

SOFTWARE ENGINEERING

By

SAMUEL EFFIONG

 samueleffiong.jacob  @samueleffiong_  samueleffiongjacob  SAMUEL EFFIONG JACOB

 effiongsamuel  samueleffiong0  samueleffiong_official



samueleffiongjacob@gmail.com



samueleffiong685@gmail.com

Topics to cover

- Software engineering
- Software developer
- Types of programming languages
- Database

What is a Software?

- ***Software** is defined as the set of instructions or programs that are used by a computer for performing specific tasks. With some sets of instructions, we can create different software applications like – calculators, web browsers, operating systems, websites. etc.*

What is a Computer Program

- A computer program is a **sequence or set of instructions in a programming language for a computer to execute**

Difference between Program and Software

Computer Program

A set of instructions is called a program.

A program is smaller in size.

It lacks a proper interface.

It lacks proper documentation.

It is normally designed by a single developer.

Software

A set of programs to create an application is called software.

The software is larger in size.

It has a well-defined interface.

It is well documented.

A proper team of developers is required to design the software.

What is *Engineering*

- *Engineering* is a scientific field that involves taking the scientific understanding of the natural world and using it to invent, design, and build things to solve problems and achieve practical goals.

What is Software Engineering?

- *Software engineering is the systematic design and development of software products and the management of the software process. It has as one of its primary objectives the production of programs that meet specifications, and are demonstrably accurate, produced on time, and within budget.*

Software Engineering



Software Engineering A Layered Technology

- Software engineering, basically, comprises a process, a set of methods, and a collection of tools for managing and developing a software system.

Layered Technology



- **Quality** – Quality focus is an organization's goal that aims at improving software engineering processes.
- **Process** – A framework that must be established for the effective delivery of software that includes the timely development of the software, management, and control of software projects.
- **Methods** – Methods provide technical aspects i.e. “how-to” for building software e.g. requirement analysis, testing support, maintainability, etc.
- **Tools** – Tools provide automated/semi-automated support for the implementation of ‘process’ and ‘methods’. Simply put, tools help in developing software applications e.g. Adobe Creative Cloud helps in Graphics designing or animation

Why is Software Engineering required?

- **Changes in requirement** – With frequent changes in the business requirements and the environment, a properly documented and well-defined system of software development is required which makes it eminent to use Software Engineering.
- **Large and complex software** – With large and complex software, it becomes difficult to handle the software product but with the concept of software engineering, it is comparatively easier to build and manage them.
- **Scalability** – Scalability is used to describe the ability of the software to grow and manage according to the increasing demand of the market.
- **Adaptability** -It also helps in creating software that is adaptable to the changing needs of software usage efficiently.
- **To reduce the time** – More often than not software product development exceeds the time this can be because of various reasons, some of which can be avoidable. When software engineering concepts are employed, proper development process and tools are used that makes it possible to develop the software within the time-lines.

Attributes of Software Engineering

- **Maintainability** – Maintainability is the ability of the software to be modified, repaired or enhanced easily with the changing needs.
- **Reliability** – It is an attribute of software quality that provides the assurance that a product will produce the same results when used in a similar operational environment.
- **Efficiency** – Efficiency attribute provides a measure of the resource requirement of a software product in an efficient way. More the efficiency of the software, lesser will be the resource requirement which will, in turn, lead to lesser cost of development and operation.
- **Reusability** – Software engineering methods provide the feature of reusability through which a module can be used in multiple applications.
- **Flexibility** – Flexibility refers to the ability of a software product to adapt to the changing requirement and environment in which it is going to be used.
- **Portability** – Portability refers to the ability of the software to be platform-independent so that it can be used in a different environment.
- **Correctness** – The ability of software to adhere to its requirement specifications.
- **Testability** – It is simply referred to as how software can be tested easily. It checks whether the software operates properly or not.
- **Effectiveness** – Effectiveness refers to doing the right things. In terms of software engineering, effectiveness refers to the process followed in software development to meet its requirements.
- **Modularity** – It refers to the extent to which the software may be divided into smaller modules which in turn, enhances the design, implementation, debugging, testing, documentation, and maintenance of the software product.

What are the Software Crisis?

- Software crisis is referred to as the inability to write efficient software programs within the required time-frame. These crises occur when proper software engineering concepts are not employed in the development of a software application.

Some of the examples of software crisis are given below

- Software projects running over-time and not getting delivered within the specified time.
- Projects getting late, over budget and providing unreliable software that is expensive to maintain.
- Software applications becoming obsolete.
- Software becoming more complex when trying to scale them.

NOTE

- 'Software engineer' is catch of all phrase for several career paths in tech that work with software. However, taking a closer look reveals that 'software engineer' doesn't begin to accurately describe the in and outs of the many software engineer roles available.

»or

- Software engineer is a combination of several stacks in the software industry

TYPES OF SOFTWARE ENGINEER

- Backend Engineer : Backend software engineers are a type of software engineers that focus on building the pieces that power a website or mobile app. Think of the backend as the website's backbone: it is where the data is stored and most of the business logic is executed. It's where everything comes together. For example if you are using a task management app, the app itself (frontend) is responsible for displaying the data and providing functionality to edit the data – but what data you are allowed to edit, and how it gets stored, is handled by the backend and built by backend software engineers.

Backend software skills

- Experience with coding languages such as C, C++, Golang, PHP, Java, JavaScript or Typescript Python, Ruby,
- Familiarity with server-side frameworks like Laravel, Express.js, or Django, Ruby On Rails, Spring, Flask, and [ASP.NET](#)
- Good grasp of MySQL, MongoDB, or Cassandra
- Ability to write efficient and scalable code
- Familiarity with cloud services such as Amazon Web Services, Google Cloud Platform, or Heroku
- Capacity to meet with clients and stakeholders to understand needs, vision, and goals

Frontend Engineer

- A frontend developers is a software developer that builds the visual interface you interact with. Anything you interact with in your browser is programmed by a frontend engineer. Frontend software development takes different components of a website, like text, design, photos, etc., and assemble them into one cohesive site that users can easily navigate.
- Frontend is really short for "Frontend Web Developer", because developers who build mobile apps are typically called a Mobile Developers – even though in both cases they are both building an application that "sits in front" of a backend application.

Skills

- Strong knowledge of HTML, CSS, and JavaScript
- Ability to code responsive designs that work across various devices and screen sizes
- Familiarity with frontend frameworks such as React, Angular, or Vue.js
- CSS pre-processors such as Sass or Less
- Good understanding of UX/UI design principles

Full Stack Engineer

- Frontend + Backend = Full Stack. Full stack software engineers are engineers that work across the full "stack" (front and backend) of dev technology. Often considered web generalists, these software developers wear many hats and can quickly adapt to different demands of web development, whether that be user-facing code, database queries, or anything in between. Full stack developers may not always have as much in-depth knowledge as their one-sided counterparts, but their ability to work on both sides of the application allow them to complete entire application features in many cases completely on their own.

Skills

- Strong knowledge of HTML, CSS, and JavaScript
- Familiarity with frontend frameworks such as React, Angular, or Vue.js
- CSS pre-processors such as Sass or Less
- Strong knowledge of programming languages such as PHP, Java, Python, or Ruby
- Familiarity with server-side frameworks such as Laravel, Express.js, or Django
- Good understanding of MySQL, MongoDB, or Cassandra
- Ability to write efficient and scalable code
- Good understanding of UX/UI design principles

Read more here

- <https://www.codewars.com/post/10-different-types-of-software-engineers-a-comprehensive-guide>

What is full-stack developer

- A full-stack developer is a **developer or engineer who can build both the front end and the back end of a website.**

What is what is backend developer

- Backend developer : is a **developer who can build back end of a website.**

What is front end developer

- **front end developer is a developer who can build the front end of a website.**

What is library

- libraries with prewritten code snippets that we can use and reuse to build applications.
- Examples: **JQUERY**
- **REACT JS**
- **VUE**
- **ANGULAR**
- Etc.

What is framework

- frameworks are a full toolset that helps shape and organize your website or application
- Examples :
- Next.js
- **Nuxt.js**
- **Ionic**
- Etc.

What is database

- **A database is information that is set up for easy access, management and updating.**

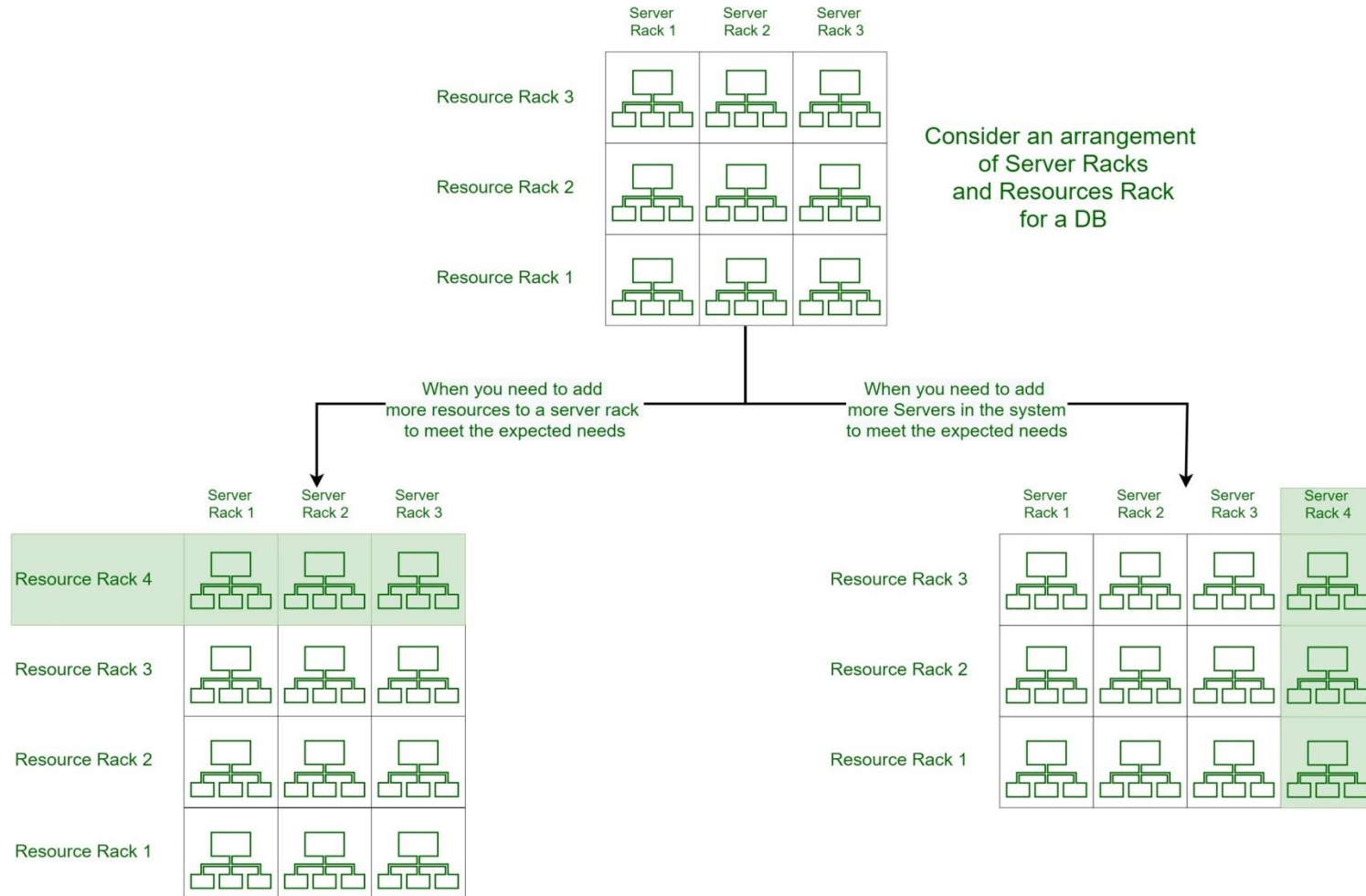
Types of database

- **vertically** — storing different tables & columns in a separate database. In general It is a very strong organize / structured database with low performance..
- Place that uses vertical database are banks, MS word, etc.
- **horizontally** — storing rows of a same table in multiple database nodes. In general it is an unstructured / unorganized database with high performance.. Any organisation

Horizontal and Vertical Scaling In Databases

- **Scaling** alters size of a system. In the scaling process, we either compress or expand the system to meet the expected needs. The scaling operation can be achieved by adding resources to meet the smaller expectation in the current system, or by adding a new system in the existing one, or both.

Horizontal and Vertical Scaling In Databases



↕ **Vertical Scaling**

↔ **Horizontal Scaling**



Difference between Horizontal and Vertical Scaling:

Horizontal Scaling

When new server racks are added in the existing system to meet the higher expectation, it is known as horizontal scaling.

It expands the size of the existing system horizontally.

It is easier to upgrade.

It is difficult to implement

It is costlier, as new server racks comprises of a lot of resources

It takes more time to be done

Vertical Scaling

When new resources are added in the existing system to meet the expectation, it is known as vertical scaling

It expands the size of the existing system vertically.

It is harder to upgrade and may involve downtime.

It is easy to implement

It is cheaper as we need to just add new resources

It takes less time to be done

examples

- Mongo dB, Cassandra dB, Radis.
- E.g. of vertical db.
- MySQL, Sql, Maria dB.

Questions

- Ask your questions

To learn is very expensive

- You will need a sponsor
- And time

Types of programming language

- C,
- C++
- Golang
- Java
- Python
- Rust
- JavaScript
- Html
- Css

Thank u for time

- Yeap we are the end of the tour..
- Enjoy your journey..
- Please follow me on my social media



[samueleffiong.jacob](https://www.facebook.com/samueleffiong.jacob)



[@samueleffiong_](https://twitter.com/@samueleffiong_)



[samueleffiongjacob](https://github.com/samueleffiongjacob)



[SAMUEL EFFIONG JACOB](https://www.linkedin.com/in/SAMUEL EFFIONG JACOB)



[effionsamuel](https://www.instagram.com/effionsamuel)



[samueleffiong0](https://www.instagram.com/samueleffiong0)



[samueleffiong_official](https://www.instagram.com/samueleffiong_official)



samueleffiongjacob@gmail.com



samueleffiong685@gmail.com