Ejercicio N° 4

Lea e interprete el siguiente texto y en clase se realizara un cuestionario

Software engineering

Software engineering is a systematic engineering approach to software development.

A software engineer is a person who applies the principles of software engineering to design, develop, maintain, test, and evaluate computer software. The term ***programmer*** is sometimes used as a synonym, but may also lack connotations of engineering education or skills.

Engineering techniques are used to inform the software development process which involves the definition, implementation, assessment, measurement, management, change, and improvement of the software life cycle process itself. It heavily uses software configuration management which is about systematically controlling changes to the configuration, and maintaining the integrity and traceability of the configuration and code throughout the system life cycle. Modern processes use software versioning.

Definitions and terminology controversies

Notable definitions of software engineering include:

- "The systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software"—The Bureau of Labor Statistics—IEEE *Systems and software engineering – Vocabulary*
- "The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software"—IEEE *Standard Glossary of Software Engineering Terminology*
- "an engineering discipline that is concerned with all aspects of software production"—Ian Sommerville
- "the establishment and use of sound engineering principles in order to economically obtain software that is reliable and works efficiently on real machines"—Fritz Bauer
- "a branch of computer science that deals with the design, implementation, and maintenance of complex computer programs"—Merriam-Webster
- "'software engineering' encompasses not just the act of writing code, but all of the tools and processes an organization uses to build and maintain that code over time. [...] Software engineering can be thought of as 'programming integrated over time.'"—Software Engineering at Google

The term has also been used less formally:

- as the informal contemporary term for the broad range of activities that were formerly called computer programming and systems analysis.
- as the broad term for all aspects of the practice of computer programming, as opposed to the theory of computer programming, which is formally studied as a sub-discipline of computer science

- as the term embodying the advocacy of a specific approach to computer programming, one that urges that it be treated as an engineering discipline rather than an art or a craft, and advocates the codification of recommended practices.

Etymology of "software engineer"

Margaret Hamilton promoted the term "software engineering" during her work on the Apollo program. The term "engineering" was used to acknowledge that the work should be taken just as seriously as other contributions toward the advancement of technology. Hamilton details her use of the term:

When I first came up with the term, no one had heard of it before, at least in our world. It was an ongoing joke for a long time. They liked to kid me about my radical ideas. It was a memorable day when one of the most respected hardware gurus explained to everyone in a meeting that he agreed with me that the process of building software should also be considered an engineering discipline, just like with hardware. Not because of his acceptance of the new "term" per se, but because we had earned his and the acceptance of the others in the room as being in an engineering field in its own right.

Suitability of the term

Individual commentators have disagreed sharply on how to define software engineering or its legitimacy as an engineering discipline. David Parnas has said that software engineering is, in fact, a form of engineering.

Steve McConnell has said that it is not, but that it should be.]Donald Knuth has said that programming is an art and a science. Edsger W. Dijkstra claimed that the terms software engineering and software engineer have been misused and should be considered harmful, particularly in the United States.

Tasks in large scale projects

Software requirements

Requirements engineering is about the elicitation, analysis, specification, and validation of requirements for software. Software requirements can be of three different types. There are functional requirements, non-functional requirements, and domain requirements. The operation of the software should be performed and the proper output should be expected for the user to use. Non-functional requirements deal with issues like portability, security, maintainability, reliability, scalability, performance, reusability, and flexibility. They are classified into the following types: interference constraints, performance constraints (such as response time, security, storage space, etc.), operating constraints, life cycle constraints (maintainability, portability, etc.), and economic constraints. Knowledge of how the system or software works is needed when it comes to specifying non-functional requirements. Domain requirements have to do with the characteristic of a certain category or domain of projects.

Software design

Software design is about the process of defining the architecture, components, interfaces, and other characteristics of a system or component. This is also called software architecture. Software design is divided into three different levels of design. The three levels are interface design, architectural design, and detailed design. Interface design is the interaction between a system and its environment. This happens at a high level of abstraction along with the inner workings of the system. Architectural design has to do with the major components of a system and their responsibilities, properties, interfaces, and their relationships and interactions that occur between them. Detailed design is the internal elements of all the major system components, their properties, relationships, processing, and usually their algorithms and the data structures.

Software construction

Software construction, the main activity of software development, is the combination of programming, unit testing, integration testing, and debugging. Testing during this phase is generally performed by the programmer while the software is under construction, to verify what was just written and decide when the code is ready to be sent to the next step.

Software testing

Software testing is an empirical, technical investigation conducted to provide stakeholders with information about the quality of the product or service under test, with different approaches such as unit testing and integration testing. It is one aspect of software quality. As a separate phase in software development, it is typically performed by quality assurance staff or a developer other than the one who wrote the code.

Software maintenance

Software maintenance refers to the activities required to provide cost-effective support after shipping the software product. Software maintenance is modifying and updating software applications after distribution to correct faults and to improve its performance. Software has a lot to do with the real world and when the real world changes, software maintenance is required. Software maintenance includes: error correction, optimization, deletion of unused and discarded features, and enhancement of features that already exist. Usually, maintenance takes up about 40% to 80% of the project cost therefore, focusing on maintenance keeps the costs down.