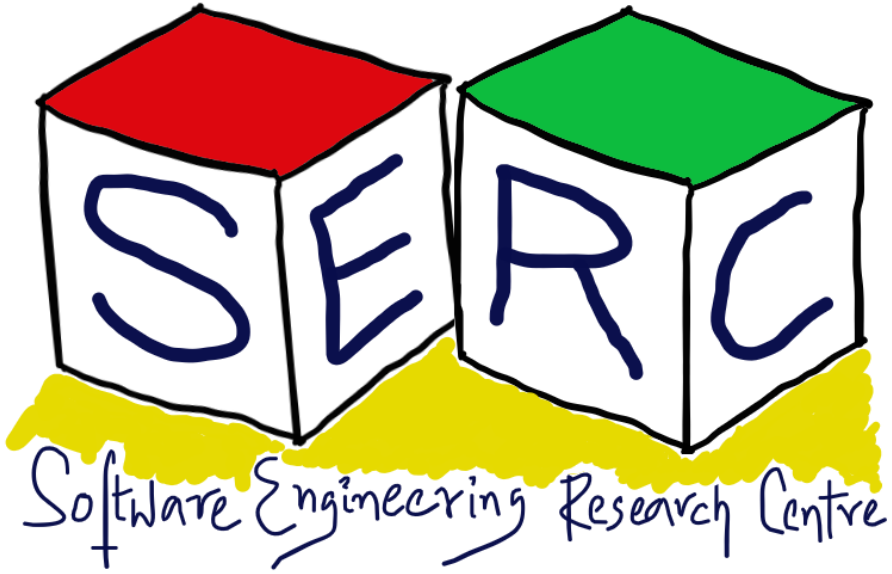
Introduction and Course Overview

CS6.401 Software Engineering



Dr. Karthik Vaidhyanthan

karthik.Vaidhyanathan@iiit.ac.in

https://karthikvaidhyanathan.com

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Meet the Team

Karthik

Aditya Hari

Annapoorani

Pranavasri

Shambhavi

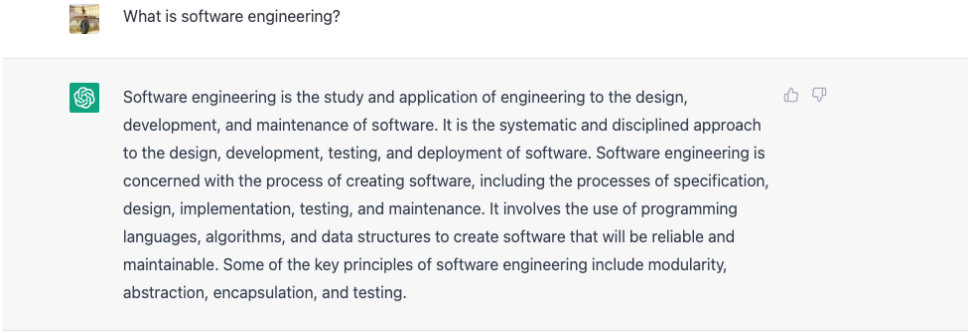
Shlok

Sidharth

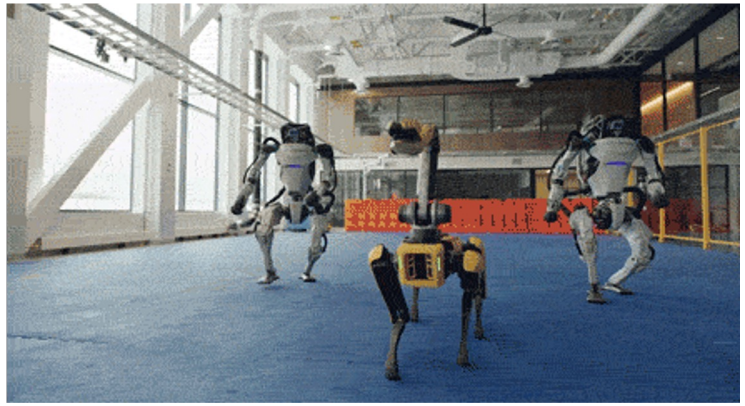
World of Smart Apps and Complex Software

Everything is increasingly powered by software

Smart bottles, watches



Chat GPT 3



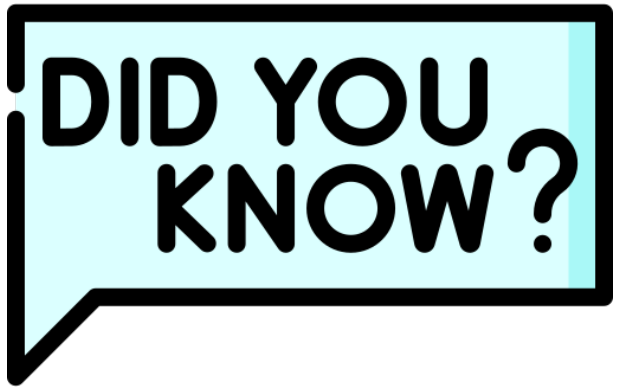
Boston Dynamics

Tesla Car in Autopilot mode

Source: Google images, tesla, openai

Some Interesting yet Serious Facts

1. About **70%** of the cost goes for software maintainence
2. Modern car runs **100 million** lines of code - Only going to increase!
3. By 2025, we will have **27 billion** connected IoT devices
4. **80%** of the ML systems considered failures in production
5. ICT is expected to consume 30% of worlds electricity and contributes to 10% of global carbon footprint
6. ……

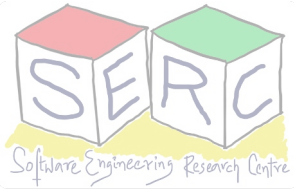


Sources: Gartner, Cisco

What is the big deal?

Let’s draw some parallels

When do you say something is well engineered?



Is Software that Critical?

Few of the many software issues

TECH \ AMAZON \

Amazon Web Services says overwhelmed network devices triggered outage

Amazon says it plans to improve its response to outages

By Emma Roth | Dec 11, 2021, 3:34pm EST

Meta services are back ONLINE! Facebook, Instagram and WhatsApp were down for more than one hour worldwide that impacted thousands of users

- **Facebook, Instagram and WhatsApp crashed around 11:30am ET on Friday**
- **The outage hit users worldwide who cited issues with websites and apps**
- **However, users also said they are unable to post on Facebook and Instagram**

By STACY LIBERATORE FOR DAILYMAIL.COM
PUBLISHED: 17:34 GMT, 19 November 2021 | **UPDATED:** 18:28 GMT, 19 November 2021

Huge economical impact!!

Source: dailymail, google news

Few of the many software issues

Vox recode

Tesla needs to fix its deadly Autopilot problem

Tesla is facing heat from federal officials following an investigation into a fatal crash involving its Autopilot.

By Rebecca Heilweil | Feb 26, 2020, 1:50pm EST

Catastrophic software errors doomed Boeing's airplanes and nearly destroyed its NASA spaceship. Experts blame the leadership's 'lack of engineering culture.'

MORGAN MCFALL-JOHNSEN | FEB 29, 2020, 18:41 IST

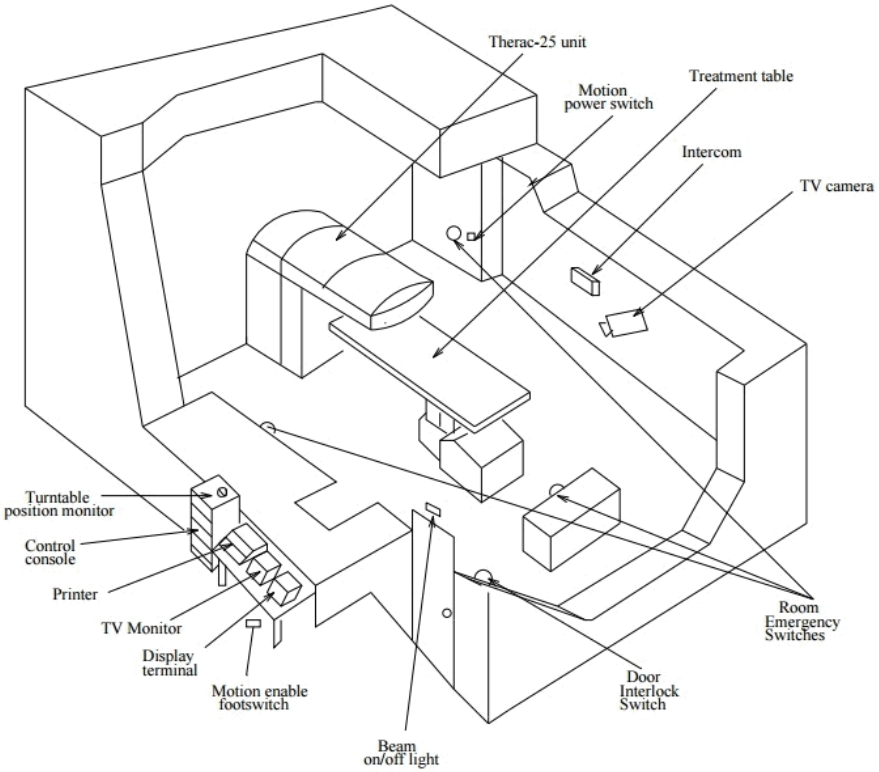


Figure 5: A typical Therac-25 facility after the final CAP.

Can be fatal!!

Source: Googe images, semantic scholar

Why Software Engineering?

Why Study Software Engineering?

- Acquire skills to understand and build high quality software systems
  - Systems could be large and complex
  - It could consist of millions of lines of code
  - How to comprehend an existing system?
  - How to break a program into smaller and manageable parts?
  - Development of large and complex software requires teamwork
  - Achieve sufficient quality (eg: maintainability, performance, reliability etc.)

- Learn techniques: specification, designing, development, testing, project management, etc.

Source: Googe images

Software Engineer is not just another title

- Software engineer is considered as just another role sometimes
- Many countries require software engineers to go through a certification or licensing process
- Going forward software engineers will be dealing with more and more critical systems

Licensing is already in place in some countries for some specific SE roles!!

Licensing Professional Software Engineers: Seize the Opportunity

By Phillip A. Laplante
Communications of the ACM, July 2014, Vol. 57 No. 7, Pages 38-40
10.1145/2618111
Comments (1)

VIEW AS: SHARE:

In his July 2013 *Communications* column, ACM President Vint Cerf revisited the controversy of licensing professional software engineers in the U.S. This issue has been one that divides the profession since reasonable cases can be made both pro and con. Officially, ACM has opposed and IEEE has supported such licensure. Even within both organizations, though, there is substantial support and opposition.[1,2,3,5] I think President Cerf was right in suggesting that, because of the dramatic growth in interacting software in both devices and in applications that significantly impact peoples' lives, and more importantly, because licensing is happening, we have reached a tipping point.

Source: Googe images, c-ACM

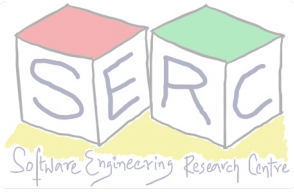
Don’t worry we have SE course!!

What probably you have done so far?

- IIIT Undergrad: Introduction to software systems (ISS), low level OO principles and some high level design in Design and Analysis of Software Systems (DASS)

- Grads at IIIT: Problem solving course and Software Systems Development

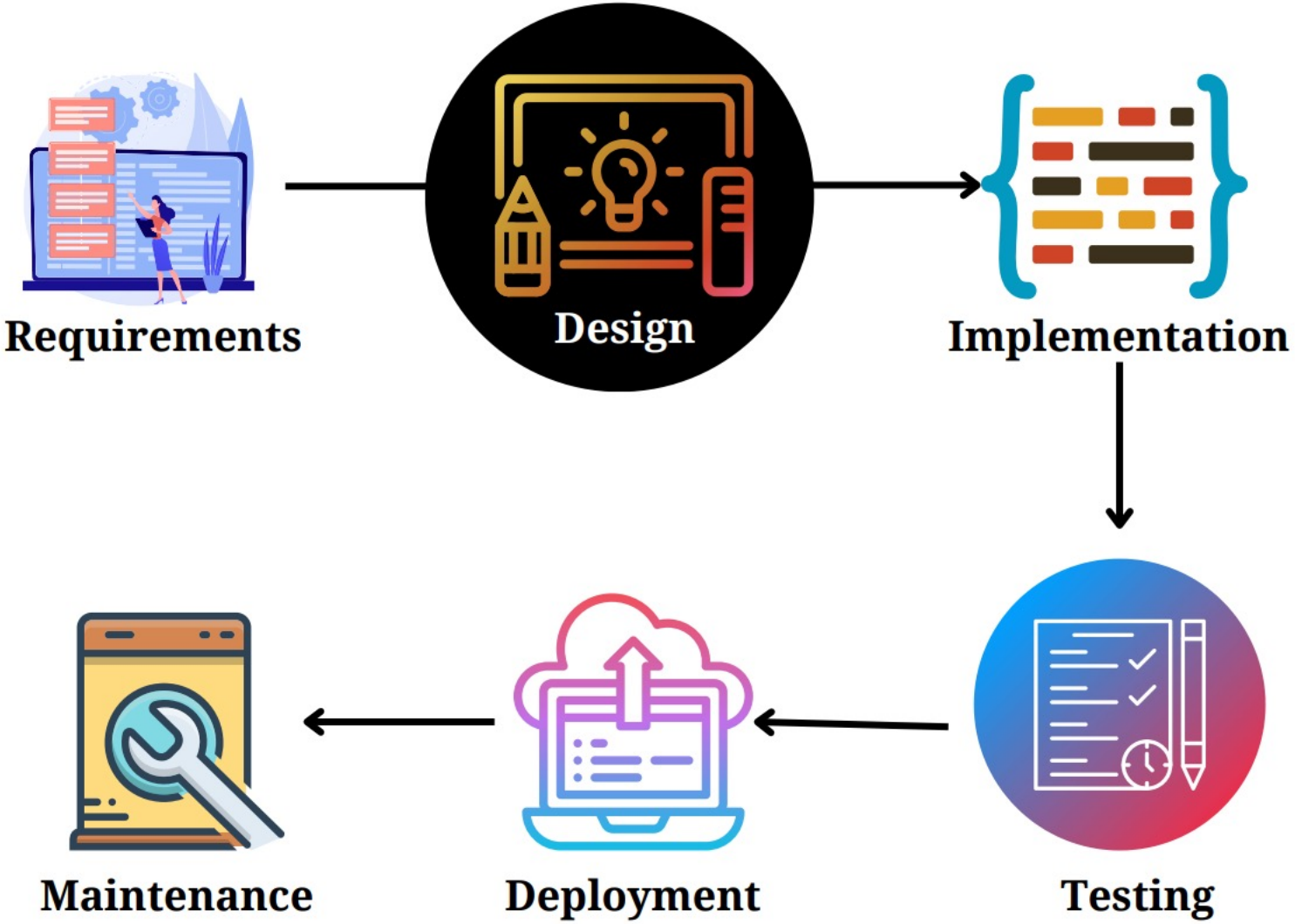
- Others: Some software development experience or an undergrad course in software/systems engineering or similar courses



Minimum Prerequisites

1. Basic idea on: Abstraction, Modularization/Decomposition, Coupling, Cohesion, etc.
2. Some languages (eg:, python, JS, etc.)
3. Basic OO principles and implementations: Encapsulation, inheritance, Polymorphism
4. At least 1 OOP language, & 1 RDBMS.
5. Idea about various phases in SDLC
6. Minimal knowledge of practices: Version control, bug tracking, task management, etc.

Software Development Lifecycle



This course is about
Software Design!

What kind of design? - Lets' take an Example

Assume that there are two strings, string1 = "ur34" and string2 = "asdasd". Write a simple program that takes two strings, inputString1 and inputString2 as inputs and compares them with string1 and string 2 respectively. If the strings match, then output should be true, else False

Sample input

ur34 asdasd

Sample Output

True

This is a simple string match program!!!

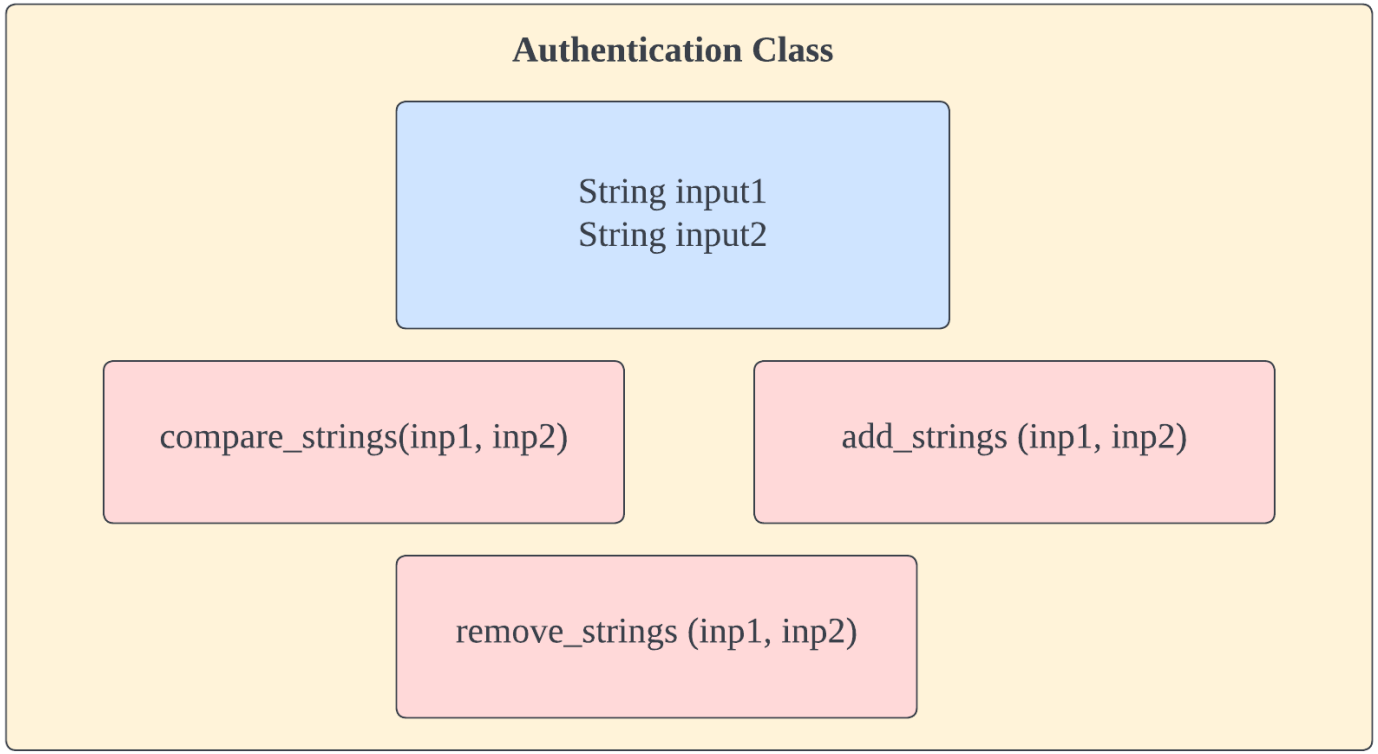
```
compare_strings(inp1, inp2)
```

Not at this level!!

What kind of design? - Lets' go little more higher

Modify the above into an authentication module. You should be able to add two new strings (username, password) and the program should be able to find a match given two strings.

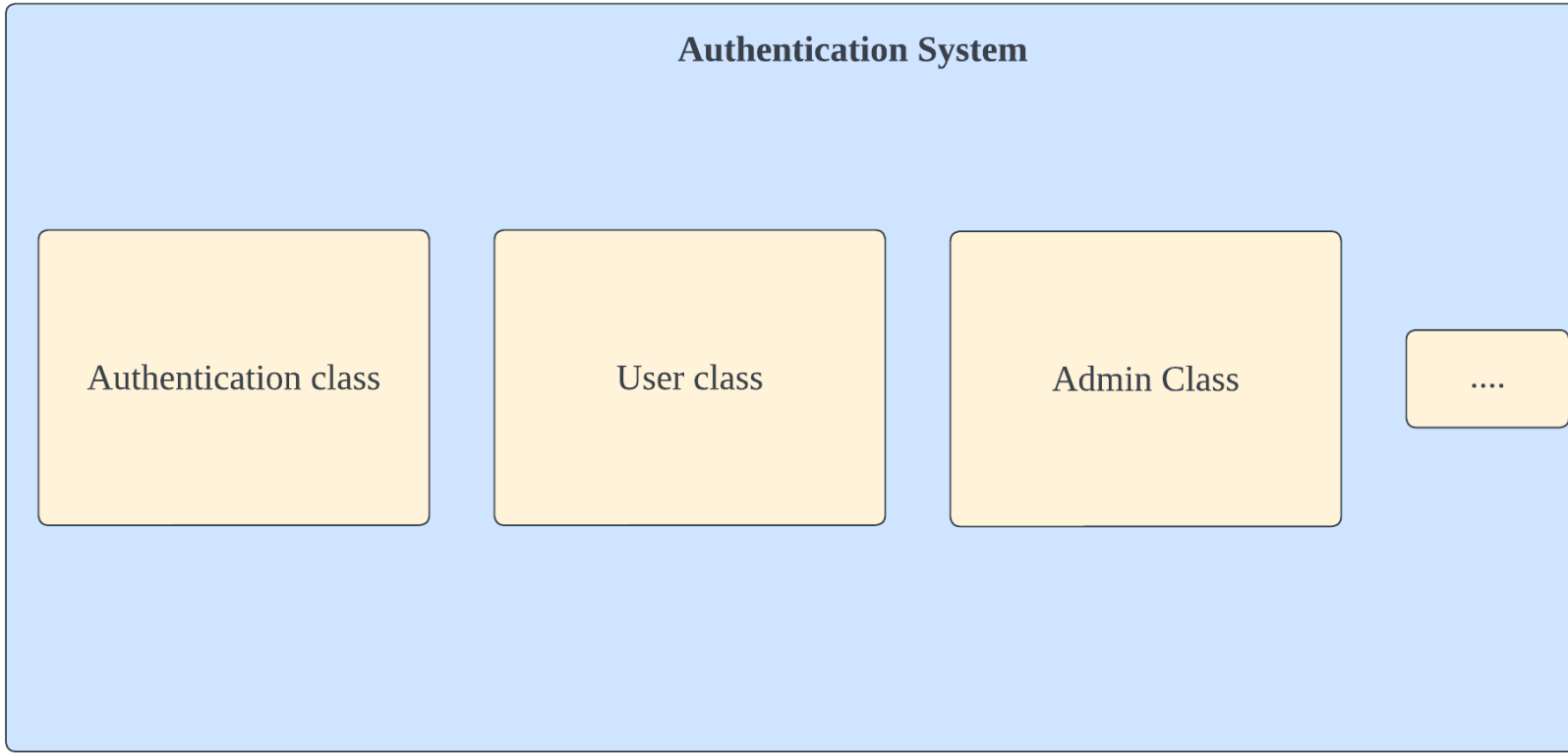
Little at this level!!



What kind of design? - Lets' go little more higher

Modify the above into an authentication system. Any user should be able to register and login. The system should support multiple types of user

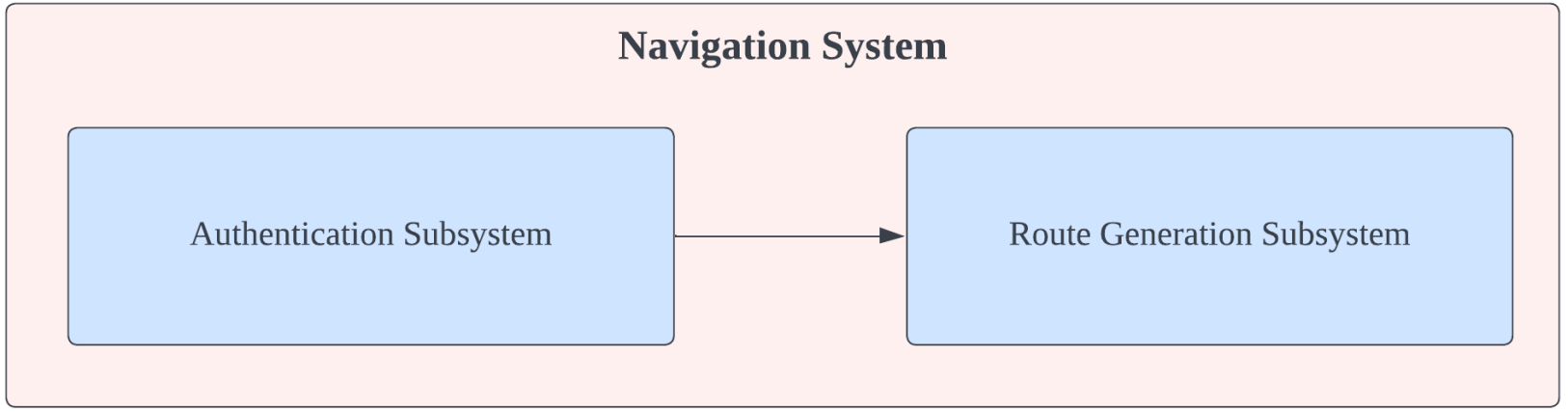
Yes, this level!!



What kind of design? - Lets' go even further higher

Assume that there is another sub system that allows users to find the shortest path between two given cities. Now, integrate this with the authentication system.

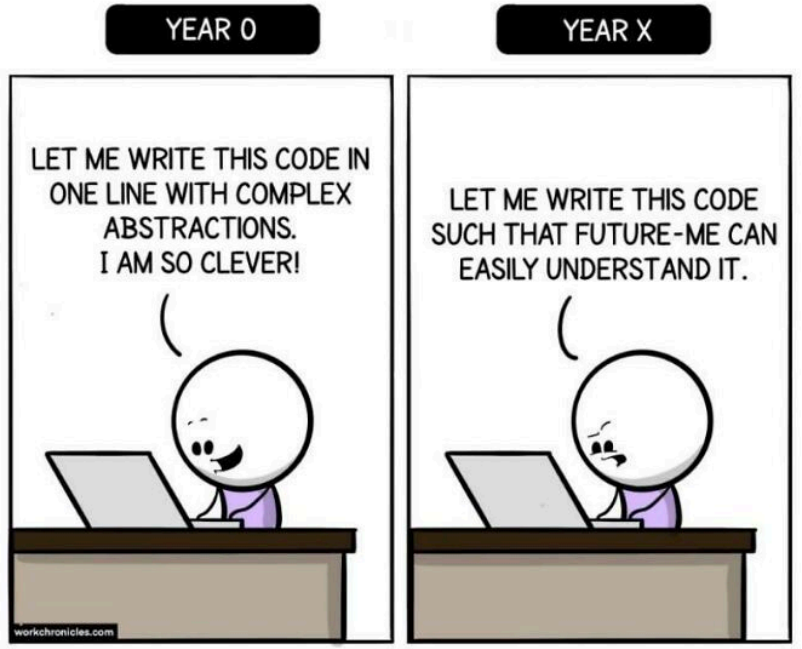
Yes, this level!!



What all needs to be considered? – scalability, performance, reliability,....

Why Design is important?

- It is like technical kernel of Software engineering
- Ensures traceability between requirements and final product
- Allows to have a complete picture of the software before construction
- Allows to discuss, confirm, reason and perform different analysis even before development - **why?**



Source: google images

*"Software is **not limited by physics**, like buildings are. It is **limited by imagination**, **by design**, by organization. In short, it is limited by properties of people, not by properties of the world. We have **met the enemy, and he is us**"*

Ralph Jhonson

Martin Fowler, ***Who needs an Architect?*** IEEE Software, 2003

Source: Googe images

SE - Major Principles

Abstraction

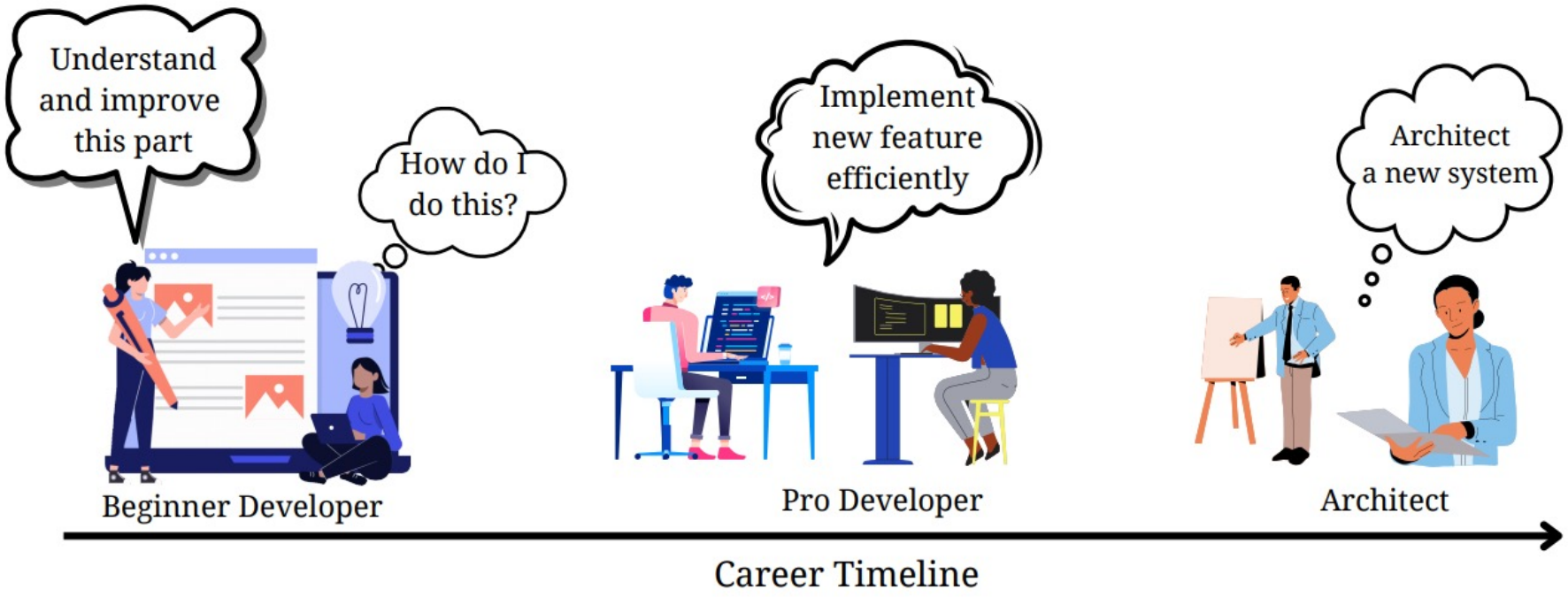
1. Focus on the relevant parts of the problem
2. Helps simplifying the problem and look at a larger picture

Decomposition

1. Decompose problem into small and independent pieces
2. Each pieces can be solved separately
3. The overall solution can be attained by solving smaller pieces and putting them together

What will you learn?

Course Outline



This Course

Modeling	Design Principles	Architectural Framework
Refactoring	Design Patterns	Architectural Patterns

Course Outline

How to comprehend existing software subsystems?

- Applying principles of refactoring

How to design software systems through abstractions and decompositions?

- Design Patterns and Architectural Patterns

How to apply them to your application?

- Deal with subsystems at higher level of abstraction provided by patterns

What if there is no exact match for a given problem?

- Evaluate alternatives and analyze trade-offs

How to capture and document the design decisions?

- Both at low level and high level

Teaching and Learning Methodology

- Problem based learning methodology

- A blend of conceptual lectures with group/individual activities

- This has been a working formula
  - Leaners engage in the materials
  - Hands on activities enables deeper understanding
  - Resembles the true career situation

Start forming the teams!!

Distribution of Grades

Component	Weightage
Final Exam	20%
Mid-term Quiz	10%
Unit Assignments	15%
3 Unit Projects (3*15)	45%
Other In-class activities	10%

Note: The instructor reserve the right to make minor changes
No extension requests please – **Extra days and soft deadlines!!**

Course Logistics

- Course announcements, management - moodle
- All resources, information and materials - https://karthikv1392.github.io/cs6401_se/
- At any point, feel free to contact either the instructor or TA
- Instructor office hours (Friday 11:00 AM to 12:00 PM)
- Feedbacks are always welcome!!

SERC Software Engineering Research Centre

Thank You

Course website: karthikv1392.github.io/cs6401_se

Email: karthik.vaidhyanathan@iiit.ac.in
Web: https://karthikvaidhyanathan.com
Twitter: @karthi_ishere