COMP 433 Software Engineering

Module 0: *Course Overview*

Ahmed Tamrawi

 atamrawi  atamrawi.github.io  ahmedtamrawi@gmail.com

B.Eng. Computer Engineering
(Class of 2007)

IOWA STATE UNIVERSITY

M.Sc. Computer Engineering
(Class of 2011)

IOWA STATE UNIVERSITY

Ph.D. Computer Engineering
(Class of 2016)

amazon

Secure Programming

Static Program Analysis

Data & Pattern Mining

Software Analysis & Security

Bug finding and Malware detection

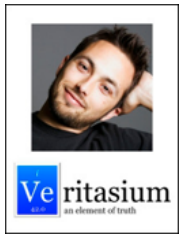
Build System Analysis

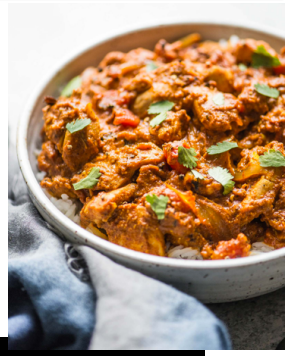
Abstractions and Symbolic Evaluations

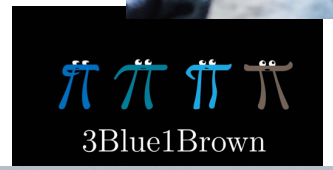
Quantum Physics

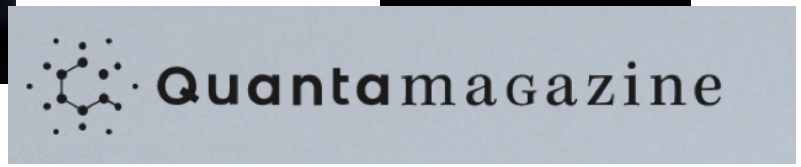
Biology

Astronomy

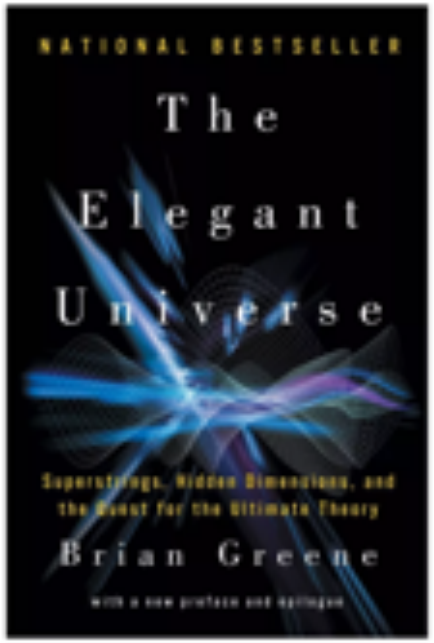








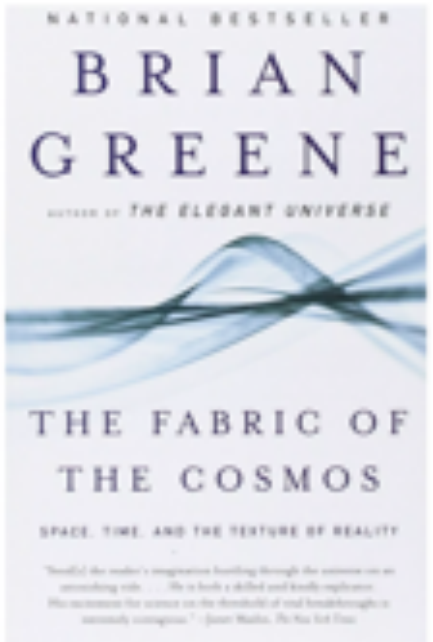
Brian Green's Books



The Elegant Universe

A rare blend of scientific insight and writing as elegant as the theories it explains

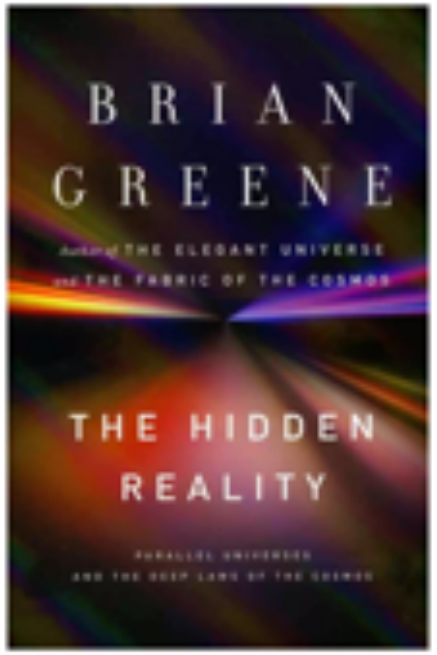
Learn more ›



The Fabric of the Cosmos

An irresistible and revelatory journey to the new layers of reality that modern physics has discovered

Learn More ›



The Hidden Reality

A remarkable adventure through a world more vast and strange than anything we could have imagined

Learn More ›

Icarus at the Edge of Time

A futuristic reimagining of the classic Greek myth, for younger audiences

Learn More ›

What does the program print?

```
1  public class JavaPuzzle {
2
3      private JavaPuzzle internalInstance = new JavaPuzzle();
4
5      public JavaPuzzle() throws Exception {
6          throw new Exception("I'm not coming out!");
7      }
8
9      public static void main(String[] args) {
10         try {
11             JavaPuzzle p = new JavaPuzzle();
12             System.out.println("Surprise!");
13         } catch (Exception e) {
14             System.out.println("I told you so!");
15         }
16     }
17 }
```

Does the method work?

```
1 public static boolean isOdd(int i) {
2         return i % 2 == 1;
3 }
```

Does the method work?

Unfortunately, it doesn't; it returns the wrong answer one quarter of the time.

```
1 public static boolean isOdd(int i) {
2         return i % 2 == 1;
3 }
```

How to fix it?

Does the method work?

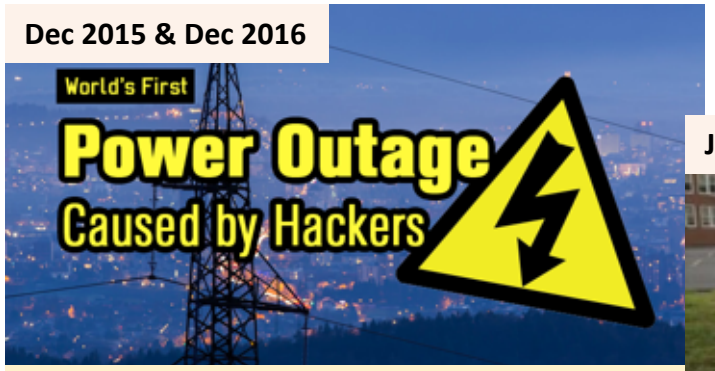
```
1 public static boolean isOdd(int i) {
2         return i % 2 != 0;
3 }
```

Can we do better?

Does the method work?

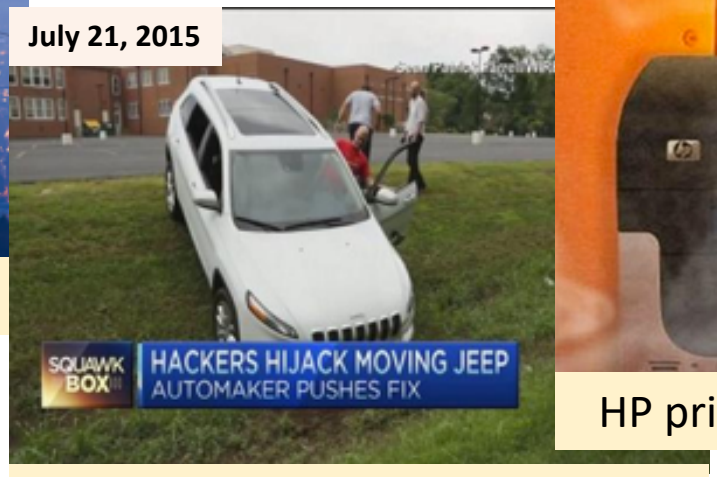
```
1 public static boolean isOdd(int i) {
2         return (i & 1) != 0;
3 }
```

Dec 2015 & Dec 2016



Ukraine power grid attacks

July 21, 2015

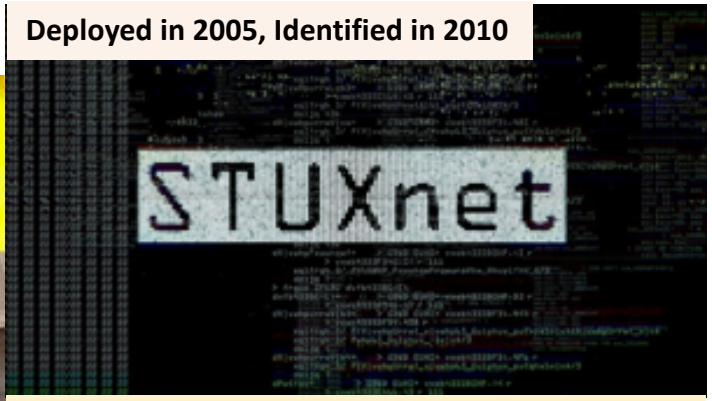


Jeep remotely hijacked

November 29, 2011

HP printers remotely set on fire

Deployed in 2005, Identified in 2010

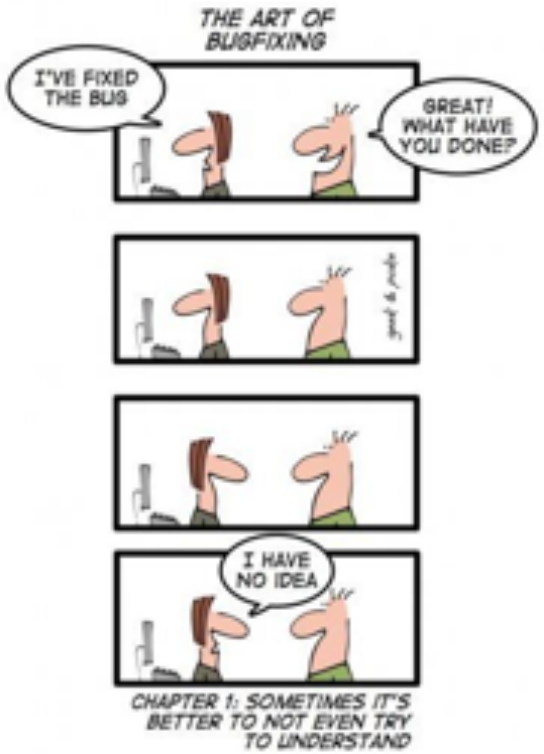


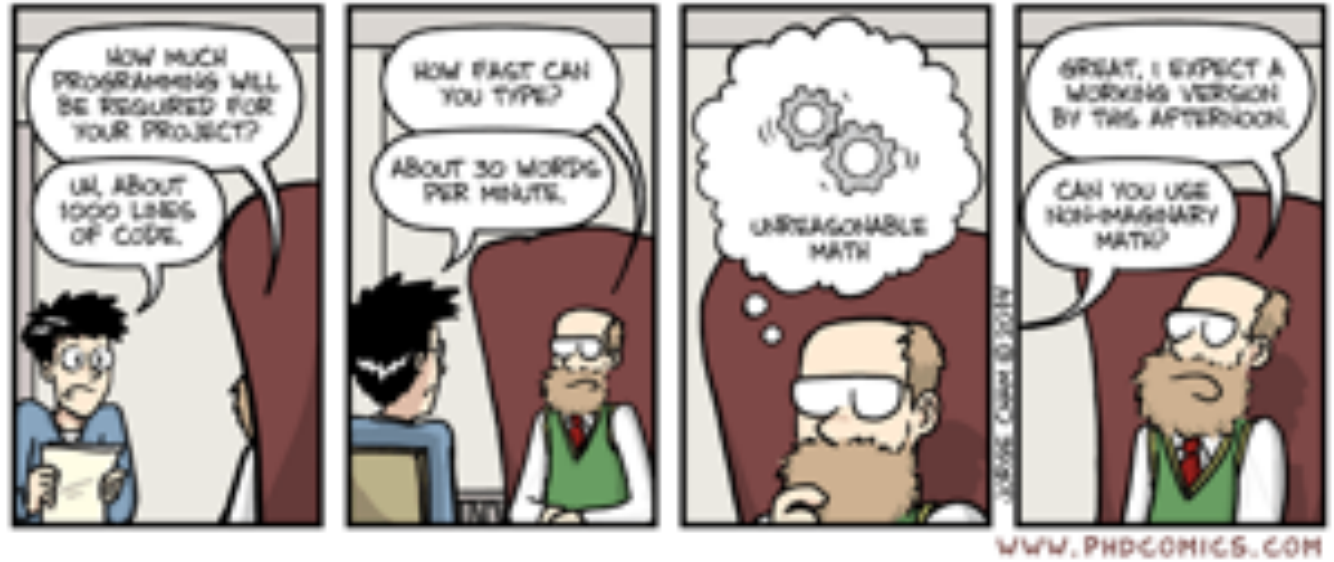
STUXnet Worm

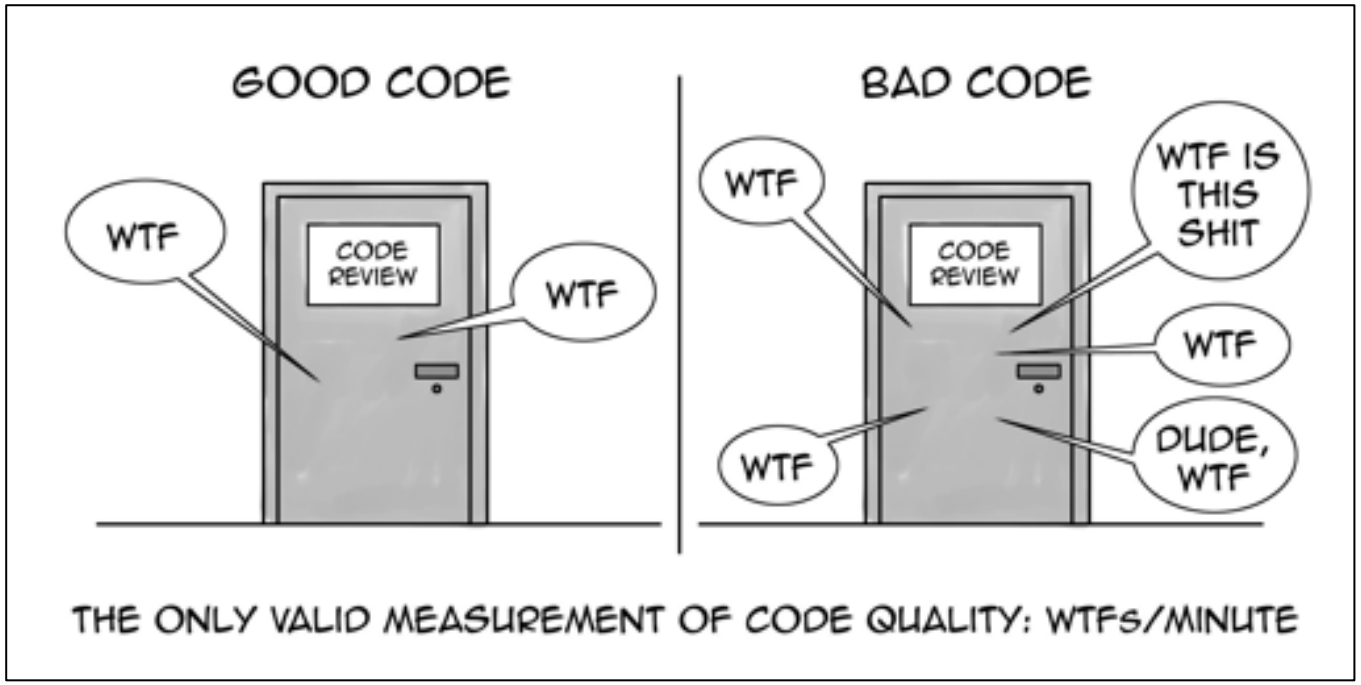
{ Complex Software }

Although software development practice has *advanced rapidly* in recent years, common practice hasn't

Many programs are still **buggy**, **late**, and **over budget**, and many **fail** to satisfy the needs of their users


THE ART OF BUGFIXING
I'VE FIXED THE BUG
GREAT! WHAT HAVE YOU DONE?
I HAVE NO IDEA
CHAPTER 1: SOMETIMES IT'S BETTER TO NOT EVEN TRY TO UNDERSTAND


HOW MUCH PROGRAMMING WILL BE REQUIRED FOR YOUR PROJECT?
UH, ABOUT 1000 LINES OF CODE.
HOW FAST CAN YOU TYPE?
ABOUT 30 WORDS PER MINUTE.
UNREASONABLE MATH
GREAT, I EXPECT A WORKING VERSION BY THIS AFTERNOON.
CAN YOU USE NON-IMAGINARY MATH?
WWW.PHDCOMICS.COM


GOOD CODE
WTF
CODE REVIEW
WTF
BAD CODE
WTF
WTF IS THIS SHIT
CODE REVIEW
WTF
WTF
DUDE, WTF
THE ONLY VALID MEASUREMENT OF CODE QUALITY: WTFS/MINUTE


HAVE YOU FIXED THE BUG?
IN PRINCIPLE YES
YES
YOU MEAN: NO
geek & poke

Course URL: https://atamrawi.github.io/teaching/**comp433_spring21**

BIRZEIT UNIVERSITY
Computer Science Department

Software Engineering (COMP 433)
Semester 2nd 2020/2021

Instructors:

Section 1	Dr. Adel Taweel (ataweel@birzeit.edu)
Section 2	Dr. Samer Zein (szein@birzeit.edu) *Course Coordinator*
Section 3	Dr. Adel Taweel (ataweel@birzeit.edu)
Section 4	Dr. Ahmad Tamrawi (atamrawi@birzeit.edu)

A. **Sommerville I. (2001,2004-2010) *Software Engineering*, 9th Edition, Addison-Wesley, Harlow, Essex, UK (older editions can also be suitable for this course)**

Suggested reading:

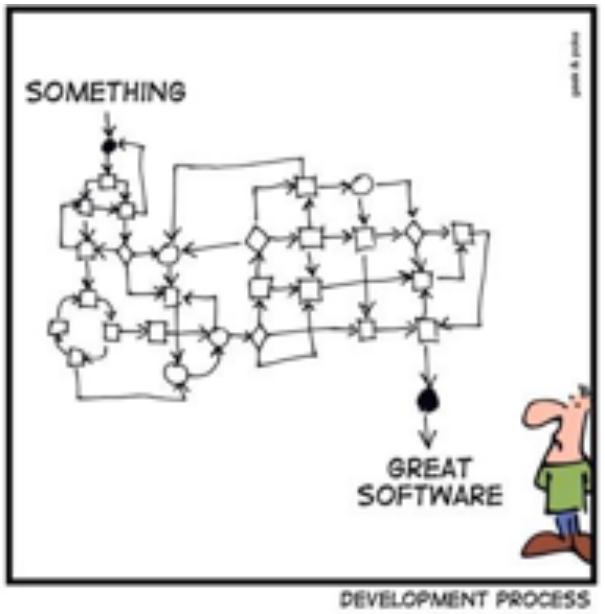
B. Bruegge and Dutoit, ***Object-Oriented Software Engineering Using UML, Patterns, and Java***, 3rd Edition, Prentice Hall, 2013.
C. Stevens P. with Pooley, R. (2005) ***Using UML: Software Engineering with Objects and Components***, 2nd Edition, Addison-Wesley, Harlow, Essex, UK
D. Jeffrey A. Hoffer, Joey F. George, Joseph S. Valacich. (2005) ***Modern System Analysis and Design*** 4th - 7th Edition, (2013), Prentice Hall.
E. Roger Pressman, ***Software Engineering: A Practitioner's Approach***, 7-8th edition, McGraw-Hill; 2014.
F. L.A. Maciaszek, ***Requirements Analysis and System Design: Developing Information Systems with UML***, 1-3rd Edition, Addison Wesley, 2007.

Introduction:
Software engineering is the discipline concerned with the application of theory, knowledge, and practice for effectively and efficiently building software systems that satisfy the requirements of users and customers. Software engineering is applicable to small, medium, and large-scale systems. It encompasses all phases of the life cycle of a software system.
Software engineering employs engineering methods, processes, techniques, and measurement. It benefits from the use of tools for managing software development; analyzing and modeling software artifacts; assessing and controlling quality; and for ensuring a disciplined, controlled approach to software evolution and reuse. Software development, which often involves a team of developers, requires choosing the suitable tools, methods, and approaches that are most applicable for a given development environment.
The elements of software engineering are applicable to the development of software in any computing application domain where professionalism, quality, schedule, and cost are important in producing a software system.

Aims:
To provide an overall understanding of the fundamental concepts, and practical methods for engineering software. It will provide an in-depth understanding of software engineering approaches centred around and grounded into practical context. The course will equip students with foundational knowledge and practical skills to apply software engineering methods and techniques for engineering reasonably large software systems. It will also provide students with analytical means for assessing and evaluating factors that influence the selection and use of appropriate software engineering methods to appreciate their practical applications and their limitations.

Past due assignment submissions will be penalized with **20% deduction** for each late day!

Goal of the Class



Improve your ability to **create** higher-quality software that is *robust*, *extensible*, *scalable*, *maintainable*, and *secure* by **understanding** what are **common software engineering practices**

My **Real Goal** for Lectures

Provide **context** and **meaning** for the things you have or will later **learn on your own**


فلـــــة شـمعـة منـــــورة


Software Engineering
a very complicated task

How complicated is **Software Construction** for a Tesla car?

1

REQUIREMENTS

Understand the technical requirements of this project. Every piece of software—whether it's an app, website redesign, or new feature—needs to solve a customer problem.

ANALYSIS

2

This step is about analyzing the performance of the software at various stages and making notes on additional requirements. Analysis is very important to proceed further to the next step.

3

DESIGN

The objective in this step is to define internal structure and algorithms for components that meet client-oriented specifications.

CODING

4

After the best or the most appropriate design has been selected, implementation starts immediately. The team develops and implements software according to the pre-defined specifications.

5

TESTING

The testing stage assesses the software for errors and documents bugs if there are any.

DEPLOY

6

This means installing the software on user devices. At times, software needs post-installation configurations at user end. Software is tested for portability and adaptability & integration related issues are solved during implementation.