



**ZIMBABWE OPEN UNIVERSITY**

*"Empowerment Through Open Learning"*



## ASSIGNMENT COVER

REGION:

\_\_\_MDLANDS\_\_\_

—

PROGRAMME: \_\_\_BSSE 241\_\_\_

INTAKE: \_\_\_1\_\_\_

FULL NAME OF STUDENT: \_\_\_PRIVILEGE

KURURA\_\_\_ PIN: \_\_\_P2289097N\_\_\_

MAILING ADDRESS:

\_\_\_p2289097n@students.zou.ac.zw\_\_\_

\_\_\_\_\_

CONTACT TELEPHONE/CELL: \_\_\_0778111517\_\_\_ ID. NO.: \_\_\_67-  
133735-A-67\_\_\_

COURSE NAME: \_\_\_REQUIREMENTS

ENGINEERING\_\_\_ COURSE CODE:

\_\_\_BSSEZ\_\_\_

ASSIGNMENT NO. e.g. 1 or 2: \_\_\_1\_\_\_ DUE DATE:

\_\_\_\_\_

ASSIGNMENT TITLE: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

MARKER'S COMMENTS: \_\_\_\_\_

---

---

OVERALL MARK: \_\_\_\_\_ MARKER'S NAME: \_\_\_\_\_

MARKER'S SIGNATURE: \_\_\_\_\_ DATE: \_\_\_\_\_

**Issue Date:** 3 October 2013

Revision 0

## QUESTION 1

For a small language training company planning to replace its call center and mailouts with online technology, it's important to keep refining the project requirements even after the initial phase. Here's why:

### Adapting to Changes

Online technology and client needs can change quickly. By regularly updating the project requirements, the company can adjust to these changes, ensuring the final product stays relevant and meets current client needs (Sommerville, 2011).

### Better Communication

Continuously discussing requirements with clients and the development team helps clear up misunderstandings and ensures everyone is on the same page. This reduces the risk of problems later on (Kotonya & Sommerville, 1998).

### Improved Quality

Regular feedback and testing during development help create a product that is user-friendly and effective, which is essential since it will replace important client interaction methods like call centers (Nuseibeh & Easterbrook, 2000).

### Reducing Risks

Regularly revisiting the requirements helps identify and fix potential problems early, preventing costly issues or delays later in the project (Lamsweerde, 2000).

### Staying Competitive

In the competitive language training market, continuously improving the product based on new needs and technologies helps the company offer better services and attract more clients (Boehm, 1981).

### Saving Costs

Although it may seem more expensive to keep refining the requirements, it actually saves money in the long run by avoiding costly fixes after the system is built (Boehm, 1981). Continuously refining the project requirements throughout development is key to ensuring the success of the new online system and helping the company improve its services and grow its client base (Sommerville, 2011).

## QUESTION 2

Requirements engineering is a key part of software development that involves a structured approach to defining, documenting, and managing software requirements. The main goal of RE is to ensure the final software product meets the needs of its users. The process is typically broken down into several stages: requirements elicitation, requirements analysis, requirements specification, requirements validation, and requirements management.

### Requirements Elicitation

This is the process of gathering information from stakeholders to understand what they need from the system. Techniques like interviews, surveys, workshops, and observations are used to collect this information. It's crucial to involve all relevant stakeholders to ensure a complete set of requirements is captured. The quality of the information gathered here is vital to the success of the entire project (Nuseibeh, 2010).

### Requirements Analysis

After gathering requirements, the next step is to analyze them to find any conflicts, overlaps, or ambiguities. This step also involves prioritizing the requirements based on their importance and how feasible they are to implement. The goal is to refine the requirements into a clear and consistent set that will guide the software's design and development (Pohl & Rupp, 2011).

## Requirements Specification

This stage involves documenting the requirements in detail. The result is usually a Software Requirements Specification (SRS) document, which includes functional and non-functional requirements, use cases, and user stories. A well-written SRS helps ensure everyone involved understands what the system is supposed to do (Wiegers & Beatty, 2013).

## Requirements Validation

In this step, the documented requirements are reviewed to make sure they accurately reflect what stakeholders need. This is important to catch any errors or inconsistencies before moving on to design and development. Techniques like requirements reviews, prototyping, and model validation are often used to validate the requirements (Kotonya & Sommerville, 2010).

## Requirements Management

This is the ongoing process of tracking and managing changes to the requirements throughout the project. It ensures that any changes are carefully considered, documented, and communicated to everyone involved. Effective management of requirements helps maintain their integrity and prevents scope creep, which can lead to delays and increased costs (Young, 2020).

## QUESTION 3

A Software Requirements Specification (SRS) document is essential for guiding the development of a software project. It lays out everything the software should do and the conditions it must meet. Here are the key issues that an SRS must address:

### 1. Functional Requirements

The SRS must clearly describe the functional requirements of the software. These are the specific features and functions the system must perform to meet the needs of its users. Functional requirements include things like what tasks the software will automate, how data will be processed, and what outputs will be generated. Clear functional requirements ensure that the development team knows exactly what to build (Wiegers & Beatty, 2013).

## 2. Non-Functional Requirements

Non-functional requirements define the system's quality attributes, such as performance, security, reliability, and usability. These are crucial because they affect how well the software operates in the real world. For example, the SRS might specify how fast the system should respond to user actions or how secure the data must be (Chung, Nixon, & Yu, 2012).

## 3. User Interfaces

The SRS should describe how the user interface (UI) will look and behave. This includes details about screen layouts, navigation, input methods, and how the system will interact with the user. A well-defined UI in the SRS helps ensure that the software will be user-friendly and meet the expectations of its users (Preece, Rogers, & Sharp, 2015).

## 4. Constraints

The SRS must outline any constraints that will impact the development and operation of the software. These could include technical limitations, regulatory requirements, or budgetary constraints. Identifying these constraints early helps the development team plan accordingly and avoid potential issues later in the project (Sommerville, 2011).

## 5. Assumptions and Dependencies

The SRS should also list any assumptions or dependencies that could affect the project. Assumptions are conditions believed to be true for the project, such as the availability of certain technologies. Dependencies refer to external factors that the project relies on, like third-party software or hardware. Addressing these in the SRS helps in managing risks and setting realistic expectations (Thayer & Dorfman, 2010).

## 6. Acceptance Criteria

The SRS should define the criteria for accepting the final software product. This includes specific conditions that the software must meet to be considered complete and acceptable by

the stakeholders. Acceptance criteria ensure that there is a clear understanding between the developers and stakeholders about what success looks like (Wiegiers & Beatty, 2013).

A well-crafted SRS is critical for the success of a software project. It must clearly address functional and non-functional requirements, user interfaces, constraints, assumptions, dependencies, and acceptance criteria. By thoroughly covering these issues, the SRS provides a solid foundation for the development process and helps ensure that the final product meets all stakeholder expectations.

#### QUESTION 4

When developing a new proprietary software application, especially one that needs to compete with existing social media platforms, it's crucial to follow a systematic approach to requirements engineering. Here's an outline of the different stages involved and the tools and techniques you can use to derive a complete and consistent requirements specification.

##### 1. Requirements Elicitation:

This is the first stage where you gather information about what the software needs to do. It involves talking to stakeholders such as potential users, business managers, and technical experts to understand their needs and expectations.

At this stage we use tools and techniques like:

You Conduct One-on-one discussions with stakeholders to get detailed information (Nuseibeh, 2010).

You take Surveys and Questionnaires Collecting data from a larger group to understand common needs (Dix, Finlay, Abowd, & Beale, 2004).

Workshops: Group sessions where stakeholders can discuss and refine requirements together (Pohl & Rupp, 2011).

##### 2. Requirements Analysis

Analyzing the collected information to identify any conflicts, overlaps, or gaps. This stage helps prioritize requirements based on their importance and feasibility.

At this stage we use tools and techniques like:

Use Case Analysis: Creating use cases to describe how users will interact with the system

Modeling: Using diagrams like flowcharts or data models to visualize and analyze requirements (Sommerville, 2011).

### 3. Requirements Specification

Documenting the requirements in detail to create a clear and formal description of what the software should do.

At this stage we use tools and techniques like:

Requirements Specification Document: Writing a detailed document that includes functional and non-functional requirements (Wieggers & Beatty, 2013).

Prototyping: Creating a prototype or mock-up of the application to illustrate how the requirements will be implemented (Sommerville, 2011).

### 4. Requirements Validation

Ensuring that the documented requirements accurately reflect the needs of stakeholders and are free from errors.

At this stage we use tools and techniques like:

Reviews: Regularly reviewing requirements documents with stakeholders to catch any mistakes or misunderstandings (Kotonya & Sommerville, 2010).

Validation Workshops: Holding workshops where stakeholders can validate and provide feedback on the requirements (Pohl & Rupp, 2011).

### 5. Requirements Management

Tracking and managing changes to the requirements throughout the project lifecycle.

At this stage we use tools and techniques like:

Change Management Tools: Using tools to document and manage changes to requirements (Young, 2020).

Traceability Matrix: Creating a matrix to ensure that all requirements are covered by the design and that changes are tracked (Thayer & Dorfman, 2010).

To develop a successful social media application, it's essential to follow these stages of requirements engineering: elicitation, analysis, specification, validation, and management. By using appropriate tools and techniques, you can ensure that the requirements are complete, consistent, and aligned with the stakeholders' needs. This structured approach helps in building a product that meets expectations and stands out in the competitive social media market.