

Contents

[1. Introduction to Software Engineering 3](#_Toc155537841)

[A. What is Software Engineering 3](#_Toc155537842)

[B. Professional software development 3](#_Toc155537843)

[C. Software engineering ethics 4](#_Toc155537844)

# Introduction to Software Engineering

## What is Software Engineering

Software engineering is the process of developing, testing and deploying computer applications to solve real-world problems by adhering to a set of engineering principles and best practices. The field of software engineering applies a disciplined and organized approach to software development with the stated goal of improving quality, time and budget efficiency, along with the assurance of structured testing and engineer Certification.

## Professional software development

Professional software development is a complex and dynamic field that involves the process of designing, creating, testing, and maintaining software systems. It is a discipline that requires a combination of technical expertise, problem-solving skills, and effective communication. Here are key aspects of professional software development:

* Requirements Analysis:

Understanding and gathering requirements from stakeholders to determine the functionalities and features the software must have.

Translating business needs into technical specifications.

* Design:

Creating a high-level architecture and detailed design for the software. Considering factors like scalability, maintainability, and performance during the design phase.

Implementation/Coding:

Writing code based on the design specifications. Following coding standards and best practices to ensure code quality.

* Testing:

Conducting thorough testing, including unit testing, integration testing, and system testing, to identify and fix bugs. Implementing automated testing to improve efficiency and reliability.

* Version Control:

Using version control systems (e.g., Git) to manage code changes, collaborate with team members, and track the history of the project.

* Collaboration:

Working collaboratively with cross-functional teams, including designers, product managers, and quality assurance professionals. Effective communication is crucial for successful collaboration.

* Documentation:

Creating and maintaining documentation for code, design decisions, and project processes.

Ensuring that documentation is accessible and understandable by team members and stakeholders.

* Deployment:

Deploying software to production environments.

Implementing continuous integration and continuous deployment (CI/CD) practices to streamline the release process.

* Maintenance and Support:

Addressing and resolving issues that arise after the software is deployed.

Implementing updates, patches, and improvements based on feedback and changing requirements.

* Security:

Incorporating security best practices throughout the development lifecycle to protect against potential vulnerabilities and threats.

Adaptability and Continuous Learning:

Staying updated with new technologies, methodologies, and best practices in the rapidly evolving field of software development. Being adaptable to changing project requirements and business needs.

* Quality Assurance:

Ensuring that the software meets quality standards and complies with relevant regulations. Conducting code reviews and other quality assurance processes.

## Software engineering ethics

Software engineering ethics can be approached from three directions. First, it can describe the activity of software engineers making practical choices that affect other people in significant ways. Second, it can be used to describe a collection of principles, guidelines, or ethical imperatives that guide or legislative action, and third, it can be used to name a discipline that studies the relationship between the other two senses of ethics. Software engineering ethics is clearly both an activity and a body of principles. The discipline of software engineering ethics that studies this activity and formalizes these principles, however, is in its infancy.

# Software Evolution

Software evolution is the process of developing software initially and then updating it for various reasons, such as adding new features, removing obsolete functionalities, or meeting changing requirements. It is an important process because organizations invest large amounts of money in their software and are completely dependent on this software. Software evolution helps software adapt to changing business requirements, fix defects, and integrate with other changing systems in a software system environment. The process of software evolution is ongoing and requires continuous investment of time and resources.

# DevOps

DevOps is a set of practices, tools, and a cultural philosophy that automate and integrate the processes between software development and IT teams. It emphasizes team empowerment, cross-team communication and collaboration, and technology automation.

The DevOps movement began around 2007 when the software development and IT operations communities raised concerns about the traditional software development model, where developers who wrote code worked apart from operations who deployed and supported the code. The term DevOps, a combination of the words development and operations, reflects the process of integrating these disciplines into one, continuous process.

## Process of DevOps

