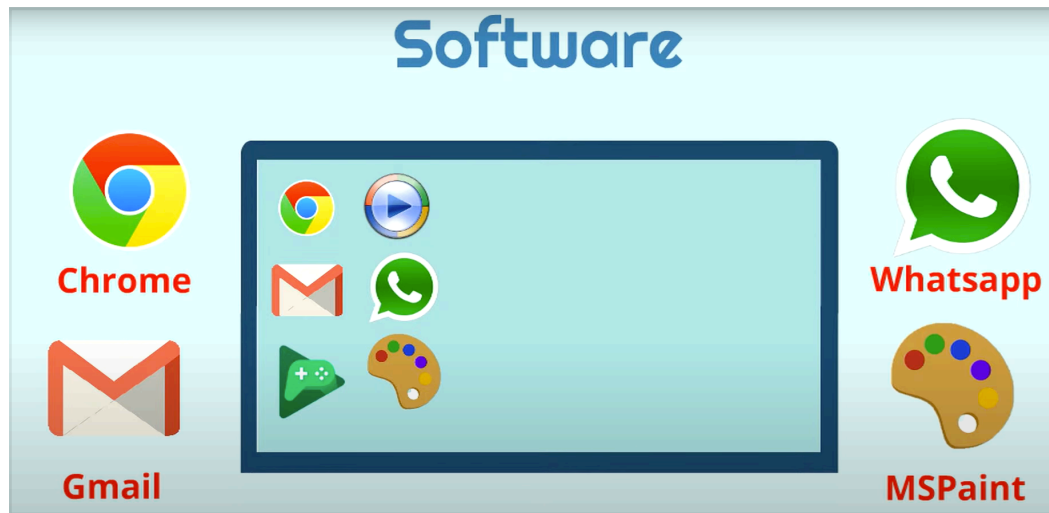


## What is a Software

Set of instructions or programs that enable a user to interact with the computer

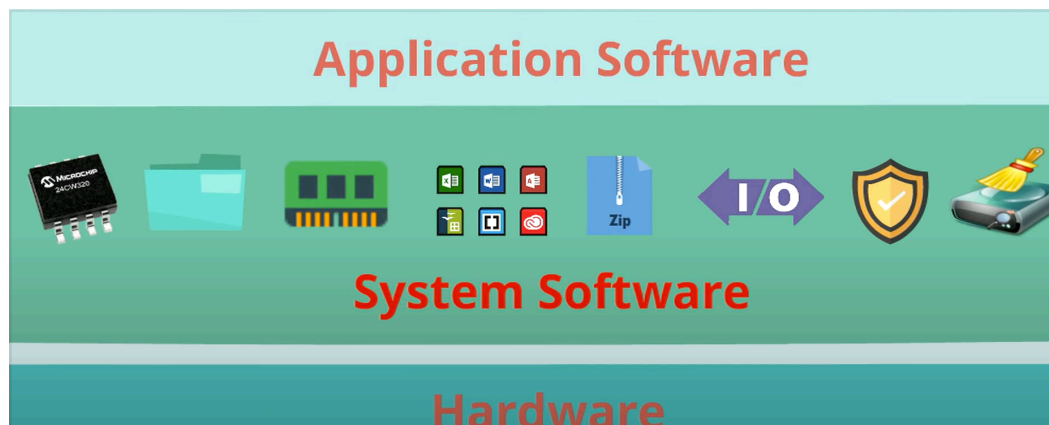


Two types of Software

1. Application Software
2. System Software

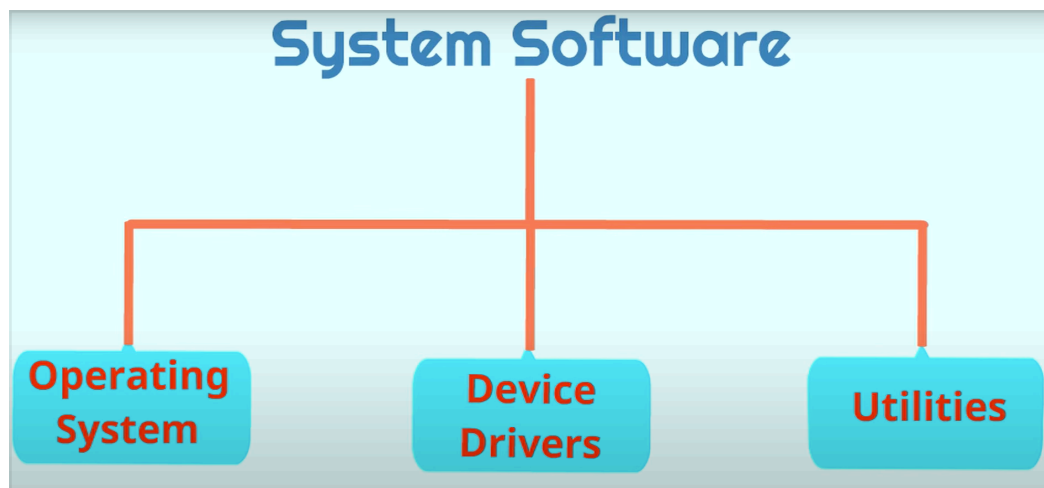
Examples of Application Softwares

1. Gmail - We use GMAIL for sending emails
2. Google Chrome - To browse the internet
3. WhatsApp - To chat, for audio and video calling
4. Facebook - Social Media



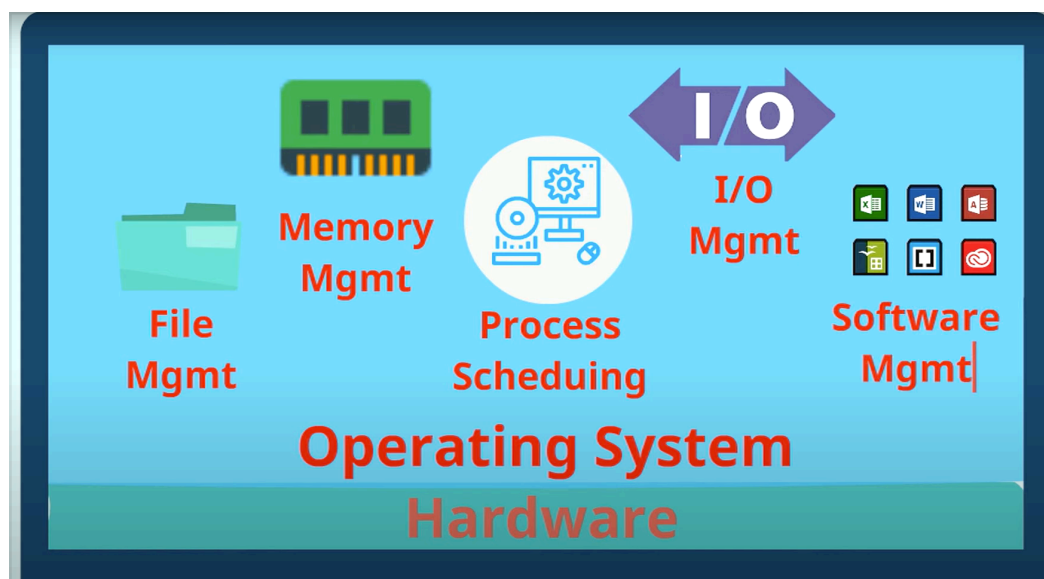
System Software typically comes with a computer and manages the hardware

of the computer and provides basic functionality required by the user.



Operating System - helps us to start up or boot up the system.

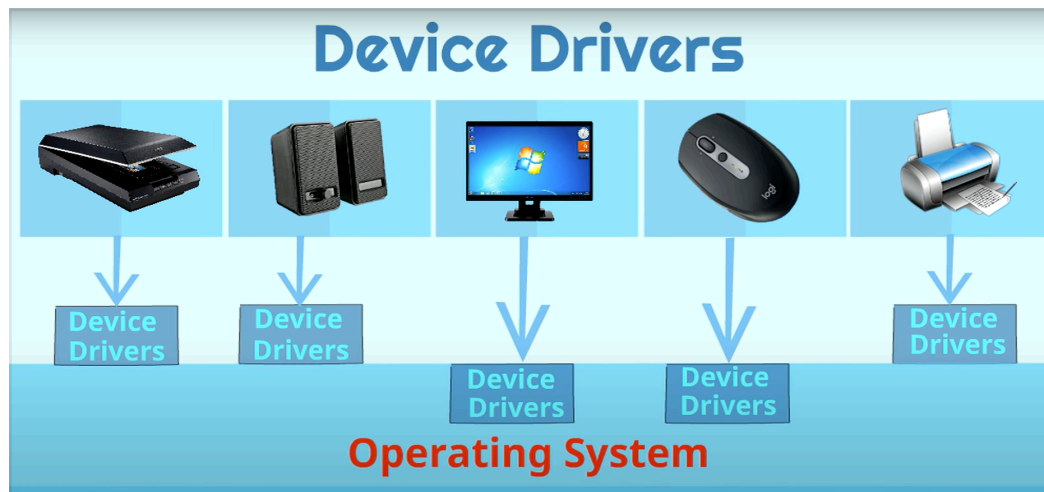
OS are a set of programs which manage the comp hardware and provide common services such as file and memory management, process scheduling, I/O management, and managing other softwares



Examples of Operating Systems are

1. Windows
2. Android
3. Linux
4. Macintosh

Device Drivers - programs which controls the various devices which are connected to the computer. The operating System comes bundled with device drivers, but if we install a new printer we have to install the device driver which comes bundled with the device.



Utilities are programs that assist users in performing some regular maintenance tasks such as antivirus, compressing or cleaning of files, backup etc.

## WHAT IS SOFTWARE

### DEVELOPMENT

Software development is the process of creating all those amazing applications, programs, games that we use in our daily lives for example: Facebook, Insta, WhatsApp, or lets say Angry Birds, Call of Duty, or a Phonepe or Google Pay that we use everyday.

Now these applications that we use daily had to be developed and those applications are developed by folks known as developers.

So inorder to ship a software or in simple words make it available for the common users to use, there are a lot of steps that are involved.

Let's try to find analogy from building a house.

Let's say we are planning to build a house for ourselves with a heated swimming pool and sports area and all the different types of luxuries possible. - This would be the high-level idea of how we want our house to look like right?

Its similar with software too - for example when Mark Zuckerberg decided to build Facebook he had an idea and what was that? The idea was to connect the world over the internet?

Now coming back to our house - we know what we want, we then have to come up with the idea about how to build it?? So what do we need? We might take help of an architect to come up with a design plan, so that the house meets all our requirements, is cost effective?

Similarly even in software - we know what we want, but we will have to layout a

design plan first right?? How we layout a design plan and all we will learn in our further classes, but I hope you are getting an understanding of what I am trying to explain right?

Then once we have the design in place, what would be the next steps for the house??

We have to start building it right? Similarly in software it is calling coding, we will start coding the software.

Now once the house has been constructed - we need to make sure that the house is safe, there is no malfunctioning of power supply, check for water leakages etc??

In software development - we call it Testing - we test the software to make sure that there are no loopholes before handing it over to our users.

Now that we have tested everything and things are looking good, we have to hand over the keys to the owner, provide the owner with No Objection Certificate, Clearance certificate and all of that stuff

In software engineering we call it deployment, we deploy the software in such a way that it is now available for the users to use it.

Now imagine we have started staying the apartment and then there are some faulty connection, or water leakage etc. what would we do, we would call the electrician or the plumber who had worked at the time of building the house to get the issues fixed right.

In Software Engineering we would call it maintenance, where we maintain the software so that is available to the users all the time.

POLL - Have you heard of the term SDLC? Not to worry I will give a brief overview of the entire thing anyways

## **WHY SHOULD WE CHOOSE TO BECOME A SOFTWARE DEVELOPER AT ALL?**

To be honest it gives me goosebumps when I talk about the power a software engineer possesses.

Can you imagine you write code on a laptop with 15 inch screen, with a configuration of 16 GB RAM, 512 GB SSD, and 2.1 GhZ processor dual core or quad core or octane core processor which is being used by people across the world??

For example - when I was working in Microsoft Teams we had built a software from the scratch which is used by 145 million users daily. 145 million users is

equivalent to 14.5 crore people across the world.

Or let's say when I worked on Phonepe, I developed the software for the cash back which is being used across the country by millions of people again.

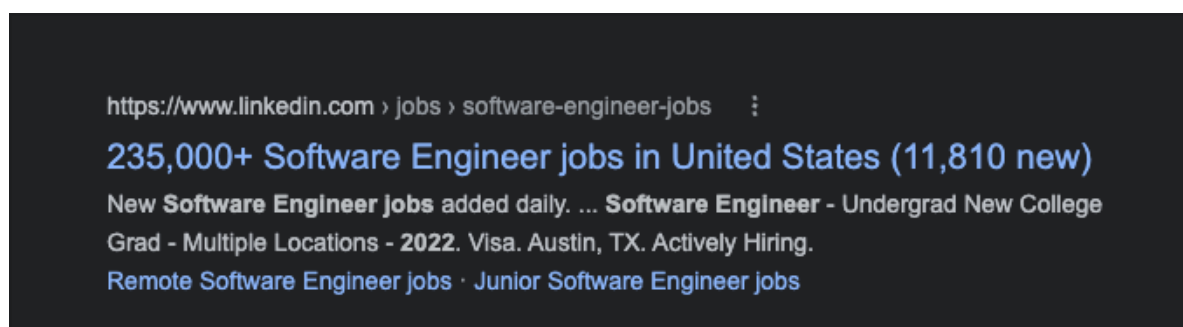
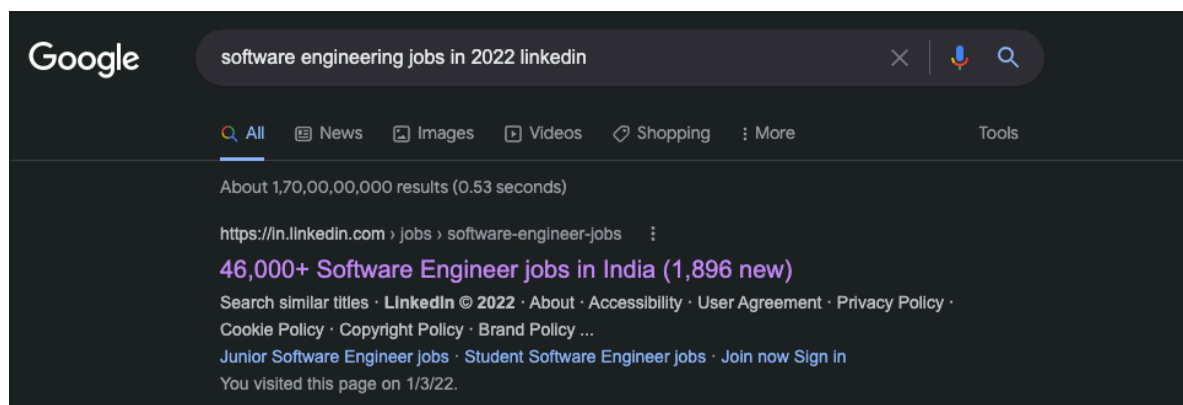
To be honest in the recent few years there has been a huge revolution of skills over degree - where if you have the right skills, it really doesn't matter what your background is and companies will hire you, without any discrimination in the paycheck. And not only that you can actually work on your own idea and make it available to the entire nation or even the world for it to use.

Apart from that, Software Engineering jobs are really highly paid jobs. If you can become a good software engineer, which I pretty much assume you all will, if you can stick to the course, then I am quite certain you can land up with really well paid jobs

## SOFTWARE JOBS IN 2022

### AND BEYOND

To give a perspective about the job market world wide, I thought of doing a google search and this is the result that I found. I just wanted to show you folks in numbers the openings and you won't believe me this is just increasing everyday. Almost every other day there is a new startup coming up. They need engineers, really smart engineers.



When the covid waves had started we saw that all the sectors started to suffer. In fact we thought that there will be dearth of jobs in IT sector as well.. But that wasn't to be, organisations moved to a work from home mode and that

eventually became the new normal in course of time. And in fact in the last couple of years there has been a revolution in the number of software engineering jobs. And I will tell you the reason why.

1. Let's talk about meetings - people used to have meetings in a meeting room previously, but now they had to move to an alternative, so apps like teams, zoom, bluejeans etc. saw a massive spike in adoption
2. Going out to buy groceries - with such high surge in covid cases, it would be very difficult for people to go out and do the daily grocery Shopping, and so companies like bigBasket, Swiggy, dunzo saw a surge in user adoption. Many new organisations also came up like Zepto. Etc etc
3. Going to banks - going to banks had become a problem right, so automatically people started relying on online transfers let's say using Phonepe, Google Pay, Paytm etc.
4. No access to movie theatres - because of no access to movie theatres all these OTT companies like hotstar, prime videos, Netflix etc. have seen an even higher adoption.
5. Purchasing items online - There was a time when only essential workers were allowed to do deliveries so we chose to purchase items more from amazon, Flipkart, etc. etc.
6. Medical - People would be scared to go to hospitals because of chances of coming in close proximity of covid patients, so online consultations with doctors became a good medium of treatment.

These are only few of the many problems that covid had raised and how software industry has solved such problems for people.

So if you see, during Covid, the IT sector actually a huge uprise in adoption and when more the requirements, the more the number of problems to be solved and when there are more number of problems to be solved, organisations would need more and more people for the same.

Also during Covid, employees and organisations have beautifully adopted the concept of remote working or work from home which has opened up opportunities for work globally. Which means that we can be in a small town and still get our work done sitting at home, with just the help of internet.

## **IS SOFTWARE ENGINEERING REALLY**

### **EASY??**

Well Software Engineering is not very easy.

There are major challenges

1. There is always a very high competition - if you take a look at LinkedIn itself, you will realise the increase in the number of candidates applying for software engineering roles, and hence if someone wants to nail the interviews, and do well at their job, it will be really important for them to really good at what they do.
2. Evolving technologies - Technology is ever evolving, if someone is

ready to learn about new technologies and explore new technologies, that can definitely act as a bottleneck in their careers. Its very important that every individual has that forward mentality of trying to get adopt to new technologies, explore them. For example, In the initial days, people would code more in Java, for application development because it was the most stable language. And most widely adopted. But the development process using Java can be slightly time consuming and for start ups that can act as a bottleneck. As a result people started exploring other languages such as Python, Node js. Etc. depending on their requirements that would be efficient as well as scalable in future. Same goes with databases, there was wide adoption of Relational Databases till the late 2000's but with increase in users, it became difficult for users to stick to RDBMS and had to explore other databases such as NoSql databases such as In-memory Key Value stores, Document databases, etc. etc. So in every field of technology there has been a major evolutions and it becomes important for engineers to stay on top of these to make sure they are choosing the right technologies because as I said there has been surge in adoption of apps and it is very important for these organisations to be able to serve their customers with the same smoothness and not spoil the user experience.

3. Lack of awareness of the right technologies - People maybe stuck with very old tech stacks which is another problem, people should be flexible enough as I said to adopt to new technologies. Because it is important to have the right technologies at the right place to get the best intended outcome considering a lot of cost efficiency.

## **LETS TAKE A SLIGHTLY DEEPER DIVE INTO HOW APPLICATION SOFTWARE DEVELOPMENT LOOKS LIKE**

When we talk about Application Development, we will very frequently come across the term called Web Application

Now what is a web application? - web application is basically a piece of software which makes a webpage alive or in other words, it actually lets end users like us interact with the application. For example: amazon - you can purchase items on amazon, post a review comment, leave a rating for the product etc. Facebook or LinkedIn - where you can put up posts, pictures, comment on other's posts, chat with people on messenger etc. etc.

On the other hand a website is usually static in nature, when I say static I mean an end user can view the contents of the page but not interact with it, for example blogs - you can read through blogs usually, but not make changes and stuffs, menu card of restaurants etc.

Now that we have a high level understanding of what a Web Application is - let's try to break down the application into different parts.

1. We interact with an application for example: Facebook through a browser or mobile application right? What is that?? It's basically the front end of the application we also know it as the UI which stands for User Interface.
2. Now let's say we comment on friend's post, something happens behind the scenes right?? Facebook should know that okay person A has made a post P by person B. So the application has been written in such a way that it can identify this process right.
3. Now the next time you visit the same friend's post, you should be able to view the comment that you had made on the post? Which means the application should have it saved somewhere right? Otherwise it will lose it and will not be able to show it the next time you visit the friend's post. So what is the application doing? It is persisting the data somewhere.

Now if we talk about all the above three points that we discussed, we can essentially break down a web application into three different layers?

1. The UI layer is known as the Presentation Layer
2. The logic layer where the application processes the comment that we posted on a friend's post is called - the Business Layer
3. And finally, the layer where the data is being persisted somewhere is called - the Data Access Layer.

In future you will come across three tier architecture or multi-tier architecture over and over again. This that I explained to you just now is what is called a three tier architecture



# WEB APPLICATION ARCHITECTURE

## Client

Browser

## Server

Presentation layer

Business/Application layer

Persistent storage layer

Cross-Cutting

Data sources

3<sup>rd</sup>-party integrations

Services

