# CEN 5016: Software Engineering

Spring 2024

University of Central Florida

Dr. Kevin Moran

## *Week 1 - Class 1:* Course Overview

# Welcome to CEN 5016!

- **<u>Welcome to the First Lecture!</u>**
- *This lecture is being recorded*

# Introductions

***Instructor:*** Kevin Moran

***Education:*** Ph.D. from William & Mary - 2018

***Research Interests:*** Software Engineering , UI Analysis, Machine Learning

***Office Hours:*** *TBA (Fill Out Entrance Survey!)*

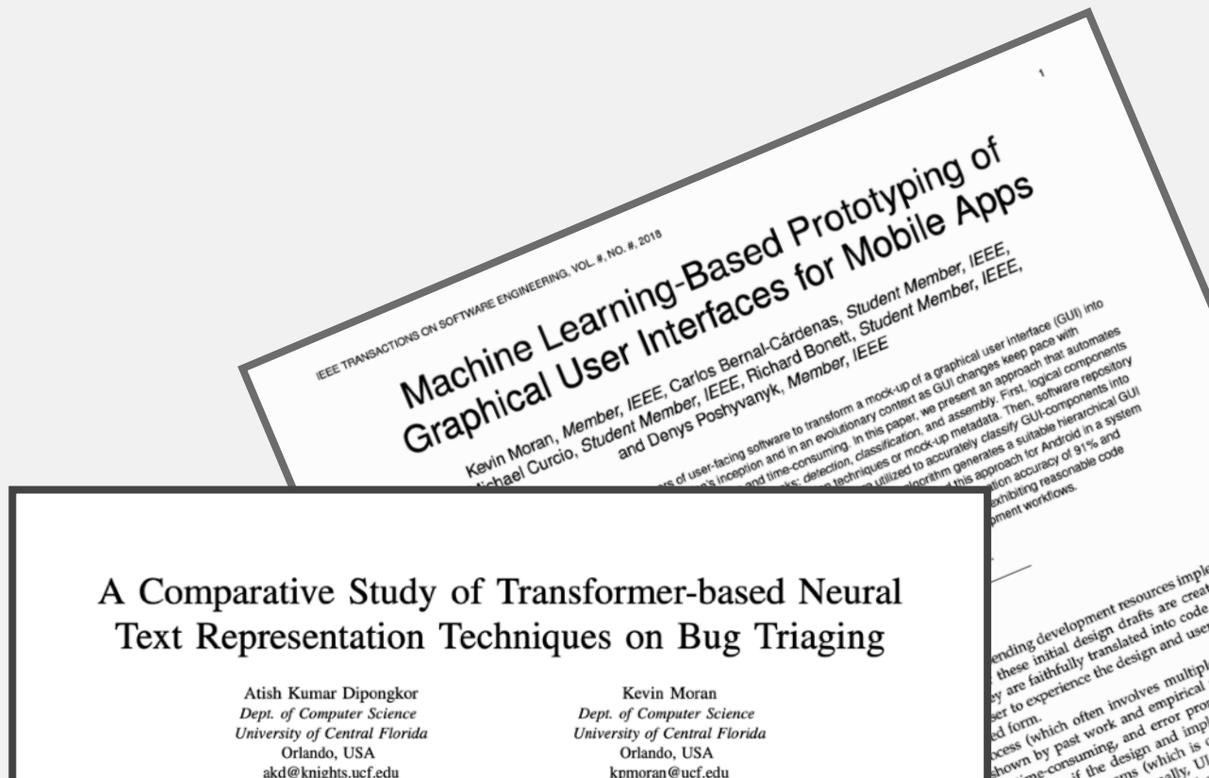
# Introductions

***Instructor:*** Kevin Moran

***Education:*** Ph.D. from William & Mary - 2018

***Research Interests:*** Software Engineering , UI Analysis, Machine Learning

***Office Hours:*** *TBA (Fill Out Entrance Survey!)*



# Introductions

***Instructor:*** Kevin Moran

***Education:*** Ph.D. from William & Mary - 2018

***Research Interests:*** Software Engineering , UI Analysis, Machine Learning

***Office Hours:*** *TBA (Fill Out Entrance Survey!)*

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. #, NO. #, 2018

Machine Learning-Based Prototyping of Graphical User Interfaces for Mobile Apps

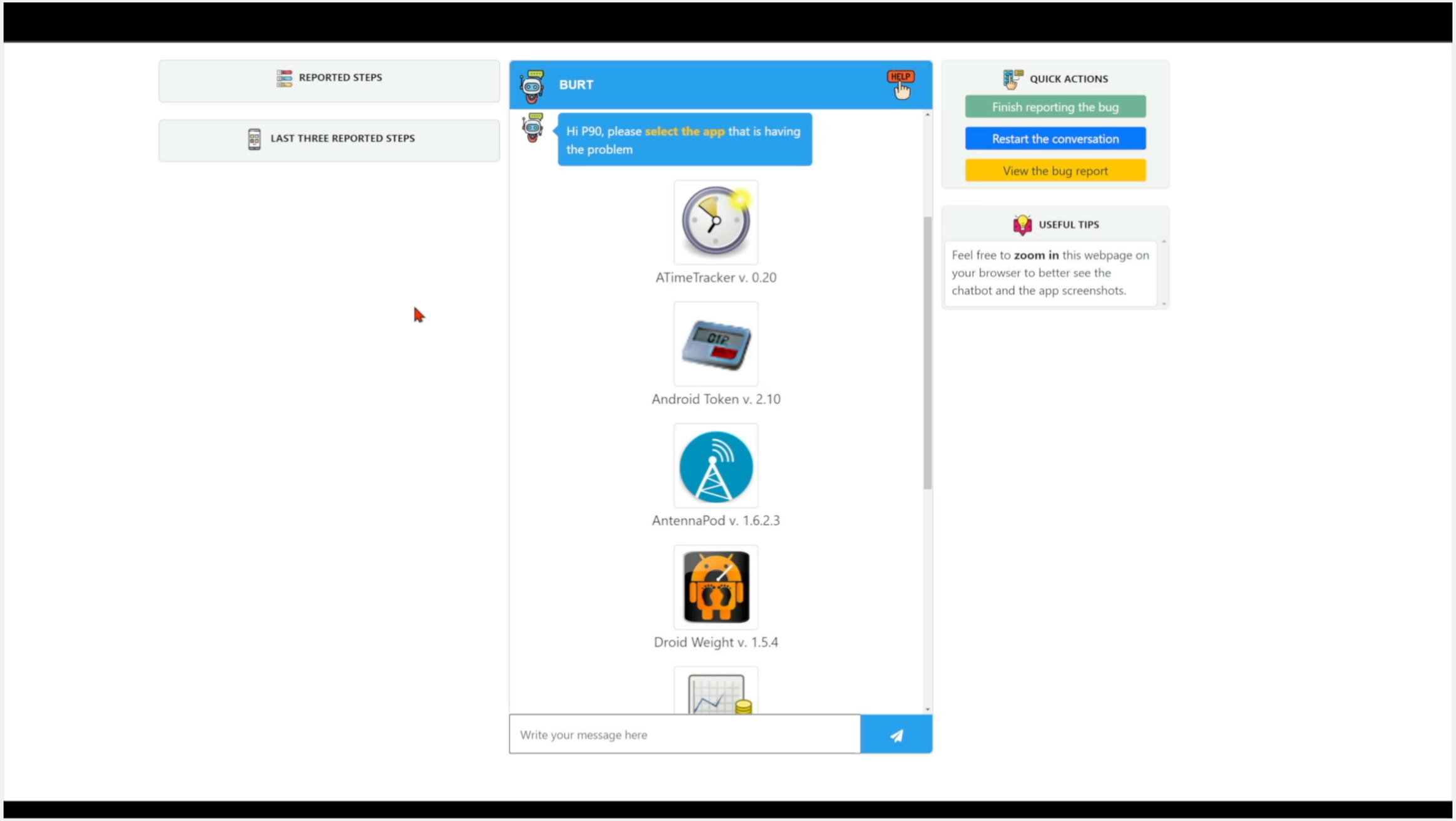
Kevin Moran, *Member, IEEE*, Carlos Bernal-Cárdenas, *Student Member, IEEE*, Michael Curcio, *Student Member, IEEE*, Richard Bonett, *Student Member, IEEE*, and Denys Poshyvanyk, *Member, IEEE*

A Comparative Study of Transformer-based Neural Text Representation Techniques on Bug Triaging

Atish Kumar Dipongkor
*Dept. of Computer Science*
*University of Central Florida*
Orlando, USA

Kevin Moran
*Dept. of Computer Science*
*University of Central Florida*
Orlando, USA

# Recent Research



# Today's Agenda

- Provide an overview of the Course Logistics - (20-30 mins)
- Provide an Introduction to the Course Philosophy and Software Engineering more broadly

# Course Logistics

# Course Lectures

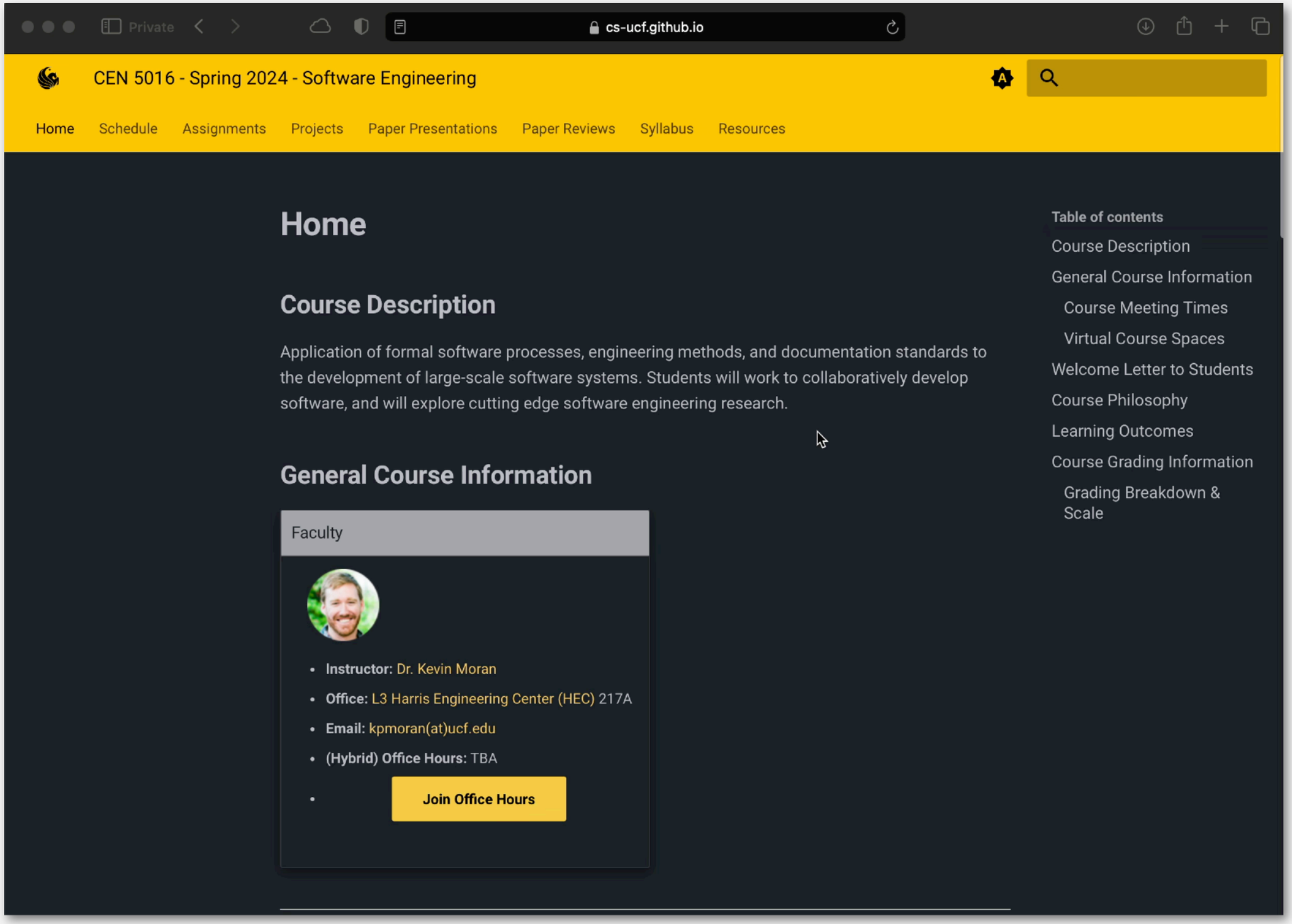
- I will not take attendance, however, you should come to class to learn and practice the concepts

- There will be in-class *quizzes* that will be only graded for completion. These quizzes will not be announced and will count as a form of a participation grade.

# Course Resources

- Course Website: Syllabus, Schedule, Assignments, Lecture slides/recordings
- Webcourses (Canvas): Grades, Assignments, Discussions
- Zoom: Hybrid/Virtual Office Hours
- Ed Discussions: Course Communications *(You should have received an invite link)*

# Course Website



# Course Materials

- There is no course textbook, however notes will be posted to the course website.
- There will be in-class activities for many lectures (***bring your laptop!***)
- I will post supplementary readings from the "*Software Engineering at Google*" book, which is freely available at: ***https://abseil.io/resources/swe-book***

# Course Structure

- This course is primarily broken down into two halves:
  - *Part 1: Practical Software Engineering Knowledge*
    - In this part of the course, we will go explore advanced strategies for building software in team-based settings at scale, and you will apply what you learn to the development project.
  - *Part 2: Exposure to Software Engineering Research*
    - In this part of the course, I will provide an overview of current developments in research on software engineering, and we will present and discuss recent research papers. You will apply your practical software engineering to improve some open-source research code.

# Grading Breakdown

- Midterm Exam - 15%
- Quizes - (5%)
- Individual Assignments - (10%)
- Research Paper Presentation - (5%)
- Software Development Project - (30%)
- Research Reproduction Project - (30%)
- Research Paper Reviews - (5%)

# In-Class Activities

- Work together in small pairs/groups to gain experience trying out methods and concepts with examples

- No grades, but very important, as you will learn a lot from your classmates during these exercises

# Midterm Exam (15%)

- Essay-based questions from the material presented in class
- Will test knowledge of concepts presented in class
- Will ask you to apply concepts to various scenarios
- All exams are closed book
- See the Syllabus for more policies

# Individual Assignments (10%)

- These assignments are to be completed individually and will help you gain some experience with background tools and knowledge needed to complete the projects.
- The first assignment will help get you familiar with the git and GitHub workflow that we will be using for this class.

# Quizzes (5%)

- Will be taken live in class
- Graded for completion only
- Mainly for me to see how the class is understanding various concepts that will be introduced throughout the semester
- Quizzes will not be announced ahead of time
- *You can miss up to two quizzes without penalty*

# Software Development Project (30%)

- This will be a development project broken down into multiple parts.
- This semester, the project will involve working on and extending a web application written in Javascript and Typescript.
- You will complete this project in teams of 4-5 students.
- There will be both technical and non-technical deliverables that will be clearly communicated on the course webpage.

# Research Reproduction Project (30%)

- The second half of this course will provide you exposure to cutting-edge research that is being done on topics in software engineering.
- The main goal of this project is to take code artifacts that are associated with a research paper, reproduce the paper's results, and improve the code.
- This will be done in the same teams of 4-5 students as the software development project, but will come later in the semester.
- This will also include both technical and non-technical deliverables.

# Research Paper Presentations (5%)

- In the second half of the course, students will present research papers, and we will discuss the practicality, impact, and rigor of these papers in class.
- Each project project group will present a single research paper live in class.
- I will provide a detailed lecture with presentation tips.

# Research Paper Reviews (5%)

- During the second half of the course, you will be asked to write two research paper reviews
- I have provided detailed instructions for this on the course website, and will also revisit this when we start the second half of the semester
- These reviews will be completed *individually* and you are free to choose whichever papers you like to review.

# Late Policy

- All assignments must be submitted through Webcourses
- All programming assignments can be turned in *up to 24 hours late*, for a *10% penalty* or *up to 48 hours late* for a *20% penalty*
- *Quizzes cannot be turned in late*
- If you become ill, if you have a family emergency, or if something else happens that prevents you from completing work on time (either assignments or exams), please talk with me.

# Academic Integrity

- Learn the University and Course Policies for Academic Integrity
- HW Assignments are 100% individual or within your group
- Discussing assignments at high level: ***OK***, sharing code: ***NOT OK***
- If in doubt, ask the instructor
- If you copy code, we ***WILL*** notice (see some of my recent research results on Code Traceability)

# Academic Integrity (cont.)

- OK Questions to ask your classmates:
  - *My code compiles with this strange error message. What does that mean?*
  - *Can you clarify what the instructor means in the directions for this assignment?*
  - *Can you explain what happens in memory when a variable is declared?*
  - *I am not sure how all these user defined functions in the large program assignment work together. Can you explain how they are all are supposed to work together?*

# AI Tools (ChatGPT)

- The use of AI tools to assist in coding are allowed in this class - but only in specific circumstances.
- Some things to remember:
  - Put Effort into Crafting High Quality Prompts
  - Be Aware of AI Limitations
  - Give the Tool Proper Attribution
  - Know when to use and not use AI Tools
- *More on this next class*

# Policies

- My promises to you:
- Quiz results will be available in the class after the quiz is taken
- Assignments, Project Checkpoints, and Exams will be graded within 1 week of submission

# Why Software Engineering?

# Why is software Engineering Important?

- The world runs on code
- Coding is an incredible form of engineering, it allows to make abstract ideas concrete, and build incredible things.
- Programming allows us to solve a variety of problems
- It can help us to accomplish things that would otherwise be impossible through automation

# Why Software Engineering?

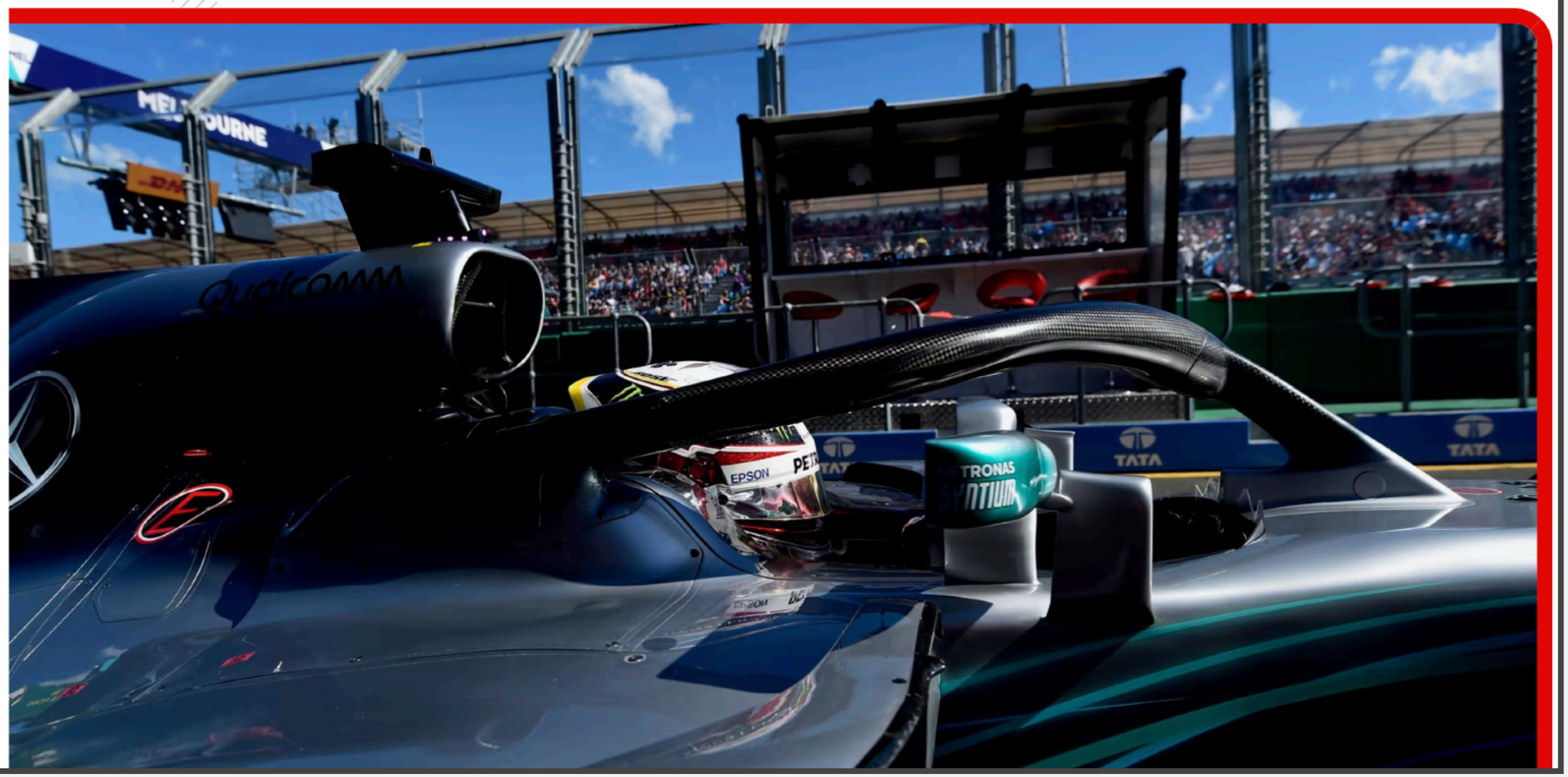
# Why Learn How to Program?

150 Million Lines of Code

# Software glitch cost Hamilton victory - Mercedes

25 March 2018

MERCEDES AUSTRALIA HAMILTON

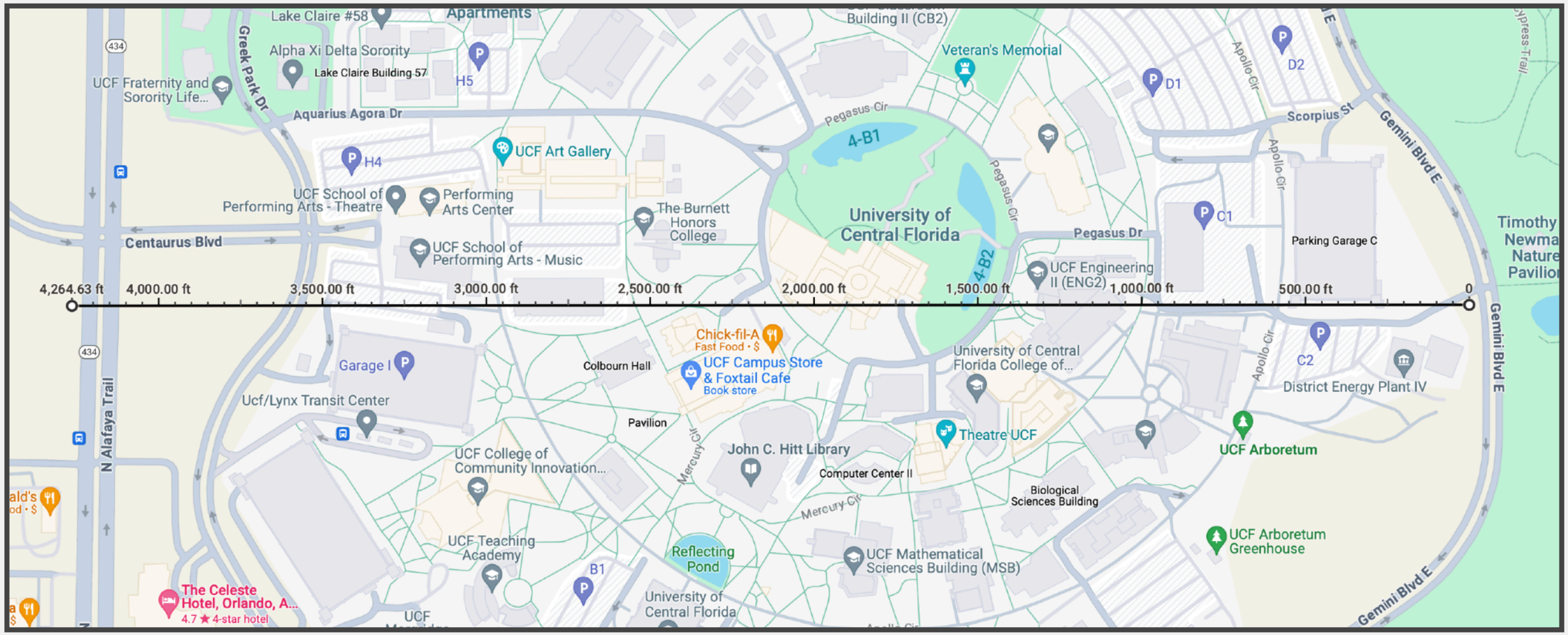


# The Harm of Bad Software Engineering

# The Harm of Bad Software Engineering

# Vasa

# Vasa


Lake Claire #58
Apartments
Alpha Xi Delta Sorority
Lake Claire Building 57
H5
UCF Fraternity and Sorority Life...
Greek Park Dr
Aquarius Agora Dr
H4
UCF Art Gallery
UCF School of Performing Arts - Theatre
Performing Arts Center
The Burnett Honors College
UCF School of Performing Arts - Music
Centaurus Blvd
Building II (CB2)
Veteran's Memorial
Pegasus Cir
4-B1
D1
D2
Apollo Cir
Scorpius St
Gemini Blvd E
Cypress Trail
University of Central Florida
Pegasus Dr
C1
Parking Garage C
4-B2
UCF Engineering II (ENG2)
Timothy Newma Nature Pavilio
4,264.63 ft
4,000.00 ft
3,500.00 ft
3,000.00 ft
2,500.00 ft
2,000.00 ft
1,500.00 ft
1,000.00 ft
500.00 ft
0
Chick-fil-A
Fast Food · $
UCF Campus Store & Foxtail Cafe
Book store
Colbourn Hall
Garage I
Ucf/Lynx Transit Center
Pavilion
University of Central Florida College of...
C2
District Energy Plant IV
Theatre UCF
UCF Arboretum
John C. Hitt Library
Computer Center II
Biological Sciences Building
UCF College of Community Innovation...
Mercury Cir
UCF Teaching Academy
Reflecting Pond
B1
UCF Mathematical Sciences Building (MSB)
UCF Arboretum Greenhouse
The Celeste Hotel, Orlando, A...
4.7 ★ 4-star hotel
N Alafaya Trail
Gemini Blvd E

# Vasa Syndrome

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

Requirements

- Shipwright's death

Teams

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

Metrics

- Failed pre-launch stability tests

Testing & Q/A

# Software Engineering?

- What is engineering?
- And how is it different from hacking/programming?

# 1968 NATO Conference on Software Engineering

- Provocative Title
- Call for Action
- “Software Crisis”

https://isthisit.nz/posts/2022/1968-nato-software-engineering-conference/

# Margaret Hamilton

- Wrote the guidance software for the Apollo Moon Missions
- Credited with initially coining the term “software engineering”

# Margaret Hamilton

# Where does Software Engineering Fit in?

- Computer science: focusing on computer hardware, compilers, operating systems, and programming languages
- Software engineering: a discipline that uses computer and software technologies as a problem-solving tools

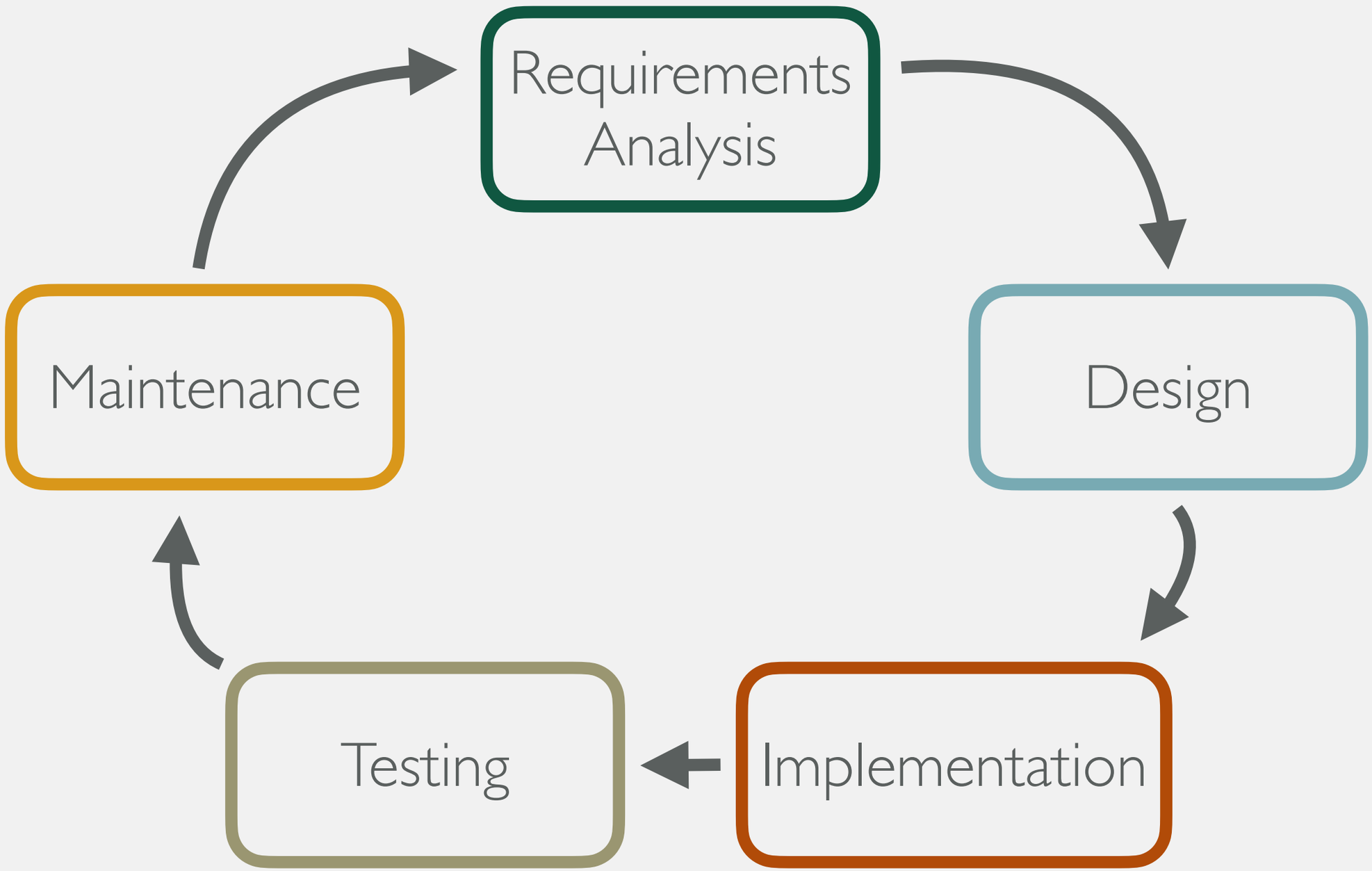
# An Engineering Approach

- Requirement analysis and definition
- System design
- Program design
- Writing the programs
- Unit testing
- Integration testing
- System testing
- System delivery
- Maintenance

# The Software Engineering "Process"

- A process: a series of steps involving activities, constrains, and resources that produce an intended output of some kind
- A process involves a set of tools and techniques

# The Software Engineering "Process"



# Software Engineering Research

**AI and software engineering:**

- Search-based software engineering
- Machine learning with and for SE
- Recommender systems
- Autonomic systems and self adaptation
- Program synthesis
- Program repair

**Testing and analysis:**

- Software testing
- Program analysis
- Debugging and Fault localization
- Programming languages
- Performance
- Mobile applications

**Software analytics:**

- Mining software repositories
- Apps and app store analysis
- Software ecosystems
- Configuration management
- Software visualization

**Dependability:**

- Formal methods
- Validation and Verification
- Reliability and Safety
- Privacy and Security
- Embedded and cyber-physical systems

**Software evolution:**

- Evolution and maintenance
- API design and evolution
- Release engineering and DevOps
- Software reuse
- Refactoring
- Program comprehension
- Reverse engineering

**Social aspects of software engineering:**

- Human aspects of software engineering
- Human-computer interaction
- Distributed and collaborative software engineering
- Agile methods and software processes
- Software economics
- Crowd-based software engineering
- Ethics in software engineering
- Green and sustainable technologies

**Requirements, modeling, and design:**

- Requirements Engineering
- Privacy and Security by Design
- Modeling and Model-Driven Engineering
- Software Architecture and Design
- Variability and product lines
- Software services

# For This Week

- *Assignment 1:* Due next Tuesday at 11:59pm
  - Will be released tomorrow
- *Course Entrance Survey:* Due Friday at 11:59pm
  - Will be released today after class