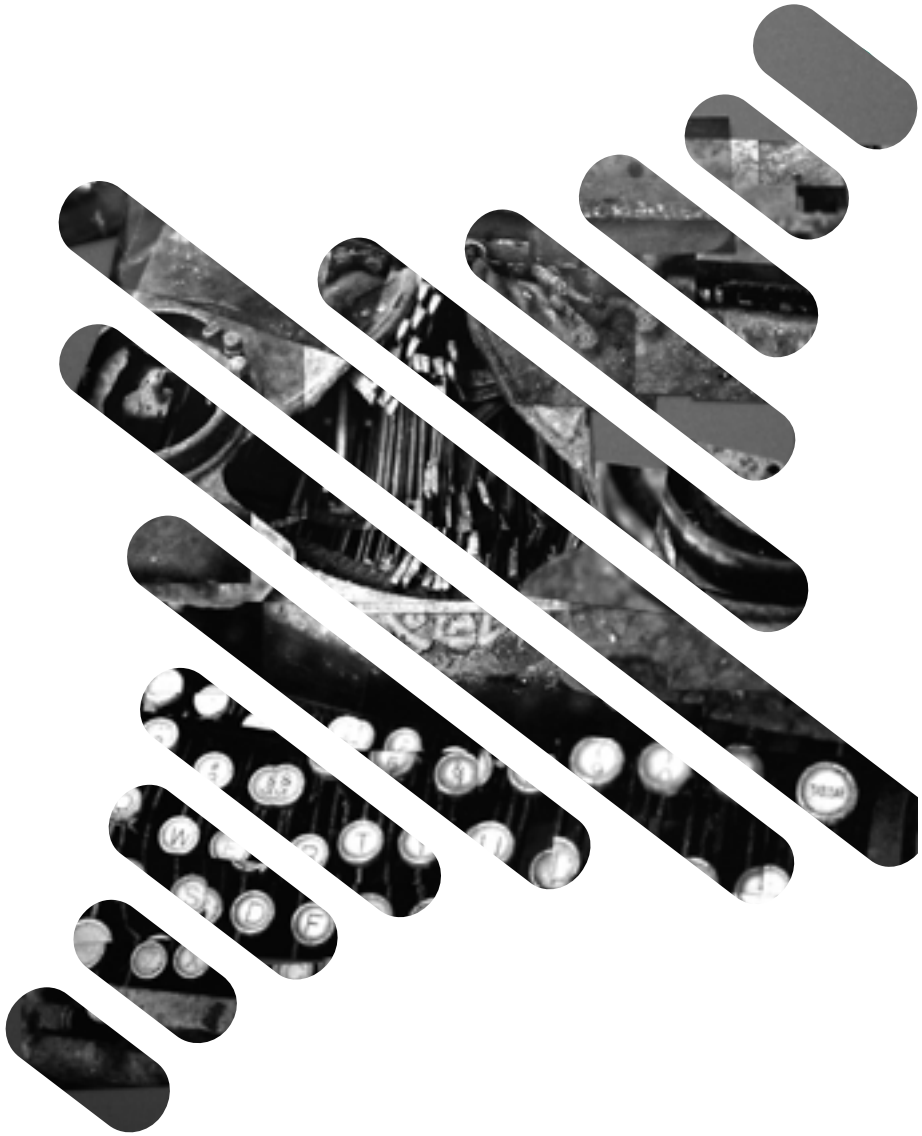
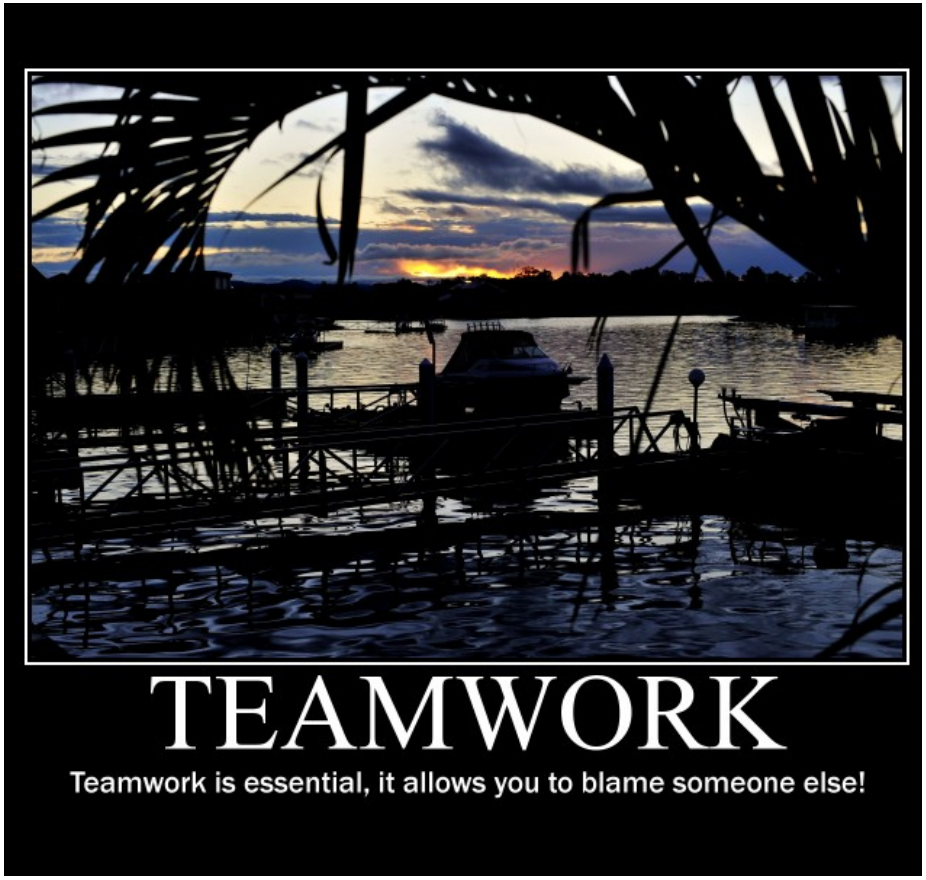
CS x278
UNIV x278

Principles of Software Engineering & Tackling Big Problems with Mobile Cloud Computing



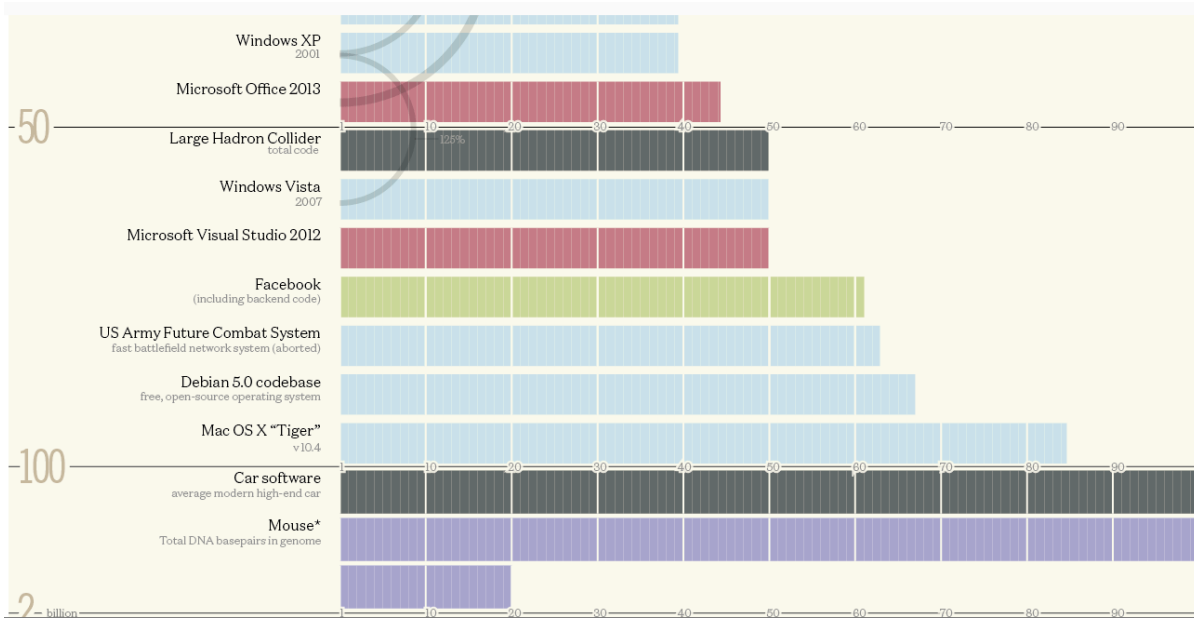


www.theextraedge.org

Yes, you are in the right place

Both UNIV x278 and CS x278 will be meeting in the same classroom this year. You will be part of a new interdisciplinary approach to learning about building software.

Lines of Software Source Code



www.informationisbeautiful.net/visualizations/million-**lines**-of-**code**

> "Software is eating the world"
>
> MARC ADREESEN

http://peace-love-run.tumblr.com/

Don’t get eaten.

This class is designed to help you understand how to create software to solve real problems. If you don’t understand how to create software, you will be at a significant disadvantage in the future.

1. Risk of Dying in a Car Accident (lifetime)
2. Risk of Dying if Infected with the Bubonic Plague
3. Risk of a Software Project Being Considered Unsuccessful
4. Risk of Dying if You Have a Heart Attack
5. Risk of Dying if Infected with Ebola

Please order the likelihood of these outcomes.

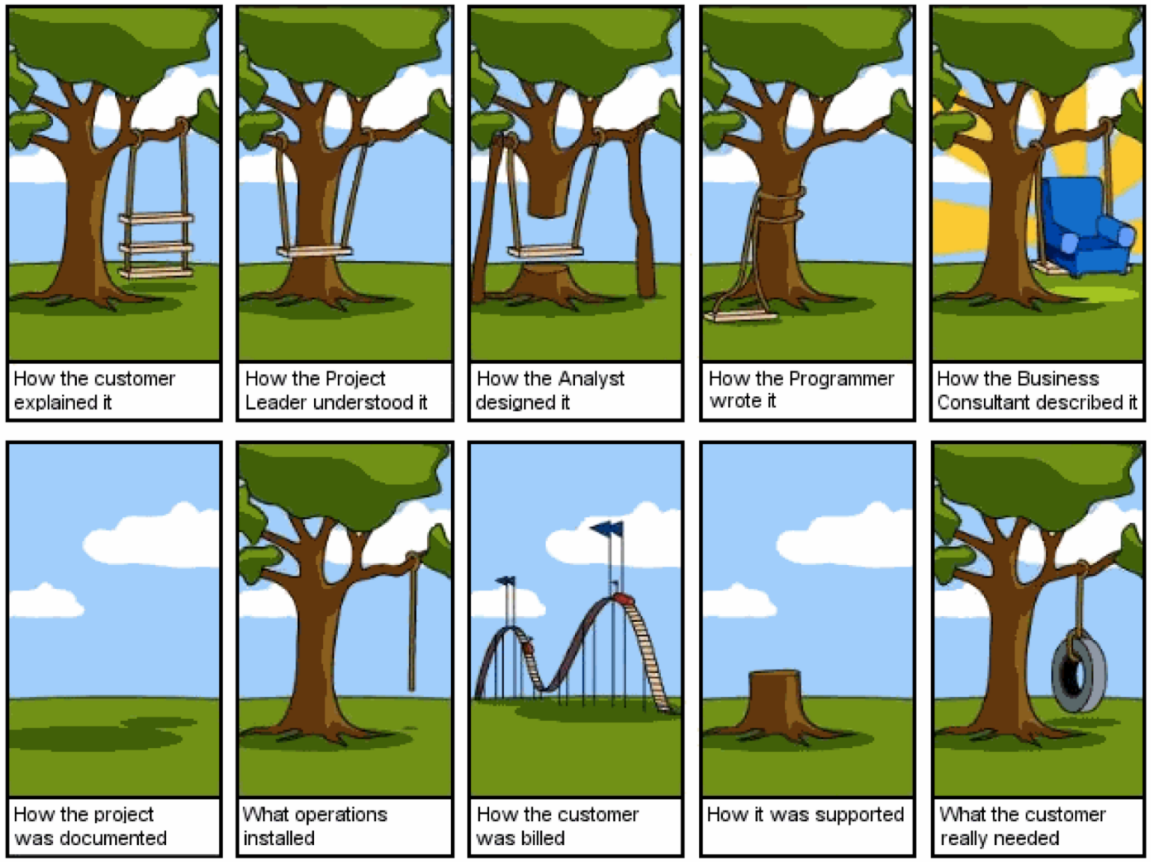
Don't over think this. This is not scientific.

1. Risk of a Software Project Being Considered Unsuccessful ~60%
2. Risk of Dying if Infected with Ebola ~50%
3. Risk of Dying if Infected with the Bubonic Plague ~11%
4. Risk of Dying if You Have a Heart Attack ~4.7%
5. Risk of Dying in a Car Accident (lifetime) < 1%

Uh oh.

Don't over think this. This is not scientific.

It’s their fault!



blog.codinghorror.com

Learn by doing.

You can't learn about building software just by reading.

http://vi.sualize.us/ikea_facile_da_montare_o_no_humor_ikea_funny_picture_Jgf.html

Course Structure

Step 1

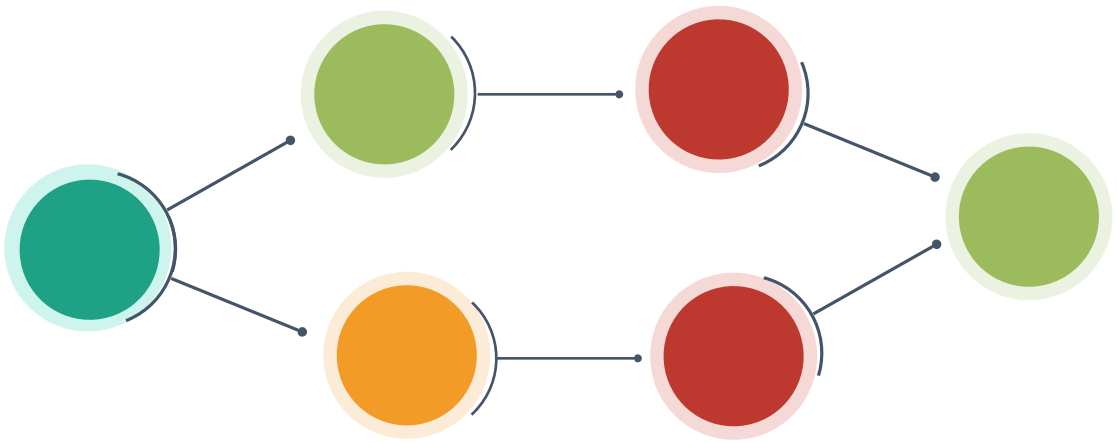
UNIV & CS Students - Software Creation Fundamentals

Step 2

UNIV Students - Faculty Mentor Research Challenges

Step 3

UNIV Students – Design Solutions for Mentor Challenges



Step 2

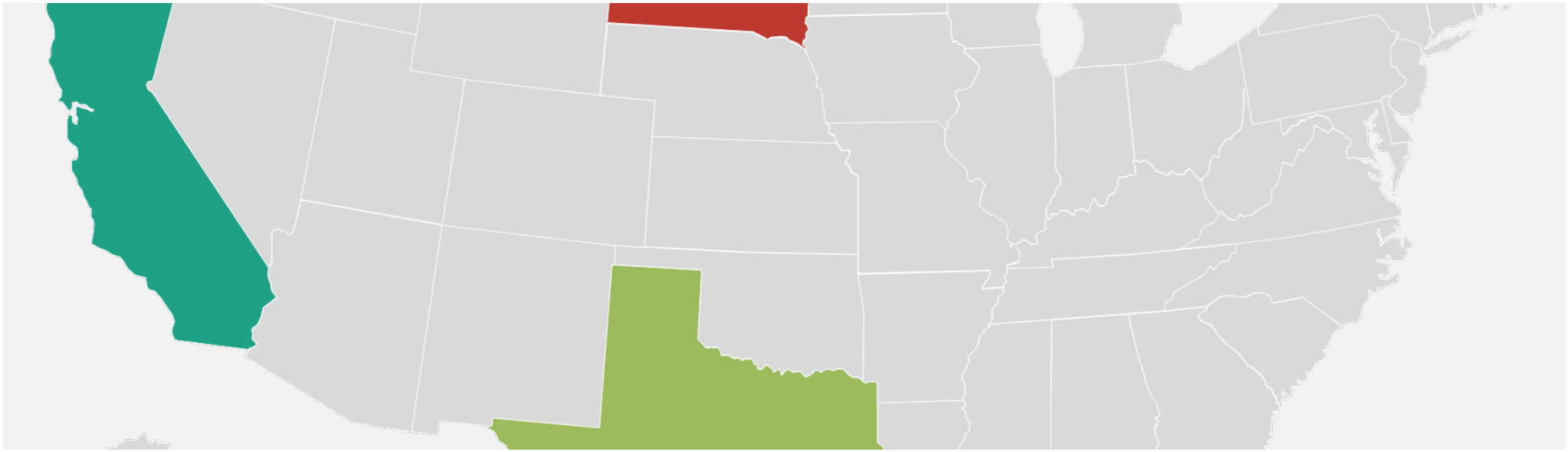
CS Students - Critical Software Engineering Tools

Step 3

CS Students – Design Programming Assignments for Peers

Step 4

UNIV & CS Students – Complete Mentor Projects



What happens when you graduate?

No more classes

Expected to be able to figure things out on your own

Will be passed by if you don't keep learning

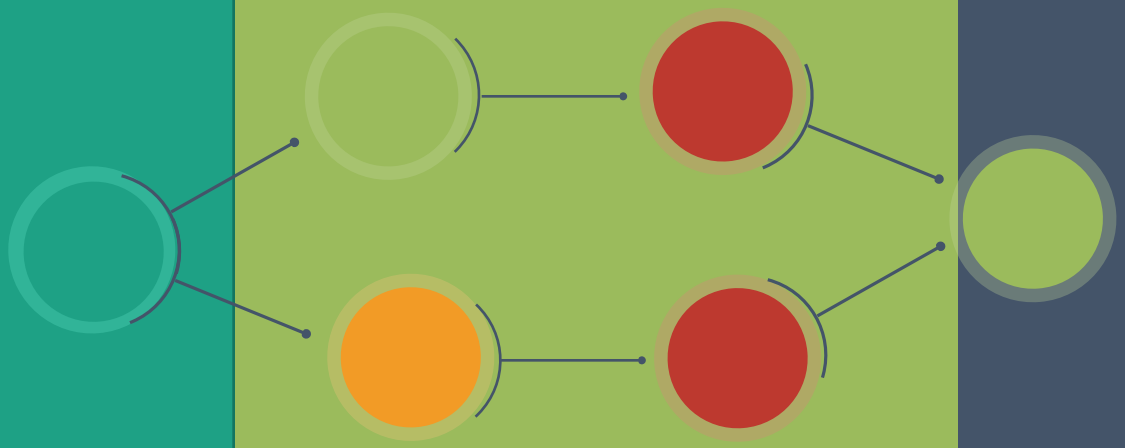
Will be expected to know how to start from scratch

Course Phases

Advocate for Reality

Understand the challenges of software development & be able to explain them to others

Self-Sufficiency



Be able to Google for solutions, learn new languages, or figure out what an SOW is on your own

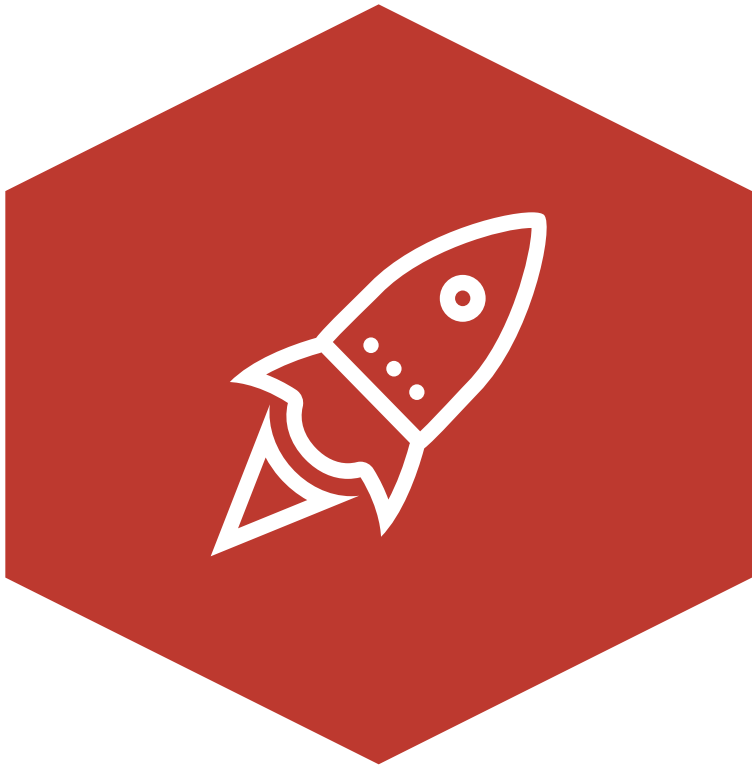
Immersion

Start a real project from scratch and support it with real users

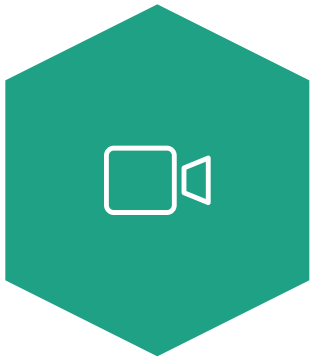
Launch Products to Learn

The most effective way to learn how to build software is to launch a product.

The more products that you launch and support, the more that you will learn. The assignments are designed to be as close to a real product launch as possible with real customers. For many of the assignments, your peers will be your customers and you will have to support them.

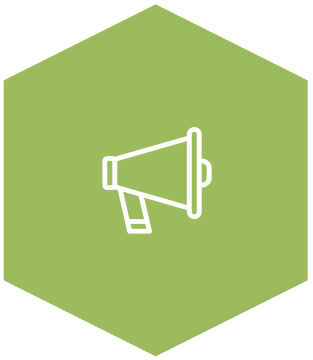


Things You Will Do On Your Own



Learn New Frameworks

Spring Boot, React, Docker, Gradle, JHipster, etc.

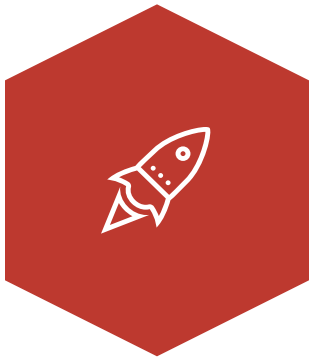


Write Specifications

Learn how challenging it is to capture the details of real solutions

Start from Scratch

Be able to tackle any problem without hand holding or step-by-step instructions

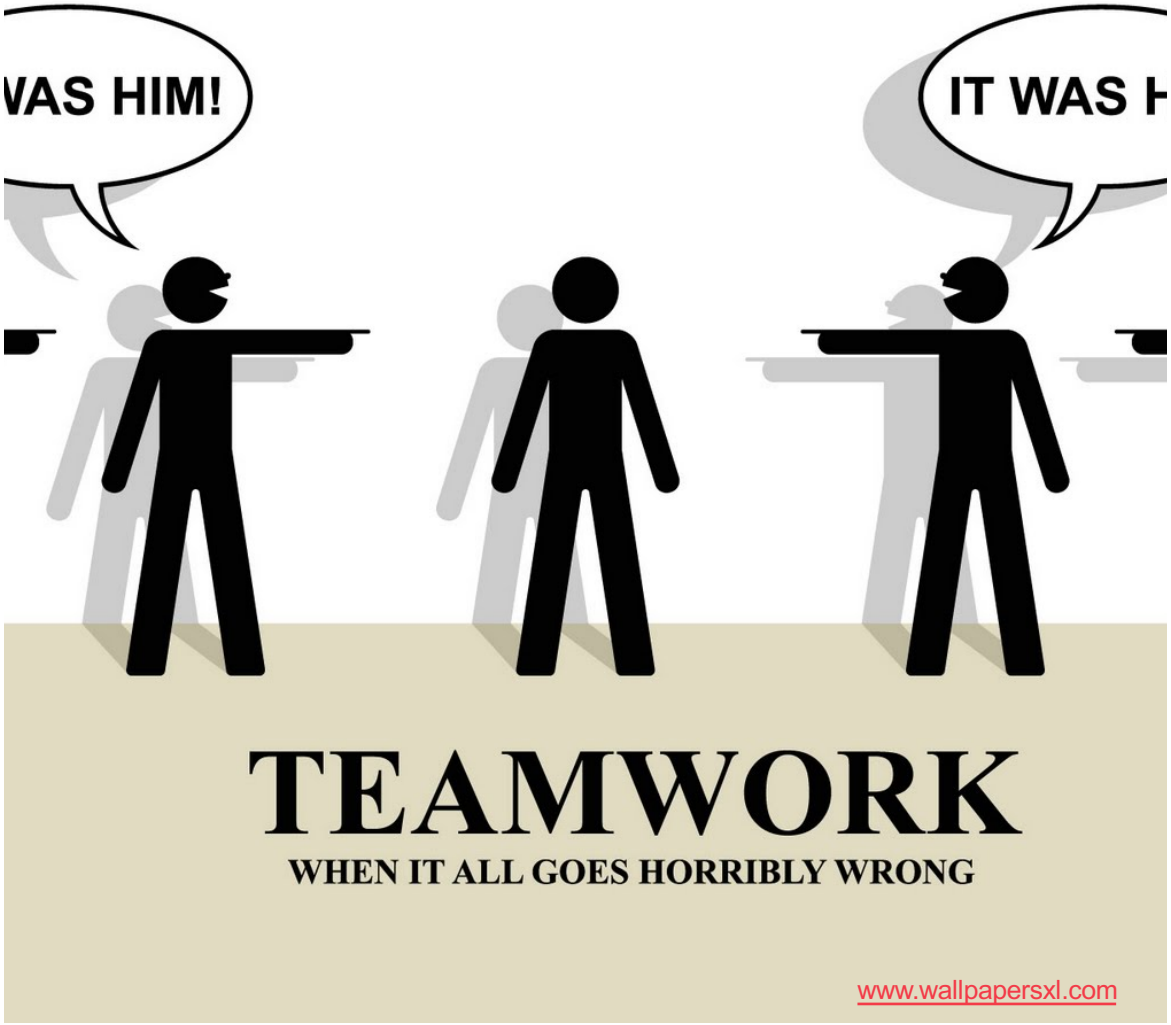


Launch & Support

Supporting real users is hard, you will find out why.

Creating team environments with in-class exercises

A lot of the work of software creation happens in meetings, conference calls, and other group interactions where teams communicate. A lot of the classes will be dedicated to team exercises to simulate these types of scenarios.



Hands-on

Outside Research

Requires Engagement

Few Lectures

Course Content

Please write these down

GitHub

https://github.com/juleswhite/CSX278

Piazza

https://piazza.com/class/irt1fu5fyvc3n