**Abstract**

**University Admission System** is self- explanatory. This system is a crucial and useful system for the students who are to apply for admission for certain university. This system helps students to apply for multiple courses, pay the admission fees, view the admission history, appear for the admission test, print the receipt of fees.

This system is helpful for the Admission department as well, as it helps them to insert **Available** Admission courses, remove courses, print report for the admission, conduct test, get the result of the test.

In this report we will discuss about the project and how it was developed throughout the given time period.

**Table of Content**

1. Introduction
   1. Existing System
   2. Problem Definition
   3. Proposed System
2. Software Development Methodology
3. Software Project Management
   1. Objective
   2. Scope
   3. Cost Estimation
4. Software Risk Management
   1. Strategy of Risk
   2. Risk Identification
5. Software Quality Management
   1. Concept
   2. Quality
   3. Quality Control
   4. Quality Checking Activity
   5. Cost of Quality
6. Software Configuration Management
7. Conclusion
8. References

**Introduction**

**Existing System:** The existing system allows you to apply for admission only once without login, which makes the database bulky. The existing system does not have a user profile, thus user cannot login to see the details of admission. The existing system is not “mobile-friendly”, meaning, it is not responsive. The existing system allows administrator to print reports.

**Problem Statement:** The current system has many problems such as; it does not have an integrated quiz for admission. It is not mobile friendly. It does not have admin panel, authorities have to ask system administrator to make changes. It does not have a good user interface.

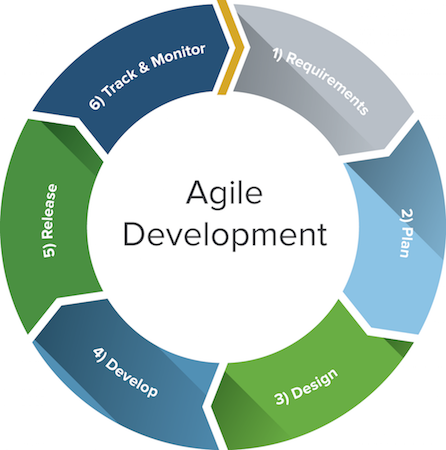
**Proposed System:** We as a team have proposed a system which overcomes most of the problems in the existing system; the new system is mobile friendly. The new system has an integrated quiz system. The new system has an administrator interface.

**Software Development Methodology**

We have developed this project by following the **AGILE** methodology.

As software development is a **continuous** process, AGILE methodology is the best option to select among all Software Development Technology.

Below is how AGILE methodology works: -



**Software Project Management**

**Objective:**

The Objective of this project was to set goals and achieve those goals in the given time period.

We were in continuous communication with the client and were getting constant constructive feedback.

As a part of continuous development we used to Develop - > Test -> Debug -> Deploy and repeat the same process.

We did proper planning and followed that plan throughout the project.

**Scope**

The scope of project was to **develop** and **deploy** an Admission system for the University, about which the details are mentioned in **Introduction** part.

For the ease of project we used GitHub as version control, so that we can track all the changes made in the project, and if the something would go wrong we would roll back to previous version.

For the ease we divided the project in smaller parts i.e. **Design – Develop (includes Debug) – Test.**

**Cost Estimation**

Cost Estimation is considered as the most crucial of all, as it depends more than one element.

We ensure our end product which is deliverable is in reasonable size, so that it does not faces performance issues.

The said software has passed all possible manual testing, it is hack proof as it is using Object Oriented methodology and is using latest hack proof methods such as PDO and MySQLi.

This software is fully open-source, i.e. anyone can use it/ distribute/ and contribute.

The said software has MIT License.

Any additional software used to develop this project had not expense as we have used all open source libraries and software.

Miscellaneous expenses such as Travel, Skill, and Training and support etc. are not part of this project.

We are using **CoCoMo** (Constructive Cost Model) as a Cost Estimation model.

**Software Risk Management**

In this section we will discuss possible risks which occur or may occur or have occurred during the development of this software. By definition, software risk means, a possibility of suffering from loss in software development process is called a software risk.

Generally risk is handled by the Project Manager and the Team, client or end-user has nothing to do with Risk which occurs during development. The known risks which have occurred are as bellows

**Compatibility Issues:** Software which is developed in certain said environment i.e. Windows, Linux or Mac OS is limited to that environment only. We cannot run that software on other platforms. This risk can be managed by making that as platform independent.

**Time Constraint:** Time is a critical factor in a project, which is limited during the project. We need to deliver the product in the given time. To solve this constraint we divided the work, so that each and every one of us gets equal work and no one gets burdened with work.

**Budget:** Budget gets decided during the planning phase, and cannot get altered or changed during the project. We cannot get over-budget in the middle of the project. To avoid this issue we did project planning properly and made a project charter.

Above are the “**known**” issues which we faced during the project and resolved for the same.

**Strategy of Risk:** In a small project like this, the risks are low and limited but risks still occur. As discussed above the risks which occurred, are of “**dual nature**”, meaning it affects both the developer (team) and the client.