Introduction to Software Engineering

Introduction to Software Engineering

Software engineering is an engineering discipline that is concerned with all aspects of software production from the early stages of system specification through to maintaining the system after it has gone into use. In this definition, there are two key phrases:

1. Engineering discipline - Engineers make things work. They apply theories, methods, and tools where these are appropriate. However, they use them selectively. and always try to discover solutions to problems even when there are no applicable theories and methods. Engineers also recognise that they must work to organisational and financial constraints so they look for solutions within these constraints.

2. All aspects of software production - Software Engineering is not just concerned with the technical processes of software development. It also includes activities such as software project management and the development of tools, methods, and theories to support software production.

The systematic approach that is used in software engineering is sometimes called a software process. A software process is a sequence of activities that leads to the production of a software product. There are four fundamental activities that are common to all software processes. These activities are:

1. **Software specification** - Where customers and engineers define the software that is to be produced and the constraints on its operation.
2. **Software development** - Where the software is designed and programmed.
3. **Software validation** - Where the software is checked to ensure that it is what the customer requires.
4. **Software evolution** - Where the software is modified to reflect changing customer and market requirements.

There are two kinds of software products:

1. **Generic products** - These are stand-alone systems that are produced by a development organisation and sold on the open market to any customer who is able to buy them.

2. **Customised (or bespoke) products** - These are systems that are commissioned by a particular customer. A software contractor develops the software especially for that customer.

Software Engineering Diversity

Software engineering is a systematic approach to the production of software that takes into account practical cost, schedule, and dependability issues, as well as the needs of software customers and producers. There are no universal software engineering methods and techniques that are suitable for all systems and all companies. Perhaps the most significant factor in determining which software engineering methods and techniques are most important is the type of application that is being developed. There are many different types of application as below.

- Stand-alone applications
- Interactive transaction-based applications
- Embedded control systems
- Batch processing systems
- Entertainment systems
- Systems for modelling and simulation
- Data collection systems

Software Development Team Roles

- Project Manager
- Product Owner
- Tech Lead
- Business Analyst
- Team Lead
- Full-Stack Developer
- Front-end Developer
- Back-end Developer
- QA Lead
- QA Engineer
- Tester
- UI Designer
- UX Designer

Software Engineering Ethics

Like other engineering disciplines, software engineering is carried out within a social and legal framework that limits the freedom of people working in that area. As a software engineer, you must accept that your job involves wider responsibilities than simply the application of technical skills. You must also behave in an ethical and morally responsible way if you are to be respected as a professional engineer.

1. Confidentiality - You should normally respect the confidentiality of your employers or clients irrespective of whether or not a formal confidentiality agreement has been signed.

2. Competence - You should not misrepresent your level of competence. You should not knowingly accept work that is outside your competence.

3. Intellectual property rights - You should be aware of local laws governing the use of intellectual property such as patents and copyright. You should be careful to ensure that the intellectual property of employers and clients is protected.

4. Computer misuse - You should not use your technical skills to misuse other people's computers. Computer misuse ranges from relatively trivial (game playing on an employer's machine, say) to extremely serious (dissemination of viruses or other malware).

Frequently asked questions

1. What is software?

Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market

2. What are the essential attributes of good software?

Maintainability, Dependability and security, efficiency and acceptability

3. What is software engineering?

Software engineering is an engineering discipline that is concerned with all aspects of software production.

4. What are the fundamental software engineering activities?

Software specification, software development, software validation, and software evolution.

5. What is the difference between software engineering and computer science?

Computer science focuses on theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.

6. What is the difference between software engineering and system engineering?

System engineering is concerned with all aspects of computer-based systems development including hardware, software, and process engineering. Software engineering is part of this more general process.

7. What are the key challenges facing software engineering?

Coping with increasing diversity, demands for reduced delivery times, and developing trustworthy software.

8. What are the costs of software engineering?

Roughly 60% of software costs are development costs; 40% are testing costs. For custom software, evolution costs often exceed development costs.

9. What are the best software engineering techniques and methods?

While all software projects have to be professionally managed and developed, different techniques are appropriate for different types of system. For example, games should always be developed using a series of prototypes whereas safety critical control systems require a complete and analysable specification to be developed. You can't, therefore, say that one method is better than another.

10. What differences has the Web made to software engineering?

The Web has led to the availability of software services and the possibility of developing highly distributed service-based systems. Web-based systems development has led to important advances in programming languages and software reuse.

Importance of Software Engineering

1. Individuals and society rely on advanced software systems. We need to be able to produce reliable and trustworthy systems economically and quickly.

2. It is usually cheaper, in the long run, to use software engineering methods and techniques for software systems rather than just write the programs. For most types of systems, the majority of costs are the costs of changing the software after it has gone into use.