Software Engineering Basics

The true work of **software engineering** begins before the product has even been designed – and the software engineering basics dictate that it continues long after the "work" has been completed. It all begins with a thorough and complete understanding of what your software needs to have

– this includes what the software needs to do, the system in which it needs to operate, and all of the security that it entails. **Security** is one of the software engineering basics because it is so essential to all aspects of development. Without tools to help you better understand how your code is being built and where any security problems may fall, your team can easily become lost in the development stage.

Software engineering design basics require creating the instructions for the computer and the systems. Much of this will take place at the **coding** level by professionals who have comprehensive training. Still, it is important to understanding that software engineering isn't always a **linear process**, which means that it requires thorough **vetting** once it has been completed.

Not all software requires software engineering. Simplistic games or programs that are used by consumers may not need engineering, depending on the risks associated with them. Almost all companies do require software engineering because of the **high-risk information** that they store and security risks that they pose.

Software engineering helps to create **customized**, **personalized** software that should look into **vulnerabilities** and risks before they even emerge. Even when the software engineering principles of safety aren't required, it can also help to reduce costs and improve **customer experience**.

Personalized: Tailored to a specific person's needs or preferences.

Coding: Writing the specific instructions that tell a computer what to do.

Software Engineering: The process of making computer programs, including planning, designing, coding, and testing.

Customer Experience: The overall feeling or satisfaction a customer has when interacting with a company or its products.

Vetting: Carefully examining or checking something for potential issues or problems.

Security: Protection against harm, damage, or unauthorized access to information or computer systems.

Linear Process: A way of working where tasks are done one after another in a straight line, rather than jumping back and forth.

Vulnerabilities: Weaknesses or flaws in a system that can be exploited by attackers.

High-Risk Information: Sensitive data that, if accessed by unauthorized people, could cause serious problems.

Customized: Made specifically to meet individual or unique requirements.

Health records contain **high-risk information**, so they must be kept confidential.

Outdated software can have **vulnerabilities** that hackers can exploit to gain access to your computer.

Online banking websites have strong **security** measures to protect your financial information.

Excellent **customer experience** is crucial for a positive customer experience when shopping online.

Learning **coding** is like learning a new language for computers, where you give them instructions.

Many online services offer **personalized** recommendations based on your preferences and behavior.

The company carefully **vets** all the job applicants to find the most qualified candidates.

Some companies offer **customized** products, allowing you to choose the features you want.

Software engineering involves creating computer programs and ensuring they work properly.

In a **linear process**, you follow steps one after the other, like building a house from the foundation up.

What does software engineering encompass, and when does it start?

Software engineering encompasses the entire process of designing, developing, testing, and maintaining software applications. It starts during the early stages of the software development life cycle, such as during requirements analysis and system design.

Why is security considered one of the software engineering basics?

Security is considered one of the software engineering basics due to its importance in all aspects of software development. It's crucial to protect software from potential threats, vulnerabilities, and unauthorized access, ensuring the integrity and trustworthiness of software applications.

What does "coding" refer to, and who usually performs this task?

"Coding" refers to the process of writing the specific instructions that tell a computer what to do. This task is typically performed by trained professionals who specialize in software development, such as software engineers, programmers, and developers.

Does all software require software engineering? Explain.

Not all software requires the full spectrum of software engineering. The level of software engineering applied may vary depending on the complexity, scale, and criticality of the software. Simple scripts or small applications may require less formal software engineering, while complex, large-scale systems necessitate thorough software engineering practices to ensure reliability, maintainability, and security.

How can software engineering benefit companies, even if they don't require the principles of safety?

Software engineering can benefit companies in various ways, even if they don't require extensive safety principles. It helps achieve better software quality, leading to improved customer satisfaction, increased efficiency, and reduced maintenance costs. Additionally, software engineering practices enhance project predictability, resulting in more successful and cost-effective software development projects.

1: What is the primary focus of software engineering?

A) Reducing costs

B) Speeding up development

C) **Ensuring security and reliability**

D) Minimizing risks

E) Outsourcing development

2: When does the work of software engineering start according to the text?

A) After product design

B) During coding

C) At the end of development

D) When security is compromised

E) **It doesn't specify**

3: Why is security considered one of the software engineering basics?

A) Because it's an optional feature

B) To slow down development

C) To add complexity

D) **Due to its importance in all aspects of development**

E) To reduce costs

4: Who typically performs coding in software engineering?

A) The Chief Information Officer

B) Independent contractors

C) **Trained professionals**

D) IT leaders

E) Maintenance teams

5: When is rework necessary in software engineering?

A) After the product has been designed

B) During the coding phase

C) Only during the maintenance phase

D) To reduce costs

E) **When errors or changes require code modifications**