Week 1: Interactive Assignment, Part I

The Software Engineering Processes and Concept Map

Eric Vara

The University of Arizona Global Campus

CST 301: Software Technology & Design

Professor John Howerton

January 09, 2023

**Diagram

Description automatically generatedSoftware Development Life Cycle Concept Map**

**Key Elements of Software Engineering**

Software engineering is the process of designing and building software systems using principles from other disciplines, such as management science and computer science. Despite having a sizable common area of concern, education in software engineering is fundamentally distinct from education in computer science, management science, or other constituent subjects. It is imperative that we define concepts, not merely haphazard collections of techniques, on which to build integrated software engineering programs. We suggest that computer science, management science, communication skills, problem solving, and design methodology should be included in any software engineering curriculum. This is based on our research, teaching, and real-world experience.

**Applicable Software Engineering Methodologies**

When implementing new features, teams employ the agile development technique to reduce risk. Teams create software in iterations that include tiny increments of new functionality according to all agile development methodologies. The agile development methodology comes in a variety of forms, including as scrum, crystal, extreme programming (XP), and feature-driven development.

**The Fundamentals of Software Engineering**

A software process, sometimes referred to as a software methodology, is a collection of related tasks that result in the creation of software. These tasks could include creating new software from scratch or altering an existing one. The following four actions must be included in any software process:

Software specifications define the primary software functionality and the restrictions that apply to them in the software specification (also known as requirements engineering). Software design and implementation is Programming and design of the software are required. Software validation the software must adhere to its specification and satisfy the needs of the user. Software evolution is the software being updated to reflect changes in customer and market demands. In actuality, they comprise sub-activities like unit testing, architectural design, and requirements validation.

The waterfall method separates the process activities of requirements specification, software design, development, validation, and testing into separate process phases. The strategy known as the incremental development method interleaves the processes of specification, development, and validation. The system is created in a series of versions, each of which includes functionality that was not present in the prior iteration. The procedure that depends on the availability of reusable systems or components is integration and configuration. The system development process focuses on setting up and integrating this component for use in a new environment.

**Software Process Models**

The process of gathering stakeholder wants and wishes and turning them into an accepted set of specific requirements that can be the foundation for all ensuing development operations is known as requirements engineering. Making the problem that is being expressed clear and comprehensive and ensuring that the solution is accurate, reasonable, and efficient are the two goals of requirements engineering approaches. This chapter lists the representative requirements engineering approaches that are currently in use, mostly concentrating on the fundamentals. Requirements engineering techniques are methods for transforming real-world issues into solutions for the digital world. Each approach uses a distinct process to develop the system specification as the solution, each with its own specialized way of thinking about the real-world problem.

**Work Cited**

Software engineering - quick guide. Tutorials Point. (n.d.). Retrieved January 09, 2023, from <https://www.tutorialspoint.com/software_engineering/software_engineering_quick_guide.htm>

8 best software development methodologies. app development company. (n.d.). Retrieved January 9, 2023, from <https://www.uptech.team/blog/software-development-methodologies>

Kay, R. M. (2021, April 29). How to learn the fundamentals of software engineering – in a more interesting and less painful way. freeCodeCamp.org. Retrieved January 9, 2023, from <https://www.freecodecamp.org/news/learn-the-fundamentals-of-software-engineering/>