

1

Chapter



INTRODUCTION



- After studying this chapter, the reader will be able to understand the
- Software
 - Software Characteristics
 - Attributes of Good Software
 - Software Engineering
 - Key Challenges Facing Software Engineering
 - Professional and Ethical Responsibility
 - Software Engineering Diversity
 - Internet and Software Engineering

CHAPTER OUTLINE

At present, there are a plethora of high-tech technologies along with software accessible to the people who outline the certain way we all lead our lives and also houses are constantly changing as well as increasing needs and demands. This has certainly led to a massive demand for custom software development. There are endless types of software which can be overwhelming for any person, especially who doesn't understand these different types of software as well as their users in a thorough fashion. So, what is software, and what are the types of software that exist today? Let's have a look.

SOFTWARE

A software or computer software essentially a group of programs together with configuration files and associated documentation which enable the users to perform some particular specific task or actually used to operate their computer. It essentially directs all of the peripheral devices on the entire computer system- what exactly to do and how exactly to perform a task. Software plays a key role of a mediator between the user and the computer hardware. In the absence of software, a user essentially can't perform any task on a computer. A software product development company is the one which develops software for the users.

Types of Software

Generally, there are two main classifications of software, which are namely, System Software along with the Application Software. Let's discuss them.

System Software

System software is a group of computer programs that direct the internal operations of computer system such as controlling I/O devices, managing the storage area within the computer etc. Essentially, it is software which is used to manage the behavior of the computer hardware in order to offer basic functionalities which are needed by the user. In simpler word, it can be said that system software is essentially an intermediary or even a middle layer between the user as well as the hardware. This software sanctions an environment or platform for the other software to easily work in. The purpose of the system software is to make the use of computer more efficient and easier. Hence, it is the reason why the system software is quite important in the management of the entire computer system. Whenever you turn on the computer first, it is this system software which gets initialized and then gets loaded in the system's memory. System software essentially runs in the background, and it isn't actually utilized by the end-users. Due to this reason, the system software is also known popularly as "low-level software". System software is of following types:

- a. **Operating System:** An operating system is an integrated set of programs that is used to manage the various resources and overall operation of the computer system. It also provides the basis for application programs and act as intermediary

(mediator) between users of a computer and computer hardware. It provides the user with an interface that is easy to use than bare hardware.

It is essentially a collection of software which handles resources as well as offers general services for various other applications which actually run over them. There are different types of operating systems like embedded, real-time, distributed, single-user, multi-user, mobile, internet and much more. Some of the key examples of operating systems are as follows: MS Windows macOS, Linux, iOS Android, CentOS, Ubuntu, Unix etc.

- b. **Translator (language processor):** Computer only understands the program written in machine language. Hence the program written in other language must be translated into machine language program before executing them. Translator is system software that converts the program written in other language into an equivalent machine language program before executing them. Some of the translating programs are:- compiler, interpreter and assembler.
- c. **Device Drivers:** This type of software controls particular hardware which is essentially attached to the system. Different hardware devices which require a driver to connect to a system easily consist of displays, printers, sound cards, hard disks, keyboard, and mice. Few of the examples of such drivers are: BIOS Driver Motherboard Drivers Display Drivers ROM Drivers Printer Drivers USB Drivers, Sound Card Driver, VGA Drivers etc.
- d. **Firmware:** It is actually permanent software which is embedded in the system's read-only memory. It is essentially a set of instructions which are permanently stored onto to the hardware device. It offers vital information regarding how a particular device interacts with different other hardware. Some of the examples of firmware are: Computer Peripherals Embedded Systems UEFI BIOS.
- e. **Utility Software:** Utility programs is also called system support software or service program, which provide useful services to the user of the computer. These are the programs that enhance or extend the operating system's capabilities or which simply offer new feature not provided by operating system itself. Utility software are specially used to perform routine task in the computer. These software are designed to assist in analyzing, as well as optimizing, along with configuring and maintaining a given computer system. It provides support to the computer infrastructure. Software like disk cleanup and management tools, anti-viruses, defragmenters, compression tools etc. are all utility software. Some of its examples are: Norton Antivirus, McAfee, Antivirus, WinRAR, WinZip, Windows File Explorer Directory etc.

Application Software

The software that is written by the user to solve a specific user oriented problem using the computer is known as application software. The application software may be used for official purpose of the government, engineering design, teaching aid and so on. They are also popularly known as end-user programs or even productivity programs which assist the user in completing various tasks like conducting online research, making notes, designing graphics, maintaining accounts, carrying out calculations or even playing computer games. They essentially lie above the system software. They are actually used by the end-user as well as have specific functionality or tasks which they are designed to perform. These software are often developed through custom software development, based on the requirements of the users.

There are two type of application software:-

Customized or Tailored Software

Customized or Tailored software is the software designed to meet the specific requirements of an organization or individuals. Tailored software is developed on the demand of customer by a software contractor. It serves only one user or organization. The tailored software made for one organization cannot be in another organization. It is written using a high level programming language such as C, C++, java, VB etc. For example, software for air traffic controls system, software for billing school fees etc.

Packaged Software

Packaged software is that software which is generalized set of programs designed and developed for general purpose. These are the software which are produced by development organization and sold on the open market to any customer who is able to buy them. It is also called universal software as it can be used by users and organizations all over the world. Example of this type software includes; word processing package, spreadsheet package, database package etc.

There is a variety of application software. Some of them are:

1. **Word Processors:** Such applications are meant for documentation. It also assists in storing as well as formatting and even printing of the documents. Key examples of such software are: MS Word, WordPerfect, Google Docs etc.
2. **Database Software:** It is used to create as well as manage a database and also known as Database Management System or in short, DBMS. Such software assists in the data organization. Some of the examples of DBMS are: MS Access FileMaker dBase Clipper MySQL FoxPro
3. **Multimedia Software:** This is software which is able to play, create as well as record images, audio or even video files. This type of software are utilized for animation, video editing, graphics as well as image editing. Due to the high demand for such software, every software product development company has vast avenues in developing them. Some of the examples of such software are: Adobe Photoshop, VLC Media Player, Windows Media Player, Windows Movie Maker etc.

4. **Web Browsers:** These are the software utilized to browse the internet. Web browsers assist the users in locating as well as retrieving data well across the web. Some of the key examples of them are: Google Chrome, Mozilla Firefox, Internet Explorer, Opera, UC Browser, Safari etc.

However, there also exists another classification of the software. They can easily be classified on the basis of their availability as well as sharability. Their classification is as below:

1. **Freeware:** These software are available free of cost. A user can easily download them from the internet and can easily use them without paying any charges or fees. However, they don't provide any type of liberty to modify the entire software or charging a fixed fee for its distribution. Examples of this software are: Adobe Reader, Skype Team Viewer, Yahoo Messenger etc.
2. **Shareware:** This software is distributed freely to users on a fixed trial basis. It generally comes with a set time limit, and on the expiration of the time limit, the user is finally asked to pay a fixed fee for the continued services. Few of the examples of shareware are: Adobe Acrobat, PHP Debugger, WinZip etc.
3. **Open-source software:** Such types of software are usually available to users along with their source code which means that the user can easily modify and distribute the software as well as add additional features to them. They can either be chargeable or free. Few of the examples of such software are: Mozilla Firefox Thunderbird, Moodle, and Apache Web Server

Web based and Mobile Application Software

Application software can be computer based, Web based and mobile application. Computer based application software are those application software that are installed on local computer. In the upcoming section we are explaining web based and mobile application.

Web Based Application

A web-based application (also called web app) is application software that is accessed over a network connection using Hyper Text Transfer protocol (HTTP), rather than existing within a device's memory. That is web app runs on a web server unlike computer-based software programs that are stored locally on the Operating system (OS) of the device. Web applications web-based applications also may be client-based, where a small part of the program is downloaded to a user's desktop, but processing is done over the internet on an external server.

Web apps includes light applications like Flash games, online calculators, calendars, generating color palettes, displaying currency exchange rates and so on, as well as more intensive applications such as web-based word processors and spreadsheet applications.

Advantages of web based application

- **Flexible access:** Web based application can be accessed from any place and any computers that have a web browser and active internet connection.
- **Platform independent:** Developers don't have to write multiple versions of the same application for different operating systems and machines.
- **Storage Increase:** With the availability of the cloud, storage space is virtually infinite.
- **Vendors are responsible for backup:** users do not have to worry for backup.
- **With web:** Based apps, users don't have to install additional software.

Disadvantages of web based application

- Users can lose access to critical data due to Internet outage or down of server.
- Slower data entry in web based application than computer-based application.
- Security - Someone else holds or controls your data.
- More expensive over time - due to needs on Internet.
- Browser Support - Unfortunately, we don't all use the same browser. This means during development you'll need to ensure your app is supported across a variety of browsers

Mobile Application

A mobile application, most commonly referred to as mobile app or simply app, is a type of application software designed to run on a mobile device, such as a smartphone or tablet or watch. Mobile applications frequently serve to provide users with similar services to those accessed on PCs. Originally mobile apps were intended for productivity assistance such as email, calendar, contact database, but due to public demands apps are expanded to other areas such mobile games, mobile wallet, GPS, order-tracking, ticket purchasing, mobile banking etc. so that there are millions of apps available.

Apps are generally distributed from application distribution platforms called **app store**, so users have to download apps from **app store**. The three biggest app stores are **Google Play** - an online software store for Android devices, **App Store** for iOS, and **Microsoft Store** for Windows 10, Windows 10 mobile platform. Examples of other app store are: **Amazon app store**, **Blackberry World**, **Samsung Apps** etc.

There are different categories of apps. Some major categories of apps with examples are:

1. **Game Apps:** Pokemon Go, PUBG mobile, Temple Run2 etc.
2. **Business App:** Adobe acrobat reader, Indeed job search, Facebook pages manager etc.
3. **Educational app:** Duolingo (learn language for free), Lumosity (brain training), YouTube etc.

4. **Entertainment apps:** Netflix, NetTV, JOOX etc.
5. **Travel apps:** Google map, Airbnb etc.
6. **Utility:** Reminder, calculator, flash light, weather etc.

Advantage of Mobile Application

- **Speed:** Apps usually have faster loading times than web pages as they aren't focus with unnecessary site elements. Also the user of web based application need to open browser, enter URL.
- **Ability to Work Offline:** Although require Internet to complete most of the jobs; it give basic functionality in offline mode.
- **Security:** With the mobile application, app owners can verify **real users** by Fingerprint scan, retina identification, and biometrics.
- The mobile application gives easy **User Interface** and better **User Experience**.
- **Mobile Apps:** Give better personalization users can customize the application based on their interest.
- **Functionality:** Apps can easily integrate with other features and tools in a device as well as support functions that aren't possible with responsive website.

Disadvantages

- Need to develop different apps for different types of mobile devices – separate apps for Android and iOS.
- High cost and lengthy development period.
- Security issue due to use of personal information when using mobile apps.
- Mobile apps developer needs the knowledge of different programming languages.

Differences between system software and application software

System software	Application software
1. System software is a group of programs that direct the internal operations of computer system such as controlling I/O devices, managing the storage area within the computer etc.	1. The software that is written by the user to solve a specific user oriented problem using the computer is known as application software..
2. Generally users do not interact with system software.	2. Generally users interact with application software.
3. System software runs independently.	3. Application can't run without the present of the system software.
4. System software perform several task..	4. Application software perform specific task.
5. It is of three types <ul style="list-style-type: none"> • Operating system • Language processor • Utility software Example: Windows XP, Linux.	5. It is of two types. <ul style="list-style-type: none"> • Tailored software • Packaged software Example: Adobe, Herosoft, Microsoft.

Differences between customized and packaged software

Customized software	Packaged software
1. Designed for specific purpose.	1. Designed for general purpose.
2. Developed on the demand of particular customer.	2. Developed by software companies on its own. (not on the demand of particular customer)
3. After the development, directly handover to the customer.	3. After the development, put on open market.
4. Do not have online help facility.	4. Have online help facility.
5. Do not come in updated version.	5. Frequently comes in new updated version.
6. Partially tested.	6. Well tested and verified.
7. Expensive	7. Cheaper
8. Complex to use	8. Easy to use.
9. Example: Microsoft office	9. Software developed for billing school fee.

SOFTWARE CHARACTERISTICS

As we know that software is any computer program which can also be defined as a set of instructions which are responsible for guiding the computer to perform certain tasks. The following are the **characteristics of software**:

- Software does not wear out
- Software is not manufactured
- Usability of Software
- Reusability of components
- Flexibility of software
- Maintainability of software
- Portability of software
- Reliability of Software

- **Software does not wear out:**
Different things like clothes, shoes, ornaments etc. do wear out after some time. But, software once created never wears out. It can be used for as long as it is needed and in case of need for any updating, required changes can be made in the same software. Consequently then it can be used further with updated features.

- **Software is not manufactured**
Software is not manufactured but is developed. So, it does not require any raw material for its development.

- **Usability of Software**

The usability of the software depends upon the simplicity of the software in terms of the user. The easier the software is to use for the user, depends on its usability as more number of user will be able to user it more efficiently.

- **Reusability of components**

As the software never wears out, neither does its components, i.e. code segments. So, if any particular segment of code is required in some other software, we can reuse the existing code form the software in which it is already present. This reduces your work and also saves time and money.

- **Flexibility of software**

Software is flexible. We can make necessary changes in software in the future according to the need and usability of same software.

- **Maintainability of software**

Every software is maintainable. It means that if any errors or bugs appear in the software, then they can be fixed without any undue effort.

- **Portability of software**

Portability of the software means that you can transfer software from one platform to another easily. Due to this, the sharing of the software among the developers and other members can be done flexibly.

- **Reliability of software**

This is the ability of the software to provide the desired functionalities under every condition. This means that software should work properly in each condition whatever user expects.

ATTRIBUTES OF GOOD SOFTWARE

The characteristics of any software product include features which are displayed by the product when it is installed and put in use. They are not the services which are provided by the product. Instead, they have related to the products dynamic behavior and the use made of the product. So, apart from the services that the software provides, software products have a number of other associated attributes that reflect the quality of the software. The software should deliver the required functionality and performance to the uses and should be maintainable, dependable efficient and usable. These attributes are not directly concerned with what the software does, they reflect its behavior while it is executing. The essential attributes of good software are as follows:

1. **Maintainability:** Software must evolve to meet changing need of customer. It is called maintainability. So, software should be written in such a way that, it may evolve to meet the changing needs of customer. This is a critical attribute because software change is an inevitable consequence of a changing business environment.

2. **Dependability:** Dependability of software is a property of Software that reflects its trustworthiness. It is the degree of confidence a user has that the system will operate as they expect and the system will not fail in normal use. Dependability has a range of characteristics including availability, reliability, security and safety.
- **Availability:** The ability of the system to deliver services when requested by the user is called availability. It is the probability that it will be up and running and able to deliver useful services to users at any given time.
 - **Reliability:** The ability of the software to deliver services as specified is called reliability. It is a probability over a given period of time, that the software will correctly deliver services as expected by the user.
 - **Safety:** The ability of software to operate correctly without failure is called safety. It is the judgment of how likely it is that the system will cause damage to people or its environment.
 - **Security:** The ability of software to protect itself against accidental or unauthorized users is called security. It is judgment of how likely it is that the system can resist accidental or deliberate intrusions.
3. **Efficiency:** Software should not make wasteful use of system resources such as memory, processor cycle. So, efficiency means, responsiveness, processing time, memory utilization.
4. **Usability:** Software must be usable without on due effort, by the type of user for whom it is designed It means, it should have adequate documentation and appropriate interface.

SOFTWARE ENGINEERING

The term Software Engineering is composed of two words, software and engineering. Software is a program code or set of executable codes containing instructions that give desired functionality. Engineering is the process of designing and constructing something which assists a particular purpose and seeks a cost-effective solution to issues. So, Software engineering is an engineering discipline which is concerned with all aspect of software production from the early stage of requirement specification to maintaining the software after it has gone into use. Software engineering was introduced to address the issues of low-quality software projects. Problems arise when software generally exceeds timelines, budgets, and reduced levels of quality. It ensures that the application is built consistently, correctly, on time and on budget and within requirements.

- a. **Engineering discipline:** Engineers make the thing work by applying theories, methods and tools where these are appropriate. They use them selectively and always try to discover solutions to problems even when there are no applicable theories and methods

- b. **All aspects of software production:** Software engineering is not just concerned with the technical processes of software development but also with activities such as software project management and with the development of tools, methods and theories to support software production.

Need of Software Engineering

The need of software engineering arises because of higher rate of change in user requirements and environment on which the software is working.

- **Large software:** It is easier to build a wall than to a house or building, likewise, as the size of software becomes large engineering has to step to give it a scientific process.
- **Scalability:** If the software process were not based on scientific and engineering concepts, it would be easier to re-create new software than to scale an existing one.
- **Cost:** As hardware industry has shown its skills and huge manufacturing has lowered down the price of computer and electronic hardware. But the cost of software remains high if proper process is not adapted.
- **Dynamic Nature:** The always growing and adapting nature of software hugely depends upon the environment in which user works. If the nature of software is always changing, new enhancements need to be done in the existing one. This is where software engineering plays a good role.
- **Quality Management:** Better process of software development provides better and quality software product.

Computer Science and Software Engineering

Computer science is concerned with theories and methods that underlie computers and software system. But software engineering is concerned with the practicalities of developing and delivering useful software. Some Knowledge of Computer Science is essential for software engineers in the same way that some knowledge of physics is essential for electrical engineers.

Software Engineering and System Engineering

System engineering is concerned with all aspect of the development and evolution of complex systems where software plays a major role. So, it mainly concerned with, hardware development, policy and process design and system development as well as software engineering. Software engineering is only part of this process.

Software Engineering Cost

The distribution of cost across the different activities in the software process depends on the process used and type of software that is being developed. Roughly 60 percent of costs are development costs and 40 percent are testing costs. For customer software evolution casts often exceeds development casts. Distribution of costs depends on the development model that is used.

KEY CHALLENGES FACING SOFTWARE ENGINEERING

1. The legacy challenge

The majority of software systems which are in use today was developed many years ago yet they perform critical business functions. The legacy challenge is the challenge of maintaining and updating this software in such a way that excessive costs are avoided and essential business services continue to be delivered.

2. Heterogeneity challenge

Increasingly, systems are required to operate as distributed systems across networks that include different types of computer and with different kinds of support systems. The heterogeneity challenge is the challenge of developing techniques to build dependable software which is flexible enough to cope with this heterogeneity.

3. The delivery challenge

Many traditional software engineering techniques are time-consuming. The time they take is required to achieve software quality. However, businesses today must be responsive and change very rapidly. Their supporting software must change equally rapidly. The delivery challenge is the challenge of shortening delivery times for large and complex systems without compromising system quality.

4. Trust challenge

As software is intertwined will all aspects of our lives, it is essential that we can trust that software, so, the trust challenge is to develop techniques that demonstrate that software can be trusted by its users.

5. Risk challenge

In safety-critical areas such as space, aviation, nuclear power plants, etc. the cost of software failure can be massive because lives are at risk. Dealing with the increased complexity of software need for new applications.

PROFESSIONAL AND ETHICAL RESPONSIBILITY

The Software Engineering Professional Practice concerned with the knowledge, skills, and attitudes that software engineers must possess to practice software engineering in a professional, responsible, and ethical manner. Because of the widespread applications of software products in social and personal life, the quality of software products can have profound impact on our personal well-being and societal harmony. Software engineers must handle unique engineering problems, producing software with known characteristics and reliability. This requirement calls for software engineers who possess a proper set of knowledge, skills, training, and experience in professional practice.

The term "professional practice" refers to a way of conducting services so as to achieve certain standards or criteria in both the process of performing a service and the end product resulting from the service. These standards and criteria can include both technical and nontechnical aspects. The concept of professional practice can be viewed as being more applicable within those professions that have a generally accepted body of knowledge; codes of ethics and professional conduct with penalties for violations; accepted processes for accreditation, certification, and licensing; and professional societies to provide and administer all of these. Software engineering involves wider responsibilities than simply the application of technical skills. They must behave in an honest and ethically responsible way if they are to be respected as professionals. Ethical behavior is more than simply upholding the law. Like other engineering discipline, software engineering is carried out within a legal and social framework that limits the freedom of engineers. They must behave in an ethical and morally responsible way if they are to be respected as professionals.

1. **Confidentiality:** Software engineers must have to follow the confidentiality of the organization, their current projects and products, information of their employees and other things which are included under the agreement of job by the employer.
2. **Competence:** This is the major point which we have to follow being a professional. Software engineers should never misrepresent your skills and the level of competency. Some persons try to show off their knowledge with the manipulation of contextual words. It will benefit you initially but in the long run you will get caught. So never accept any work which is out of your competency.
3. **Intellectual property rights:** You must have the knowledge of the local governing law regarding the software import and export. Intellectual property is a very serious crime and sensitive issue. So never try to reuse property of any other organization to extend your own product without giving them credit or until they will allow you to use their copyright product. One should be aware of local laws governing the use of intellectual property such as patent and copyright. We should be careful to ensure that intellectual property of employers and client is protected.
4. **Computer Misuse:** In the field of computer science, engineers try to misuse their systems which are provided by the employer to harm the integrity of organization. It can be a cause of lawsuit against you to illegally use property of the organization without their permission. For an example if organization is not allowing you to play video games upon the personal system and you have installed the game that it will become a cause of computer misuse.

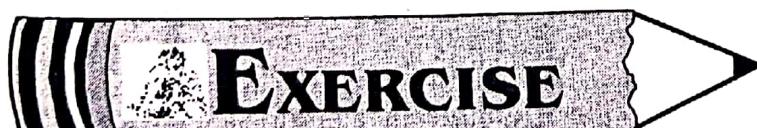
SOFTWARE ENGINEERING DIVERSITY

Software engineering is a systematic approach to development of software that takes into account practical cost, schedule and dependability issues. There are no universal software engineering methods and techniques that are suitable for all systems and all companies; rather diverse set of methods and tools has evolved. There are many different types of application diversity.

1. **Standalone Applications:** These are applications that run on a local computer such as PC; do not need to be connected to network. E.g. office package, CAD, Photo manipulation etc.
2. **Interactive transaction based application:** These are applications that execute on a remote computer and that are accessed by users from their own PCs or terminals. They include business applications such as e-commerce applications.
3. **Embedded control system:** These are software control systems that control and manage hardware devices. Example: software in mobile phones, software in microwave oven system to control cooking process.
4. **Batch processing system:** these are business systems that are designed to process data in large batches. Example: phone billing systems, salary payment systems.
5. **Entertainment system:** these are systems primarily for personal use and entertain the user. Example: Gaming software.

INTERNET AND SOFTWARE ENGINEERING

The internet is now a platform for running application and organizations are increasingly developing web applications rather than local system. Internet service allows application functionality to be accessed over the internet. With the help of internet, instead of writing software and deploying it on users PC, the software can be developed on web server that can be accessed through browsers. This made it much cheaper to change and upgrade the software as there was no need to install the software on every PC. Now a day you get every services using cloud computing technology. This is an approach to the provision of computer services where applications run remotely on the cloud. In cloud computing users do not buy software but pay according to how much the software is used or are given free access in return.



1. What is software? Explain different types of software with example.
2. Differentiate between generic and bespoke software product.
3. Explain the attributes of good software.
4. What is software engineering? How does it different from computer science?
5. What is software process? Differentiate between software engineering and system engineering.
6. What is CASE? Explain the importance of CASE tools in software development life cycle.
7. Explain challenges facing software engineering along with professional and ethical responsibilities.
8. Why software testing costs are high for generic software product? Explain.
9. What do you mean by software engineering diversity? Explain with example.

