What is Software Engineering?

The term is made of two words, software and engineering.

Software is more than just a program code.

A program is an executable code, which serves some computational purpose.

Software is considered to be collection of executable programming code, associated libraries and documentations such as requirements, design models and user manuals.

Engineering on the other hand, is all about developing products, using well-defined, scientific principles and methods.

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.

Engineering discipline

Using appropriate theories and methods to solve problems bearing in mind organizational and financial constraints.

All aspects of software production

Not just technical process of development.

Also project management and the development of tools, methods etc. to support software production.

Software Engineering could be more accurately called Software System Engineering, it builds upon System Engineering

System engineering is concerned with all aspects of systems development including hardware, software and process engineering.

Software engineering is part of this process concerned with developing the software infrastructure, control, applications and databases in the system.

Software engineers should adopt a systematic and organized approach to their work and use appropriate tools and techniques depending on the problem to be solved, the development constraints and the resources available.

Developing software without using software engineering is like building a car by just grabbing some tools and metal and building it.

Programming ≠ Software Engineering

Importance of software engineering

More and more, individuals and society rely on advanced software systems. We need to be able to produce reliable and trustworthy systems economically and quickly.

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

What is Software Engineering?

Software Engineering: Practical application of computer science, management techniques, and other skills to: design, construct, and maintain

Definition!!!

software and its documentation.

Systematic application of methods, tools, and techniques to achieve a stated requirement or objective for software system.

Uses engineering discipline to reduce problems of late delivery, cost overruns, and failure to meet requirements.

IEEE defines software engineering as:

The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.

Software engineering is an engineering branch associated with development of software product using well-defined scientific principles, methods and procedures.

The outcome of software engineering is an efficient and reliable software product.

Software products

Software, when made for a specific requirement is called software product.

Software products may be developed for a particular customer or may be developed for a general market.

Generic: Stand-alone systems developed to be sold to a range of different customers e.g. PC software such as Excel or Word.

Custom: Software developed for a single customer according to their specification e.g. UDOM Student Record

New software can be created by developing new programs, configuring generic software systems or reusing existing software.

Product specification

Generic software products

The specification of what the software should do is owned by the software developer and decisions on software change are made by the developer.

Customized software products

The specification of what the software should do is owned by the customer for the software and they make decisions on software changes that are required.

Software engineering diversity

There are many different types of software system and there is no universal set of software techniques that is applicable to all of these.

The software engineering methods and tools used depend on the type of application being developed, the requirements of the customer and the background of the development team.

Client requirements are very different

There is no standard process for software engineering

There is no best language, operating system, platform, database system, development environment, etc.

A skilled software developer knows about a wide variety of approaches, methods, tools.

The craft of software engineering is to select appropriate methods for each project and apply them effectively.

Stand-alone applications: These are application systems that run on a local computer, such as a PC. They include all necessary functionality and do not need to be connected to a network.

Interactive transaction-based applications: Applications that execute on a remote computer and are accessed by users from their own PCs or terminals. These include web applications such as e-commerce applications.

Embedded control systems: These are software control systems that control and manage hardware devices. Numerically, there are probably more embedded systems than any other type of system.

Batch processing systems: These are business systems that are designed to process data in large batches. They process large numbers of individual inputs to create corresponding outputs.

Entertainment systems: These are systems that are primarily for personal use and which are intended to entertain the user.

Systems for modelling and simulation: These are systems that are developed by scientists and engineers to model physical processes or situations, which include many, separate, interacting objects.

Data collection systems: These are systems that collect data from their environment using a set of sensors and send that data to other systems for processing.

Systems of systems: These are systems that are composed of a number of other software systems.

Software engineering fundamentals

Some fundamental principles apply to all types of software system, irrespective of the development techniques used:

1. Systems should be developed using a managed and understood development process. Of course, different processes are used for different types of software.
2. Dependability (trusted) and performance are important for all types of system.
3. Understanding and managing the software specification and requirements (what the software should do) are important.
4. Where appropriate, you should reuse software that has already been developed rather than write new software.

Web-based software engineering

The Web is now a platform for running application and organizations are increasingly developing web-based systems rather than local systems (Stand-alone applications).

Web services allow application functionality to be accessed over the web.

Cloud computing is an approach to the provision of computer services where applications run remotely on the 'cloud'.

Users do not buy software but pay according to use.

Web-based systems are complex distributed systems but the fundamental principles of software engineering discussed previously are as applicable to them as they are to any other types of system.

The fundamental ideas of software engineering apply to web-based software in the same way that they apply to other types of software system.

Software reuse: Software reuse is the dominant approach for constructing web-based systems. When building these systems, you think about how you can assemble them from pre-existing software components and systems.

Incremental and agile development: Web-based systems should be developed and delivered incrementally. It is now generally recognized that it is impractical to specify all the requirements for such systems in advance.

Service-oriented systems: Software may be implemented using service-oriented software engineering, where the software components are stand-alone web services.

Rich interfaces: Interface development technologies such as AJAX and HTML5 have emerged that support the creation of rich interfaces within a web browser.

General issues that affect software

Heterogeneity: Developing techniques for building software that can cope with heterogeneous platforms and execution environments. Increasingly, systems are required to operate as distributed systems across networks that include different types of computer and mobile devices.

Business and social change: Developing techniques that lead to faster delivery of software. Business and society are changing incredibly quickly as emerging economies develop and new technologies become available. They need to be able to change their existing software and to rapidly develop new software.

Security and trust: As software is intertwined with all aspects of our lives, it is essential that we can trust that software.

Scale: Software has to be developed across a very wide range of scales, from very small embedded systems in portable or wearable devices through to Internet-scale, cloud-based systems that serve a global community.

Software Engineering: A Problem Solving Activity

Analysis: Understand the nature of the problem and break the problem into pieces

Problem cannot be solved without understanding it first

Software should be written for people first: To nurture software, people must be able to understand it

Synthesis: Put the pieces together into a large structure