

# Introduction to SE

## Assignment 1

Submitted To: Ms. Anosha Khan

Submitted By: Haider Abbas Moazzam

Class: SE-2B

Roll No: 22L-7862

### *Question 1:*

**(A)** No, the Software does not wear out like Hardware. It does deteriorate but it does not wear out like hardware, because hardware experiences environmental factors like dust, temperature or overheating and other similar factors, while software does not face these kinds of conditions, this is the reason hardware wears out and software does not.

According to the bath tub curve the failure rate of Hardware in the beginning is relatively high which is attributable to manufacturing defects, but a time comes when it becomes stable but soon after it starts to wear out and deteriorate again.

**(B)** When a software is newly made, it can have some potential errors and bugs which cause high failure rates in its first stage. When changes are made to it along with updates it also introduces some defects. However, if a software is tested multiple times with different test cases,

then it can help remove potential defects, bugs or even viruses. Apart from this if we go to the low level and programmers use the right algorithms and keep in mind the time complexity and the memory management of a software then it can also help to reduce such defects. The Lehman laws of 'continuous change' can also be proven fruitful in this case.

**(C)** The main difference between the bathtub curve of hardware and failure curve of Software is that Hardware failures are at high in the beginning, and they become stable but soon they shoot up again, however in software systems the failure rates are reduced but making changes to it and reducing the errors along with testing and a time comes when it becomes stable, and the curve becomes ideal.

## *Question 2:*

**(A)** A legacy system is a computer system, application, or technology that is outdated or obsolete and is no longer being actively developed or maintained. There are several characteristics that define a legacy system, the reason large companies continue to use legacy systems despite the high maintenance cost is because they become familiar to it, and it often takes a lot of time for employees to shift on new systems and learn them. Apart from this, new systems require a large amount of money to program them, which is not feasible for companies. Some systems have been heavily customized to fit the specific needs of the company and there may not be an equivalent solution available.

**(B)** Some common software engineering problems in Legacy systems are that they may have security vulnerabilities that were not addressed when they were originally developed, making them vulnerable to cyber-attacks. Maintaining a legacy system can also be expensive and time-consuming, requiring a significant investment of resources to keep

it functioning properly. They may not be designed to handle substantial amounts of data or high levels of traffic, leading to performance issues.

(C) There are several reasons which indicate the need to evolve a legacy system such as Security needs. If we take the example of Facebook. When it was created, it was mainly marketed as a social application. But as the time passed it was developed into a marketing app and along with it some gaming applications were also added into it. This shows how the business needs of a system change with time. Improved efficiency and productivity are also a reason to update legacy systems. For example, if we consider google. At the start the number of users was less. But as the time passed the number of customers using google and its services increased so it had to update their data management techniques and introduce technologies like cloud to manage it.

### *Question 3:*

(A) If software engineers ignore or violate well-accepted software engineering (SE) guidelines, a variety of problems can occur. Developing software that does not follow SE guidelines can result in higher costs due to the need for extensive debugging, rework, and maintenance. If the budget allocated by the stakeholders is exceeded, then it becomes difficult to produce the software as desired by the customer .The Software that does not follow SE guidelines may not be scalable, making it difficult to add new features or accommodate growth in the user base. For example, if the developer does not use the right guidelines of indenting the code or commenting on the code then it would be difficult for someone else to understand it and make changes or updates to the software.

(B) No other fields are also affected if the right guidelines are not followed. Like medicine, construction, or finance. Violations in medical practice can include prescribing incorrect medications, misdiagnosing patients, or not following proper sanitation procedures. Violation in Finance can lead to frauds or insider trading.

### *Question 4:*

(A)

1. Angular JS (I have keen interest in JavaScript libraries that's why I have chosen this one)
2. Firebase SDK (Looks interesting as it helps developers to create web apps and monetize them and I have interest in Full Stack development, so it looks useful for that)
3. TensorFlow (With the world moving towards AI (Artificial Intelligence), this library full of tools, and resources for ML looks interesting)

(B)

[GitHub Profile](#)

(C)

GitHub can be proven useful if I want to contribute to Open-source projects as it has multiple repositories that can help me manage my code well and If I want any of my teammates to help me then we can easily have peer work. We can also use it to find any kind of

programming help because it is rich in repositories supporting different languages.

## (D)

Google Summer of Code (GSoC) is a global program that offers students the opportunity to work on open-source software projects over the summer. The program was launched in 2005 by Google and has since become one of the largest open-source programs in the world.

GSoC offers students the opportunity to work with some of the most innovative open-source projects, including those related to software engineering, machine learning, big data, cloud computing, and more. During the program, students work with experienced mentors to develop their skills and contribute to real-world projects. They receive a stipend for their work, which provides them with the financial support they need to focus on their projects full-time.

GSoC provides students with a unique opportunity to gain practical experience in software development, work with leading open-source projects, and make valuable contributions to the open-source community. It is a competitive program, with thousands of applicants from around the world applying each year. However, for those accepted, it is an opportunity to significantly impact the world of open-source software.