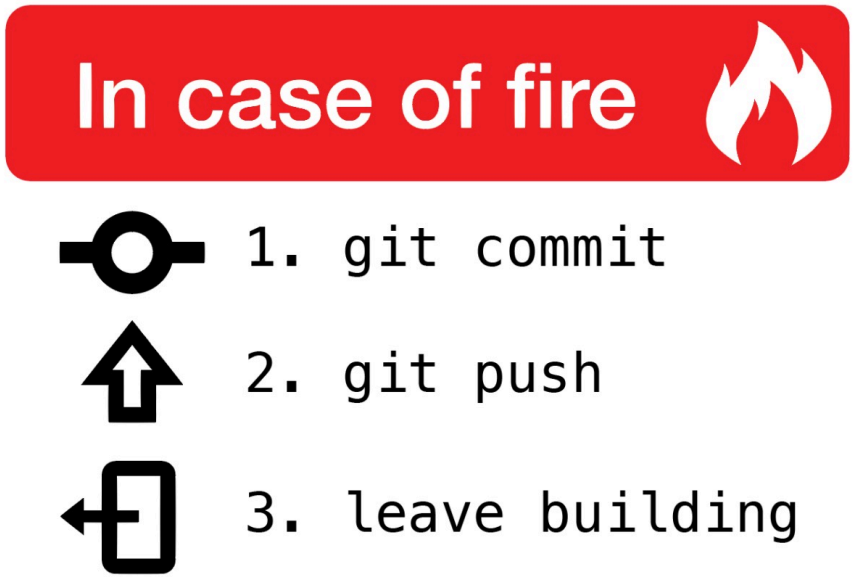


Example of Process Decisions

- Writing down all requirements
- Require approval for all changes to requirements

Example of Process Decisions

- Writing down all requirements
- Require approval for all changes to requirements
- Use version control for all changes

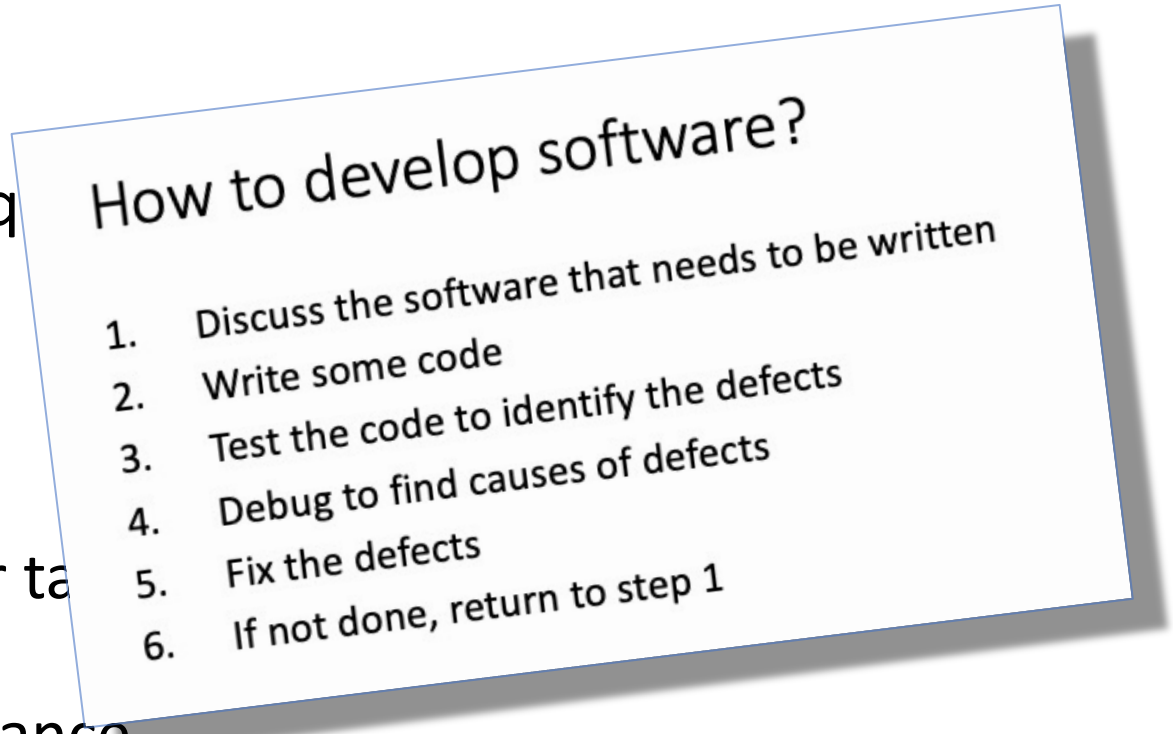


Example of Process Decisions

- Writing down all requirements
- Require approval for all changes to requirements
- Use version control for all changes
- Track all reported bugs
- Review requirements and code
- Break down development into smaller tasks and schedule and monitor them
- Planning and conducting quality assurance
- Have daily status meetings
- Use Docker containers to push code between developers and operation

Example of Process Decisions

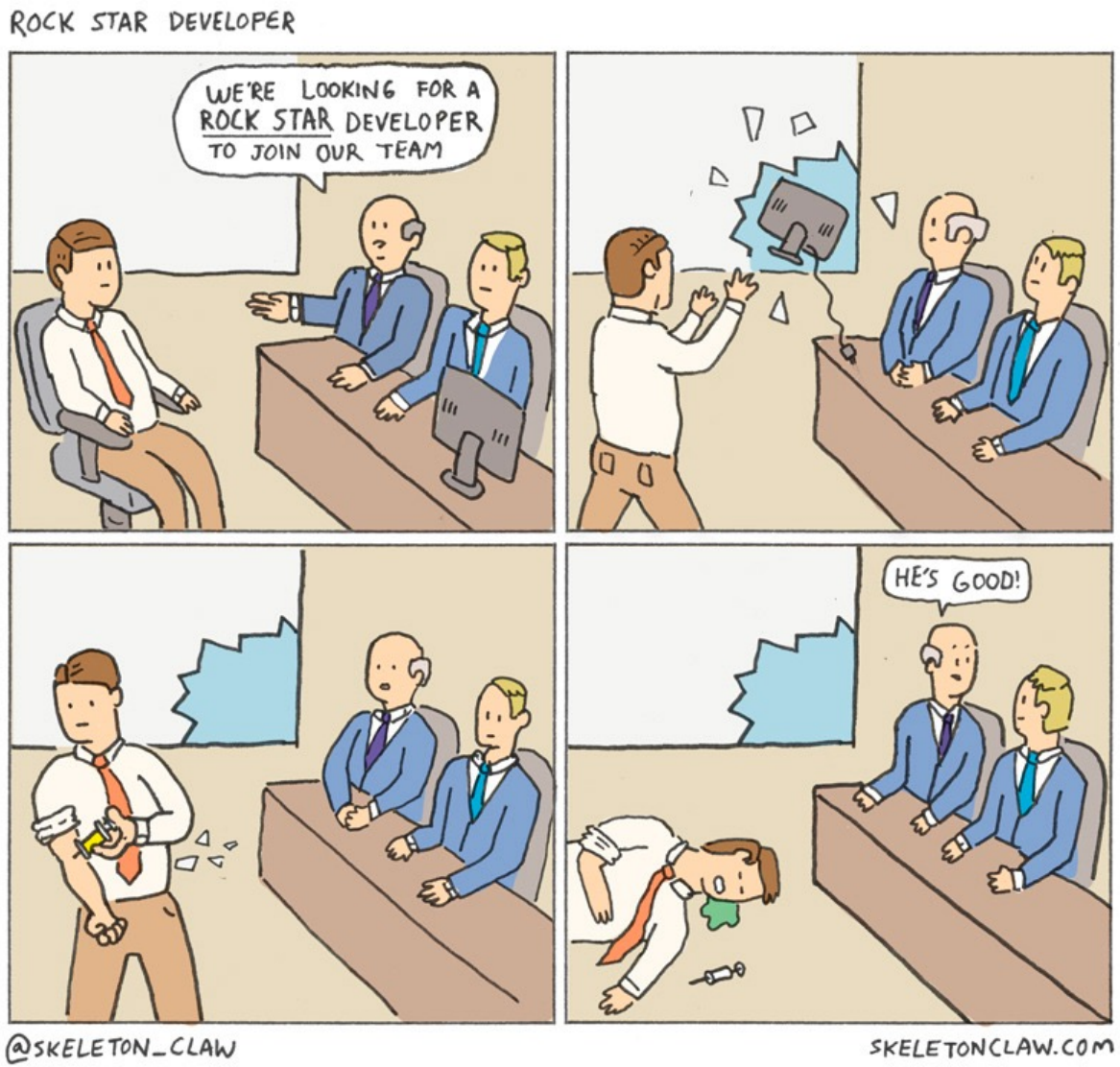
- Writing down all requirements
- Require approval for all changes to req
- Use version control for all changes
- Track all reported bugs
- Review requirements and code
- Break down development into smaller ta
  them
- Planning and conducting quality assurance
- Have daily status meetings
- Use Docker containers to push code between developers and operation



The word “process” was viewed as negative...

10X Engineers

- Aka “rock-star”, “ninja”



Shekhar Kirani
@skirani

10x engineers

Founders if you ever come across this
engineers, grab them. If you have a 10x
of your first few engineers, you increas
startup success significantly.

OK, here is a tough question.

How do you spot a 10x engineer?

6:02 AM · Jul 11, 2019 · Twitter Web App

1.3K Retweets **4K** Quote Tweets **4.6K** Likes

Shekhar Kirani @skirani · Jul 11, 2019
Replying to @skirani
1. 10x engineers hate meetings. They think it is
obvious things are being discussed. They atter
manager has called for a "Staff meeting" to dis
status.

50 187 700

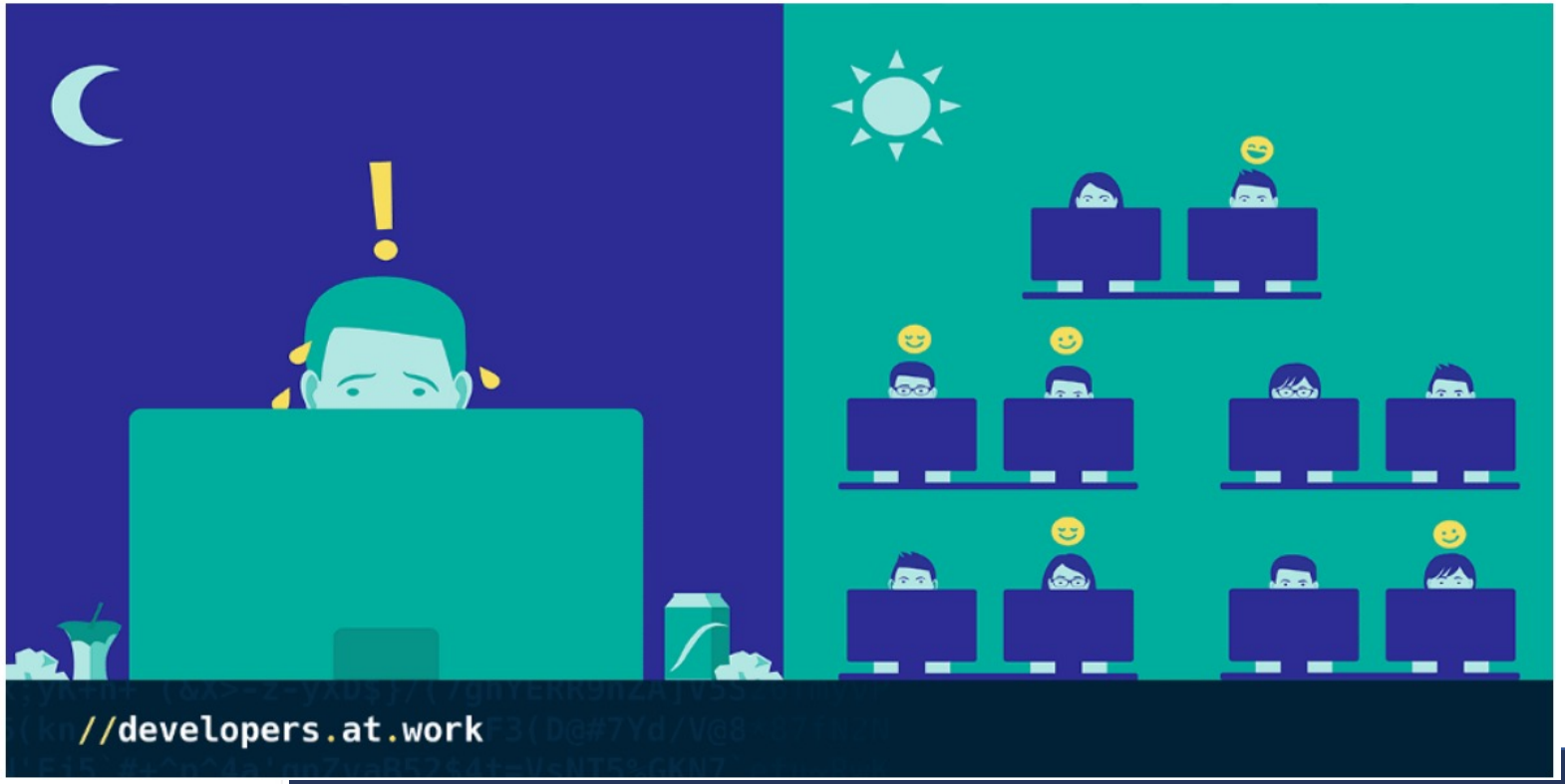
Shekhar Kirani @skirani · Jul 11, 2019
2. Timings in the office for 10x engineers is hig
work when very few folks are around. If there is
meeting, they are not visible. Most of them are
come late to the office.

36 124 638

Shekhar Kirani @skirani · Jul 11, 2019
3. 10x engineers laptop screen background color is typically black (they
always change defaults). Their keyboard keys such as i, f, x are usually

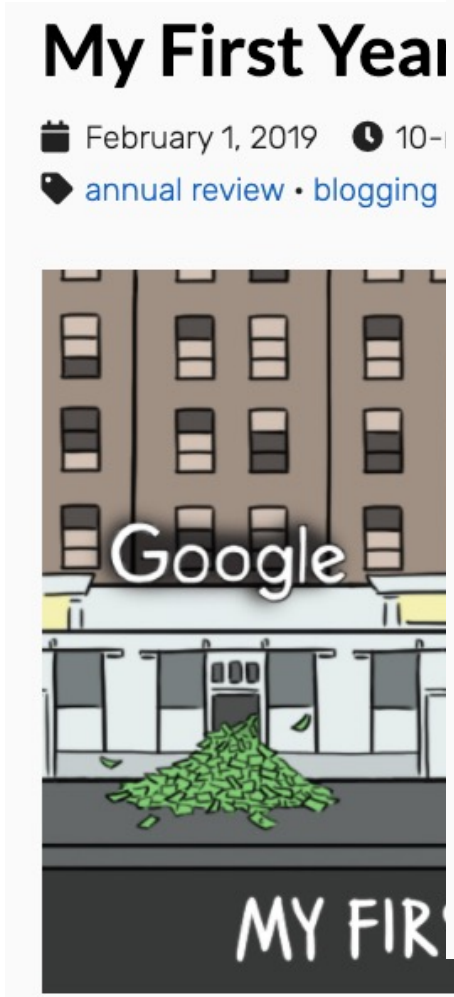
10x-ing Your Team: The End of Superstar Developer Culture

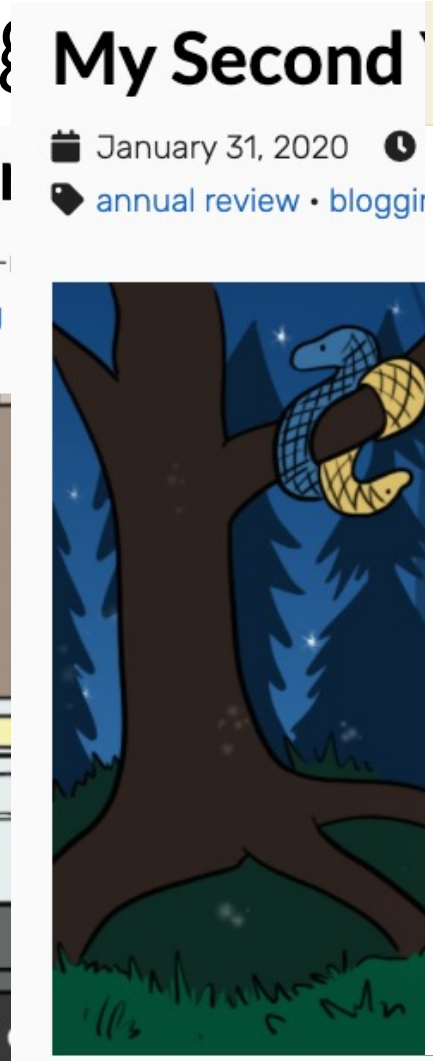
Pivotal Follow
May 31, 2017 · 8 min read

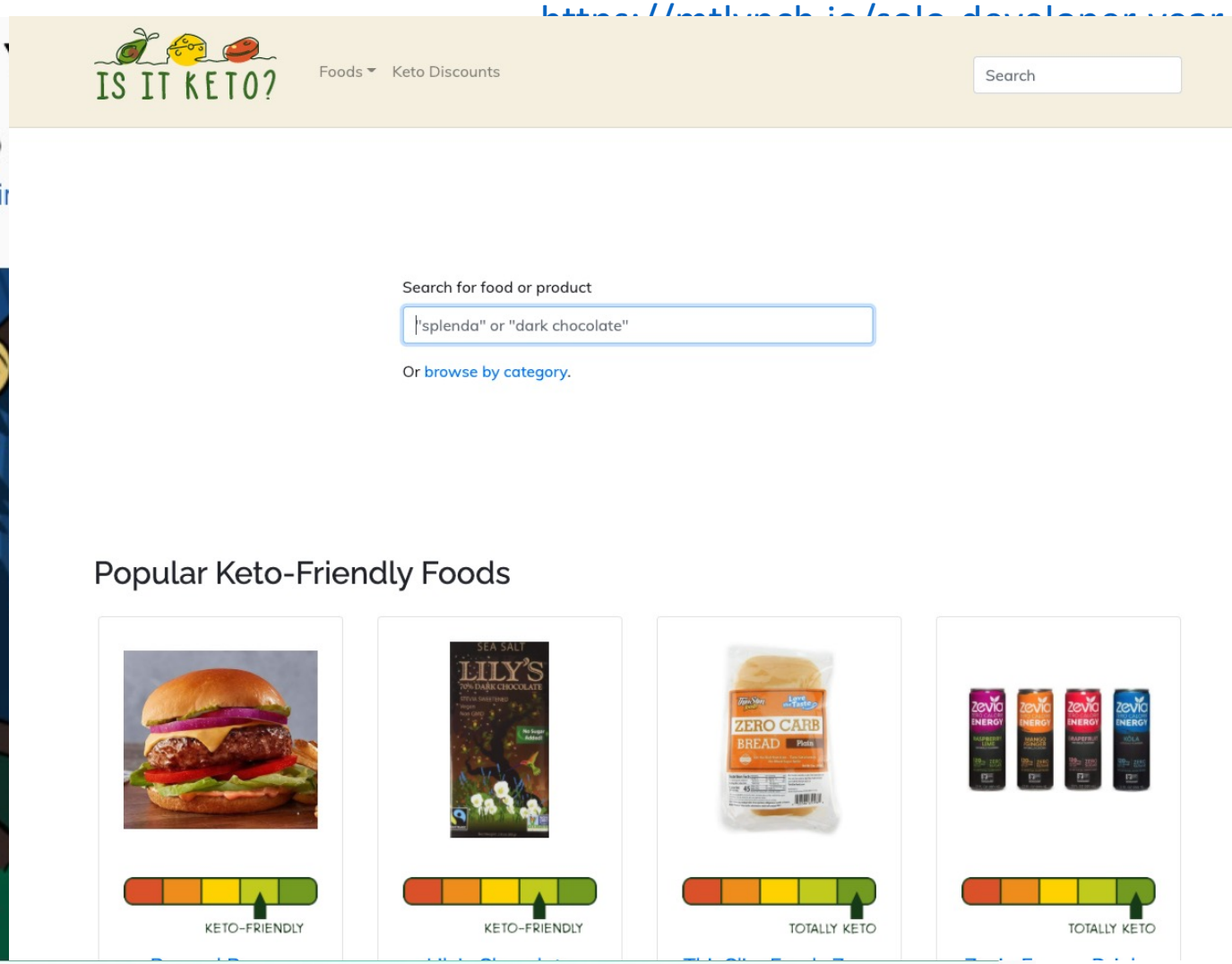


Solo prog

https://mtlynch.io/solo-developer-year-1/

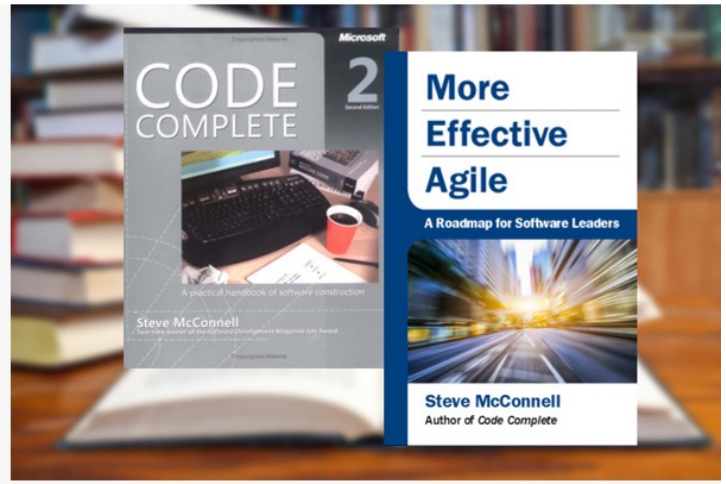




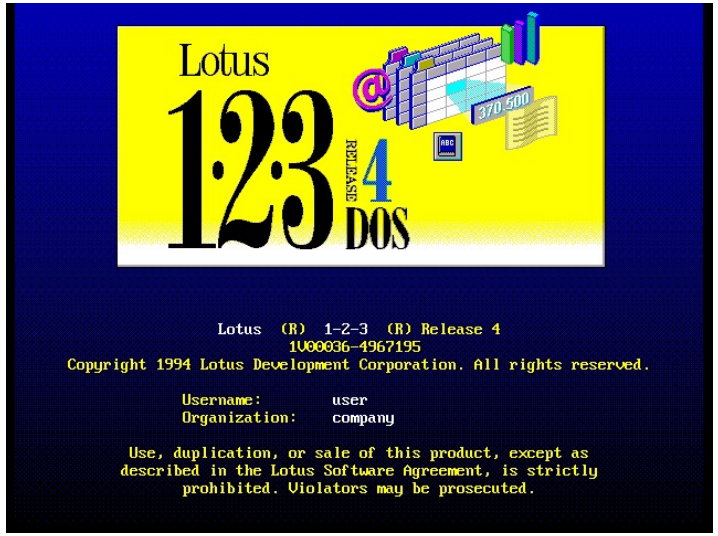


"During the time I was at Boeing in the mid 1980s, there was a project that had about 80 programmers working on it that was at risk of missing a critical deadline. The project was critical to Boeing, and so they moved most of the 80 people off that project and brought in one guy who finished all the coding and delivered the software on time."

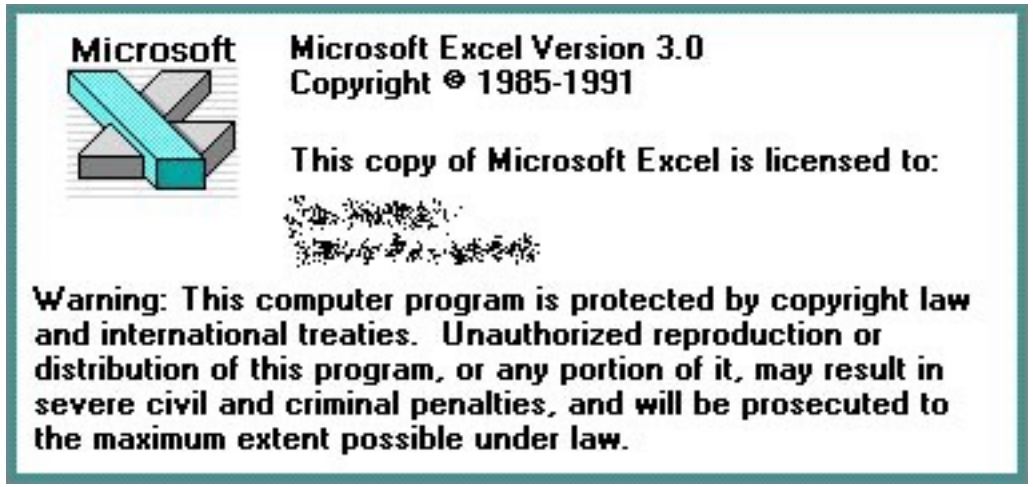
– Steve McConnell



10x of Teams

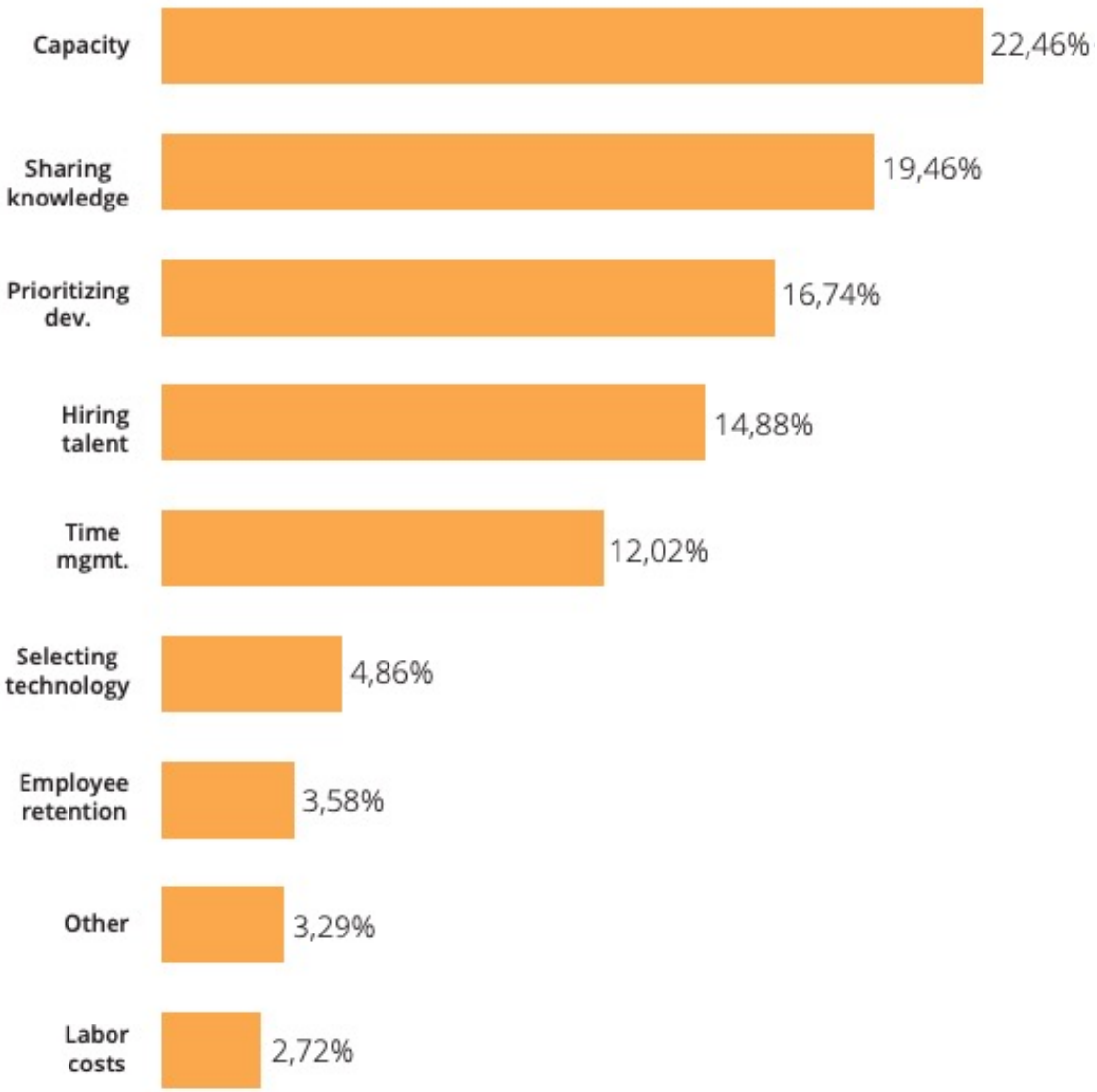


- Lotus 123 version 3
- 260 staff years
- 400,000 lines of code.



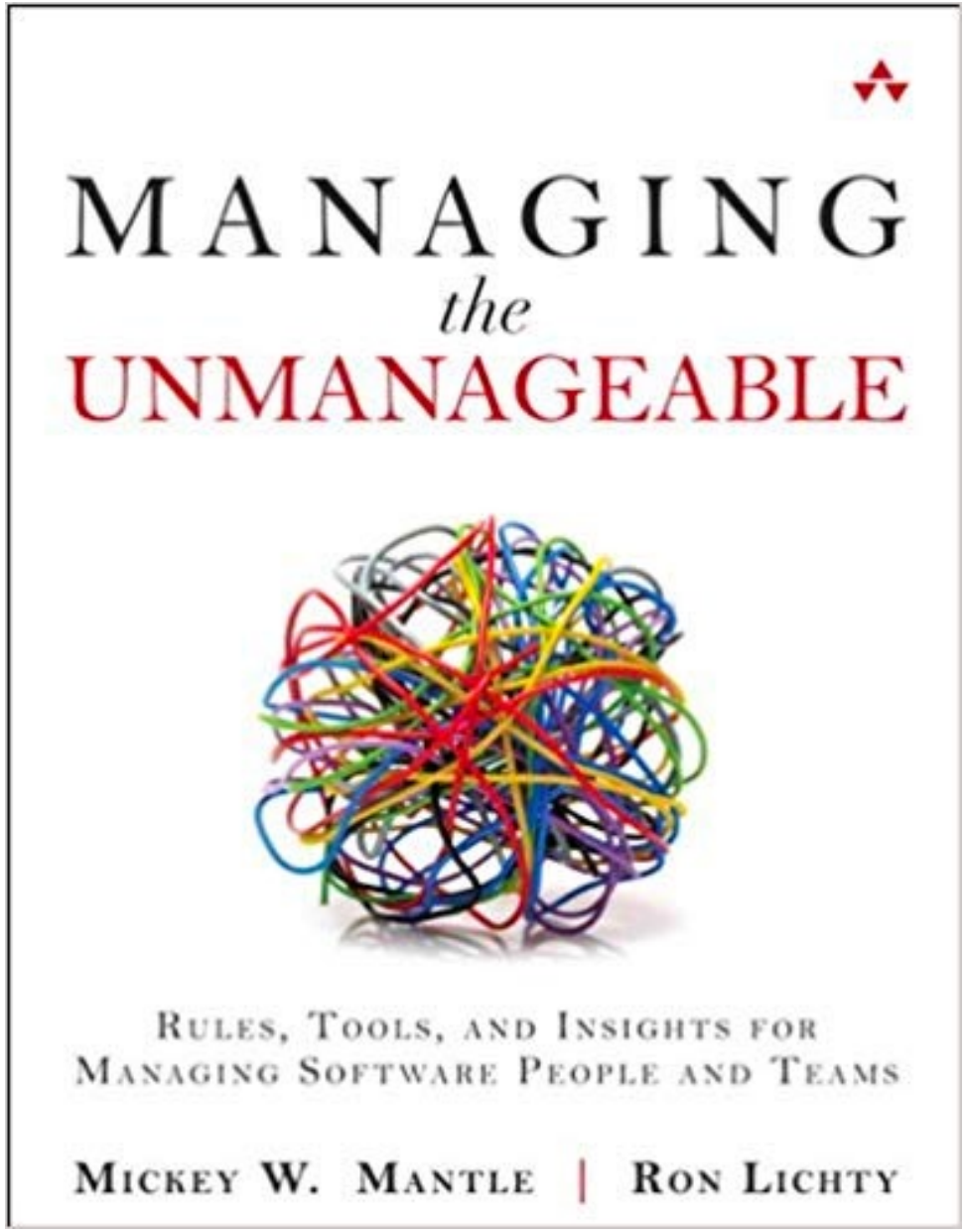
- Microsoft Excel 3.0
- 50 staff years
- 649,000 lines of code

What is your biggest challenge in software development?



https://codingsans.com/blog/recruiting-engineers

RECRUITING ENGINEERS: ASANA'S SECRETS TO HIRING TALENT (INTERVIEW WITH GREG SABO, ENGINEERING MANAGER AT ASANA)



"I just wish that I had this book when I started as a first-time manager five years ago!"

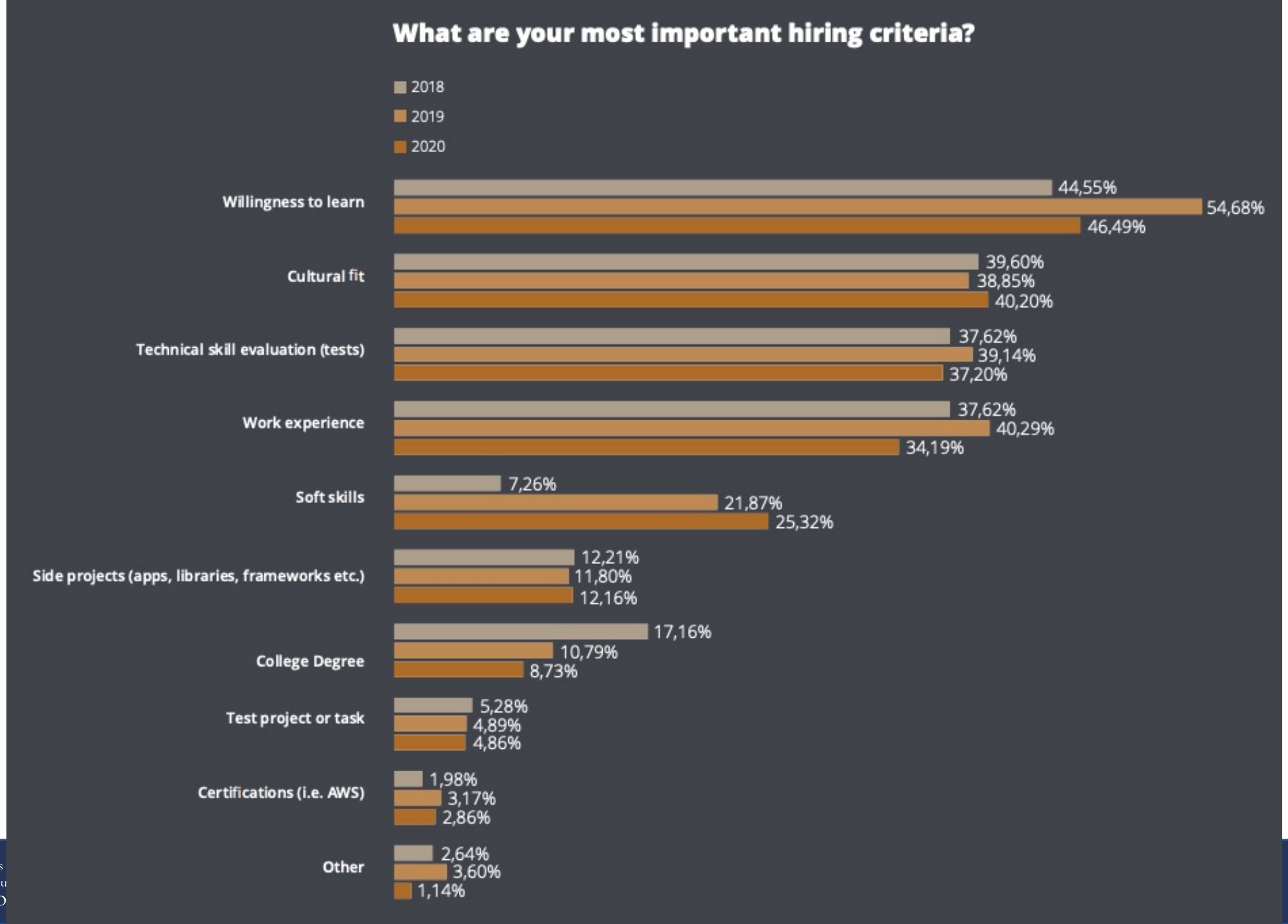
"Becoming a great engineering leader requires more than technical know-how; Ron and Mickey's book provides a practical cookbook for the important softer side of engineering leadership, which can be applied to any software development organization."

Why Programmers Seem Unmanageable?

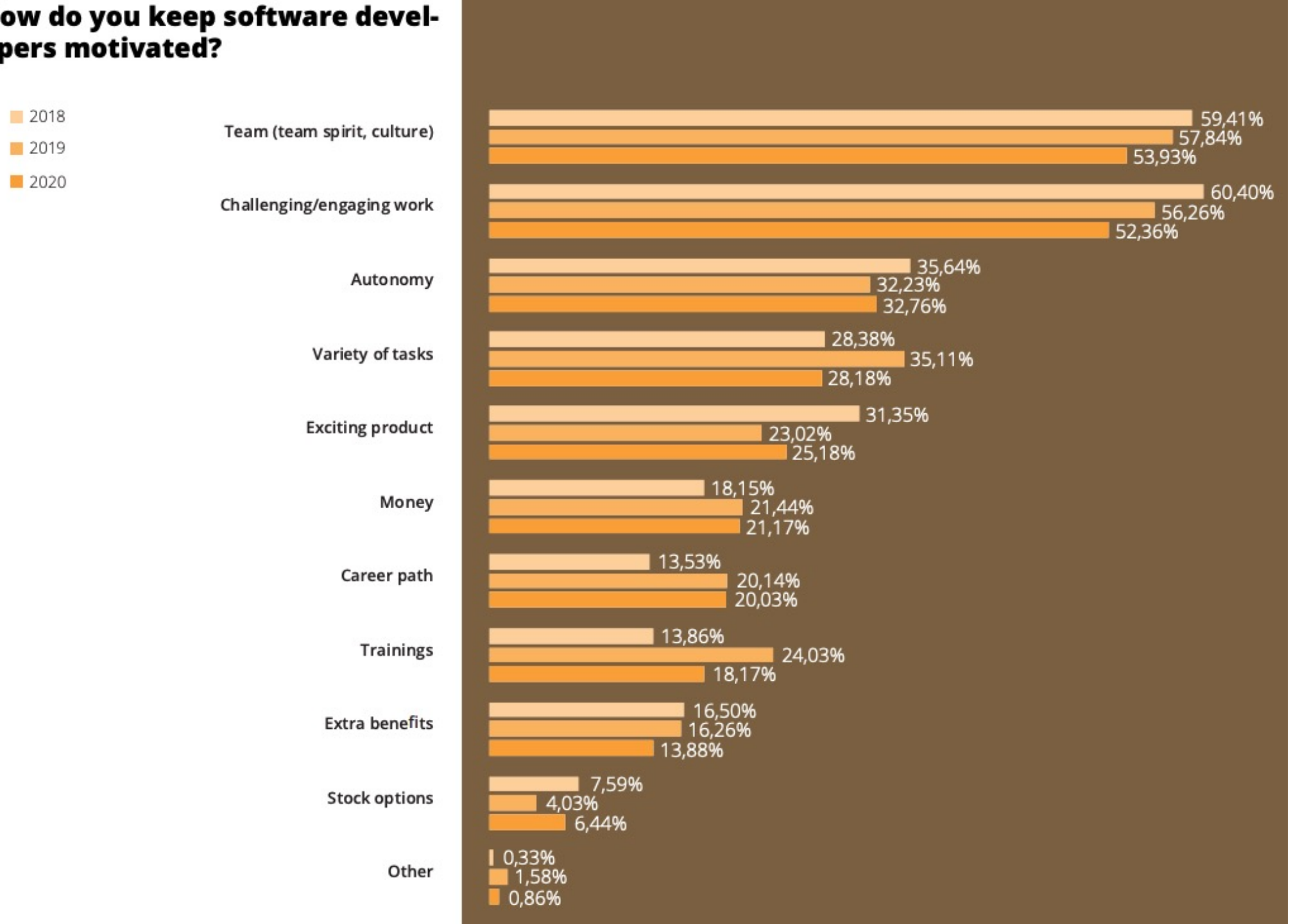
- *Writing a new program from scratch is akin to writing a novel.*
- Anyone can be a programmer
- The practices of SE have had minimal impact

"If having fun is what most programmers do, you may begin to understand why managing programmers is so challenging. If you are being paid to have fun, why would you want to be managed? Being managed takes part of the fun out of the work!"

"Managing programmers is a lot like herding cats"


What are your most important hiring criteria?
2018
2019
2020
Willingness to learn
44,55%
54,68%
46,49%
Cultural fit
39,60%
38,85%
40,20%
Technical skill evaluation (tests)
37,62%
39,14%
37,20%
Work experience
37,62%
40,29%
34,19%
Soft skills
7,26%
21,87%
25,32%
Side projects (apps, libraries, frameworks etc.)
12,21%
11,80%
12,16%
College Degree
17,16%
10,79%
8,73%
Test project or task
5,28%
4,89%
4,86%
Certifications (i.e. AWS)
1,98%
3,17%
2,86%
Other
2,64%
3,60%
1,14%

How do you keep software developers motivated?



Why do engineers choose TO JOIN particular teams?

Reasons grouped by clustering analysis	Percent
Liked new team and/or technology (exciting, manager)	**85.8%**
Coworker asked me to join (new team, old team)	**37.8%**
Joined for better opportunities (location, domain, lack of other options)	**24.5%**
Followed my manager (former or current)	**14.6%**

Why do engineers want to leave their teams?

Reasons grouped by clustering analysis	Percent
Change is coming (technology, charter, re-org, turnover)	**52.6%**
Seeking new challenges or location (role, location, challenges)	**39.0%**
Dissatisfaction with manager (priorities, goals, person, actions)	**31.6%**
The grass is always greener on the other side (novelty, escape)	**12.3%**
Not a good fit (bored, no need for my skills)	**5.3%**
Poor team dynamics (dysfunctional, no career growth)	**4.4%**

JULIA ROZOVSKY, ANALYST, GOOGLE PEOPLE OPERATIONS | NOVEMBER 17, 2015

The five keys to a successful Google team

Pod. Work group. Committee. Autonomous collective. Whatever you call it, you're part of one at Google and probably wherever you work: a team. So if we know what makes managers great, why don't we know what makes a team great?

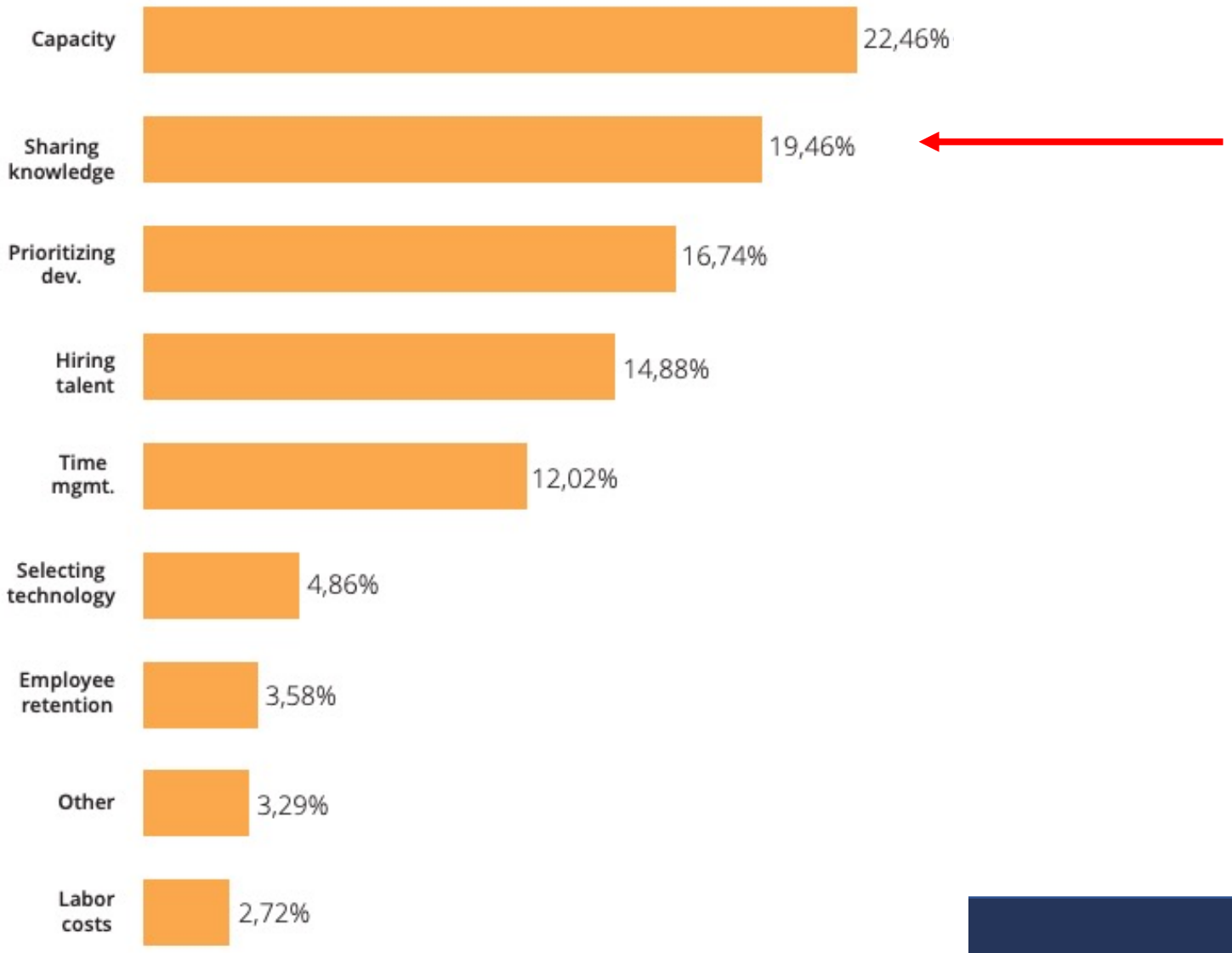
1. **Psychological safety:** Can we take risks on this team without feeling insecure or embarrassed?
2. **Dependability:** Can we count on each other to do high quality work on time?
3. **Structure & clarity:** Are goals, roles, and execution plans on our team clear?
4. **Meaning of work:** Are we working on something that is personally important for each of us?
5. **Impact of work:** Do we fundamentally believe that the work we're doing matters?

https://rework.withgoogle.com/blog/five-keys-to-a-successful-google-team/

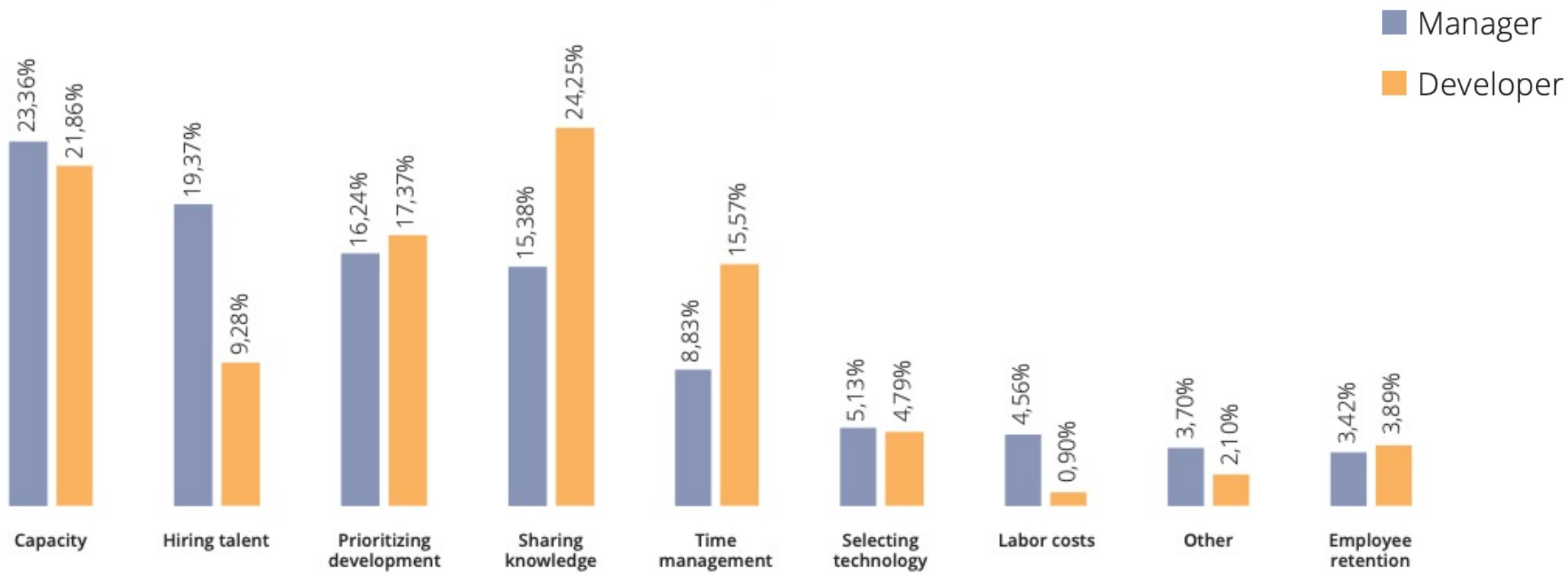
State of Software Development

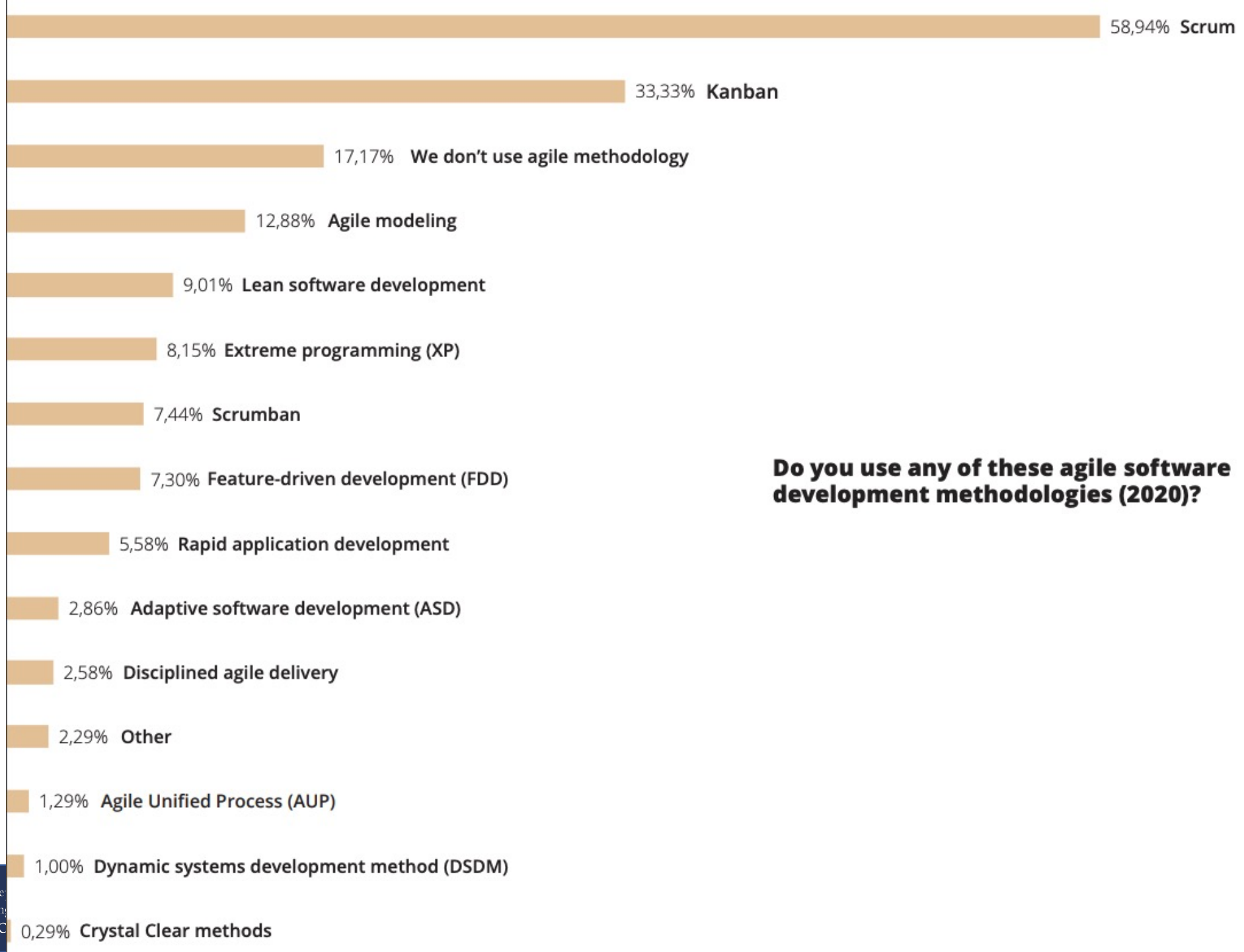
https://codingsans.com/uploads/landing/State-of-Software-Development-2020.pdf

What is your biggest challenge in software development?

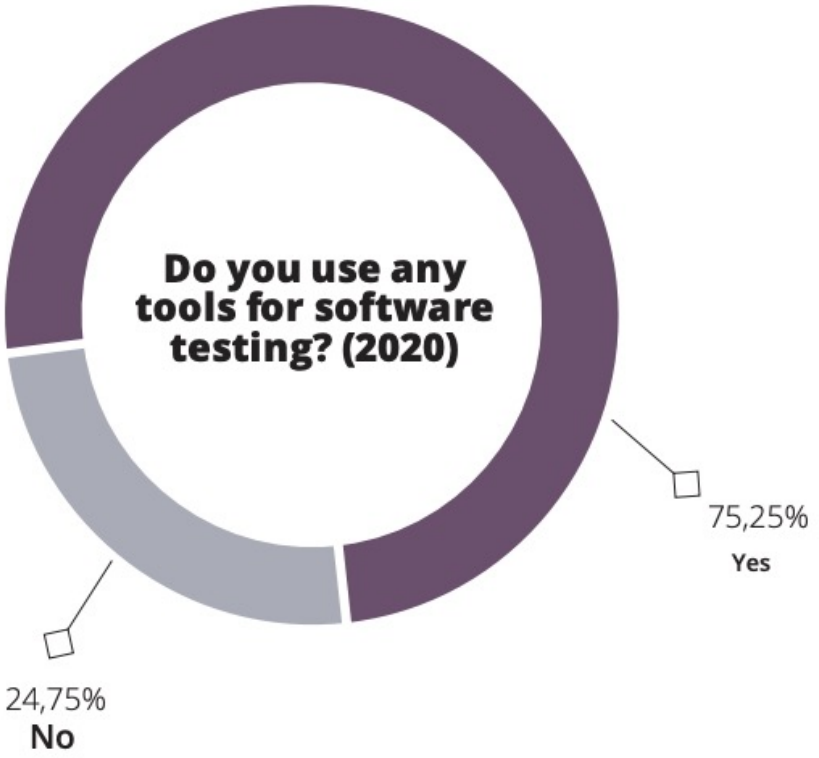


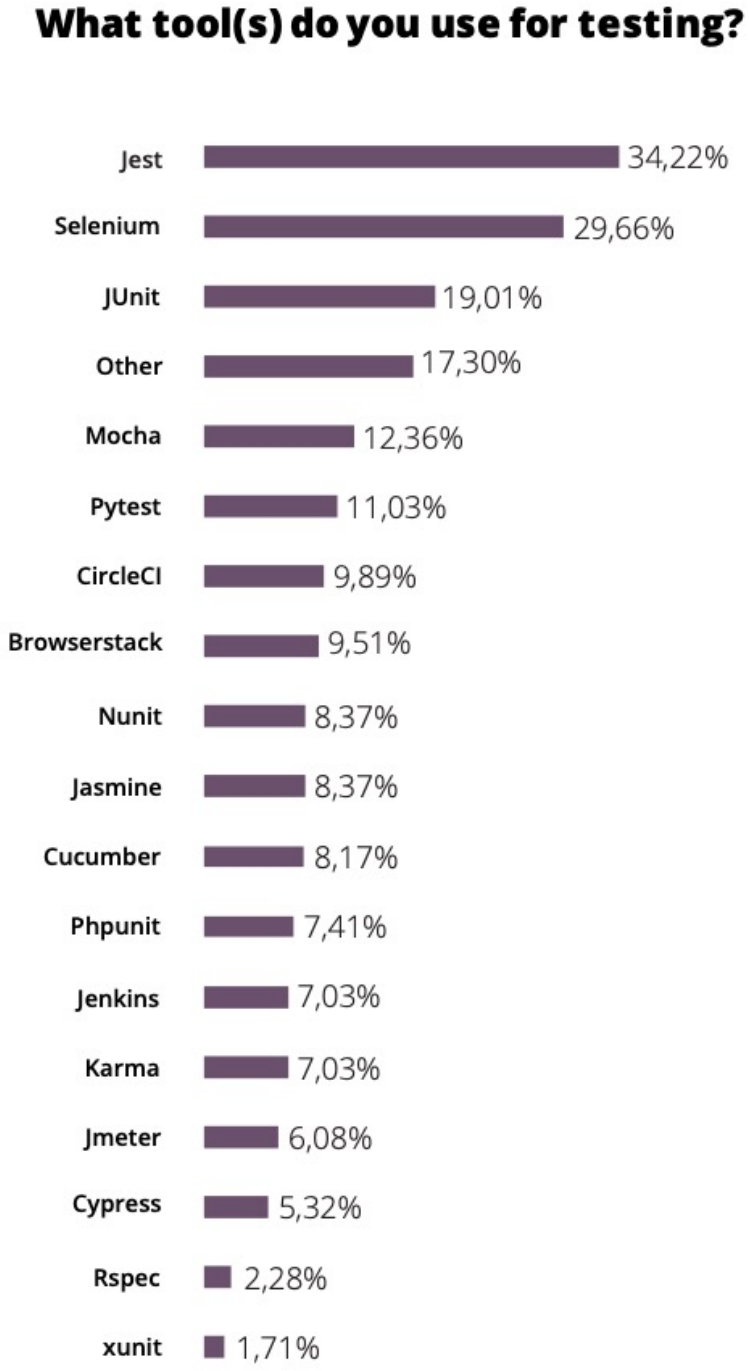
What is your biggest challenge in software development?

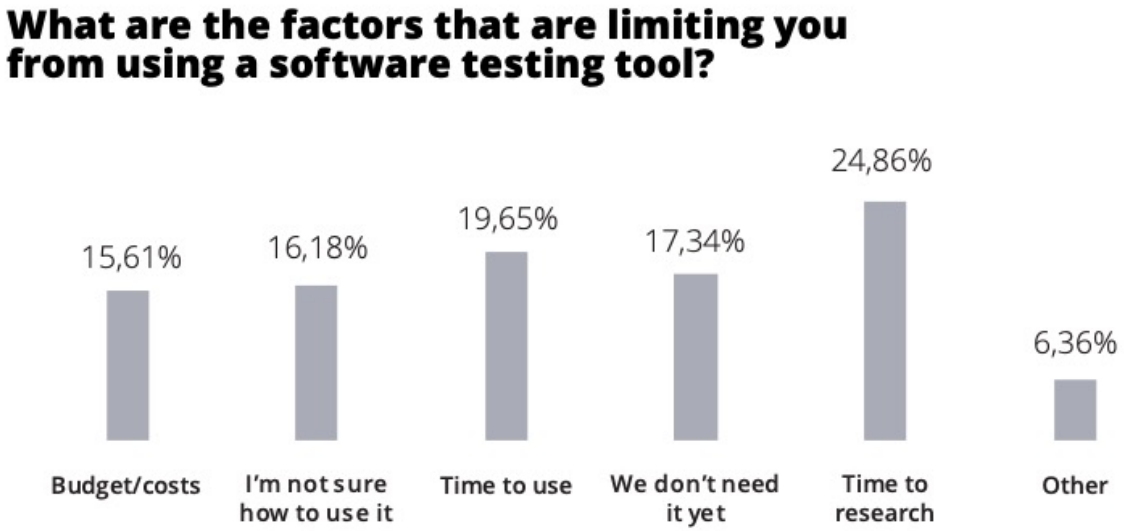




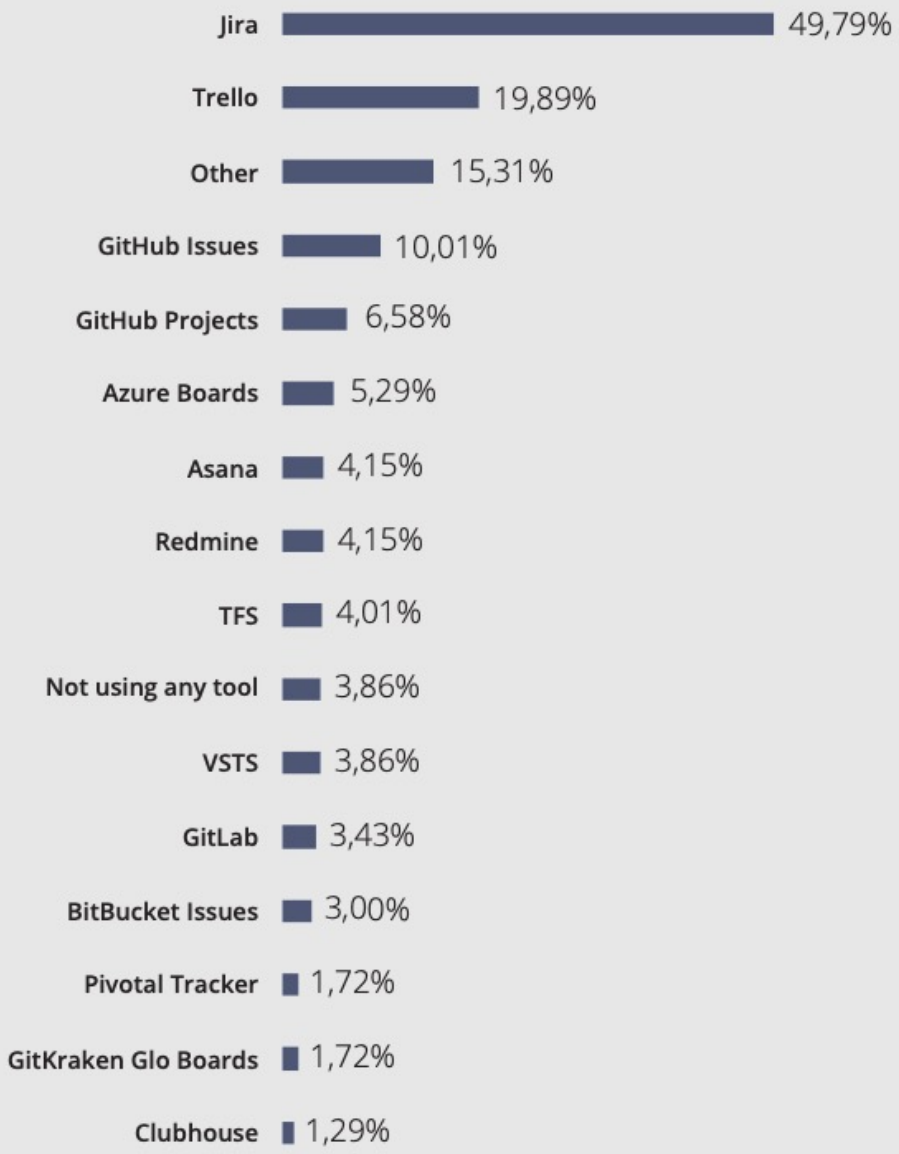
Do you use any of these agile software development methodologies (2020)?


Do you use any tools for software testing? (2020)
75,25%
Yes
24,75%
No


What tool(s) do you use for testing?
Jest 34,22%
Selenium 29,66%
JUnit 19,01%
Other 17,30%
Mocha 12,36%
Pytest 11,03%
CircleCI 9,89%
Browserstack 9,51%
Nunit 8,37%
Jasmine 8,37%
Cucumber 8,17%
Phpunit 7,41%
Jenkins 7,03%
Karma 7,03%
Jmeter 6,08%
Cypress 5,32%
Rspec 2,28%
xunit 1,71%

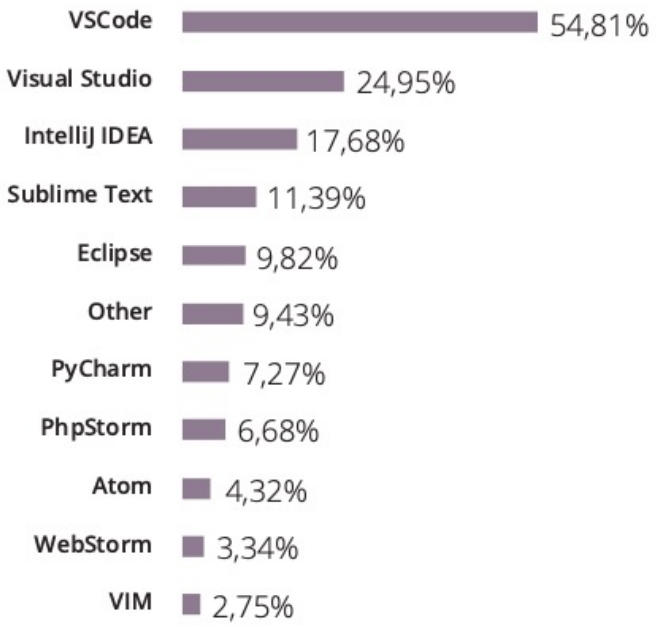

What are the factors that are limiting you from using a software testing tool?
15,61%
Budget/costs
16,18%
I'm not sure how to use it
19,65%
Time to use
17,34%
We don't need it yet
24,86%
Time to research
6,36%
Other

What tool do you use for project management?

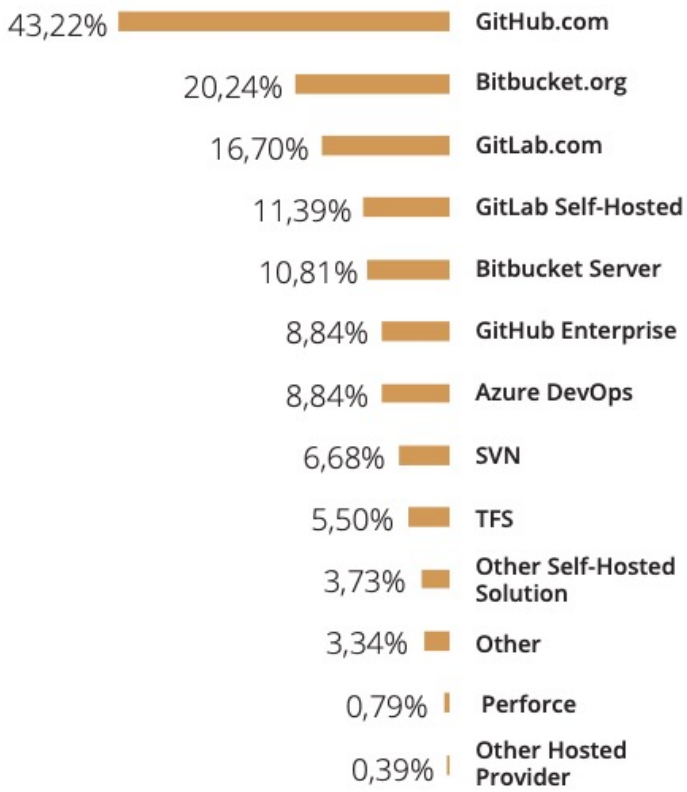


What tools do you use to communicate during a project?

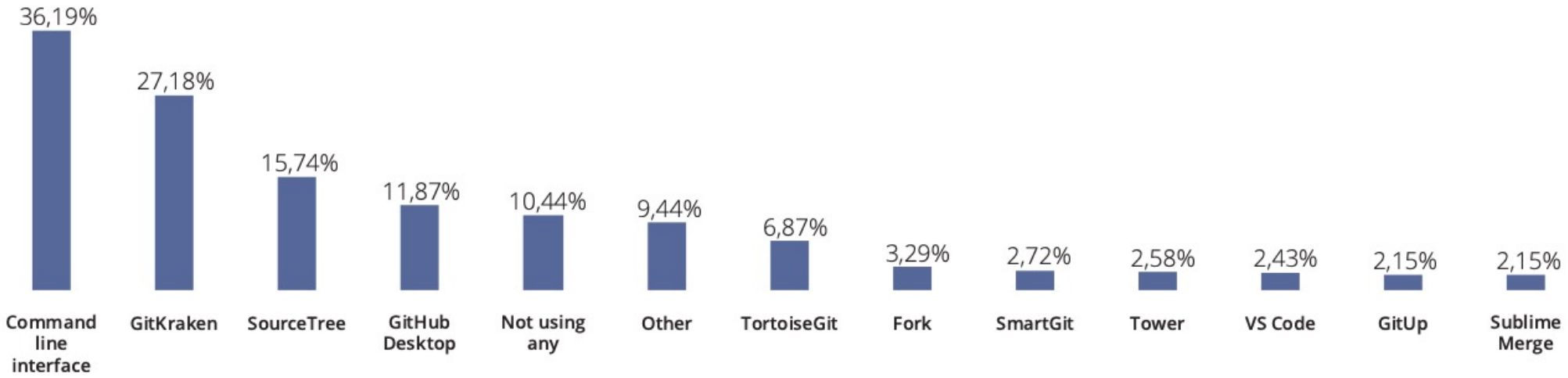
What IDE(s) do you use?



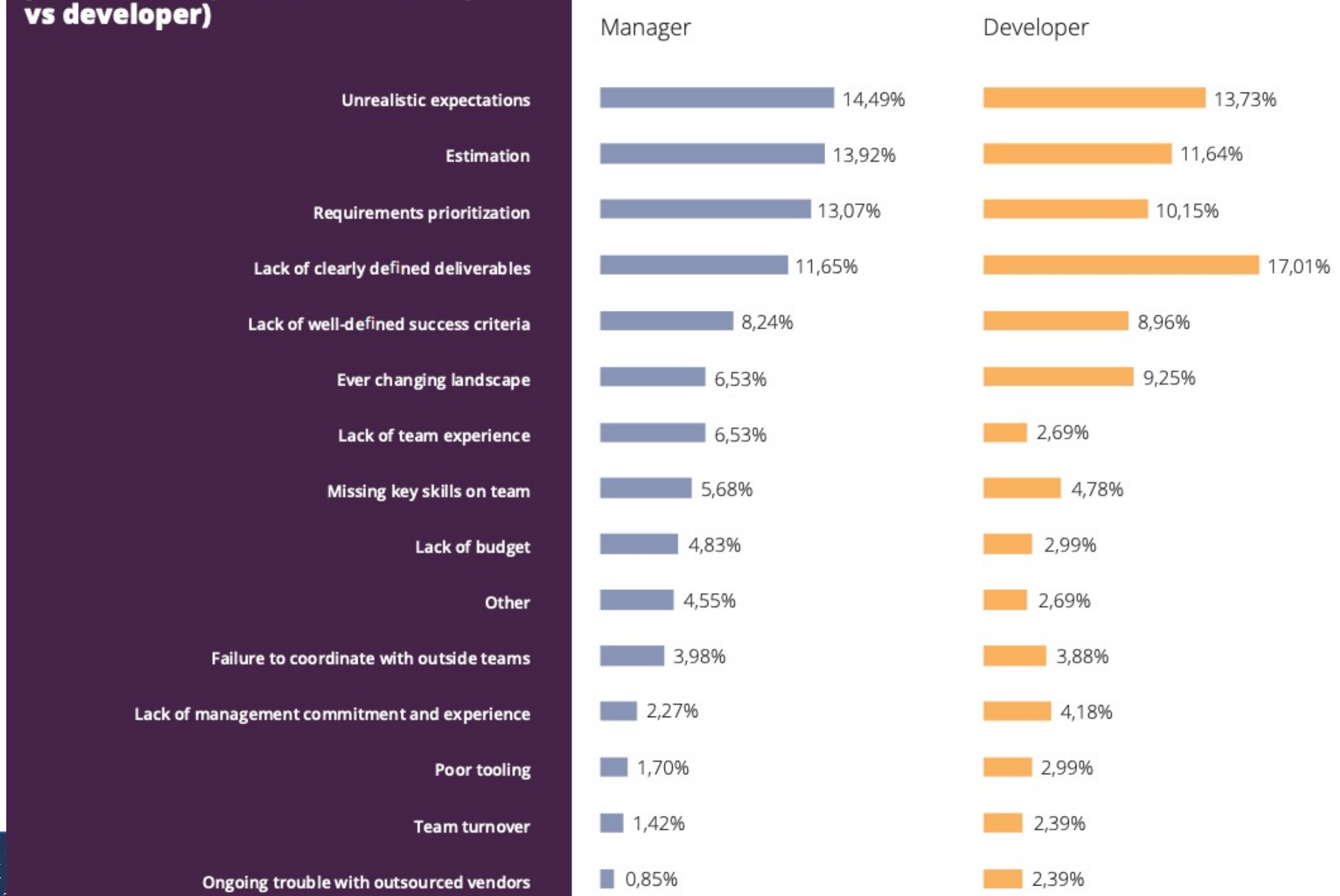
What version control system do you use?



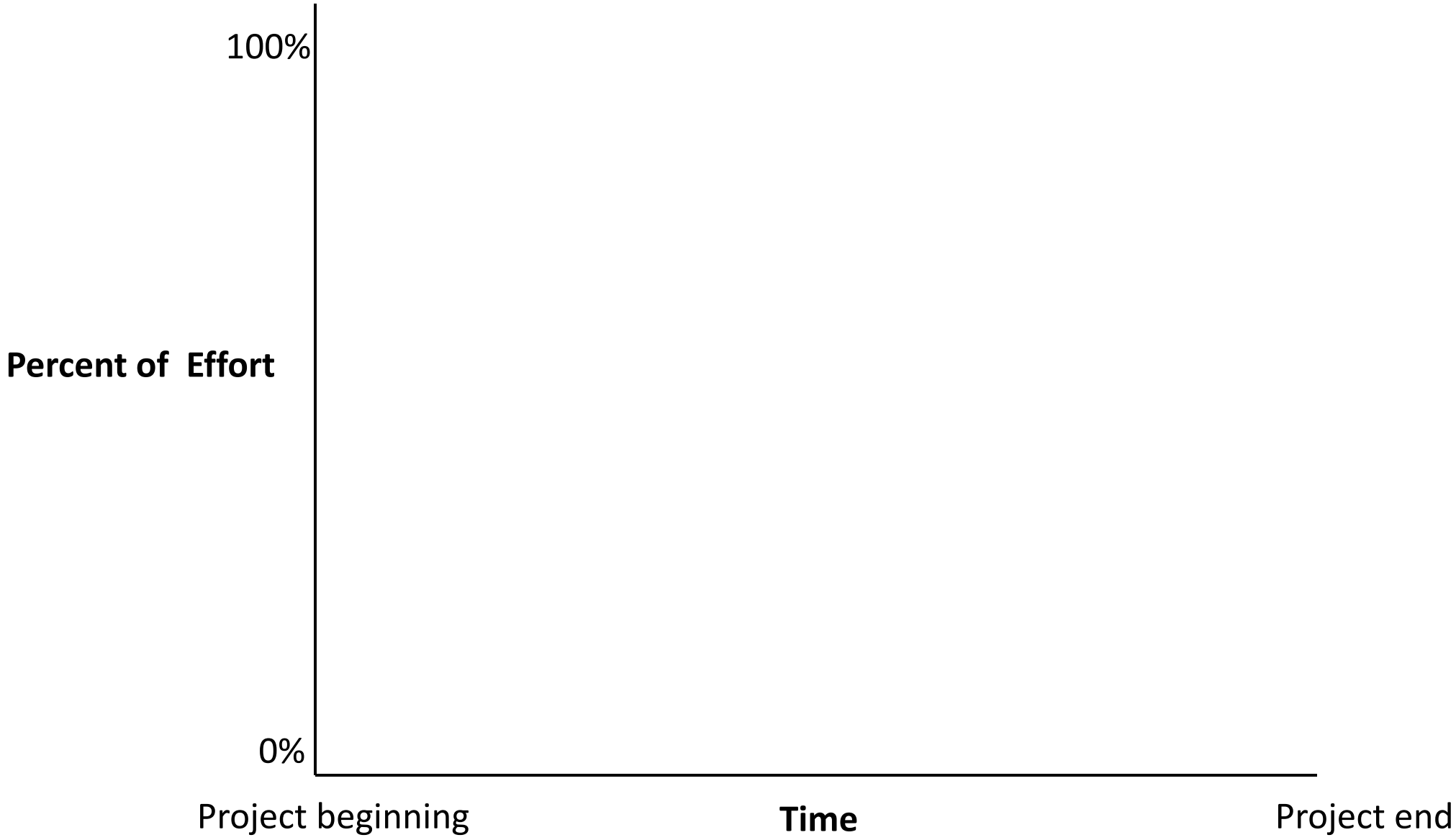
What source control client(s) do you use?

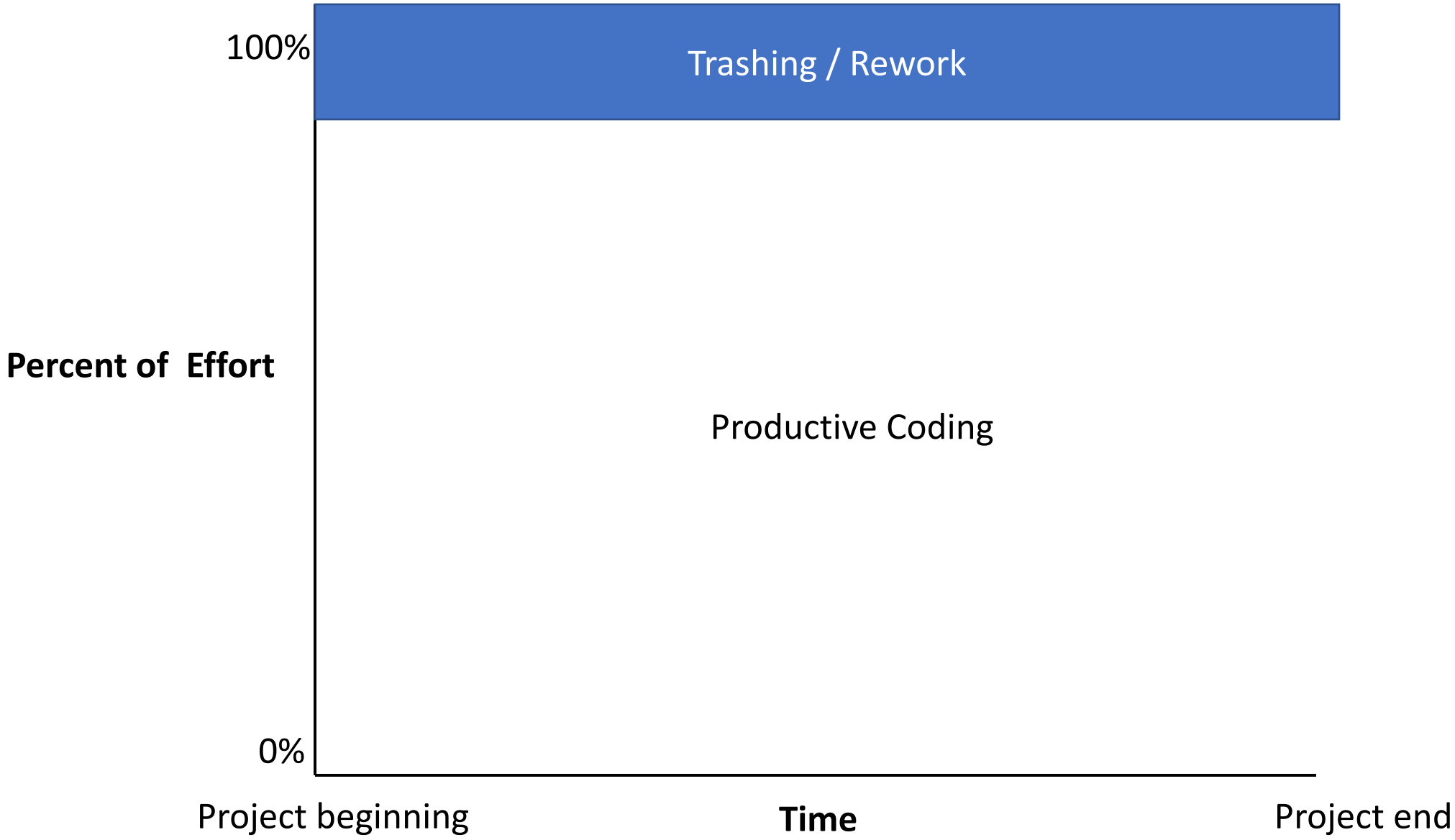


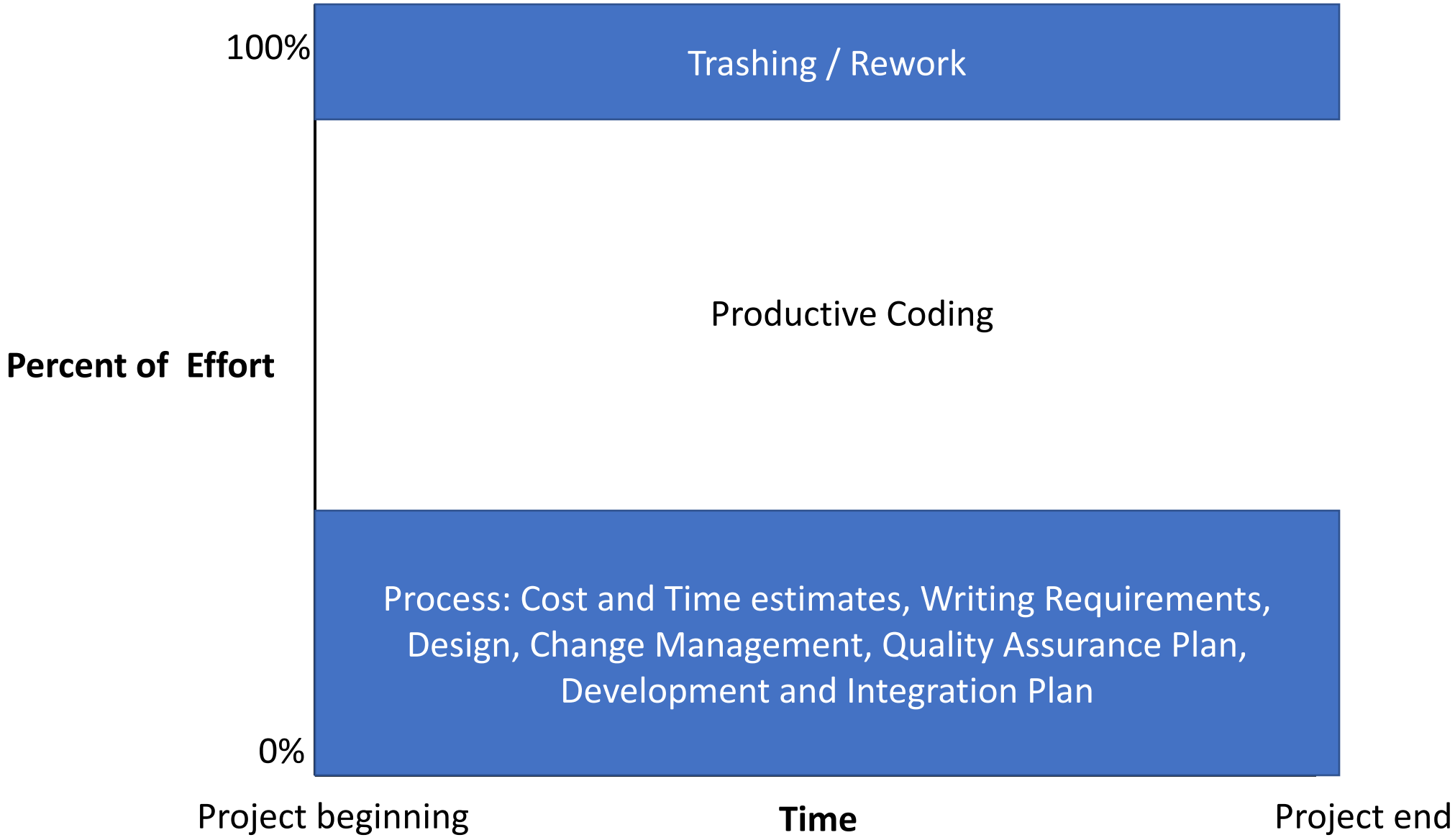
What is the #1 cause of delivery problems for your team? (manager vs developer)

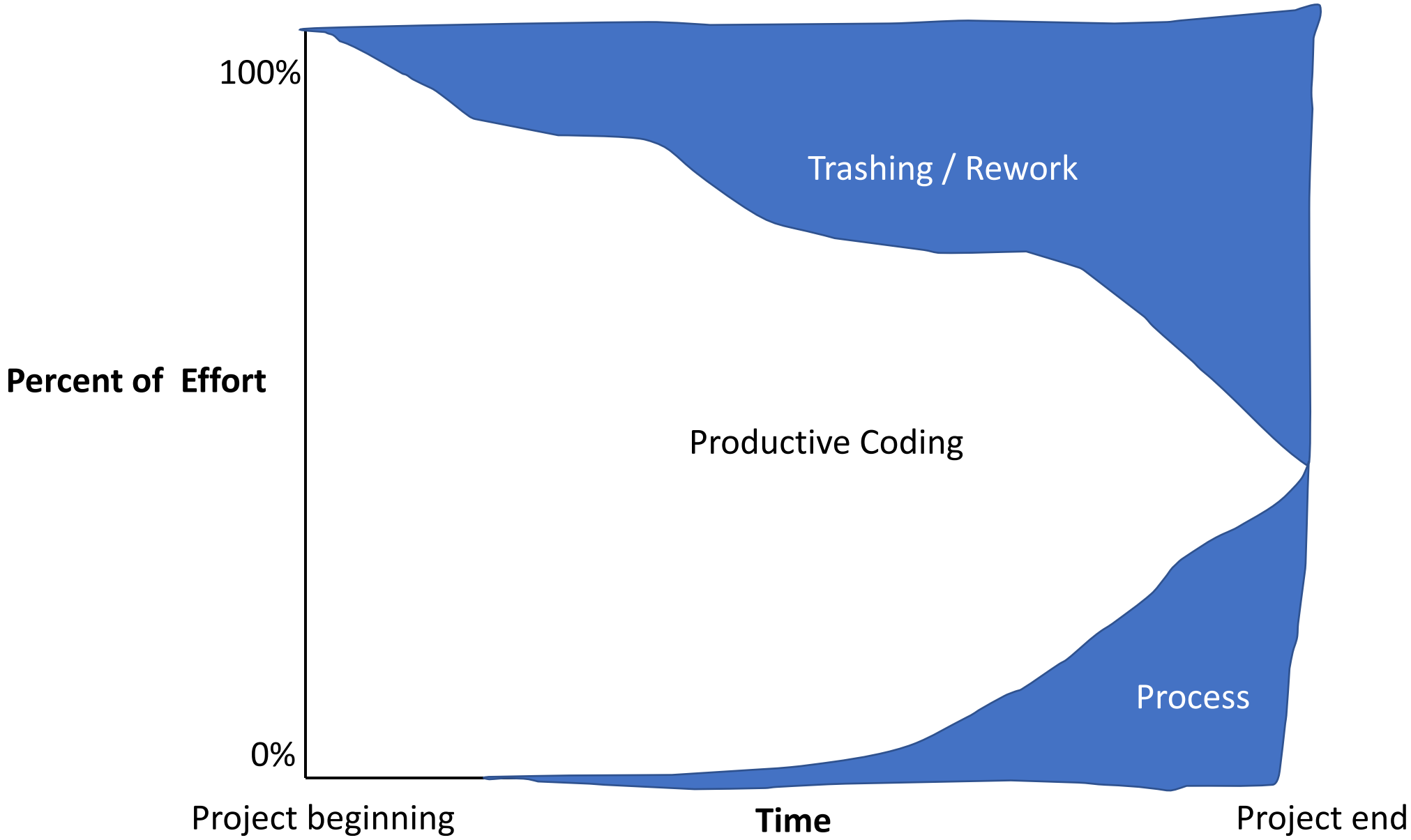


PROCESS IS IMPORTANT


100%
Percent of Effort
0%
Project beginning
Time
Project end

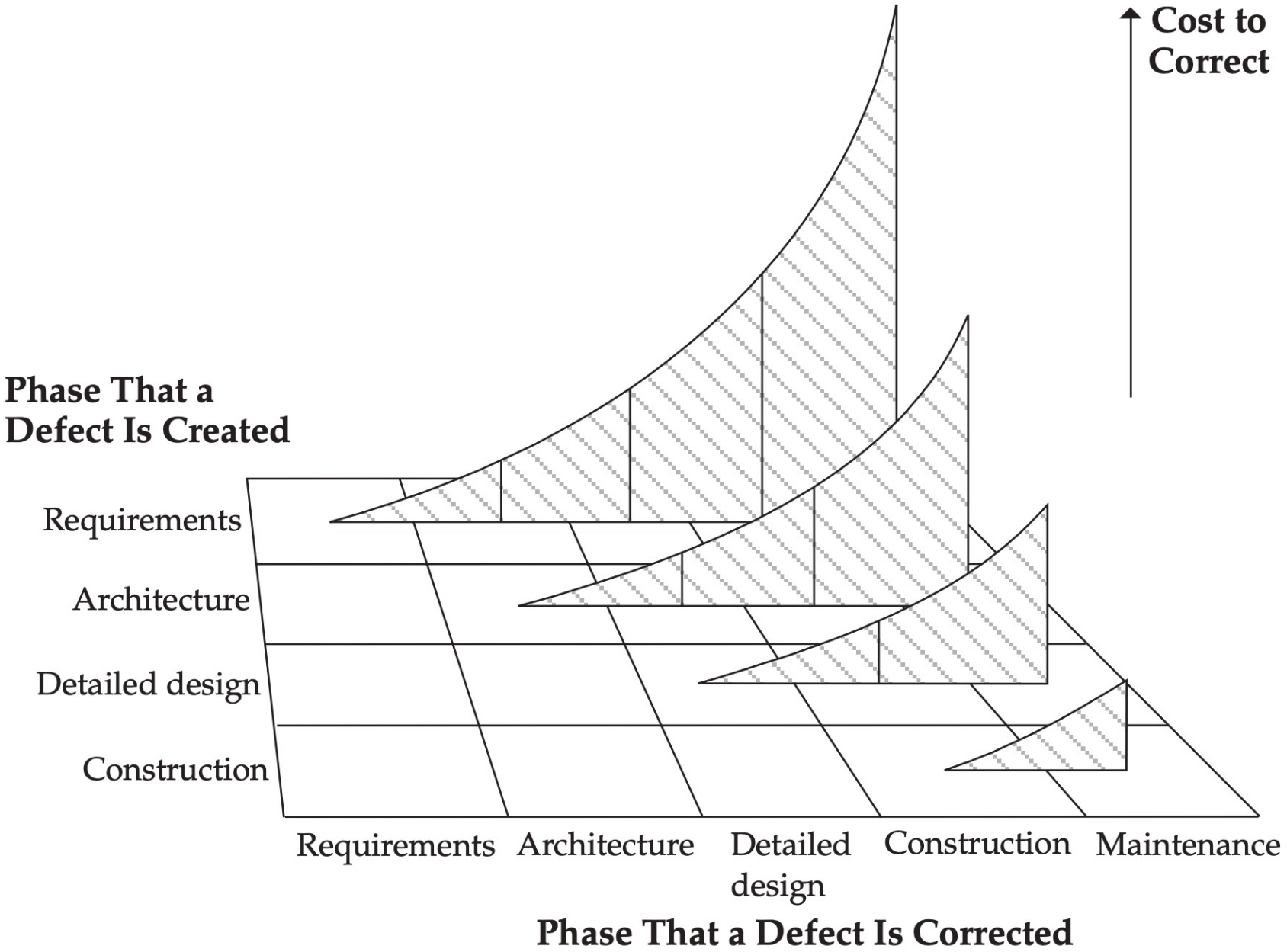

100%
Trashing / Rework
Percent of Effort
Productive Coding
0%
Project beginning
Time
Project end


100%
Trashing / Rework
Productive Coding
Percent of Effort
Process: Cost and Time estimates, Writing Requirements, Design, Change Management, Quality Assurance Plan, Development and Integration Plan
0%
Project beginning
Time
Project end


100%
Trashing / Rework
Percent of Effort
Productive Coding
Process
0%
Project beginning
Time
Project end

Survival Mode

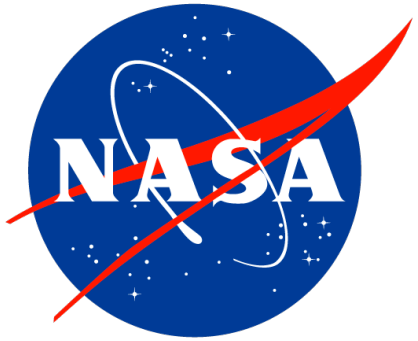
- Missed deadlines -> "solo development mode" to meet own deadlines
- Ignore integration work
- Stop interacting with testers, technical writers, managers, …


Cost to Correct
Phase That a Defect Is Created
Requirements
Architecture
Detailed design
Construction
Requirements
Architecture
Detailed design
Construction
Maintenance
Phase That a Defect Is Corrected

Real world cases

Organizations that have explicitly focused on improving their development processes have, over several years, cut their time-to-market by about one-half and reduced their costs and defects by factors of 3 to 10.

5 yr, cost -75%, time - 40%, defects - 90%



8 yr, cost -50%, defects - 75%

Planning

Task: Estimate Time

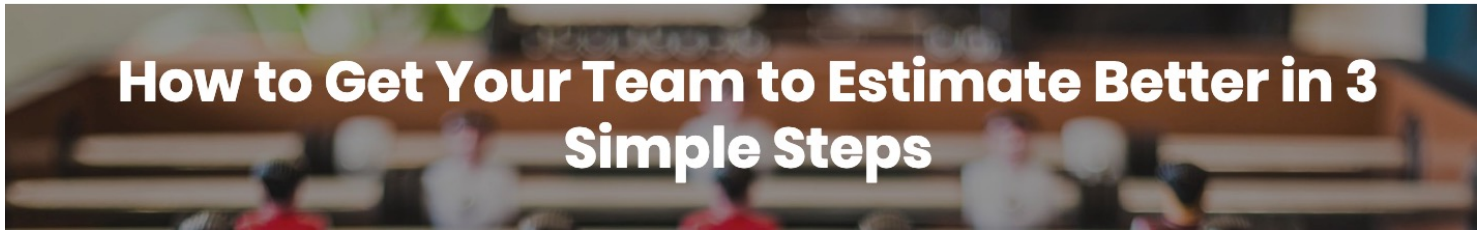
- a web application of Trip guide (booking, scheduling, route planning...)

Estimate in 8h days (20 work days in a month, 220 per year)

Revise Time Estimate

- Remember the GIS system experience?
- Is GIS similar/different/easier/more challenging/reusable?
- How much design did you do?
- Break down the task into ~5 smaller tasks and estimate them.
- Revise your overall estimate if necessary

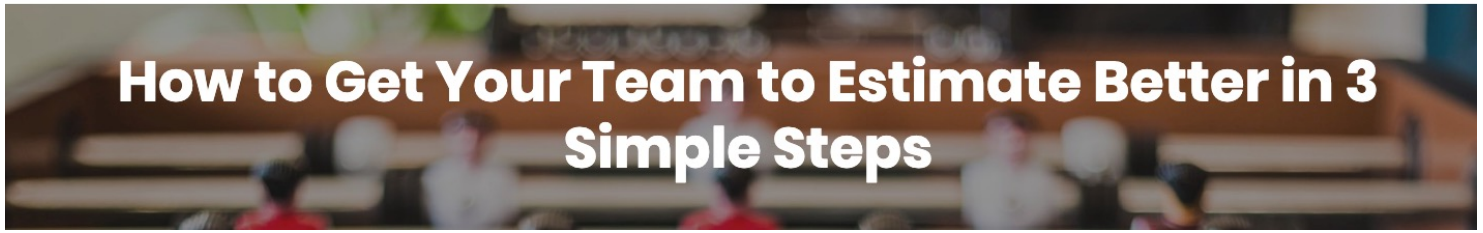
:codica



- 2 Types of Projects
  - Projects having an accurate target, technical inquiry and deadlines.
  - Projects having a general idea and no accurate visualization of further development, like products for startups or Time & Material projects.

https://www.codica.com/blog/how-to-get-better-estimates/

:codica



"It is important to concentrate on the scale of complexity, not the amount of further work."

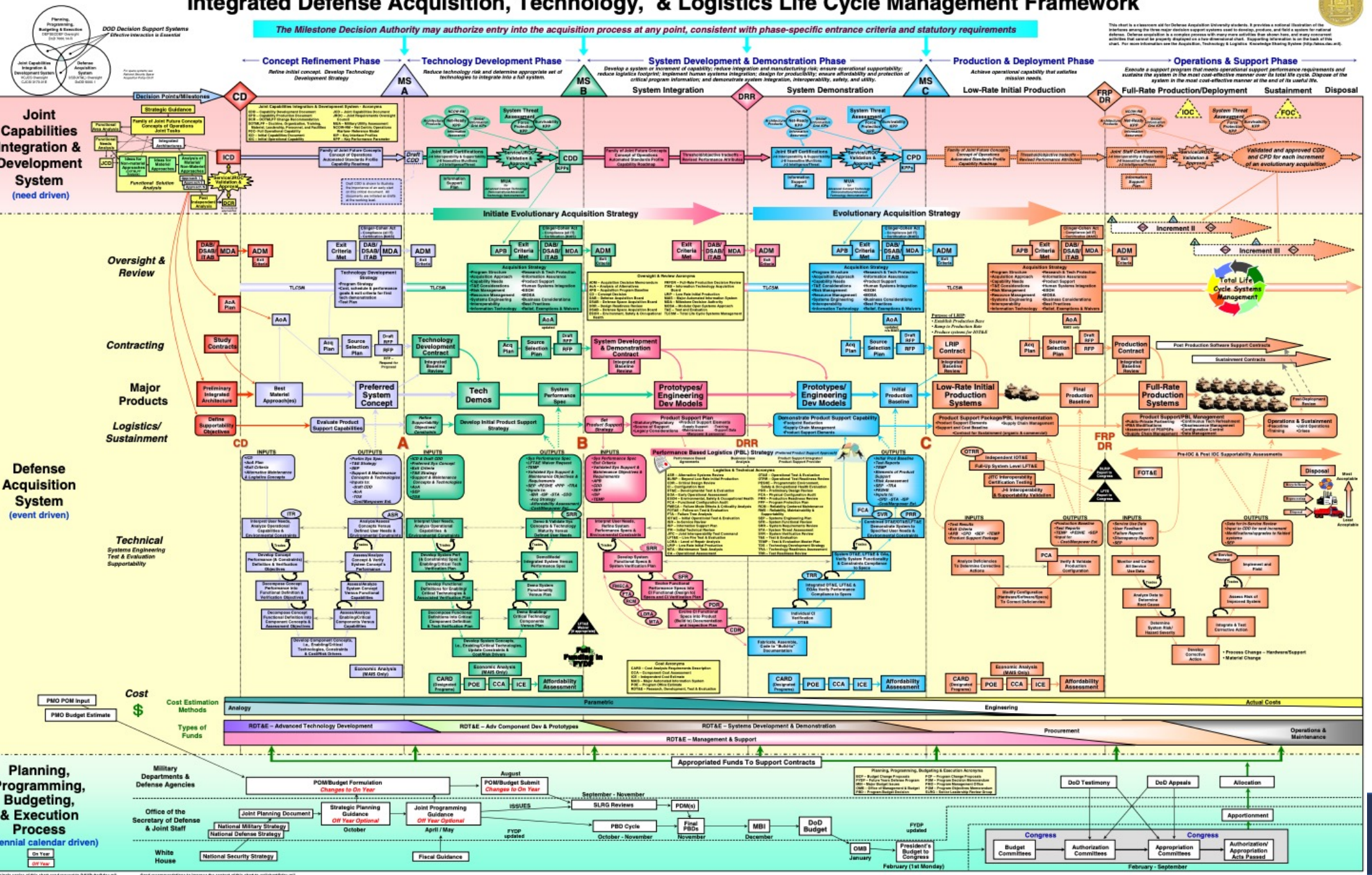
https://www.codica.com/blog/how-to-get-better-estimates/

Milestones and Deliverables

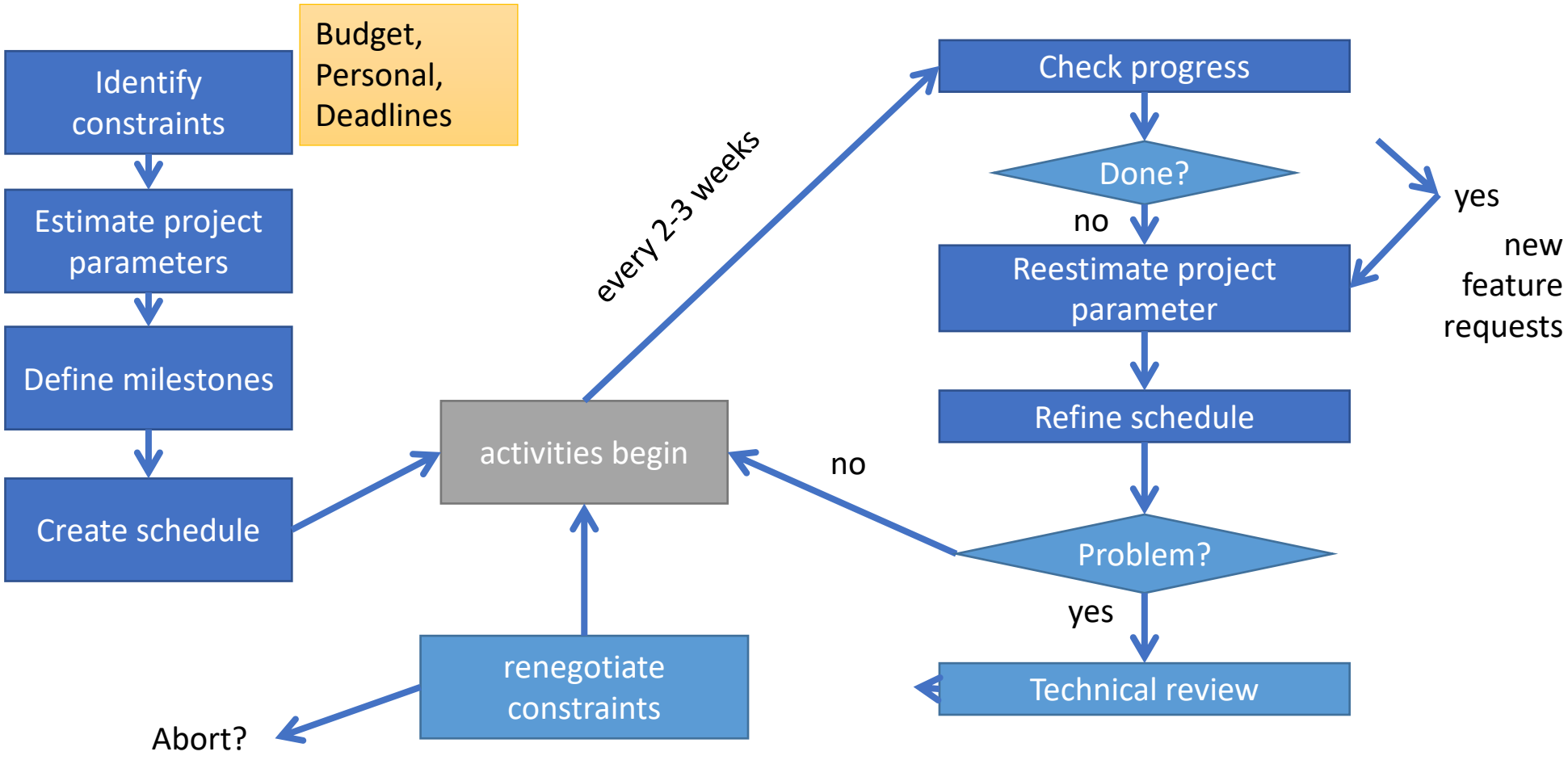
- Making progress observable, especially for software
- Milestone: clear end point of a (sub)tasks
  - For project manager
  - Reports, prototypes, completed subprojects
  - "80% done" not a suitable milestone
- Deliverable: Result for customer
  - Similar to milestone, but for customers
  - Reports, prototypes, completed subsystems

ver. 5.2. August 2005

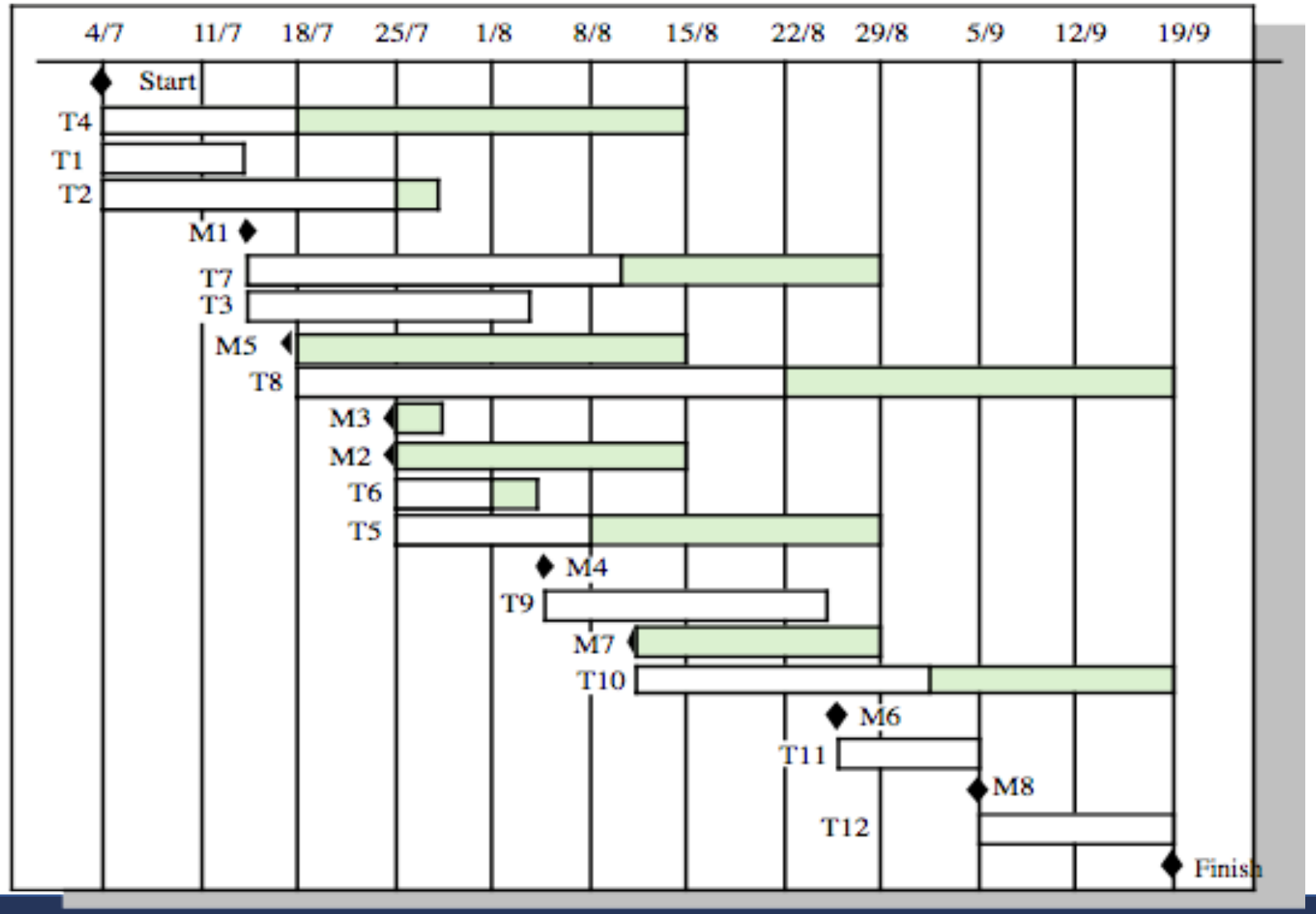
Integrated Defense Acquisition, Technology, & Logistics Life Cycle Management Framework



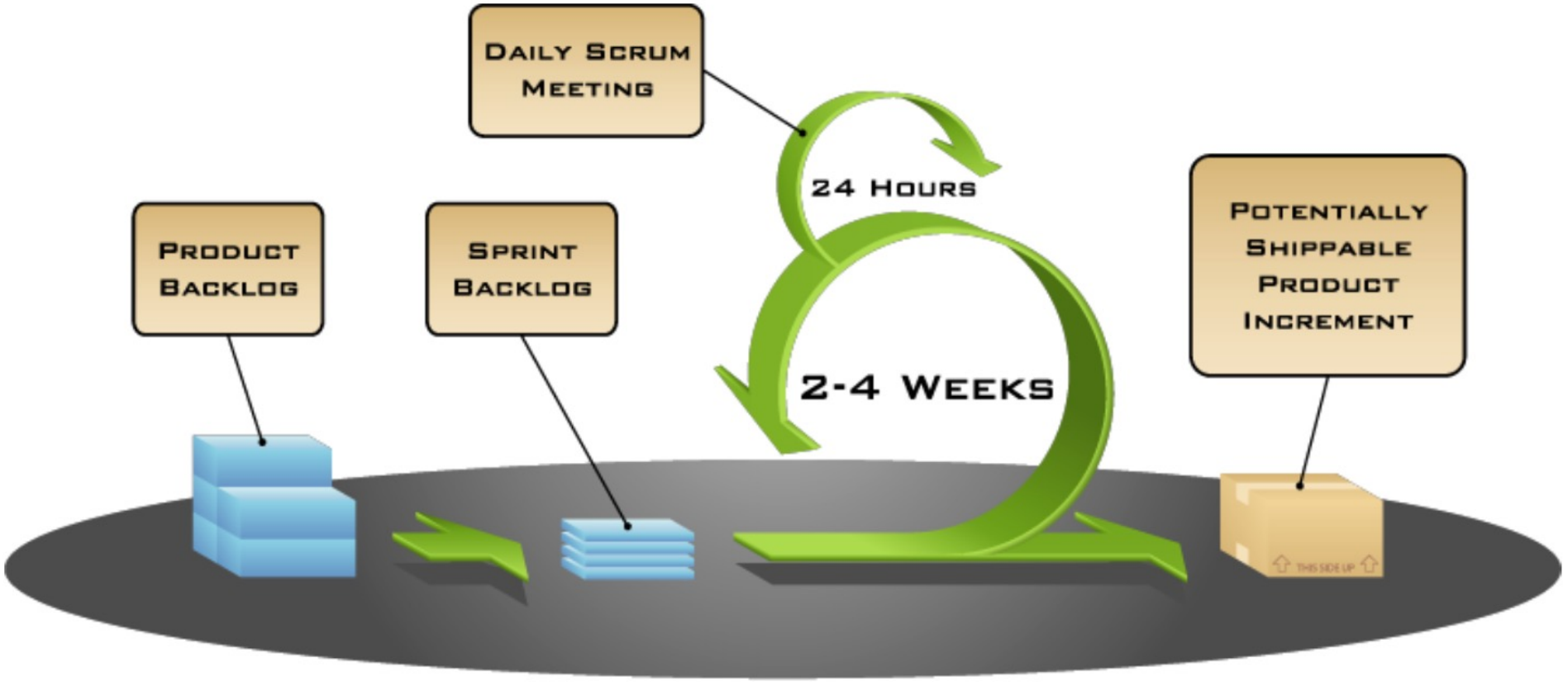
Project Planning



Gantt Diagrams



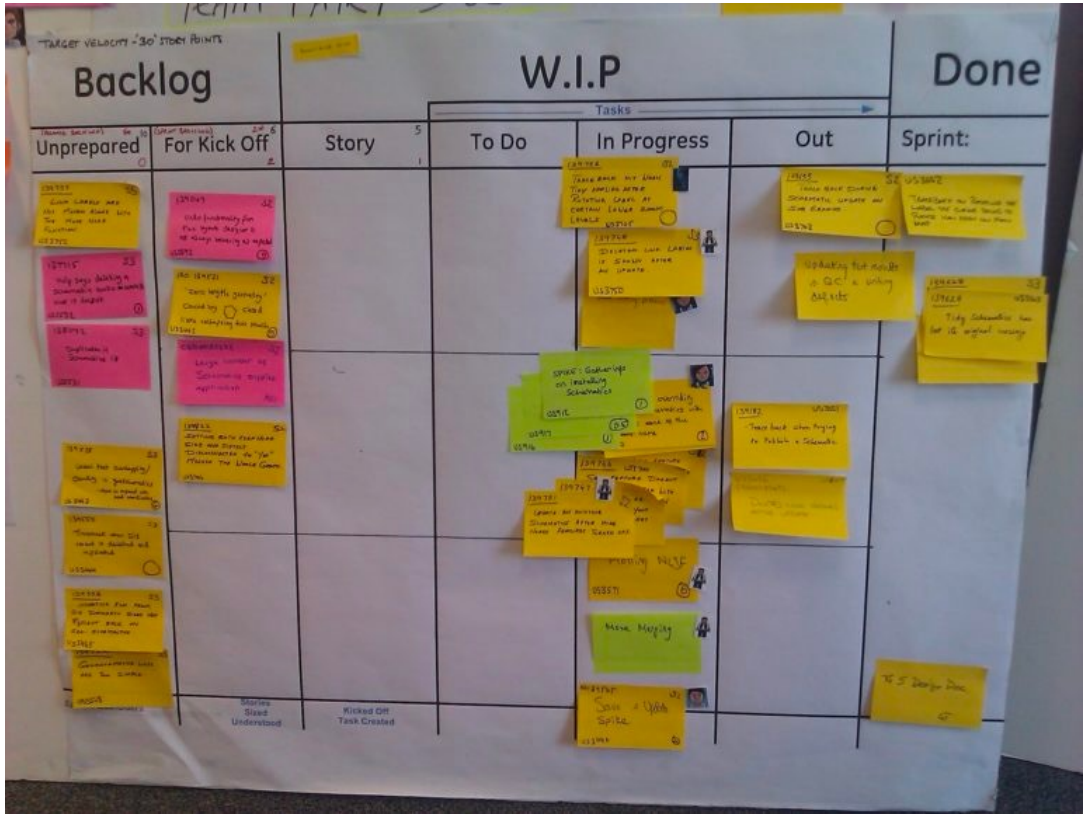
Brief intro to Scrum

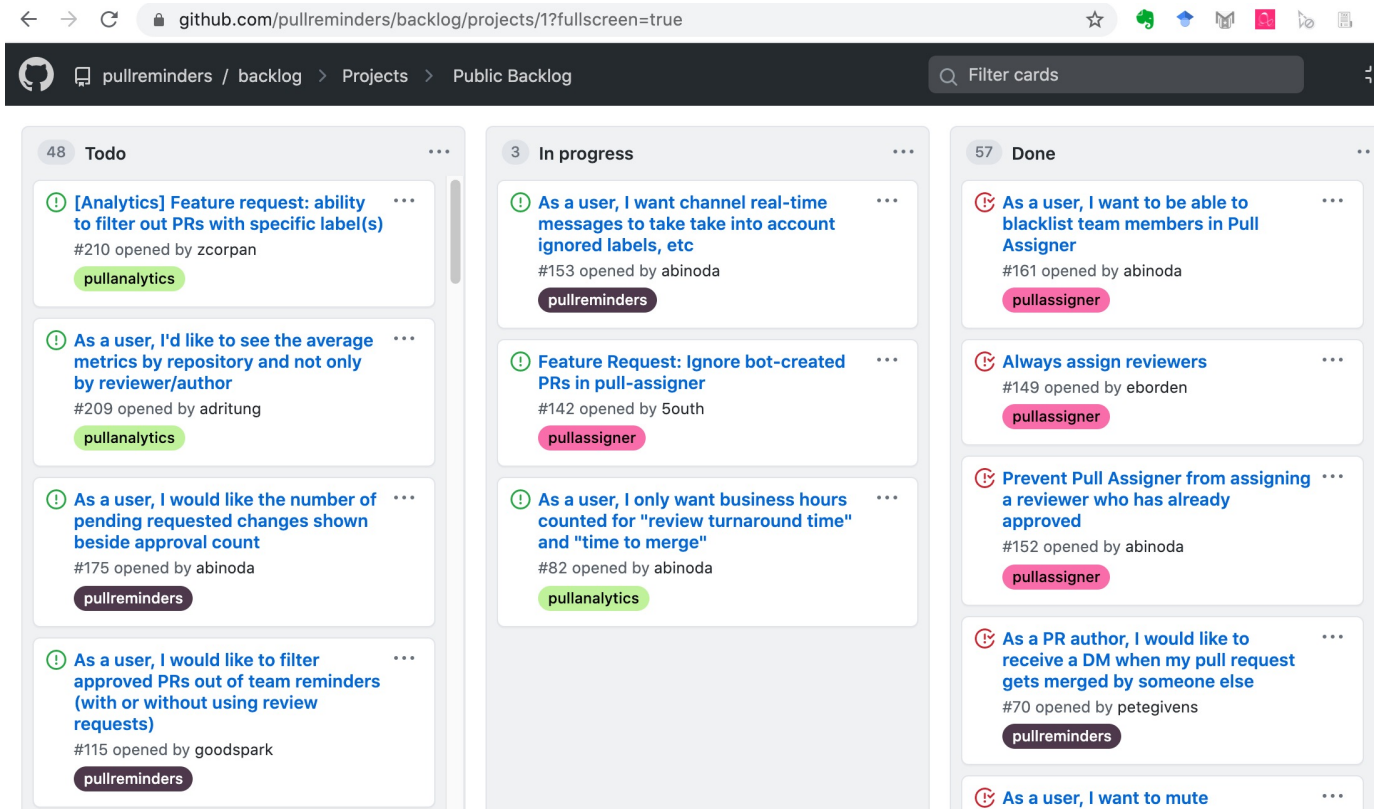


Product Backlog/Sprint Backlog

- The product backlog is all the features for the product
- The sprint backlog is all the features that will be worked on for that sprint. These should be broken down into discrete tasks:
  - Fine-grained
  - Estimated
  - Assigned to individual team members
  - Acceptance criteria should be defined
- User Stories are often used

Backlog – information radiators



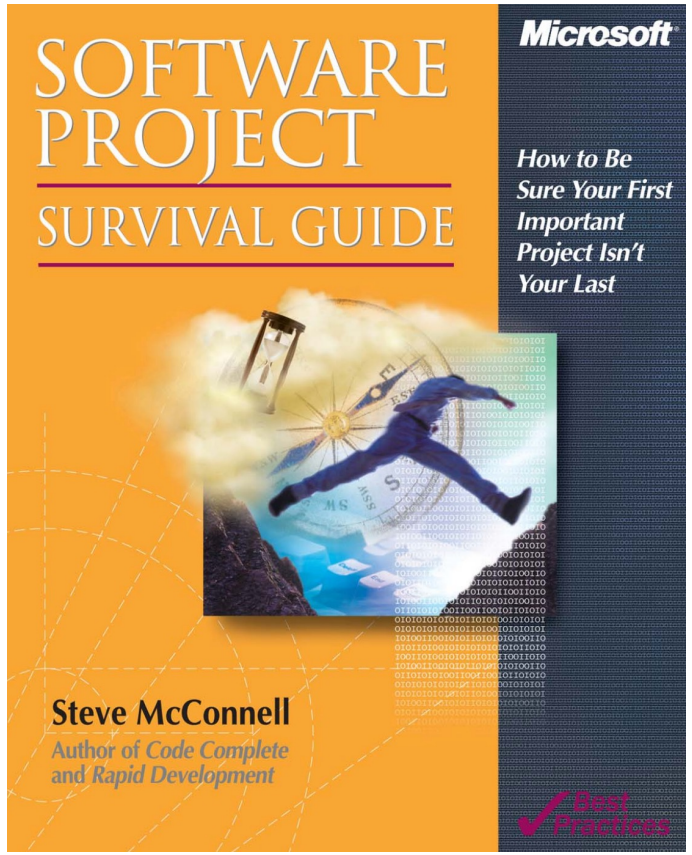


Scrum meetings

- Sprint Planning Meeting
  - Entire Team decides together what to tackle for that sprint
- Daily Scrum Meeting
  - Quick Meeting to touch base on :
    - What have I done? What am I doing next? What am I stuck on/need help?
- Sprint Retrospective
  - Review sprint process
- Sprint Review Meeting
  - Review Product

Further Reading

- McConnell. Software Project Survival Guide. Microsoft Press 1998, Chapter 3 (link)
- Sommerville. Software Engineering. 8th Edition. Addison-Wesley 2007. Chapters 5 "Project Planning" and 26 "Software Cost Estimation"



Teamwork (Student Teams)

More on teams in real projects in the course

Expectation

- Meet initially and then regularly
- Review team policy
- Divide work and integrate
- Establish a process
- **Set and document clear responsibilities and expectations**
  - Possible Roles: Coordinator, Scribe, Checker, Monitor
  - Rotate roles every assignment
- Every team member should understand the entire solution

Dealing with problems

- Openly report even minor team issues in individual part of the milestone report
- In-class discussions and case studies
- Additional material throughout semester
- We will attend one team meeting

Planning and In-Team Communication

- Asana, Trello, Microsoft Project, …
- Github Wiki, Google docs, …
- Email, Slack, Facebook groups, …

Project 1 – Milestone 1 Team Workflow

- The team workflow is a document that outlines team roles. It also gives us information about organizational issues, like team meeting times. This helps us send course staff to aid you and helps us to follow your progress.
- The main purpose of this document is to give you some rules for team process, management, tracking, and goal setting. As a general rule, groups work pretty well in this course.However, any good working group will have some measurements in place if something goes awry.