SOFTWARE ENGINEERING

THEORY-PRACTICAL CORRESPONDENCE

Report by:

Share Karo App members

Introduction

The project Share Karo App has been devised through thorough research of the Software Engineering course module. The following components have been studied by the group and henceforth incorporated into the development of the “Share Karo App”.

Software Engineering

Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software.

From the Users Perspective:

- Correctness
- Reliability
- Efficiency
- Maintainability
- Usability
- Robustness

From the Developer's Perspective:

- Consistency
- Understandability
- Testability
- Compactness
- Compatibility
- Integrity

We integrated all the above mentioned requirements into the project so as to deliver better performance on both ends.

Agile Scrum Methodology

Scrum is an agile process that allows us to focus on delivering the highest business value in the shortest time. It allows us to rapidly and repeatedly inspect actual working software (every two weeks to one month).

The business sets the priorities. Our teams self-manage to determine the best way to deliver the highest priority features. Every two weeks to a month anyone can see real working software and decide to release it as is or continue to enhance for another iteration.

We followed this module and did rigorous brainstorming every week to develop and enhance the project development process.

SDLC Module

A software development life cycle model (also called process model) is a descriptive and diagrammatic representation of the software life cycle and life cycle model represents all the activities required to make a software product transit through its life cycle phases. It also captures the order in which these activities are to be undertaken.

We incorporated the Spiral Model into the project after researching and corresponding our methodology to the module.

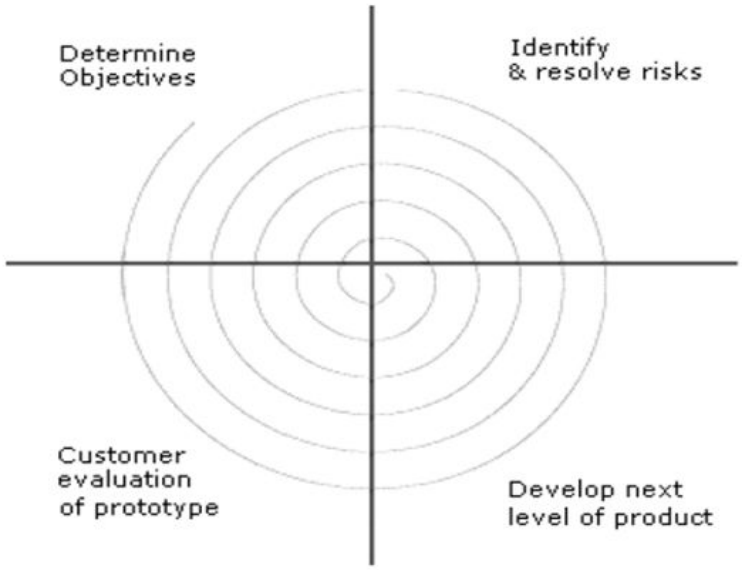

Fig. 2.2: Spiral Model

Risk handling is inherently built into this model.

The spiral model is suitable for development of technically challenging software products that are prone to several kinds of risks.

Requirement Engineering Phase

Software Requirement Specification

The main objective of the SRS document is basically to describe the principal requirements engineering activities and to introduce techniques for requirements elicitation and analysis. It also plans to describe requirements validation and to discuss the role of requirements management in support of other requirements engineering processes.

Our project involved the following key points:

- Overall Description of the project was given.
- External Interface Requirements were specified.

- System Features that presented the complete module of the app/project and have been explained.
- Other Nonfunctional Requirements like constraints and everything on the requirements.

This helped to understand the project even closely and in a better form and the requirements helped the developers to design the app even more efficiently.

Feasibility Analysis

A feasibility study decides whether or not the proposed system is worthwhile and it is a short focused study that checks:

•If the system contributes to organisational objectives.

•If the system can be engineered using current technology and within budget.

•If the system can be integrated with other systems that are used.

The team has worked hard for designing the USPs of the app that differentiates it from all contemporary substitutes as the app comes with extravagant advantages like:

1) Low cost development and management.
2) Low memory requirements for operation across systems.
3) Cross platform and Open source functioning.
4) User friendly interface.
5) Privacy protection for the users.
6) Easy interaction between users.
7) Socializing add-on to the app.

Requirement Traceability Matrix

Traceability is concerned with the relationships between requirements, their sources and the system design.

The Traceability matrix helped the project developers to draw parallels between the requirements so that they can be connected.

UML

Use Case diagram

The Use case diagram is a formal way of representing how a business system interacts with its environment and illustrates the activities that are performed by the users of the system. The diagrammatic relationship helps us design the project in a better way.

Use Class diagram

Sequence diagram