Global Journal of Engineering, Design & Technology
ISSN: 2319-7293
OPEN ACCESS Freely available online
Perspective

Software Engineering and its Associated Tasks

Peter Okechukwu*

Department of Electrical and Software Engineering, University of Calgary, 2500 University Drive NW, Alberta, Canada

DESCRIPTION

Software engineering is the methodical engineering approach to software development. A software engineer is someone who creates, maintains, tests, and reviews computer software using the principles of software engineering. Although "programmer" is occasionally used as a substitute for "engineer," this does not imply that the person has any formal engineering training or understanding.

The software development process, which also includes the definition, implementation, evaluation, measurement, management, change, and improvement of the software life cycle process itself, is guided by engineering methodologies. Software configuration management, which is concerned with stringently regulating configuration changes and upholding the integrity and traceability of the configuration and code over the duration of the system life cycle, is heavily utilized in this process. The workflows of today make use of software versioning.

Tasks in large scale projects

Software requirements: The requirements engineering process is used to elicit, analyze, specify, and validate software needs. Software requirements can be divided into three types. There are functional and non-functional requirements in addition to domain requirements. The user should be able to use the results, and the software should function correctly. Issues like portability, security, dependability, scalability, performance, reuse, and flexibility are handled by non-functional requirements.

They can be categorized into a variety of groups, including interface restrictions, performance constraints, operating restrictions, life cycle constraints, and cost constraints. Understanding the functionality of the system or programme is necessary to identify non-functional needs. We refer to the characteristics of a certain group or field of projects as the domain requirements.

Software design: Software design is the process of deciding on a system's or component's architecture, parts, interfaces, and other aspects. "Software architecture" is another name for this. The software design process is divided into three distinct phases. The three levels are interface design, architectural design, and detailed design. The phrase "interface design" describes a system's interaction with its environment. This takes place at a high degree of abstraction along with the fundamental processes of the system.

Architectural design takes into account a system's core components as well as its traits, connections, and interactions. The term "detailed designs" refers to all of the fundamental properties of the main system components, including their quality, relationships, processing, and usually their algorithms and data structures.

Software construction: The main task of software development is the creation of new software, which is done by combining coding, unit testing, integration testing, and debugging. Programmers routinely test their code as it is being created in order to assess what has just been created and to determine when the code is prepared to go to the next stage.

Software testing: To inform stakeholders about the calibre of the product or service under test, an empirical, technical research method known as software testing is used. Many strategies are employed, including unit testing and integration testing. This is one aspect of software quality. It is usually carried out as a distinct step of software development by Quality Assurance (QA) employees or a developer other than the one who wrote the code.

Software analysis: Examining how computer programmes respond in respect to a property, such as performance, robustness, or security, is the practice of software analysis. You can run it both ways: using real-time programme execution and not.

Software maintenance: The procedure required to provide reasonable support after a software product has been deployed is called maintenance. Software programmes need to be updated and modified after they are published in order to fix issues and improve performance. Because of how much the actual world affects software, maintenance is required.

Examples of software maintenance include bug fixes, optimization, the elimination of unused functionality, and enhancements to the current functions. Concentrating on maintenance reduces expenses because it often makes up 40%to 80% of project expenditures.

Correspondence to: Peter Okechukwu, Department of Electrical and Software Engineering, University of Calgary, Alberta, Canada, E-mail: peter_okechu@gmail.com

Received: 03-Mar-2022, Manuscript No. GJEDT-22-21491; **Editor assigned:** 07-Mar-2022, PreQC No. GJEDT-22-21491 (PQ); **Reviewed:** 22-Mar-2022, QC No. GJEDT-22-21491; **Revised:** 29-Mar-2022, Manuscript No. GJEDT-22-21491 (R); **Published:** 08-Apr-2022, DOI: 10.35248/2319-7293.22.11.149

Citation: Okechukwu P (2022) Software Engineering and Its Tasks in Large Scale Projects. Global J Eng Des Technol.11:149