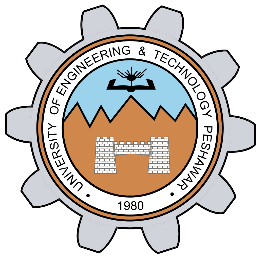
**ASSIGNMENT NO 2**

**RELATED TO CLO1 AND CLO2**



**Fall 2023**

**Software Engineering**

Submitted by: **Maaz Habib**

Registration No.:**20PWCSE1952**

Class Section: **C**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

# Dr. Samad Baseer

**D**epartment of **C**omputer **S**ystems **E**ngineering

**U**niversity of **E**ngineering and **T**echnology, **P**eshawar

**Software Engineering Assignment 02**

**Question 01:**

**What are the reasons for a successful or unsuccessful software project? give two examples.**

**Reasons for Successful Software Projects:**

1. **Clear and Well-Defined Requirements:**

Successful software projects have clear and well-defined requirements that are agreed by stakeholders at the beginning of the project. This helps ensure that the development team understands what needs to be built, reducing the risk of misunderstandings and scope creep.

1. **Effective Communication and Collaboration:**

Successful software projects prioritize effective communication and collaboration among team members, stakeholders, and end-users. Strong communication channels enable prompt feedback, alignment of expectations, and quick resolution of issues, leading to efficient project delivery.

**Reasons for Unsuccessful Software Projects:**

1. **Poorly Defined Requirements:**

Unsuccessful software projects often suffer from poorly defined requirements or frequent changes in scope, commonly referred to as scope creep. Lack of clarity and constant changes lead to confusion, delays, and budget overruns, ultimately affecting the project's success.

1. **Inadequate Planning and Project Management:**

Unsuccessful software projects may suffer from inadequate planning and project management practices. This includes insufficient resource allocation, unrealistic timelines, lack of risk management, and ineffective progress tracking. Without proper planning and management, projects are prone to delays, budget overruns, and failure to meet quality standards.

**Question 02:**

**What kind of problems may arise if software project is developed on an Adhoc Basis?**

A lot of problems may arise due to adhoc some of the problems are discussed below.

**Quality Issues:**

Ad hoc development may result in poor software quality due to insufficient testing, lack of code reviews, and rushed implementation. This can lead to bugs, errors, and system failures, ultimately affecting user satisfaction and trust in the software.

**Difficulty in Maintenance:**

Software developed in an ad hoc manner may be difficult to maintain and enhance over time. Poorly documented code, lack of proper version control, and inconsistent naming conventions can make it challenging for developers to understand and modify the software in the future.

**Communication Challenges:**

Ad hoc development may lead to communication challenges among team members, stakeholders, and end-users. Without clear channels for communication and collaboration, misunderstandings can arise, requirements may be misinterpreted, and feedback may be disregarded.

**Higher Risk of Failure:**

Overall, ad hoc software development increases the risk of project failure. Without a structured approach, it's more likely that the project will encounter issues related to scope, quality, schedule, and budget, ultimately jeopardizing its success.

**Question 03:**

**Give an example of a project that would be ok to do with a waterfall process, but definitely not an agile process, and then explain why waterfall is so preferable over agile for that project.**

A project that might be more suitable for a waterfall process than an agile process could be the construction of a physical infrastructure, such as a bridge or a dam. In a waterfall approach, the project is typically divided into distinct phases, and each phase must be completed before moving on to the next. This sequential and linear structure aligns well with the nature of large-scale construction projects.

Here's why a waterfall process might be preferable for such a project:

1. **Sequential Nature of Tasks:**

In construction projects, certain tasks depend on the completion of others. For instance, the foundation must be laid before erecting the superstructure. Waterfall's sequential approach ensures that each phase is completed before moving to the next, which aligns well with the linear nature of construction tasks.

1. **Fixed Requirements and Scope:**

In construction projects, the requirements and scope are often well-defined from the beginning. Changes to these elements are typically costly and can significantly impact the project timeline. Waterfall is more rigid and allows for a clear definition of requirements at the outset, making it suitable for projects with stable and fixed requirements.

1. **Extensive Planning and Documentation:** Waterfall methodologies

emphasize thorough planning and documentation upfront. For construction projects, detailed architectural plans, engineering specifications, and environmental impact assessments are crucial. Waterfall's emphasis on comprehensive documentation suits scenarios where these aspects need to be carefully outlined before execution.

1. **Minimal Customer Involvement During Execution:**

Construction projects might not require constant customer feedback or iteration during the execution phase. Once the planning is complete, the construction team can efficiently execute the plan without frequent adjustments, aligning with the waterfall approach that limits customer involvement until the final delivery.

**Reference:**

* *"Scrum: The Art of Doing Twice the Work in Half the Time" by Jeff Sutherland.*
* *"Agile Estimating and Planning" by Mike Cohn.*
* *"The Mythical Man-Month: Essays on Software Engineering" by Frederick P. Brooks*

*Jr.*

* *"Scrum: The Art of Doing Twice the Work in Half the Time" by Jeff Sutherland.*
* *"The Mythical Man-Month: Essays on Software Engineering" by Frederick P. Brooks Jr.*