SOFTWARE DEVELOPMENT

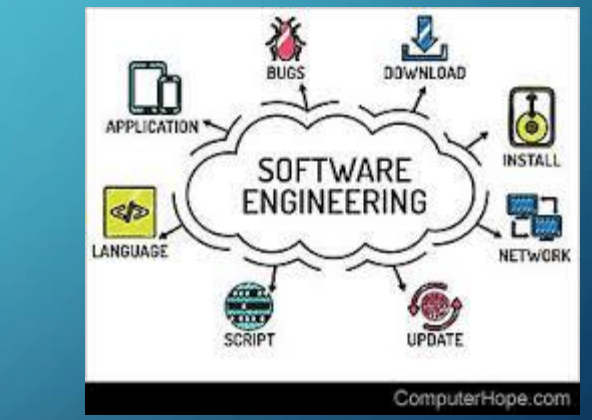
DOC. ING. MARCELA HALLOVÁ, PHD.

SOFTWARE

- ✓ **Software is a set of instructions and data that allows computers and other devices to perform specific tasks.**
  - ✓ System software – Includes operating systems (e.g., Windows, Linux, macOS) and essential programs that make a computer work. This software controls devices, manages memory, and ensures the interaction between hardware and application software.
  - ✓ Application software – Refers to programs that provide specific tools or services to users, such as web browsers, a text editor, or mobile applications. The software allows users to perform various tasks, such as document creation, data management, or internet access.

SOFTWARE ENGINEERING

- ✓Software engineering is a discipline that focuses on applying engineering principles and methodologies to the development, deployment, and maintenance of software.
- ✓It is a systematic, methodical and effective approach to software development. It covers the entire software lifecycle, from requirements analysis and specification, through architecture design and implementation, to testing, deployment, and long-term maintenance.
- ✓Software engineering uses knowledge from computer science, project management, and other fields.
- ✓The goal of software engineering is to create reliable, efficient, and maintainable software that meets the requirements of users and customers.



STAGES OF SOFTWARE DEVELOPMENT

1) Requirements Analysis

2) Proposal

3) Implementation

4) Testing

5) Deployment

6) Maintenance

1) REQUIREMENTS ANALYSIS

- ✓ Requirements analysis involves identifying all stakeholders, collecting their requirements through various methods such as interviews, questionnaires, document analysis, etc.
  - ✓ **Functional requirements** – determine what the software should do (e.g. the software allows the user to register by entering an email address and password, the manager can view issued invoices, etc.).
  - ✓ **Non-functional requirements** – relate to qualitative aspects of the software, such as performance, security and usability (e.g. card verification is carried out within two seconds, the software is supposed to be scalable, compatible with other systems, etc.).

2) PROPOSAL

- ✓The main goal is to transform the requirements into a description of the technical solution and specifications for the software being developed.
- ✓In this phase, the plan and architecture of the software solution are created so that the system is efficient, sustainable and easily expandable.
- ✓Decisions are also made about the technologies, programming language, and overall architecture of the system.
- ✓For better visualization, specification and understanding of software design, UML (Unified Modeling Language) diagrams are often used, which show the structure and functions of the software.
- ✓An important part is the design of the user interface (UI), which should be intuitive and user-friendly.

3) IMPLEMENTATION

- ✓The implementation phase in software development is the process of translating a software design into functional code that can be executed by machines (computers, smartphones…)
- ✓At this stage, programming languages, development environments (IDEs), frameworks (a predefined set of tools, components and rules that provides the structure and basis for the development of software applications, thus speeding up and simplifying the implementation process) and libraries (files of reusable code that provide functions, methods and tools to solve common tasks, thus facilitating development and reducing the need to write code from scratch) are used to implement defined functionalities of the system.
- ✓A high-quality implementation requires adherence to coding standards, clean code principles, and design patterns that increase the readability and efficiency of code creation.

4) TESTING

- ✓Software testing is a key step in the process of ensuring the quality and reliability of the software product being developed.
- ✓Testing focuses on identifying errors and flaws in the code, functionality, or behaviour of the system, while observing whether the application meets defined requirements and standards.
  - ✓**Unit tests** – verify the functionality of individual components.
  - ✓**Integration tests** – focus on the interaction between different modules.
  - ✓**System tests** – evaluate the overall functionality of the system in real conditions.

5) DEPLOYMENT

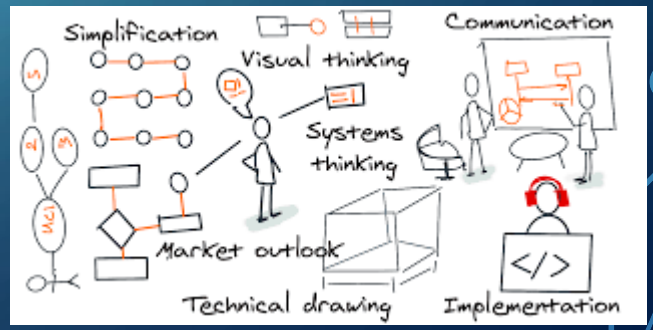
- ✓Deployment is the software development phase in which the software being developed is moved from a development or test environment to production, where it will be made available to end users.
- ✓This process includes not only the actual running of the software on the production servers, but also the other steps required to get it fully operational, such as system configuration, database setup, implementation of security measures, and integration with other systems.
- ✓The goal of deployment is to ensure that the software works reliably and efficiently in a real-world environment and meets user requirements.

6) MAINTENANCE

- ✓Maintenance includes all the activities necessary to maintain, optimize, and improve a system throughout its lifecycle.
- ✓At this stage, bug fixes are carried out that appear after deployment in the real environment, as well as software updates to consider changes in user requirements or technological conditions.
- ✓Maintenance may also include performance improvements, modifications to improve system security, and the implementation of new features that are necessary to maintain competitiveness or meet new regulatory requirements.

SOFTWARE DEVELOPER TEAM AND THEIR ROLES

- ✓ **Team leader** – Is responsible for leading and coordinating the team's work. Their job is to plan and organize work, motivate team members, and ensure that the project is completed successfully. A team leader should have strong leadership skills, technical knowledge, and the ability to communicate effectively.
- ✓ **Analyst** – Is responsible for analyzing user requirements and translating them into technical specifications. Its task is to communicate with users, collect requests and propose a solution. Analysts should have good communication skills and the ability to analyze and solve problems.
- ✓ **Software architect** – His task is to design and define the technical structure and architecture of a software product so that it is scalable, sustainable and efficient. A software architect selects the appropriate technologies, design patterns, and components to ensure optimal system performance, security, and compatibility. In addition to technical decisions, the software architect is also responsible for ensuring that the system meets the requirements for flexibility and extensibility in the future.



SOFTWARE DEVELOPER TEAM AND THEIR ROLES

- ✓ **Developer** – Is responsible for writing code and implementing software functionality. The team can include different developers specializing in different areas, such as frontend developers, backend developers, or database specialists. Developers should have good programming skills and the ability to work in a team.

- ✓ **Tester** – Is responsible for testing the software and identifying bugs. Its task is to carry out different types of tests and document the results. Testers should have good analytical skills and be able to work with testing tools.

PROGRAMMING

- ✓Programming is the process of creating and implementing algorithms through programming languages to control the operation of computer systems and various devices.
- ✓Programming is the process of creating instructions for a computer to perform required tasks.
- ✓These instructions are written in programming languages that are understandable to the computer.
- ✓Programming languages are like human languages, but with precise rules and syntax.
- ✓The entire process includes logic design, syntactic and semantic correctness of the code, performance optimization, and ensuring the reliability of the resulting solution.

THE IMPORTANCE OF LEARNING TO PROGRAMMING

- ✓ Programming is one of the most valuable skills in the digital era, as it provides a wide range of applications in various fields, from information technology to science, finance, or healthcare.
- ✓ The ability to write code and program allows individuals to create software solutions, automate repetitive tasks and processes, leading to increased efficiency and productivity at work, industry, and many other industries.
- ✓ Regardless of the field, programming literacy is a great advantage that allows you not only to better understand modern technologies, to be able to use them better and more effectively, but also to actively participate in their development and innovation.

PROGRAMMING LANGUAGES

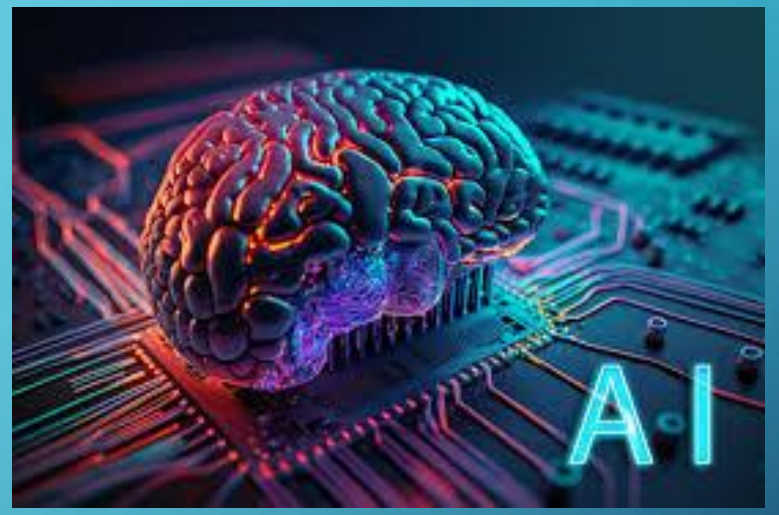
- ✓ A programming language is a formal system designed to write algorithms and instructions that a computer can run and thus control the operation of computer systems.
- ✓ An algorithm is a sequence of steps that lead to the solution of a certain problem.
  - ✓ **Low-level** – they are close to computer hardware and machine code. An example of such a programming language is Assembly.
  - ✓ **High-level** – these are more abstract and allow programmers to work with data and operations at a higher level. Examples of such programming languages are C, C++, Java, Python.

PROGRAMMING LANGUAGES

- ✓ **Python** - a high-level, versatile programming language known for its readability and simplicity. Widely adopted in data science, web development, and automation, Python's large standard library and supportive community make it a favorite for beginners and experts alike.
- ✓ **Java** - a popular, object-oriented language valued for its portability and robustness. Known for its "write once, run anywhere" capability, Java remains a top choice for enterprise-level applications and Android development.
- ✓ **C** - a foundational, low-level language that offers direct access to hardware, making it ideal for system programming. With a simple syntax and powerful features, C remains essential for operating systems and embedded systems.
- ✓ **C++** - builds on C, adding object-oriented features and libraries that make it powerful for both system and application development. Its performance and efficiency are highly valued in real-time and resource-constrained applications.
- ✓ **Structured Query Language (SQL)** - a domain-specific language used to manage data, especially in a relational database management system (RDBMS).

TRENDS AND THE FUTURE OF SOFTWARE DEVELOPMENT

- ✓ **Artificial intelligence** is currently booming and influencing the performance of activities in various industries, including programming and software development.
- ✓ The use of the so-called platforms with **no** or **minimal code** - **no-code** and **low-code** platforms. These two approaches to software development allow users without advanced programming knowledge to create applications.



THANK YOU FOR YOUR ATTENTION!