Republic of Iraq
Ministry of Higher Education
& Scientific Research
University Of Anbar
College of Computer Sciences
& Information Technology
Department of Computer Science

Lecturer Name: Dr. Shihab Hamad Khaleefah
Academic Status: BhD. In Computer Science
Qualification: - Senior Lecturer
Course Material: Software Engineering
Course Code: CSDC313
Grade Level: Third Stage
Lecture: 1-2

OVERVIEW TO THE MAIN CONCEPT IN SOFTWARE ENGINEERING

1. Introduction to the Main Concepts of Software
   - 1.1 The kinds of software products
   - 1.2 The Attributes of Good Software
2. What is System Software?
   - 2.1 System software
   - 2.2 Types of System Software
   - 2.3 Features of System Software
3. What is Application Software?
   - 3.1 Application Software
   - 3.2 Types of Application Software
   - 3.3 Features of Application Software
   - 3.4 The Difference Between System Software and Application Software
4. What is Software Engineering?
5. Importance of Software Engineering
6. What is the Difference Between Software Engineering and Computer Science?
7. Software Process
   - 7.1 Fundamental Software Engineering Process Activities
8. Selecting Suitable Software Process Model
9. Types of Application
10. Why Do We Need Software Engineering?

Republic of Iraq
Ministry of Higher Education
& Scientific Research
University Of Anbar
College of Computer Sciences
& Information Technology
Department of Computer Science

Lecturer Name: Dr. Shihab Hamad Khaleefah
Academic Status: BhD. In Computer Science
Qualification: - Senior Lecturer
Course Material: Software Engineering
Course Code: CSDC313
Grade Level: Third Stage
Lecture: 1-2

Introduction to the Main Concepts of Software

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

The opposite of software is the **Hardware** which describes the physical aspects of a computer.

Figure 1: Software and hardware devices components.

There are two kinds of software products:

Generic software products: Software that are produced by a **development organization** and sold on the open market to any customer who is able to buy them.

◦ **Example:** software for PCs such as databases, word processors, drawing packages, and projectmanagement tools, library information systems, and accounting systems.

Customized software products: Software that are requested by a **particular customer**.

◦ **Example:** software for electronic devices, software written to support a particular business process, and so on.

Attributes of Good Software

Good software should deliver the required functionality and performance to the user and should be maintainable, dependable, and usable.

These attributes obviously depend on domain application. Therefore, a **banking system** must be **secure**, an **interactive game** must be **responsive**, a **telephone switching system** must be **reliable**, and so on.

Figure 2: Basic Software Quality

What is System Software?

System software, is software that **directly operates the computer hardware and provides the basic functionality to the users** as well as to the other software to operate smoothly.

In other words,

System software basically **controls a computer's internal functioning and also controls hardware devices** such as monitors, printers, and storage devices.

It is like an **interface between hardware and user applications**, it helps them to communicate with each other because hardware understands machine language (i.e. 1 or 0) whereas user applications are work in human-readable languages like English, and Arabic.

Figure 3: System software interface.

Types of System Software

1. **Operating System:** It is the main program of a computer system, it **manages all the resources** such as computer memory, CPU, printer, and hard disk, **and provides an interface to the user**, which helps the user to interact with the computer system. It also provides various services to other computer software. **Examples of operating systems are Linux, Apple macOS, and Microsoft Windows**.
2. **Language Processor:** The system software **converts the human-readable language into a machine language** So, the conversion is done by the language processor. It **converts programs written in high-level programming languages like Java, C, C++, Python, etc (known as source code), into sets of instructions that are easily readable by machines (known as object code or machine code).**
3. **Device Driver:** A device driver is a **program or software that controls a device and helps that device to perform its functions**. Every device **like a printer, mouse, modem, etc. needs a driver to connect with the computer system eternally**.

Features of System Software

1. System Software is **closer to the computer system**.
2. System Software is **written in a low-level language in general**.
3. System software is **difficult to design and understand**.
4. System software is **fast in speed (working speed)**.
5. System software is **less interactive to the users in comparison to application software**.

What is Application Software?

Application Software, is the software that **performs special functions or provides functions that are much more than the basic operation of the computer** is known as application software.

In other words, **application software is designed to perform a specific task for end-users**. It is a product or a program that is designed only to fulfil end-users' requirements. It includes **word processors, spreadsheets, database management, inventory, payroll programs, etc**.

Figure 4: Samples of different applications software interface.

Types of Application Software

There are different types of application software:

1. **General Purpose Software:** This type of application software is **used for a variety of tasks and it is not limited to performing a specific task only**. **For example, MS-Word, MS-Excel, PowerPoint.**

2. **Customized Software:** This type of application software is **used or designed to perform specific tasks or functions or designed for specific organizations**. **For example, airline reservation system, and Banking services management system.**

3. **Utility Software:** This type of application software is **used to support the computer infrastructure**. It is **designed to analyze, configure, optimize and maintains the system, and take care of its requirements** as well. **For example, antivirus, disk fragmenter, memory tester, disk repair, disk cleaners, registry cleaners, disk space analyzer.**

Figure 5: Samples of different Application Software.

Features of Application Software

1. An important feature of application software is it **performs more specialized tasks** like word processing, spreadsheets, and email.
2. Mostly, **the size of the software is big, so it requires more storage space**.
3. Application software is **more interactive for the users**, so it's easy to use and design.
4. The application software is **easy to design and understand**.
5. **Application software is written in a high-level language in general**.

The Difference Between System Software and Application Software

System Software	Application Software
It is designed to manage the resources of the computer system, like memory and process management, etc.	It is designed to fulfill the requirements of the user for performing specific tasks.
Written in a low-level language.	Written in a high-level language.
Less interactive for the users.	More interactive for the users.
System software plays vital role for the effective functioning of a system.	Application software is not so important for the functioning of the system, as it is task specific.
It is independent of the application software to run.	It needs system software to run.

What is Software Engineering?

Engineering: The **Application of Science to the Solution of Practical Problems**. Also, **it is about getting results of the required quality within the schedule and budget.**

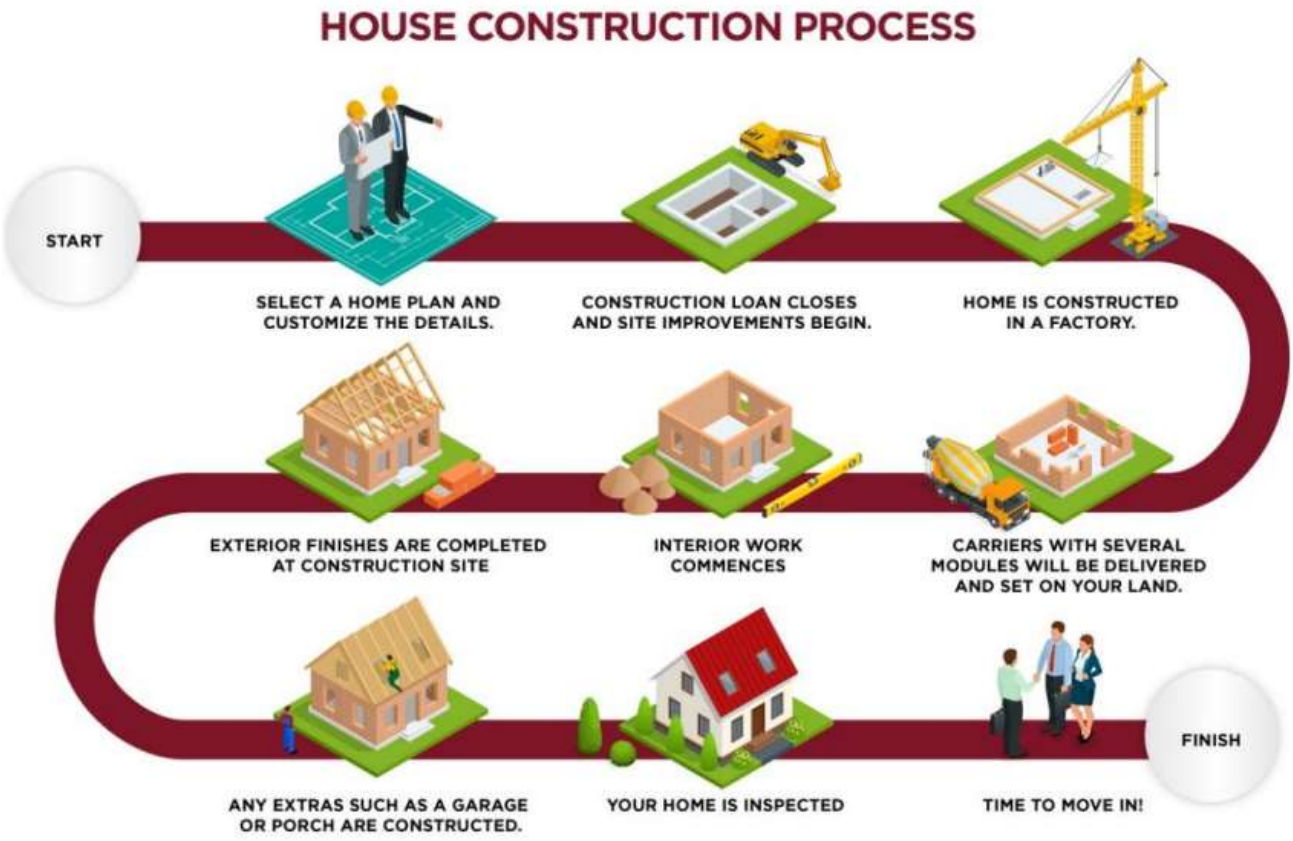


Figure 6: Samples of engineering application practical problem.

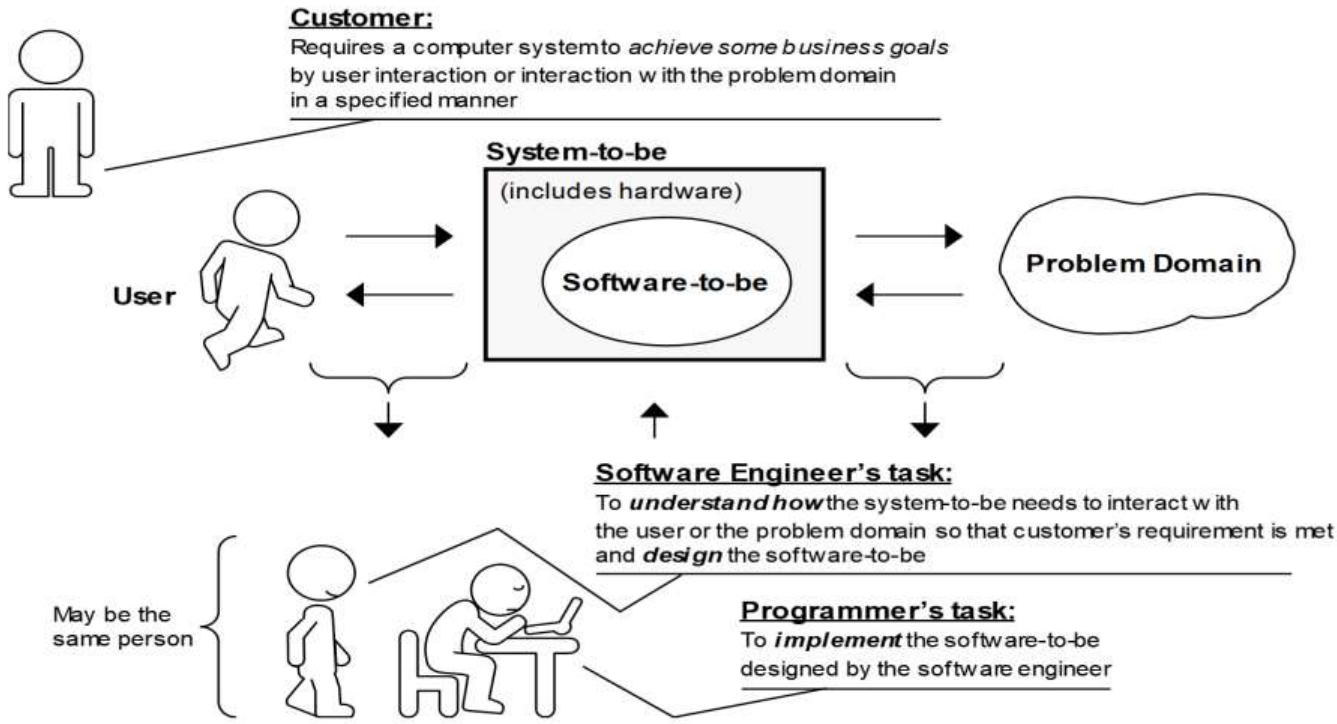
Software Engineering: The **Application of computer science to Building Practical Software Systems.**

Also, **software engineering known as a bridge from customer needs to programming implementation.**

Figure 7: Software engineering definition.

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**.

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



Importance of Software Engineering

Reduces Complexity: It **simplifies complex problems and solves those issues one by one**.

To Minimize Software Costs: Reduce the things which are not needed. As a result of the production of Software, **costs become less and more affordable**.

To Decrease Time: By following software engineering methods, it **will save your valuable time by decreasing it**.

Reliable Software: After testing and maintain, we **will ensure that the Software is secure and will work well**.

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.

Figure 8: Software engineering and computer science.

Software Process

The **systematic approach that is used in software engineering** is sometimes called a **software process**

software process, is a **set of activities whose goal is the development or evolution of software**.

Software development models or Software Development Life Cycle or (SDLC model) includes many models such as **Waterfall model, Incremental development model, Agile, Scrum, and Integration and configuration model**.

Figure 9: SDLC Fundamental processes.

Four Fundamental Software Engineering Activities:

◦ **Specification (requirements):** what the system should do and its development constraints.

◦ **Development:** where the software is designed and programmed.

◦ **Validation:** checking that the software is what the customer requires.

◦ **Evolution:** where the software is modified to reflect changing customer and market requirements.

Selecting Suitable Software Process

Choosing a suitable software process depends on the environment of the system

For example: critical software or real-time software such as an **aircraft** has to be completely specified before development begins. [***We use Waterfall Model***]

While in **e-commerce systems**, the **specification and the program are usually developed together**. [***We use Agile Model***]

The **most significant factor in determining which software engineering methods and techniques are most important** is the [**type of application that is being developed**].

Types of Application

There are many different types of application including:

1. **Stand-alone applications:** *office applications on PC.*
2. **Interactive transaction-based applications:** *web applications such as e-commerce applications.*
3. **Embedded control systems:** *systems that control and manage hardware devices such as medical devices.*
4. **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. Such as weather system*.

Why Do We Need Software Engineering?

Software engineering **allows software development organizations to produce reliable and robust software fulfilling a wide spectrum of requirements**. Today, there is a great rate of change in user requirements as well as the targeted environment, making software engineering more important than ever. **Various other reasons that have made software engineering essential when it comes to developing high-quality and reliable software products include:**

1. Cost-effectiveness

As software engineering is a systematic approach, it is **easier to estimate costs as well as to cut expenses that don't add to the overall quality of the software systems**.

2. Granting Scalability

Software systems that are not created by leveraging scientific methods and techniques suffer from scalability issues. On the other hand, **software products developed using software engineering are easier to scale**.

3. High Quality

The better the **quality of the software development process**, the **better the quality of the resulting software**.

4. Increasing Software Size

It is ok **not to use software engineering for small-size software. As the size**, however, increases, software engineering must be **employed to develop such a software system efficiently**.

Summary

In this lecture,

The student's will be able to learn about the main concepts of software which includes: kinds of good software products and its attributes. Also, will learn about the system software that includes: system software, types and the features of system software. The most important point is to understanding the application software with its types, and understanding the difference between system software and application software. These all above points led to concentrating on the software engineering, importance of software engineering, the difference between software engineering and computer science, and why do we need software engineering. Finally, software process, the fundamental software engineering process activities were described and explained in details in order to know how to select the suitable software process model for any project problem which will be explained in the next lecture.

References

- Stephens, R. (2015). Beginning software engineering. John Wiley & Sons.
- Sommerville, I. (2016). Software engineering tenth edition.
- Foster, E., & Towle Jr, B. (2021). Software engineering: a methodical approach. Auerbach Publications.
- Tsui, F., Karam, O., & Bernal, B. (2022). Essentials of software engineering. Jones & Bartlett Learning.