Object Oriented Software Engineering (OOSE)

22CS017

Introduction to Software Engineering

Index

- What is Software
- Hardware Vs Software
- The Evolving Role of Software
- The necessity of Software Evolution
- Characteristics of software
- Changing Nature of Software
- Classification of the Software
- Introduction of Software Engineering
- Key Principles of Software Engineering
- Practice Questions

What is Software

Software is a set of instructions, data or programs used to operate computers and execute specific tasks.

- Software is the opposite of hardware, which describes the physical aspects of a computer.
- Software is a generic term used to refer to applications, scripts and programs that run on a device.
- Software can be thought of as the variable part of a computer, while hardware is the invariable part.

There are two main categories of software are **application software and system software**.

- An **application** is software that fulfills a specific need or performs tasks.
- **System software** is designed to run a computer's hardware and provides a platform for applications to run on top of.

Other types of software include programming software, which provides the programming tools software developers need; middleware, which sits between system software and applications; and driver software, which operates computer devices and peripherals.

Hardware vs Software

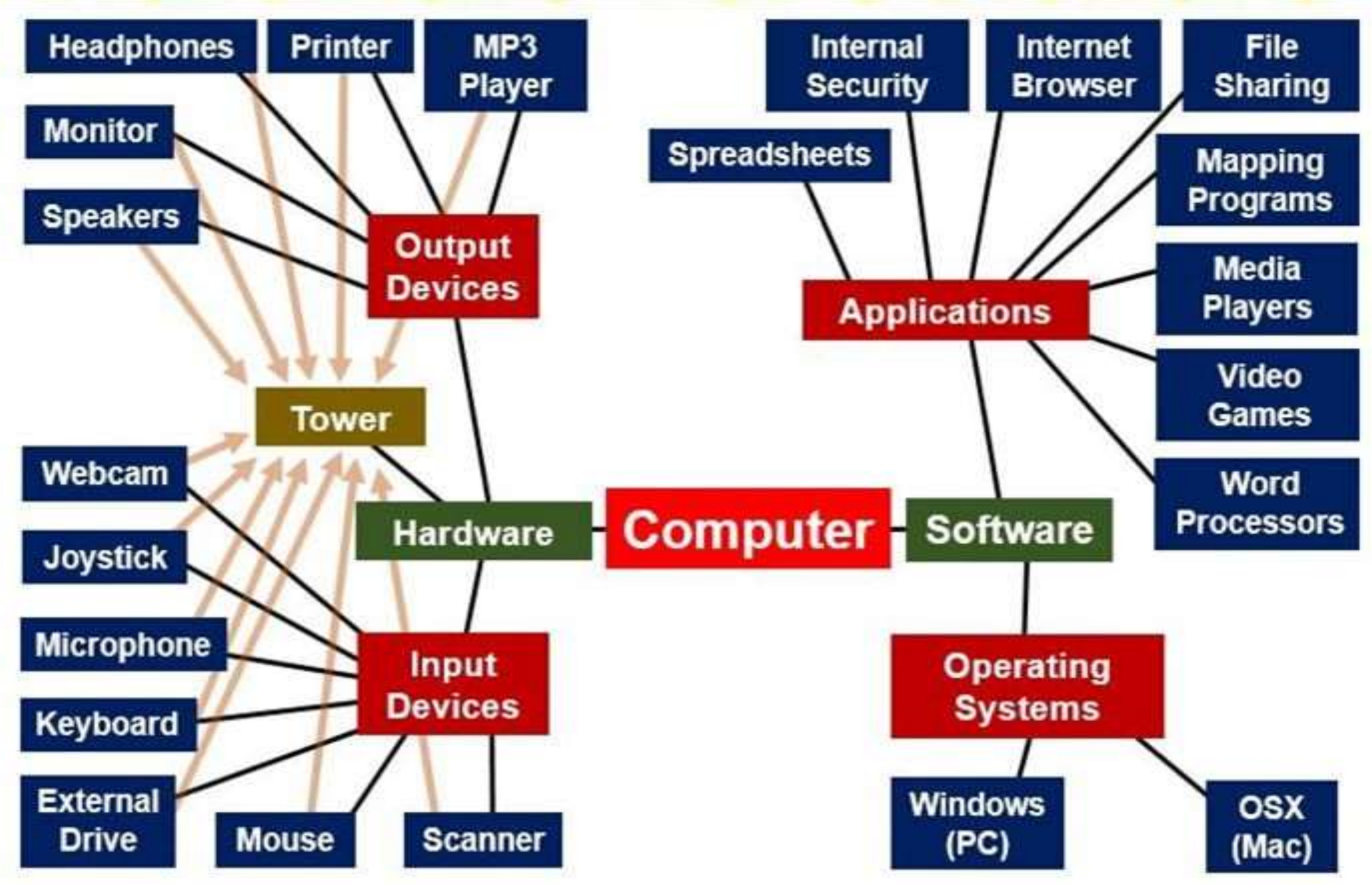


Figure 1: Hardware vs Software

The Evolving Role of Software

Software Evolution is a term which refers

- to the process of developing software initially, then timely updating it for various reasons,
- to add new features
- to remove obsolete functionalities etc.

The evolution process includes fundamental activities of change analysis, release planning, system implementation and releasing a system to customers. The cost and impact of these changes are accessed to see how much system is affected by the change and how much it might cost to implement the change. If the proposed changes are accepted, a new release of the software system is planned.

During release planning, all the proposed changes (fault repair, adaptation, and new functionality) are considered. A design is then made on which changes to implement in the next version of the system. The process of change implementation is an iteration of the development process where the revisions to the system are designed, implemented and tested.

The necessity of Software Evolution

a) Change in requirement with time: With the passes of time, the organization's needs and modus Operandi of working could substantially be changed so in this frequently changing time the tools(software) that they are using need to change for maximizing the performance.

b) Environment change: As the working environment changes the things(tools) that enable us to work in that environment also changes proportionally same happens in the software world as the working environment changes then, the organizations need reintroduction of old software with updated features and functionality to adapt the new environment.

c) Errors and bugs: As the age of the deployed software within an organization increases their preciseness or impeccability decrease and the efficiency to bear the increasing complexity workload also continually degrades. So, in that case, it becomes necessary to avoid use of obsolete and aged software. All such obsolete Softwares need to undergo the evolution process in order to become robust as per the workload complexity of the current environment.

The necessity of Software Evolution

d) Security risks: Using outdated software within an organization may lead you to at the verge of various software-based cyberattacks and could expose your confidential data illegally associated with the software that is in use. So, it becomes necessary to avoid such security breaches through regular assessment of the security patches/modules are used within the software. If the software isn't robust enough to bear the current occurring Cyber attacks so it must be changed (updated).

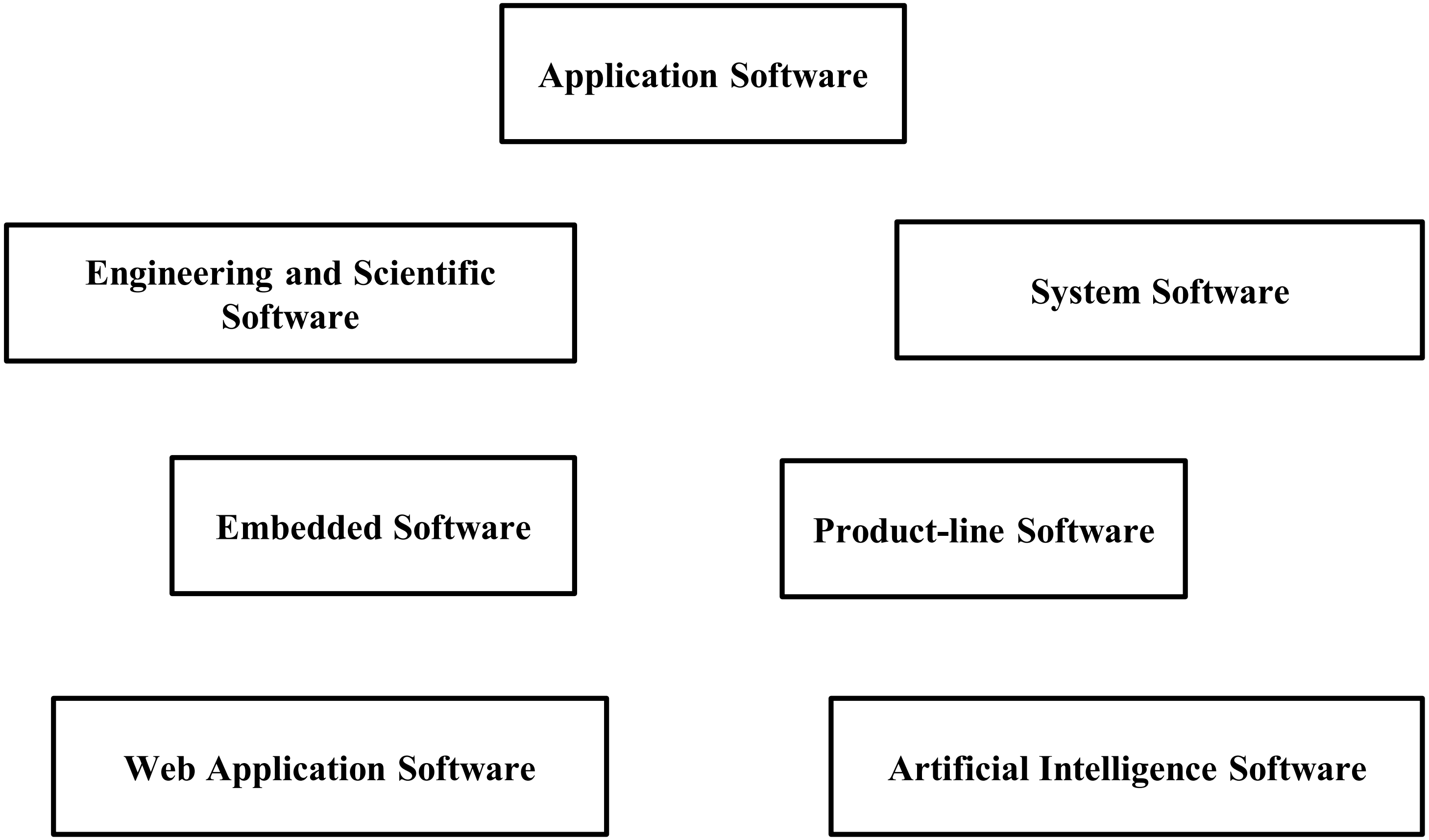
e) For having new functionality and features: In order to increase the performance and fast data processing and other functionalities, an organization need to continuously evolute the software throughout its life cycle so that stakeholders & clients of the product could work efficiently.

Characteristics / Attributes of Software

•

- **Reliability:** The capability to provide failure-free service.
- **Functionality:** The capability to provide functions that meet stated and implied needs when the software is used.
- **Usability:** The capability to be understood, learned, and used.
- **Efficiency:** The capability to provide appropriate performance relative to the amount of resources used.
- **Maintainability:** the capability to be modified for purposes of making corrections, improvements, or adaptations.
- **Portability:** The capability to be adapted for different specified environments without applying actions or means other than those provided for this purpose in the product.

Changing Nature of Software



Application Software

Engineering and Scientific Software

System Software

Embedded Software

Product-line Software

Web Application Software

Artificial Intelligence Software

Figure 2 : Changing Nature of Software

Changing Nature of Software

- **Application Software:** Application software is defined as programs that solve a specific business need. Application in this area process business or technical data in a way that facilitates business operation or management technical decision making. In addition to convention data processing application, application software is used to control business function in real time.

- **Engineering and Scientific Software:** This software is used to facilitate the engineering function and task. however modern application within the engineering and scientific area are moving away from the conventional numerical algorithms. Computer-aided design, system simulation, and other interactive applications have begun to take a real-time and even system software characteristic.

- **System Software:** System software is a collection of programs which are written to service other programs. Some system software processes complex but determinate, information structures. Other system application process largely indeterminate data. Sometimes when, the system software area is characterized by the heavy interaction with computer hardware that requires scheduling, resource sharing, and sophisticated process management.

Changing Nature of Software

- **Embedded Software:** Embedded software resides within the system or product and is used to implement and control feature and function for the end-user and for the system itself. Embedded software can perform the limited and esoteric function or provided significant function and control capability.
- **Product-line Software:** Designed to provide a specific capability for use by many different customers, product line software can focus on the limited and esoteric marketplace or address the mass consumer market.
- **Web Application:** It is a client-server computer program which the client runs on the web browser. In their simplest form, Web apps can be little more than a set of linked hypertext files that present information using text and limited graphics. However, as e-commerce and B2B application grow in importance. Web apps are evolving into a sophisticate computing environment that not only provides a standalone feature, computing function, and content to the end user.
- **Artificial Intelligence Software:** Artificial intelligence software makes use of a nonnumerical algorithm to solve a complex problem that is not amenable to computation or straightforward analysis. Application within this area includes robotics, expert system, pattern recognition, artificial neural network, theorem proving and game playing.

Classification of the Software

1. **Purpose:** Software can be classified as system software (e.g., operating systems, device drivers) or application software (e.g., word processors, games).

1. **Platform:** Software can be classified as native software (designed for a specific operating system) or cross-platform software (designed to run on multiple operating systems).

1. **Deployment:** Software can be classified as installed software (installed on the user's device) or cloud-based software (hosted on remote servers and accessed via the internet).

1. **License:** Software can be classified as proprietary software (owned by a single entity) or open-source software (available for free with the source code accessible to the public).

Classification of the Software

5.Development Model: Software can be classified as traditional software (developed using a waterfall model) or agile software (developed using an iterative and adaptive approach).

5.Size: Software can be classified as small-scale software (designed for a single user or small group) or enterprise software (designed for large organizations).

5.User Interface: Software can be classified as Graphical User Interface (GUI) software or Command-Line Interface (CLI) software.

What is Software Engineering

- Software engineering includes a variety of techniques, tools, and methodologies, including requirements analysis, design, testing, and maintenance.
- It is a rapidly evolving field, and new tools and technologies are constantly being developed to improve the software development process.
- By following the principles of software engineering and using the appropriate tools and methodologies, software developers can create high-quality, reliable, and maintainable software that meets the needs of its users.
- Software Engineering is mainly used for large projects based on software systems rather than single programs or applications.
- The main goal of Software Engineering is to develop software applications for improving quality, budget, and time efficiency.
- Software Engineering ensures that the software that has to be built should be consistent, correct, also on budget, on time, and within the required requirements.

Key Principles of Software Engineering

1.**Modularity**: Breaking the software into smaller, reusable components that can be developed and tested independently.

1.**Abstraction**: Hiding the implementation details of a component and exposing only the necessary functionality to other parts of the software.

1.**Encapsulation**: Wrapping up the data and functions of an object into a single unit, and protecting the internal state of an object from external modifications.

1.**Reusability**: Creating components that can be used in multiple projects, which can save time and resources.

Key Principles of Software Engineering

5.**Maintenance**: Regularly updating and improving the software to fix bugs, add new features, and address security vulnerabilities.

5.**Testing**: Verifying that the software meets its requirements and is free of bugs.

5.**Design Patterns**: Solving recurring problems in software design by providing templates for solving them.

5.**Agile methodologies:** Using iterative and incremental development processes that focus on customer satisfaction, rapid delivery, and flexibility.

5.**Continuous Integration & Deployment:** Continuously integrating the code changes and deploying them into the production environment

Practice Questions

- How is System Software classified?
- What are the five functions of the Software?
- What is the main difference between a computer program and computer software?
- What is computer software?
- What is mean by software scope?

Bibliography

- http://www.itinfo.am/eng/software-development-methodologies/
- https://www.tutorialspoint.com/software_engineering/software_design_strategies.htm
- https://www.softwaretestinghelp.com/types-of-software-testing/
- https://www.tutorialspoint.com/software_testing_dictionary/alpha_testing.htm
- https://www.tutorialspoint.com/software_testing_dictionary/validation_testing.htm
- https://www.tutorialspoint.com/software_testing_dictionary/acceptance_testing.htm

happy

LEARNING