Week 2: Interactive Assignment, Part II

Software Development and Requirements Engineering Concept

Eric Vara

The University of Arizona Global Campus

CST 301: Software Technology & Design

Professor John Howerton

January 16, 2023

**Software Development Life Cycle and Requirements Engineering**

**Diagram

Description automatically generated**

**Scrum, Agile Development Techniques and Scaling**

A team completes a predetermined amount of work during a timed period called a Scrum sprint cycle. Each sprint begins as soon as the previous one is over and lasts for two to four weeks on average. The Scrum sprint cycle is frequently described as a continuous development method.

It gives product releases a predictable work cadence and maintains the project's momentum till completion which leads to extreme programming. Extreme Programming (XP) is an agile software development paradigm that seeks to deliver software of a better caliber while also improving the team's quality of life. Of the agile frameworks, XP is the most explicit in terms of the proper engineering techniques for software development.

For small and medium sized projects that can be developed by a small co-located team, agile methodologies have shown to be effective. It is occasionally asserted that the effectiveness of these techniques is due to enhanced communication, which is achievable when everyone is cooperating. Agile methodologies need to be modified in order to handle bigger, lengthier projects with various development teams that may be spread out across different regions.

**User and System Requirements**

Users and system needs are not that different from one another. User requirements specify what the user is expected to do. System requirements include both functional needs such as the system should manage 100000 users at the same time and non-functional requirements that specify how the user will fulfill user requirements when engaging with the system.

**User Requirements:** The user's test results must be visible to them.

**System Requirements:** The user accesses the results page to view their test results as per the system requirement.

**Functional and Non-Functional Requirements**

Functional requirements These include expectations for the system's functionality, its responses to different inputs, and how it would need to respond in certain circumstances. The functional requirements may, in some situations, also explicitly indicate what the system should not accomplish. Requirements that are not functional These are limitations on the products or services the system has to offer. They include limitations imposed by standards, deadline restrictions, and limitations on the development process. Non-functional requirements frequently concern the entire system rather than specific features or services.

**Main Activities for Requirements Engineering**

**Elucidating Requirements**

Understanding what stakeholders do and how they could utilize a new system to support that work is the goal of the software requirements elicitation process. Software engineers collaborate with stakeholders during requirements elicitation to learn about the application domain, work activities, services, and system features that stakeholders desire, as well as the necessary system performance. hardware limitations, etc.

**Specifying Requirements**

Writing down the user and system needs in a requirements document is the process known as requirements specification. The requirements for the user and the system should be distinct, straightforward, simple to comprehend, comprehensive, and consistent.

**Validating Requirements**

The process of verifying that requirements define the system the customer actually wants is known as requirements validation. It is similar to elicitation and analysis because it focuses on identifying issues with the requirements. Requirements validation is crucial because mistakes in a requirements document can result in high rework costs if they are found during development or after the system has been put into operation.

**Requirements Always Change**

New requirements always appear once a system has been implemented and is being utilized frequently. This is partially due to mistakes and omissions in the initial specifications that must be fixed. However, changes to the system's business environment are the main driver of changes in system requirements.

**Work Cited**

Atlassian. (2022). Scrum sprints: Everything you need to know. Atlassian. Retrieved January 15, 2023, from https://www.atlassian.com/agile/scrum/sprints

GuidingCode. (2022, January 27). What are user and system requirements in software engineering. GuidingCode. Retrieved January 16, 2023, from <https://guidingcode.com/user-and-system-requirements-in-software-engineering/>

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/>