

11102

Fall 2025

**Introduction
to Computing**

What is Software Engineering?

Ibrahim Albluwi

Discussion

What is the difference between building a **house for your cat** and a **skyscraper**?



vs.



Discussion

What is the difference between building a **house for your cat** and a **skyscraper**?



Skyscraper

Requires a **large team** of builders, engineers, designers, etc.

Will be **used** by **thousands of people**.

Will live for many **years**.

High Stakes: If building fails, people might die or businesses might go bankrupt.

Must be completed within a **budget** and according to a **timeline**.

Cat House

Can be **built** by **you alone**.

Will be **used** by **your cat** only.

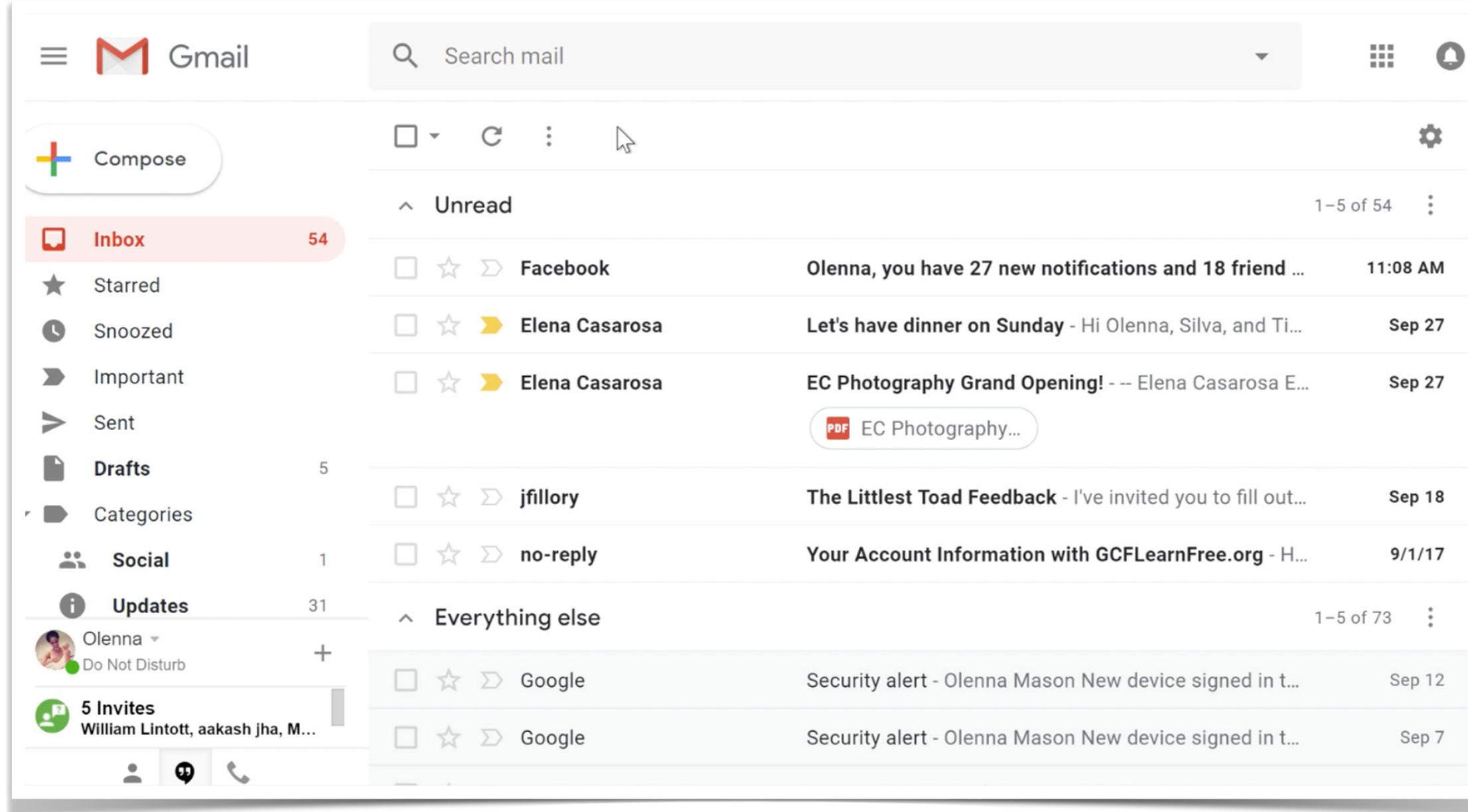
Might be thrown away next **week**.

Low Stakes: If building fails, you can build or buy another one!

Budget and **time** are not a big issue!

Discussion

What is the difference between solving a **homework exercise** and building **Gmail**?



Gmail

VS.

A screenshot of a digital workspace titled 'Basics'. It features four cards: 1) 'Say Hello' with a yellow double arrow icon and a person emoji, marked with a green checkmark. 2) 'Reversing Input' with a yellow double arrow icon and a person emoji, highlighted with a blue rounded rectangle. 3) 'Your Age Next Year' with a yellow double arrow icon and a person emoji, marked with a green checkmark. 4) 'Bug Hunt' with a yellow double arrow icon and a bug emoji.

Homework Exercise

Discussion

What is the difference between solving a **homework exercise** and building **Gmail**?



Gmail



Homework Exercise

Requires a **large team**.

Can be **solved** by **you alone**.

Will be **used** by **millions of people**.

Will be **used** by **you** only.

Will live for many **years**.

Thrown away after a few **months**.

High Stakes: People and businesses depend on it significantly.

Low Stakes: If you get it wrong, you lose 0.01 points.

Updates and bug fixes must be done under a **budget** and in a **timely** manner.

You build it for **free** and have a week to complete it in **< 15 hours**.

Requires **continuous updates**.

Why update it if there is a ✓

Summary

There is a big difference between:

- Building for **yourself** vs. Building **for others**.
- Building **alone** vs. Building in a **team**.
- Using **once** vs. Using **forever**.
- **Low** stakes vs. **High** stakes.
- No **budget** vs. Building with a **budget**.
- No **timeline** vs. Building with a **timeline**.
- **Static** requirements vs. **Changing** requirements.



Both skyscrapers and large **software** need
a structured approach (a.k.a **engineering!**)

Software Engineering

What is it about?

Building software with others, for others, with high stakes, on a budget, and on a timeline.

Formally:

"The application of a *systematic, disciplined, quantifiable* approach to the *development, operation*, and *maintenance* of software; that is, the application of engineering to software."¹

Practically:

"The job of *designing, developing, testing, and maintaining* software applications and systems."²

¹ IEEE Standard 610.12

² <https://github.com/resources/articles/what-is-software-engineering>



How Much Do SE's Get Paid?

'GLASSDOOR'

Accessed Nov. 2025

How much does a Software Engineer
make in Amman, Jordan?

Experience

0-1 year

Industries

Information Technology

Total pay range

JOD 456 - JOD 2K/mo

JOD 554/mo Median total pay

Experience

10-14 years

Industries

Information Technology

Total pay range

JOD 885 - JOD 3K/mo

JOD 2K/mo Median total pay

Anecdotal Data

PSUT **fresh** graduates working as software engineers typically earn between **450** JOD and **800** JOD. Top **fresh** graduates earn **1400+** JOD.

Examples of Software Engineering Tasks

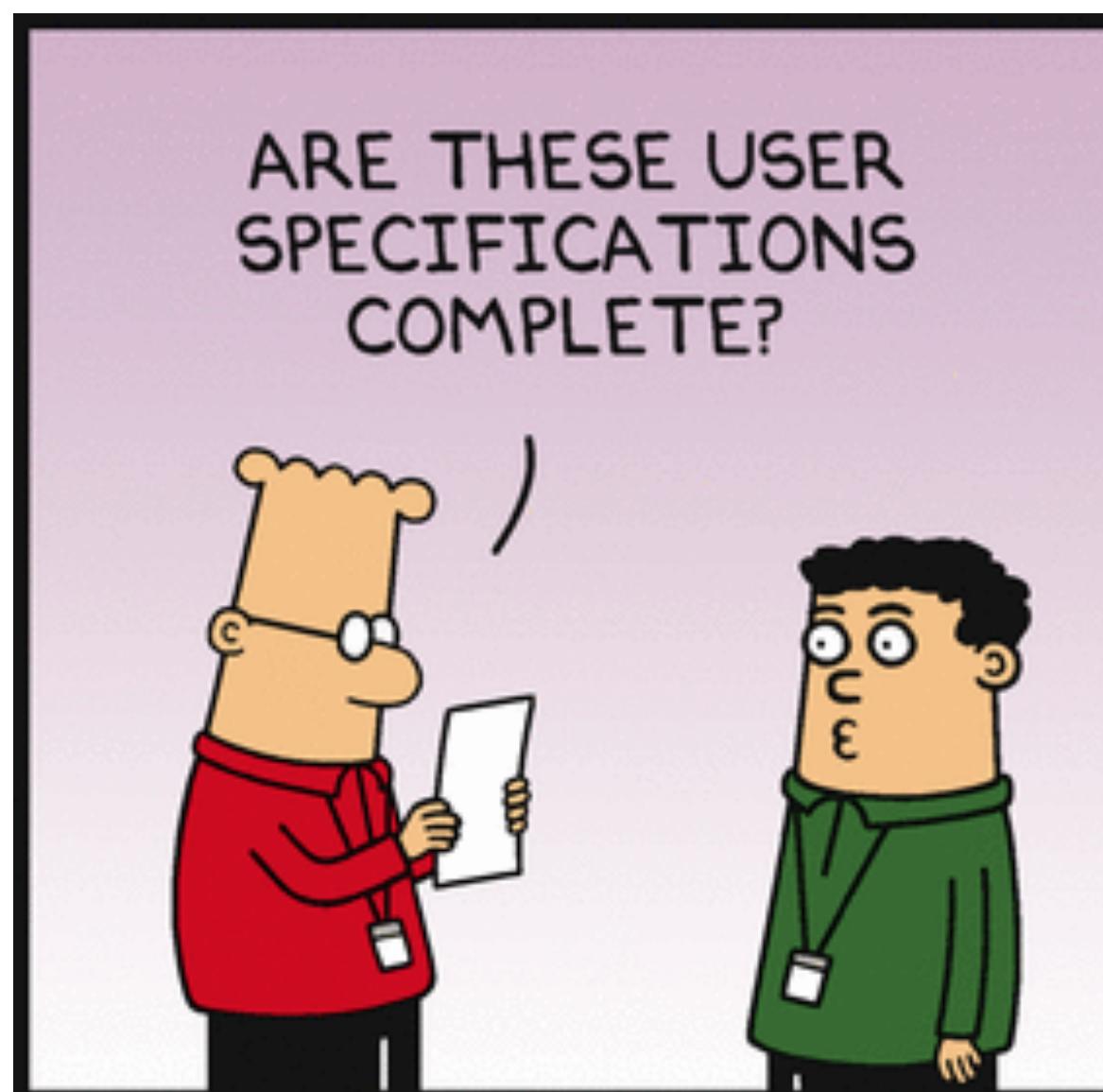
Requirements Gathering

Key questions:

What does my client want?

What do my app users need?

Example. Conduct meetings and sign contracts with a client.



Gathering requirements is hard!

Examples of SE Tasks

Requirements Gathering

Key questions:

What does my client want?

What do my app users need?

Example. Conduct meetings and sign contracts with a client.

Example. Study market needs, survey actual or potential users, collect feature requests, etc.

The screenshot shows a GitHub repository for 'microsoft / vscode'. The main navigation bar includes icons for issues, pull requests, and releases, along with search and filter options. Below the header, there are statistics: 'Open 3,534' and 'Closed 22,192', followed by buttons for 'Open all', 'Author', 'Labels', and three dots. The main content area displays five feature requests listed as cards:

- Allow to change the font size and font of the workbench**
#519 · hSDK123 opened on Nov 24, 2015 · On Deck
feature-request workbench-fonts
- Allow customization of mouse shortcuts**
#3130 · Tyriar opened on Feb 18, 2016 · Backlog
feature-request keybindings
- Feature Request: Show all errors and warnings in project for all JavaScript and TypeScript files, not just opened ones**
#13953 · kevinjreece opened on Oct 18, 2016 · Backlog
feature-request typescript
- VIM mode like sublime**
#114851 · miked508 opened on Jan 24, 2021 · Backlog
feature-request VIM
- Visual Studio Code for iPad**
#70764 · allessandrojs opened on Mar 19, 2019 · Backlog
feature-request ios-ipados

Each card includes a small profile picture, the number of comments (e.g., 574, 436, 188, 23, 283), and the number of likes (e.g., 3577, 1854, 1460, 1360, 1332).

Examples of SE Tasks

Requirements Gathering

What does my client want?
What do my app users need?

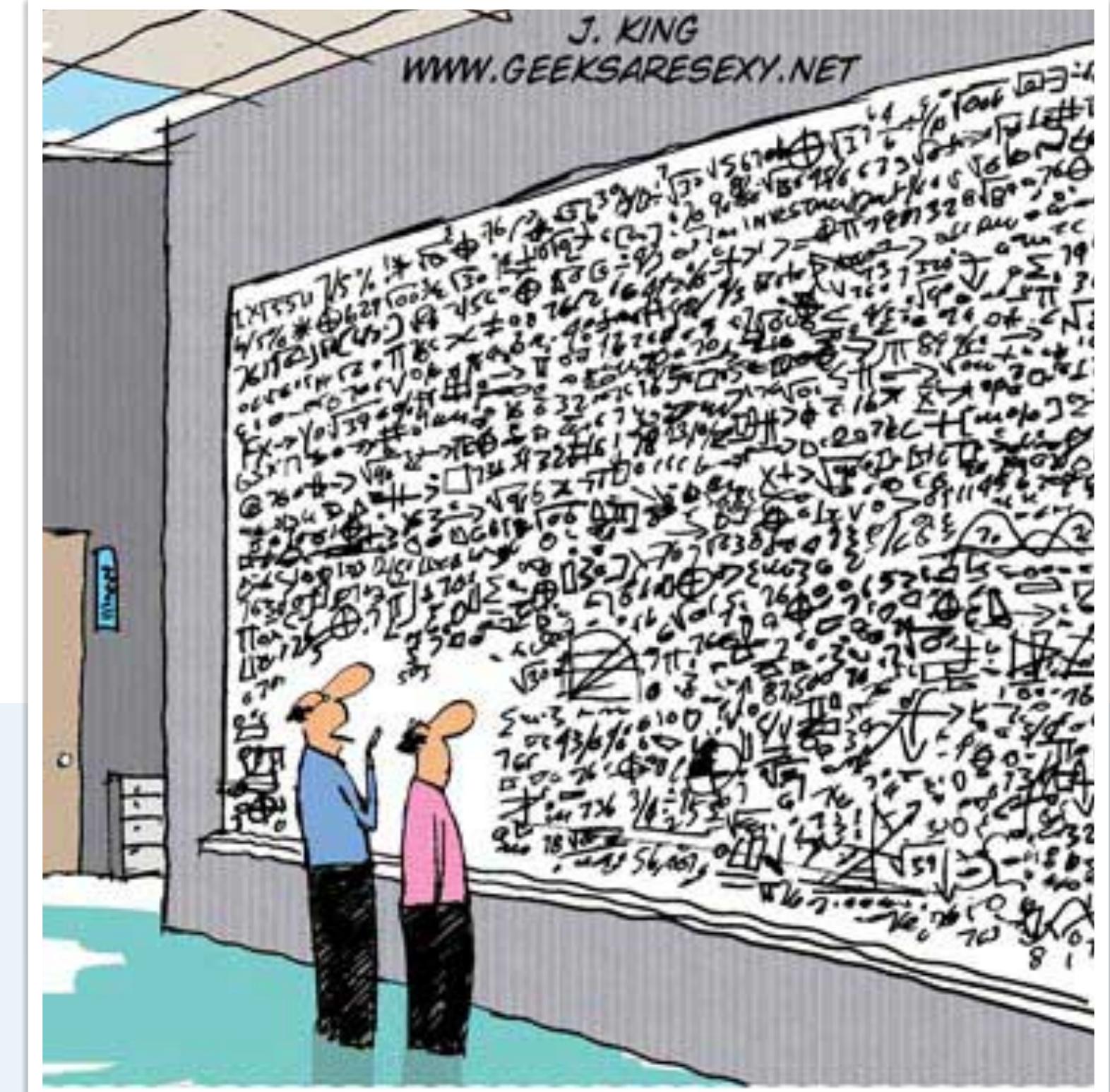
Design

Key questions:

How should the software system be developed?

What should be done to:

- Ensure the system is *secure*, *reliable*, and *efficient*.
- Ensure the system can *scale* to a large number of users in the future.
- Ensure that *updates* and *new features* can be made at low cost.
- Allow *code reuse* during development and after development.
- etc.



“...And that, in simple terms, is what’s wrong with your software design.”

Examples of SE Tasks

Requirements Gathering

What does my client want?
What do my app users need?

Design

How should the software system be developed?

Implementation

Actually writing the code!

Testing

How do we ensure the software is bug-free?

When the developer tests



When the quality team tests



When the project manager tests



When the customer tests



Examples of SE Tasks

Requirements Gathering

What does my client want?
What do my app users need?

Design

How should the software system be developed?

Implementation

Actually writing the code!

Testing

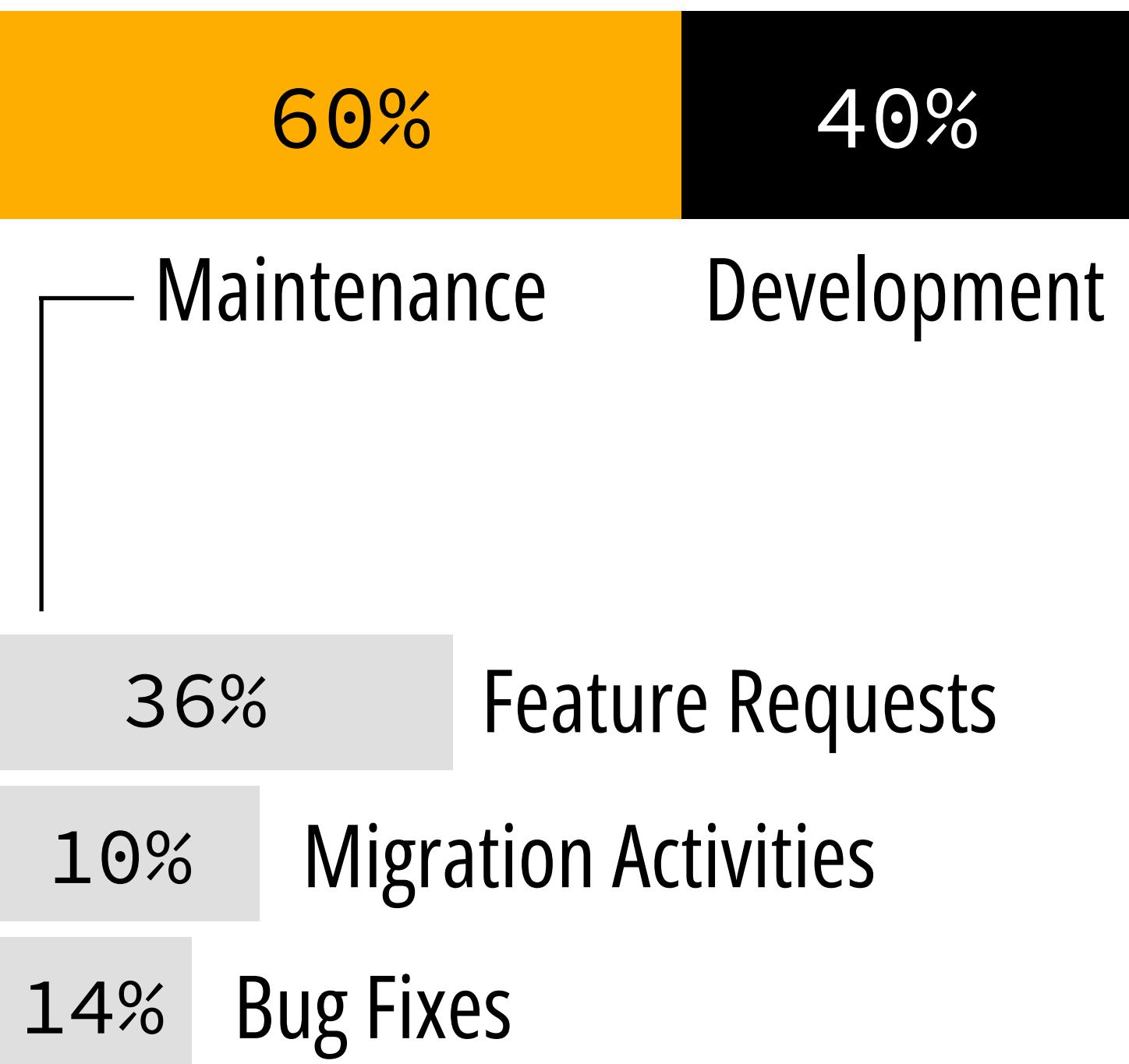
How do we ensure the software is bug-free?

Deployment, Maintenance, and Support

The larger portion of the work might be done in this phase!



Software Project Costs



A Sneak Peek at the **SE Study Plan** at PSUT

Common with Computer Science

3 11102

Introduction to Computer Science

3 11103

Structured Programming

3 11313

Algorithms Design and Analysis

3 11323

Database Systems

3 11206

Object Oriented Programming

3 11212

Data Structures and Introduction to Algorithms

3 12343

Visual Programming

3 12243

Webpage Design and Internet programming

3 22342

Computer Organization and Assembly Language

3 11335

Operating Systems

3 11435

Data Communications & Computer Networks

3 13211

Introduction to Software Engineering

Required for SE Majors

3 13325

Software Requirements Analysis

3 13324

System Analysis and Design

3 13212

Software Construction

3 13326

S.W. Eng. Approaches to Human Computer Interaction

3 13432

Software Project Management

3 13327

Software Design and Architecture

3 13428

Software Quality Assurance and Testing

A deep dive into each of the tasks performed by software engineers!

Shallow Dive # 1

Programming in a large team

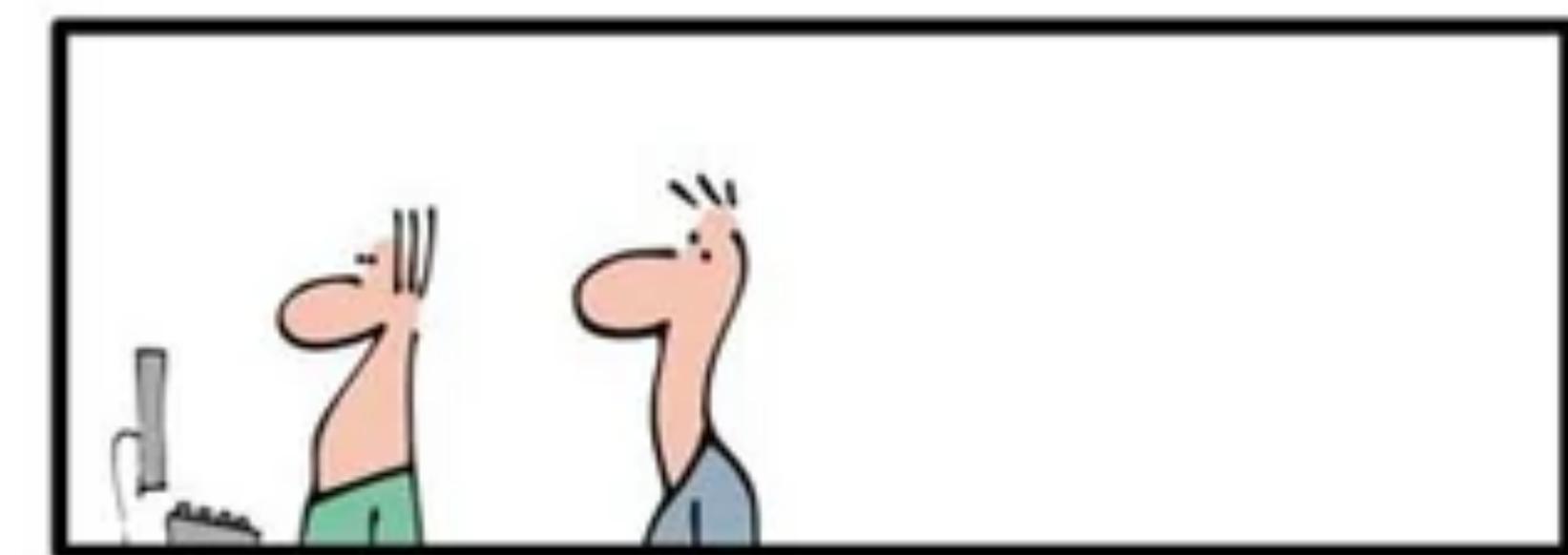
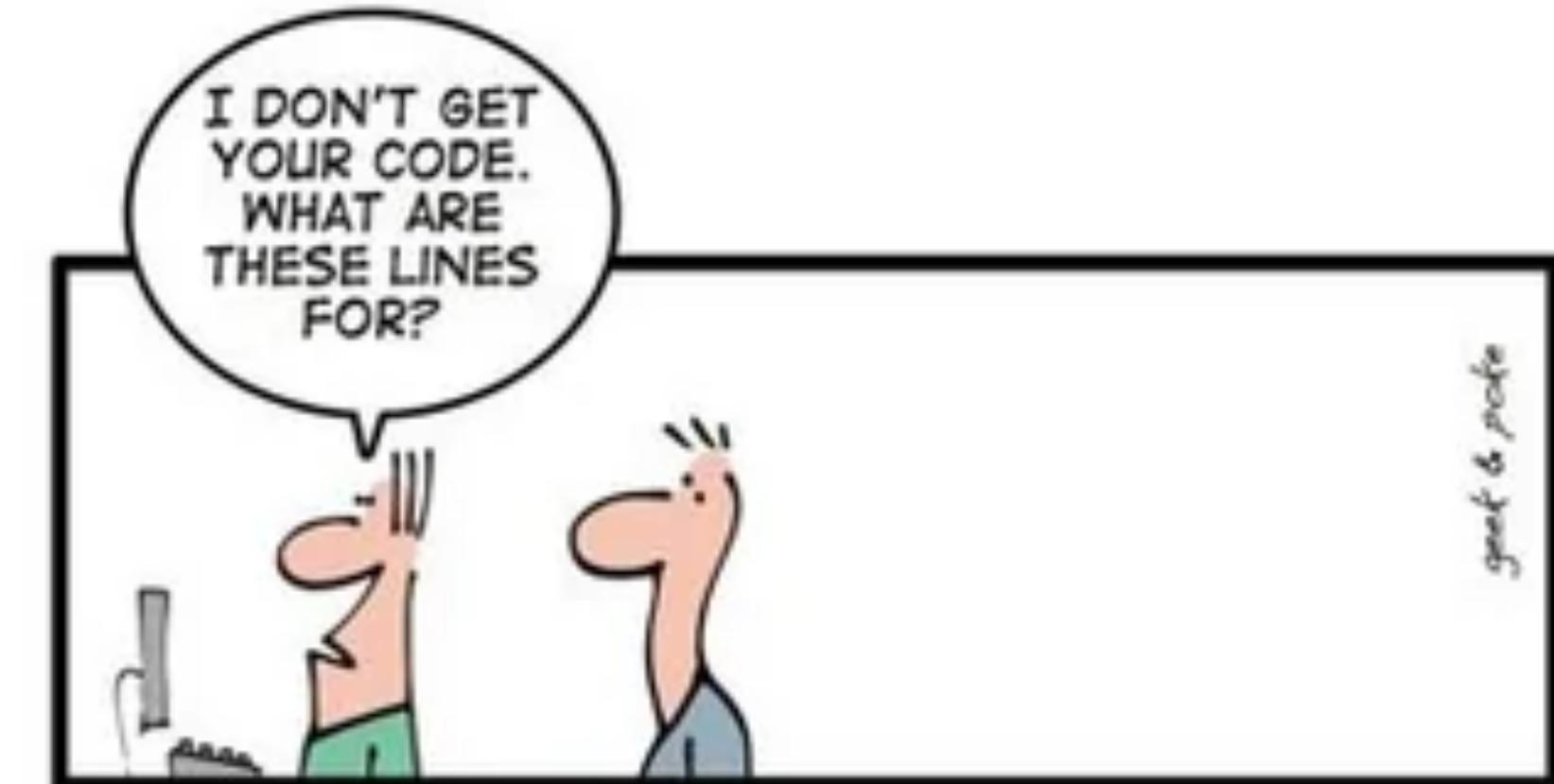
Coding Style

When working in a large team, your code will be read and used by others.

Most of the code you will be working with will be code written by others.

Teams typically agree on **coding style guidelines** for everyone to follow.

This makes reading code much easier!



THE ART OF PROGRAMMING - PART 2: KISS

Coding Style

Example 1. PEP 8 Style Guide (a community standard)

<https://peps.python.org/pep-0008/>

Example 2. Google Coding Style Guidelines (used at Google and other places)

<https://google.github.io/styleguide/pyguide.html>

Coding Style Example

PEP 8 Style Checker.

```
ialbluwi Documents $ pycodestyle test.py
test.py:2:17: E201 whitespace after '['
test.py:2:20: E203 whitespace before ',', '
test.py:2:25: E203 whitespace before ',', '
test.py:2:30: E202 whitespace before ']'
test.py:3:14: E225 missing whitespace around operator
test.py:4:21: E225 missing whitespace around operator
```

test.py

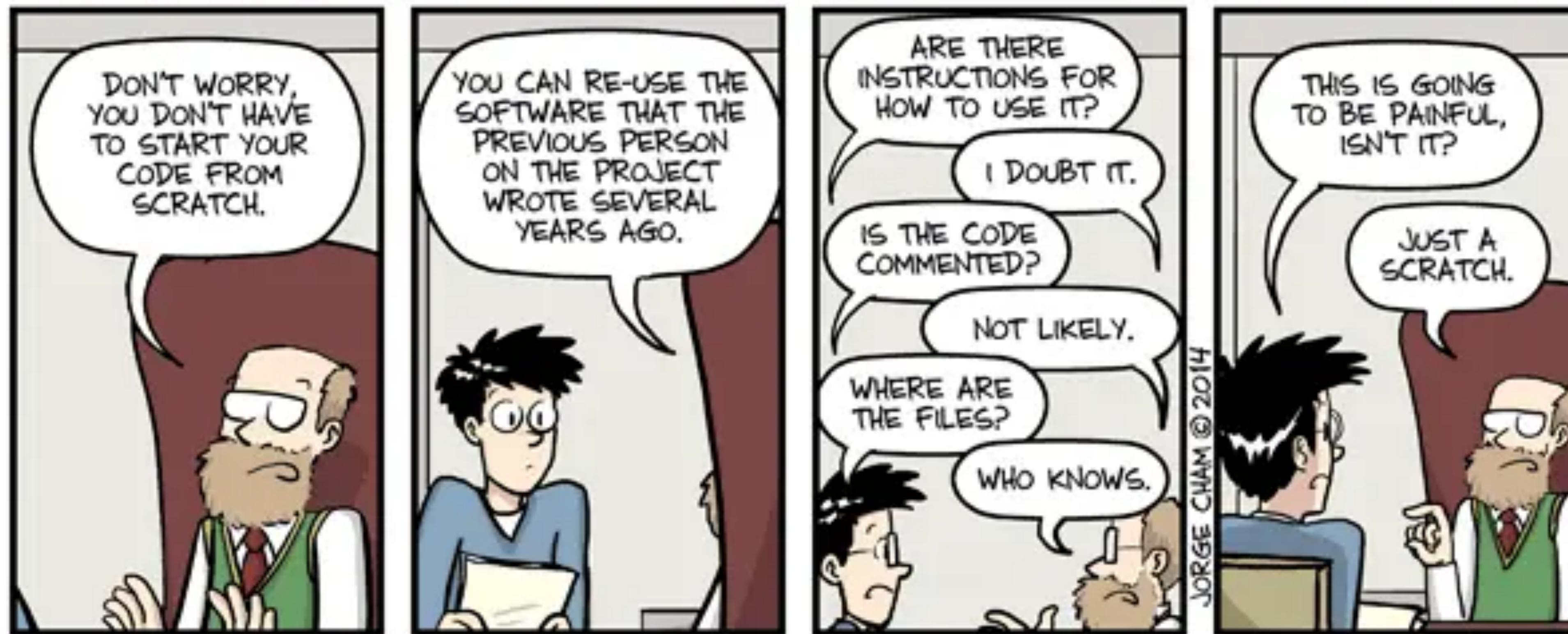
```
1 def compute(e1, e2, e3):
2     MyGrades = [ e1 , e2 , e3 ]
3     failgrade=50
4     if sum(MyGrades)>50:
5         print('PASS')
```

To install **pycodestyle** use the terminal:

- On Windows: pip install pycodestyle
- On Mac/Unix: pip3 install pycodestyle

Programming in a large team

Documentation



Documentation

Software engineers write lots of documentation! For example:

Code Documentation. Other team members need to understand the code.
Even the person who wrote the code can forget as time passes!

Documentation

Software engineers write lots of documentation! For example:

Code Documentation. Other team members need to understand the code.
Even the person who wrote the code can forget as time passes!

Design Documentation. Team members need to communicate and justify designs and decisions to team leads (and the other way around, too!)

Documentation

Software engineers write lots of documentation! For example:

Code Documentation. Other team members need to understand the code. Even the person who wrote the code can forget as time passes!

Design Documentation. Team members need to communicate and justify designs and decisions to team leads (and the other way around, too!)

Onboarding Guides. New team members need guidance on how to navigate and deal with the code base!

Documentation

Software engineers write lots of documentation! For example:

Code Documentation. Other team members need to understand the code.
Even the person who wrote the code can forget as time passes!

Design Documentation. Team members need to communicate and justify designs and decisions to team leads (and the other way around, too!)

Onboarding Guides. New team members need guidance on how to navigate and deal with the code base!

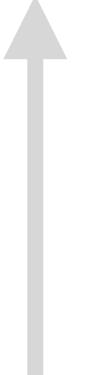
User Documentation

Users of the system need to know how to use it!

Code Documentation Example

Python provides a tool for automatically extracting documentation from code.

```
1 def is_palindrome(word):
2     """Check if a word can be read the same forwards and backwards."""
3     return word == word[::-1]
4
5 def is_sorted(lst):
6     """Check if a list is sorted in ascending order."""
7     for i in range(len(lst) - 1):
8         if lst[i] > lst[i + 1]:
9             return False
10    return True
```



A Docstring
(marked by triple quotes `""" ... """`)

Code Documentation Example

Python provides a tool for automatically extracting documentation from code!

```
ialbluwi Documents $ pydoc test
```

```
Help on module test:
```

```
NAME
```

```
test
```

```
FUNCTIONS
```

```
is_palindrome(word)
```

Check if a word can be read the same forwards and backwards.

```
is_sorted(lst)
```

Check if a list is sorted in ascending order.

automatically reads the docstrings
and generates the following:

Code Documentation Example

Python provides a tool for automatically extracting documentation from code!

Try:

```
ialbluwi Documents $ pydoc math
```

Shows the documentation for math

```
ialbluwi Documents $ pydoc random
```

Shows the documentation for random

```
ialbluwi Documents $ pydoc random.randint
```

Shows the documentation for randint

```
ialbluwi Documents $ pydoc -w random
```

Creates an HTML file containing
the documentation of random

A library that provides better HTML output is: pdoc
(requires installation: pip install pdoc then pdoc filename.py)

Version Control

Teams need to:

- Keep track of *who changed what*.
- Manage *synchronous edits* to the same file.
- Allow *going back* to older versions.
- Allow *branching out* to try features without affecting others.

Example. Git and GitHub

Stay tuned for **Assignment 3!**



Shallow Dive # 2 **Testing**

What can go wrong if software engineers don't test well?

What can go wrong ?



Patriot Missile Defense: Software Problem Led to System Failure at Dhahran, Saudi Arabia

IMTEC-92-26

Published: Feb 04, 1992. Publicly Released: Feb 27, 1992.

MIM-104 Patriot

文 A 49 languages ▾

Failure at Dhahran

On February 25, 1991, an Iraqi Al Hussein Scud missile hit the barracks in Dhahran, Saudi Arabia, killing 28 soldiers from the U.S. Army's 14th Quartermaster Detachment.^[90]

A government investigation revealed that the failed intercept at Dhahran had been caused by a software error in the system's handling of timestamps.^{[91][92]} The Patriot missile battery at Dhahran had been in operation for 100 hours by



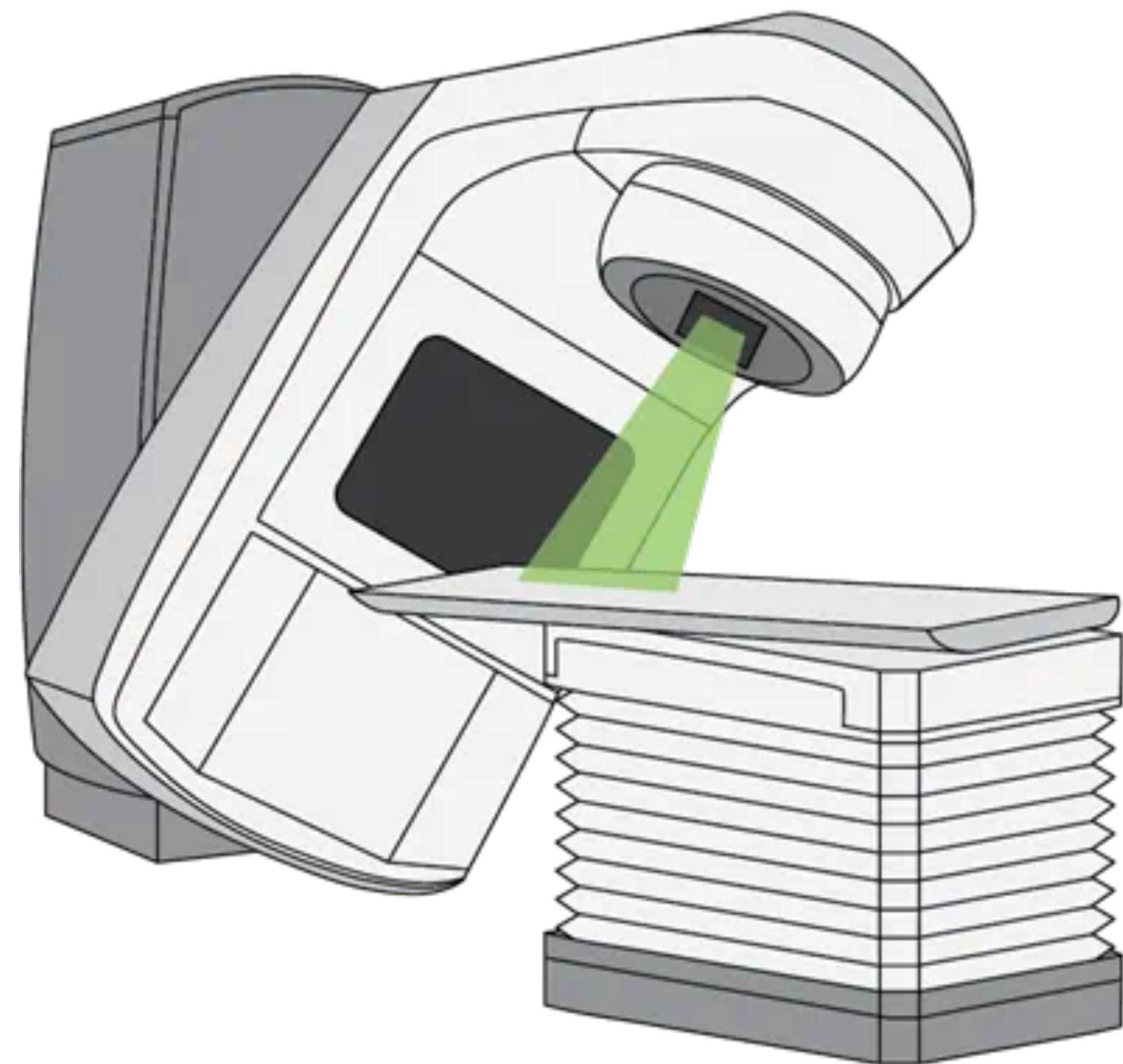
≡ Ariane flight V88

文 A 5 languages ▾

Ariane flight V88^[1] was the failed [maiden flight](#) of the [Arianespace Ariane 5](#) rocket, vehicle no. 501, on 4 June 1996. It carried the [Cluster](#) spacecraft, a constellation of four [European Space Agency](#) research satellites.

The launch ended in failure due to multiple errors in the software design: [dead code](#), intended only for [Ariane 4](#), with inadequate protection against [integer overflow](#) led to an [exception handled](#) inappropriately, halting the whole otherwise unaffected [inertial navigation system](#). This caused the rocket to veer off its flight path 37 seconds after launch, beginning to disintegrate under high aerodynamic forces, and finally self-destructing via its automated [flight termination system](#). The failure has become known as one of the most infamous and expensive [software bugs](#) in history.^[2] The failure resulted in a loss of more than US\$370 million.^[3]

≡ Multidata Systems International ⋮A Add languages



A software product of the company was involved in an accidental overexposure of patients in [Panama](#) in 2001 when the treatment planning software RTP/2 (vers. 2.11, 1991) reportedly contributed to 28 patients receiving excessive amounts of radiation at the [Instituto Oncologico Nacional](#) in [Panama City](#). At least eight patients died, while another 20 received overdoses likely to cause significant health problems. The physicians, who were legally required to double-check the computer's calculations by hand, were indicted for murder.^[2]

A panel of experts designated by the [International Atomic Energy Agency](#) delivered a comprehensive report in August 2001, finding that the software permitted incorrect forms of data entry which in turn had led to miscalculation of treatment times.^[3] Multidata began a recall

≡ Northeast blackout of 2003

⋮A Add languages ▾

The **Northeast blackout of 2003** was a widespread [power outage](#) throughout parts of the [Northeastern](#) and [Midwestern United States](#), and most parts of the Canadian province of [Ontario](#) on Thursday, August 14, 2003, beginning just after 4:10 p.m. [EDT](#).^[1]

The blackout was due to a [software bug](#) in the alarm system at the control room of [FirstEnergy](#), which rendered operators unaware of the need to redistribute load after overloaded transmission lines dropped in voltage. What should have been a manageable local blackout cascaded into the collapse of much of the Northeast regional electricity distribution system.





⋮ British Post Office scandal

The **British Post Office scandal**, also called the **Horizon IT scandal**, involved the **Post Office** pursuing thousands of innocent **subpostmasters** for apparent financial shortfalls caused by faults in **Horizon**, an accounting software system developed by **Fujitsu**.

Between 1999 and 2015, more than 900 subpostmasters were wrongfully convicted of theft, fraud and false accounting based on faulty Horizon data, with about 700 of these prosecutions carried out by the Post Office.

In 2017, 555 subpostmasters led by Bates brought a **group action** against the Post Office in the **High Court**. In 2019, the judge ruled that the subpostmasters' contracts were unfair, and that Horizon "contained bugs, errors and defects". The case was settled for £58 million, leaving the claimants with £12 million after legal costs.

What can go wrong ?



New iOS Bug Crashing iPhones Simply by Receiving a Text Message [Includes Fix]

Tuesday May 26, 2015 08:34 EAT by [Juli Clover](#)

A new bug has been discovered in the Messages app, allowing a string of characters sent to a person via iMessage or SMS to crash an iPhone and cause the Messages app to crash after being opened. The bug, which requires a [specific string of symbols and Arabic characters](#) to be sent, was first noticed on reddit earlier this afternoon and has been spreading around the Internet since then.

Sending the string of characters to an iPhone results in an immediate respring, causing an iPhone to crash and quickly reboot. From there, if the Messages app was opened at a list view, the Messages app crashes automatically when you try to open it. If it was opened to the conversation where you received the message, the app will open, but attempting to go to another conversation causes Messages to crash.

Shallow Dive # 2 **Testing**

How should we test?

Déjà vu

We need to write code that checks if our code works as intended or not!

Task

Given the *intended* behavior of a module or a function, write test cases (*without looking at the code*) that ensure the code works as intended!

Challenge

Design good test cases!

1

Not yet

TESTCASES

3 / 8 passed, 3 points

Single Row

✓ 1 point

Single Column

✓ 1 point

Main Diagonal

✓ 1 point

> Anti Diagonal

✗ 1 point

> Full Board

✗ 1 point

> Did Player Win?

✗ 1 point

> Ties

✗ 1 point

> Full Game

✗ 1 point

Exercise

Which of the following test cases are enough for testing the following function?

```
def maximum(a, b, c):
    '''Returns the maximum
    between three numbers'''
    pass
```

- A. (1, 2, 3)
- B. (1, 2, 3) (3, 2, 1) (1, 3, 2)
- C. (1, 2, 3) (3, 2, 1) (1, 3, 2)
- D. (1, 2, 3) (1, 3, 2) (2, 1, 3) (2, 3, 1) (3, 1, 2) (3, 2, 1)
- E. None of the above.

Exercise

Which of the following test cases are enough for testing the following function?

```
def maximum(a, b, c):
    '''Returns the maximum
    between three numbers'''
    pass
```

- A. (1, 2, 3)
- B. (1, 2, 3) (3, 2, 1) (1, 3, 2)
- C. (1, 2, 3) (3, 2, 1) (1, 3, 2)
- D. (1, 2, 3) (1, 3, 2) (2, 1, 3) (2, 3, 1) (3, 1, 2) (3, 2, 1)
- E. None of the above.

Exercise

The following code passes all the previous test cases although it is incorrect when all the numbers are **negative**.

```
def maximum(a, b, c):
    max = 0

    if a > max:
        max = a
    if b > max:
        max = b
    if c > max:
        max = c

    return max
```

Exercise

Which of the following test cases are enough for testing the following function?

```
def maximum(a, b, c):
    '''Returns the maximum
    between three numbers'''
    pass
```

- A. (1, 2, 3) (3, 2, 1) (1, 3, 2) (-1, -2, -3) (-3, -2, -1) (-1, -3, -2)
- B. All 3-permutations of -3, -2, -1, 1, 2, 3
- C. None of the above.

Exercise

Which of the following test cases are enough for testing the following function?

```
def maximum(a, b, c):
    '''Returns the maximum
    between three numbers'''
    pass
```

- A. (1, 2, 3) (3, 2, 1) (1, 3, 2) (-1, -2, -3) (-3, -2, -1) (-1, -3, -2)
- B. All 3-permutations of -3, -2, -1, 1, 2, 3
- C. None of the above.

Exercise

The following code passes all the previous test cases although it is incorrect when some numbers are **equal**.

```
def maximum(a, b, c):
    max = 0

    if a > c and a > b:
        max = a
    if b > c and b > a:
        max = b
    if c > a and c > b:
        max = c

    return max
```

Exercise

Is it enough to test all possible **3-tuples** (permutations with repetition) of {-3, -2, -1, 1, 2, 3}?

Exercise

Is it enough to test all possible **3-tuples** (permutations with repetition) of {-3, -2, -1, 1, 2, 3}?

Not necessarily!

The following code would pass all test cases but fail with **very small numbers!**

```
def maximum(a, b, c):
    max = -9999999
    if a > max:
        max = a
    if b > max:
        max = b
    if c > max:
        max = c
    return max
```

Exercise

What about testing with millions of permutations of small and large positive and negative numbers?

Exercise

What about testing with millions of permutations of small and large positive and negative numbers?

Definitely an **overkill!**

Exercise

What about testing with millions of permutations of small and large positive and negative numbers?

Definitely an **overkill!**

Think of **Equivalence Groups**:

- All *positive*.
 - All *negative*.
 - Mixed positive and negative.
 - *Permutations*: (min, mid, max), (min, max, mid),
(mid, min, max), (mid, max, min),
(max, mid, min), (max, min, mid)
 - With duplicates.
 - With *very large* and *very small* numbers.
 - Should you include *floating-point* numbers?
- (1, 2, 3) Covers positive
and (min, mid, max)
Testing with (5, 7, 8) might be redundant!

Equivalence Classes

Thinking about equivalence classes allows for avoiding redundancy in testing.

- All *positive*.
- All *negative*.
- Mixed positive and negative.
- *Permutations*:
(min, mid, max), (min, max, mid),
(mid, min, max), (mid, max, min),
(max, mid, min), (max, min, mid)
- With duplicates.
- With *very large* and *very small* numbers.
- Should you include *floating-point* numbers?

Example. Using **(1, 2, 3)** as a test case covers *all positive* and *(min, mid, max)*

Testing with **(5, 7, 8)** might be redundant!

It does not fall into a different equivalence class.

Best Practices

- Include tests across the **range of possible** values.
- Include tests for **boundary** values.
- **Understand the problem** to come up with tests (e.g., when ordering elements, relative order and equal values matter).
- Target all **equivalence classes**.
- Add random test cases to catch issues you might not have thought about!

Image Credits

<https://www.linkedin.com/pulse/sap-functional-specifications-vs-requirements-barry-neaves-kj5ce/>

<https://cancer.ca/en/treatments/treatment-types/radiation-therapy/external-radiation-therapy>

https://www.esa.int/ESA_Multimedia/Images/2023/06/Ariane_5_V88

https://www.flaticon.com/free-icon/skyscraper_3562457

<https://github.com/Crissov/unicode-proposals/issues/89>

<https://www.x3blackfriday.com/?path=page/ggitem&ggpid=1487145>

<https://www.ammanjo.co/print.php?id=114713>