Integrated Faculty Progress Monitoring System

BY: Trina Marie Alcantara
Jade Kevin Balocos
Richmond D. Besmonte

Introduction

In the realm of higher education, ensuring the quality and effectiveness of faculty performance is paramount to fostering an environment conducive to student success and institutional excellence. As academic institutions strive to enhance their educational offerings and maintain high standards, there is an increasing need for robust systems that can accurately track and support faculty development.

To address this need, educational institutions are turning to innovative solutions that integrate multiple facets of faculty assessment and development into cohesive platforms. Among these, the Integrated Faculty Progress Monitoring System stands out as a comprehensive tool designed to streamline and enhance the process of evaluating and supporting faculty performance.

This system is meticulously crafted to offer a unified approach to monitoring faculty progress, incorporating various components such as performance metrics, professional development tracking, and feedback mechanisms. By consolidating these elements into a single, integrated platform, the system provides a holistic view of faculty achievements and areas for improvement.

At its core, the Integrated Faculty Progress Monitoring System enables institutions to set clear performance benchmarks, regularly assess faculty

contributions, and facilitate targeted professional development opportunities. Through its user-friendly interface and data-driven insights, the system not only supports faculty members in their professional growth but also aids institutional leaders in making informed decisions that drive academic excellence.

By leveraging this sophisticated system, educational institutions can ensure a more systematic and transparent approach to faculty evaluation, ultimately contributing to a more dynamic and effective learning environment.

Purpose and description

The system is a tailored solution for the College of Computer Studies at San Pablo Colleges. It serves as a sotool that enables the dean to efficiently oversee and manage the faculty's qualifications and various academic prerequisites. With this system in place, the dean can maintain a comprehensive database of the faculty members' credentials, ensuring that all educational standards and regulations are met. Furthermore, the system can ease the process of document submission by providing faculty members with timely notifications and reminders to submit their required documents. This approach facilitates the dean's advisory role, allowing for a more organized and effective management of faculty affairs. The system's design reflects a deep understanding of the unique operational needs of the college, and it incorporates features that simplify the complexities associated with academic administration. By automating routine tasks, the system frees up the

dean's time, allowing them to focus on more strategic aspects of their role, such as faculty development and curriculum enhancement. Additionally, the system's user-friendly interface ensures that faculty members can navigate through the document submission process with ease, promoting a culture of compliance and accountability within the college. Overall, this system represents a significant advancement in the administrative capabilities of San Pablo Colleges, setting a new standard for academic excellence and operational efficiency.

Objectives

The main objective of the study is to provide San Pablo Colleges improved way submission of faculty requirements, specifically aims to;

1. Develop a system that is capable of;
    a. Accepting faculty personal data sheet in a centralized repository
    b. Monitoring submissions of faculty PDS.
    c. Faster the submission procedure.
    d. Tracking faculty submissions by means of having diagrams or progress bar
    e. Notifying faculty members about their task activity.
    f. Develop a system that will generate an editable file of employee's PDS.
    g. Develop a system that will enable to update the PDS of employees in San Pablo Colleges.
    h. To develop a system that enebles employees to print their PDS file.
2. Evaluate the system using ISO 25010 system evaluation tool

3. Create a deployment or implementation plan specifically designed to San Pablo Colleges.

Scope and Limitations

The study focuses on the progress and monitoring of the faculty members.

Scope

The system offers a comprehensive solution for assessing faculty performance in educational institutions to provide a holistic view of teaching effectiveness. IFPMS also includes a professional development module with web-based training tailored to faculty needs, facilitating ongoing skill enhancement. Its Analytics capabilities empower administrators to make informed, data-driven decisions on faculty promotions, rewards, and interventions. Furthermore, the system's customizability allows it to be adapted to various educational contexts, enabling institutions to tailor evaluation criteria, reporting formats, and feedback mechanisms to their specific requirements.

Limitations

The system, while offering a robust framework for enhancing faculty performance, has limitations. Key concerns include data privacy and security, especially given the reliance on sensitive information like attendance and evaluations. Compliance with regulations such as GDPR is complex, particularly with cross-border data flows. Additionally, resistance from faculty and administrators to new technologies may require extensive training and change management. The system's effectiveness also depends on sufficient technical infrastructure, which may be lacking in some settings. Customization can be resource-intensive, potentially delaying implementation. There's also a risk of over-relying on quantitative metrics, which might not fully capture teaching quality. Moreover, regular maintenance and updates are required, straining resources, and the professional development modules may not meet all faculty needs.

Technical Background

Details of Technology Use

This section aims to specify the technologies in use.

HTML - The Hypertext Markup Language is the established coding language used to make webpages. HTML explains how a webpage is organized, it is also comprised of a sequence of components. HTML elements provide instructions to the browser on how to show the content.

PHP - is a programming language that is executed on the server side and is capable of generating static content, and interactive, or internet-based applications. Code written in PHP can be embedded in HTML or utilized with a range of web template systems, CMS, and frameworks.

CSS - Cascading Style Sheets is a basic design language created to simplify the task of making web pages look more appealing. CSS manages the appearance and atmosphere of a website. CSS can be utilized to control text color, font style, paragraph spacing, column scaling and layout, background images or colors, layout designs, and display variations for various devices and screen sizes, as well as numerous other effects.

MYSQL - is an open-source relational database management system that can be accessed using Structured Query Language (SQL). A query language is a basic programming language that enables us to retrieve, modify, and control data within a relational database.

BOOTSTRAP - it is a front-end development framework that is free and open-source for building websites and web applications. Bootstrap is created to facilitate the flexible creation of websites optimized for mobile devices, offering a range of coding options for design templates.

JAVASCRIPT - is a coding language utilized for generating dynamic content on websites. This is accomplished by introducing fresh HTML elements and making changes to current ones. Utilizing JavaScriopt builds websites that are both user-friendly and interactive. Using this coding language the front-end developers create features such as image sliders, interactive games, and timers.

VISUAL STUDIO CODE - is a simplified code editor that includes features such as debugging, task execution, and version control. It aims to offer only the necessary tools for developers to quickly code, build, and debug while leaving more advanced processes to comprehensive IDEs.

MONGODB - this uses a scale-out architecture that is favored by developers for creating scalable applications with changing data schemas. It simplifies the storage of structured and unstructured data for developers as a document database.

PYTHON - is a high-level programming language with dynamic semantics that is interpreted and object-oriented. Its advanced built-in data structures, along with dynamic typing binding, make it a great choice for rapid application development, and also for serving as a scripting language or glue to link existing components.

Related Literature/Systems

Progress monitoring systems have emerged as a pivotal element in educational settings, providing a mechanism for continuous feedback on both faculty performance and student outcomes. Historically, these systems have evolved from basic attendance tracking and performance evaluations to sophisticated, data-driven tools that inform instructional practices and decision-making processes. According to Fuchs (2004), progress monitoring has expanded from elementary to secondary education and beyond, emphasizing the need for systematic data collection and analysis to track academic and teaching performance effectively (Ketterlin-Geller, Sparks, & McMurrer, 2022) (Kogure et al., 2015).

Integrated Systems in Education

Integrated faculty progress monitoring systems combine multiple data points to offer a comprehensive view of faculty performance. These systems incorporate various elements, such as attendance tracking, performance evaluations, and feedback loops, creating a holistic approach to monitoring. For example, the implementation of a web and mobile-based faculty performance evaluation system highlights the integration of diverse evaluation criteria and technological frameworks to ensure thorough monitoring (Salas, 2015). This multifaceted approach is essential for capturing the complexities of faculty roles and responsibilities, providing a detailed understanding of their impact on student outcomes.

Technological Advancements in Monitoring

The incorporation of advanced technologies, such as artificial intelligence (AI) and machine learning, has significantly enhanced the capabilities of progress monitoring systems. AI-based systems analyze vast amounts of data to identify patterns and predict outcomes, providing valuable insights into faculty performance. A study on AI approaches to monitoring student performance illustrates how these technologies can be adapted for faculty progress monitoring, enhancing predictive accuracy and effectiveness (Khan, Ahmad, Jabeur, & Mahdi, 2021) (Ketterlin-Geller, Sparks, & McMurrer, 2022). The use of AI not only streamlines the data analysis process but also helps in identifying areas where faculty may need additional support or training.

Web-Based Professional Development Systems

Web-based systems designed for professional development play a significant role in faculty progress monitoring. These systems offer interactive instructional modules that guide teachers through various performance measures, including administration and scoring. An evaluation of a web-based professional development system for algebra teachers demonstrated high satisfaction rates and significant improvements in teachers' knowledge and skills, ultimately benefiting their instructional practices (Lyons et al., 2019) (Steinfeld, Franklin, Mercer, Fraynt & Simon, 2016). By integrating professional development with progress monitoring, these systems ensure that teachers receive continuous support and feedback, fostering a culture of ongoing improvement.

Case Studies and Implementation

Numerous case studies highlight the successful implementation of progress monitoring systems in various educational contexts. For instance, a web-based system for managing final-year projects effectively tracks faculty involvement and progress, demonstrating the potential of these tools in higher education settings (Khan, Ahmad, Jabeur, & Mahdi, 2021) (Salas, 2015). Similarly, a study on integrated healthcare systems illustrates how progress monitoring can be adapted to track behavioral health outcomes, providing valuable insights that can be applied to educational contexts (Steinfeld, Franklin, Mercer, Fraynt & Simon, 2016). These case studies offer practical examples of how integrated progress monitoring systems can be tailored to meet the specific needs of different educational environments, showcasing the versatility and effectiveness of these systems.

Challenges and Future Directions

Despite the advancements in progress monitoring systems, several challenges remain. One significant challenge is the resistance to change among faculty and administrators, who may be skeptical of new technologies and data-driven approaches. Additionally, concerns about data privacy and security pose significant barriers to the widespread adoption of these systems. Ongoing training and support are also essential to ensure that educators are comfortable using these tools and can effectively interpret the data they generate (Ketterlin-Geller, Sparks,

& McMurrer, 2022) (Lyons et al., 2019) (Steinfeld, Franklin, Mercer, Fraynt & Simon, 2016).

Future research should focus on addressing these challenges by developing more user-friendly interfaces, enhancing data security measures, and providing comprehensive training programs. Moreover, exploring new technologies, such as blockchain for secure data management and advanced analytics for deeper insights, could further enhance the effectiveness of progress monitoring systems (Khan, Ahmad, Jabeur, & Mahdi, 2021) (Salas, 2015). By continuing to innovate and adapt these systems, educational institutions can ensure that they meet the evolving needs of both educators and students, ultimately leading to improved educational outcomes.

Integrated faculty progress monitoring systems are crucial for improving educational outcomes by providing continuous and comprehensive feedback on faculty performance. The integration of advanced technologies, the development of robust frameworks, and the effectiveness of these systems. However, addressing the challenges related to resistance to change, data privacy, and ongoing support is essential for their successful adoption and implementation. By continuing to innovate and adapt these systems, educational institutions can ensure that they meet the evolving needs of both educators and students.

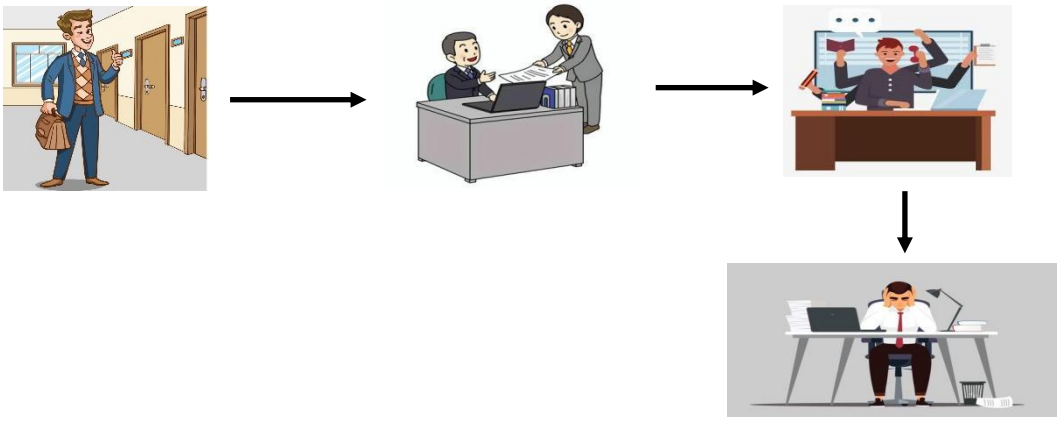
CHAPTER II

SOFTWARE DEVELOPMENT METHODOLOGY

Requirement Analysis

To enhance the comprehension of the faculty credential submission process, developers have formulated a series of questions aimed at illuminating key aspects of this procedure. Faculty members are mandated to submit their complete credentials to the dean's office. Subsequently, the dean is tasked with aggregating the documents from all faculty members. Through a manual review, the dean assesses which documents are pending submission. Additionally, the dean is responsible for identifying faculty members who have not yet submitted their credentials. This manual process can be exhaustive for the dean, potentially hindering the ability to undertake other essential duties. There is an implicit suggestion that automating this process could significantly alleviate the dean's workload, allowing for a more efficient allocation of time and resources. By streamlining the credential submission and verification process, the faculty can ensure that all necessary documentation is accurately accounted for, thereby enhancing the operational efficiency of the dean's office.

Current Technical Situation



The diagram in above illustrates the conventional method by which faculty members submit their required documents. This process involves faculty personally delivering the necessary paperwork directly to the dean's office. Upon receipt, the dean meticulously reviews each document for completeness and accuracy. In the event that there are any missing documents, the dean is responsible for identifying them and notifying the respective faculty member. This traditional, manual approach to document submission is noted to be a significant source of stress and fatigue for the dean, compounded by the ongoing need to remind faculty to provide any unsubmitted documents. The idea suggests that the current system is labour-intensive and potentially inefficient, possibly indicating the need for a more efficient, automated process to alleviate the administrative burden on the dean and improve overall workflow.

Proposed System

The diagram illustrates a modernized document submission system designed to enhance the efficiency of administrative processes within an academic institution. Through this system, each faculty member is provided with a personalized account, enabling them to upload essential documents required by the Dean's office. The innovative aspect of this system lies in its real-time functionality, allowing the Dean to instantly receive submissions. This feature also facilitates the Dean's ability to track and identify any outstanding deliverables or documents that have not been submitted. The flexibility of the system is further highlighted by its accessibility, granting faculty members the convenience of submitting documents from any location and at any time, thus transcending geographical and temporal constraints. This level of accessibility ensures that the process of document management is significantly simplified for both the Dean and the faculty members, fostering a more efficient and less cumbersome administrative workflow.

Requirement Documentation

This documentation details the capabilities and the function that this system is capable of. This will establish the agreement between the developers and the client. With this, the developers came up with the following features

MODULES

Add faculty – this feature allows the admin to add Professor

Upload – This module allows the faculty members/Professors to upload required documents.

Dash board- this allows the admin to see the different data analytics

List of Professors- this module is designed to see all the current professors

Login module- this interface will provide identification of the users

Logout module- this interface is designated to exit or close the application

Hardware Requirement

Processor	:	At least Intel Core i3 4170 3.7Ghz
Hard Disk Drive	:	At least 1GB of free space for server and at least 5 client computers
Memory Module	:	Any brand of memory with at least 2GB capacity
Router	:	Any brand of router

Software Requirements

Operating System	:	Microsoft Windows 10 or higher 32/64bit

Application software : Visual Studio 2013 or higher

Database Software : MySql/PHPmyadmin

Screen Resolution : 1366 x 766 (Recommended)

Internet Service Provider : 5 Mbps

Software Requirement

- Operating System Requirements:

  Windows: 10 or most recent

  MAC: OS X10 or earlier

  Linux: Ubuntu

- Browser Requirement:

  Microsoft Edge

  Google Chrome

  Mozilla Firefox

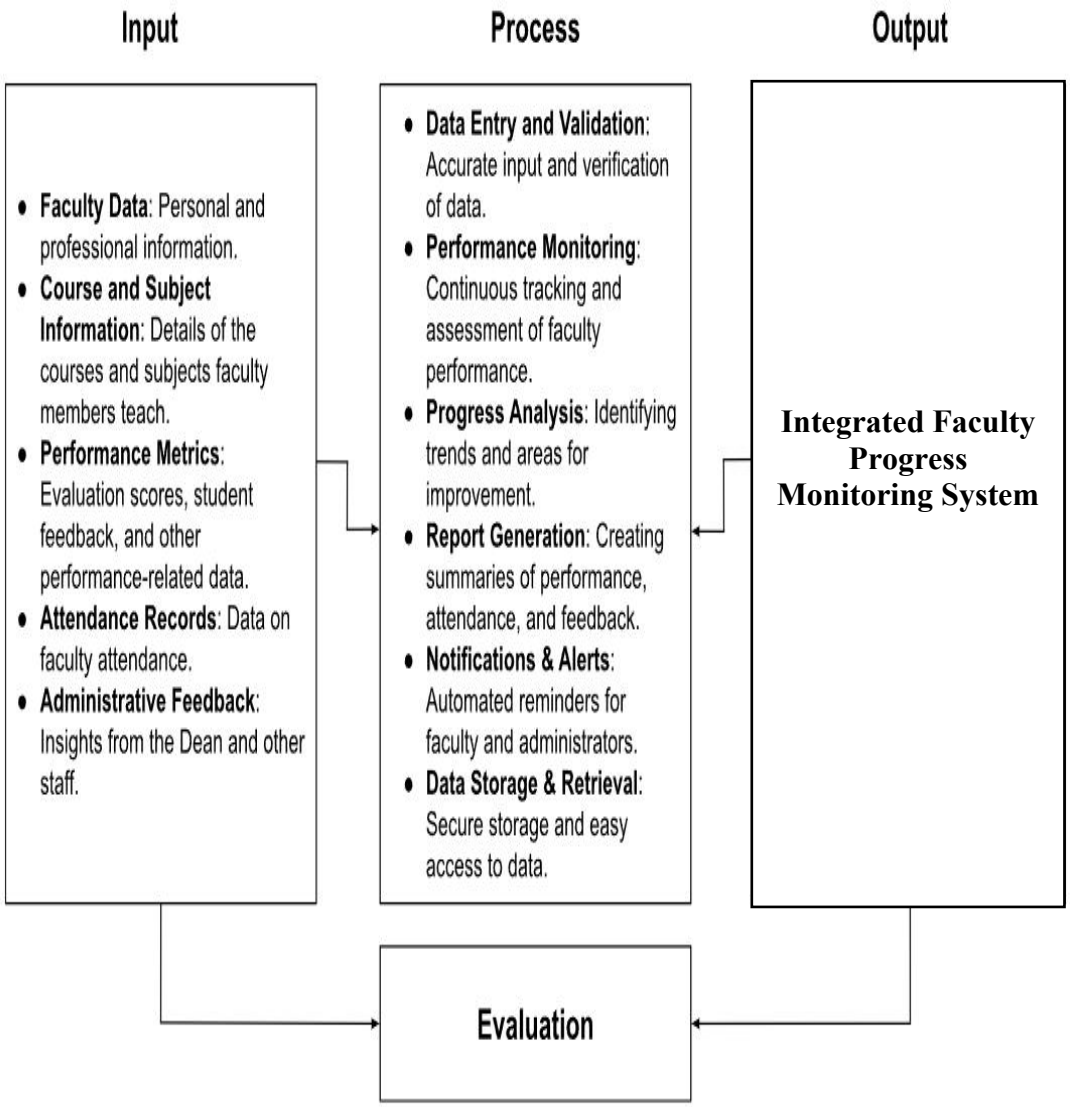
Other Web Browsers that support HTML 5.

Peopleware

Administrator (Dean), San Pablo Colleges Professors, and Future Researchers are the expected users of this system.

Design of Software, Systems, Product and/or Processes

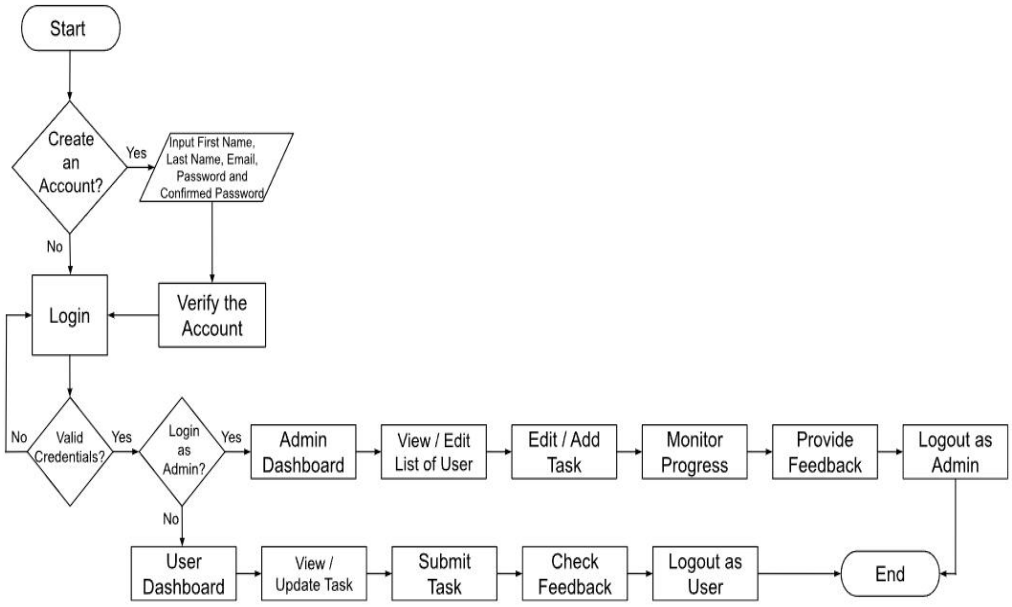
Conceptual Design



Block Diagram

Input	Process	Output
• Faculty Data Entry • Course Details Input • Performance Metrics Submission • Attendance Records Entry • Feedback Entry • Login System	• Data Validation & Processing Engine • Progress Monitoring Module • Report Generation Engine • Alert & Notification System • Data Storage System • Authentication & Security Module	• Faculty Performance Reports • Progress Tracking Charts • Attendance Summaries • Feedback Reports • Alerts & Notifications • User Dashboard

Flow Chart



Development and Testing

The developers used python as their programming language with incorporating PHP and Mysql to their project. The programming languages that have been used have been successfully incorporated with the proposed project which is the evaluation system for the professors and students with not ruling out the possible room for improvements for the betterment outcome of the system being develop.

Agile Methodology

1. Project Overview: Develop a system to track and evaluate faculty progress, performance, and development, including goal setting, performance metrics, feedback, and reporting.

2. Key Phases:

Initial Planning (2 Weeks):

- Gather requirements from stakeholders.
- Define project scope and create a product backlog.

Sprint 1 – Foundations (3 Weeks):

- Set up the development environment.
- Implement core features like user authentication and a basic dashboard.
- Collect initial user feedback.

Sprint 2 – Advanced Features (3 Weeks):

- Develop advanced features such as performance metrics and feedback mechanisms.
- Integrate with existing systems if necessary.
- Refine based on user feedback.

Sprint 3 – Refinement (2 Weeks):

- Enhance UI/UX based on feedback.
- Conduct thorough testing.
- Prepare for deployment.

Deployment & Maintenance (Ongoing):

- Deploy the system.
- Provide training and support.
- Collect feedback for future updates.

3. Roles:

- **Product Owner:** Manages backlog and ensures alignment with stakeholder needs.
- **Scrum Master:** Facilitates the Agile process and removes obstacles.
- **Development Team:** Designs, develops, and tests the system.
- **Stakeholders:** Provide input and feedback.

4. Artifacts:

- **Product Backlog:** List of features and requirements.
- **Sprint Backlog:** Tasks for the current sprint.
- **Increment:** Working product at the end of each sprint.

5. Ceremonies:

- **Sprint Planning:** Define sprint goals and tasks.
- **Daily Stand-ups:** Update on progress and issues.
- **Sprint Review:** Demonstrate completed work and gather feedback.
- **Sprint Retrospective:** Reflect on the process and suggest improvements.

6. Success Metrics:

- **User Satisfaction:** Positive feedback from users.
- **Performance Metrics:** Accurate tracking and reporting.
- **System Stability:** Few bugs/issues post-deployment.
- **Adoption Rate:** High usage and engagement.

Evaluation

Following the system's development, an evaluation procedure will conduct. There are three types of evaluators: Librarian and staff, and IT professionals. The review will place to see if the system met the international standard. The ISO 250101, named "Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQUARE) - System and Software Quality Models," is a software quality standard. It describes the models, which comprise of characteristics and sub-characteristics, that were utilized for both software product quality and software quality, as well as practical guidance on how to use the quality models.

ISO 25010 describes the two quality models, (1) the quality in use model composes of eight (8) characteristics, and some are further sub-divided into sub characteristics that related to the outline of interaction when a product was used in a particular context of use.

The ISO 25010 is made up of eight product quality characteristics 27 sub characteristics: Functionality, Reliability, Usability, Performance Efficiency, Security, Compatibility, Maintainability, Portability.

Functionality

Functional Completeness - Refers to the set of functions that covers all of the specified tasks and user objectives.

Functional Correctness - This refers to how well a product or system provides the correct results with the needed degree of precision.

Functional Appropriateness - Refers to how well functions are able to accomplish specified tasks and objectives.

Reliability

Maturity - Refers to how well a system, product, or component is able to meet your needs for reliability.

Availability - The ability to whether a system, product, or component is operational and accessible.

Fault Tolerance - Ability how well a system, product, or component operates despite hardware and/or software faults.

Recoverability - Ability to how well a product or system can recover data in the event of an interruption or failure.

Usability

Appropriateness Recognizability - Refers to how well you can recognize whether a product or system is appropriate for your needs.

Learnability - Ability to how easy to learn how to use a product or system.

Operability - Refers to whether a product or system has attributes that make it easy to operate and control.

User Error Protection - Refers to how well a system protects users against making errors.

User Interface Aesthetics - Refers to whether a user interface is pleasing.

Accessibility - Refers to how well a product or system can be used with the widest range of characteristics and capabilities.

Performance Efficiency

Time Behavior - Characterizes response times for a given thru put i.e. transaction rate.

Resource Behavior - Refers to the amount and types of resources used by a product or system while performing its functions.

Capacity - Refers to the maximum limits of a product or system parameter.

Security

Confidentiality - Refers to how well a product or system is able to ensure that data is only accessible to those who have authorized access.

Integrity - Ability to how well system, product, or component is able to prevent unauthorized access and modification to computer programs and/or data.

Non-repudiation - Ability to how well actions or events can be proven to have taken place.

Accountability - Ability to the actions of an unauthorized user can be traced back to them.

Authenticity - Ability to how well the identify of a subject or resources can be proved.

Compatibility

Co-existence - Ability how well a product can perform its required functions efficiently while sharing a common environment and resources with products, without negatively impacting any other product.

Interoperability - Ability to how well two or more systems, products, or components are able to exchange information and use that information.

Maintainability

Modularity - Refers to whether the components of a system or program can be changed with minimal impact on the other components.

Reusability - Refers to how well an asset can be used in more than one system.

Analysability - Refers to the effectiveness of an impact assessment on intended changes. In addition, it also refers to the diagnosis of deficiencies or causes of failures, or to identify parts to be modified.

Modifiability - Refers to how well a product or system can be modified without introducing defects or degrading existing product quality.

Testability - Ability how effective the test criteria is for a system, product, or component. In addition, it also refers to the tests that can be performed to determine whether the test criteria has been met.

Portability

Adaptability - Ability how well a product or system can be adapted for different or evolving hardware, software, or other usage environments.

Installability - Ability how successfully a product or system can be installed.

Replaceability - Ability how well a product can replace another comparable product

Implementation Plan

Table 1. Deployment Plan

Strategy	Activities	Person(s) Involved	Duration
Approval from the client (SPC CCS)	Approval letter from the Dean	Developers, CCS Dean	1 hour
Implementation of System	Installation of the proposed system	Developers, CCS Dean	3 hours
Actual Demonstration	Demonstration of the developed system	Developers, Faculty, CCS Dean	2 hours
Training of the faculty	Actual training	Developers, faculty, CCS Dean	2 hours

Table 1 shows the plan of the developers in developing the system until it is implemented. To ensure a smooth progression of the system's implementation, the following statements outline the strategic steps and their respective durations:

1. The client's approval will be secured within one hour, evidenced by an approval letter from the Dean, with the developers and the CCS Dean being the key personnel involved in this phase.

2. System implementation will be executed over a three-hour period, during which the proposed system will be installed by the developers in collaboration with the CCS Dean.

3. An actual demonstration of the developed system is scheduled for two hours, where developers will present the system to the faculty and the CCS Dean.

4. Faculty training is allocated a two-hour window, wherein developers will conduct hands-on training sessions with the faculty under the supervision of the CCS Dean.