Software Development Life-Cycle Model for

Mess Management System

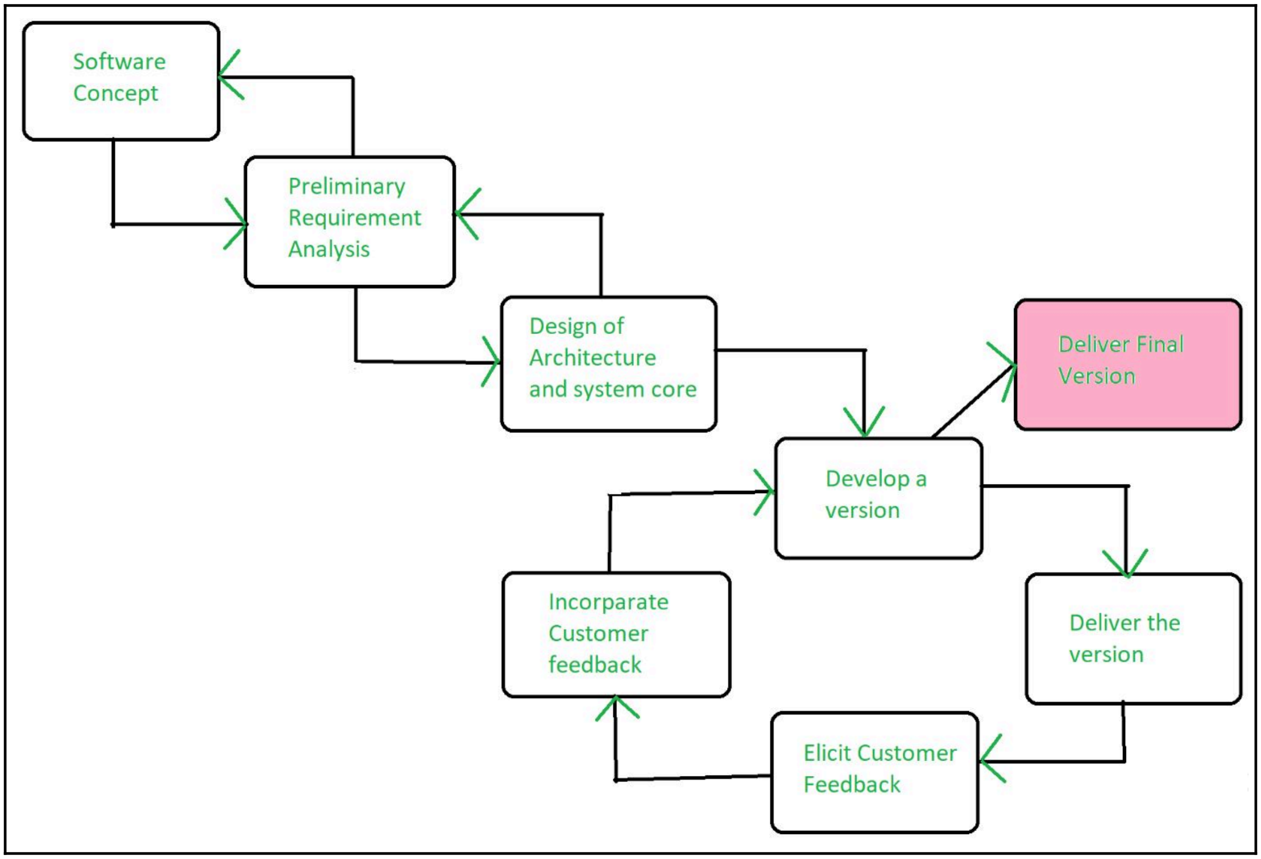
Indian Institute of Information Technology Guwahati

Prepared by:

- **Abhishek Kumar (2101012)**
- **Aman Soni (2101029)**
- **Aman Verma (2101031)**

29th Jan, 2024

DEVELOPMENT MODEL: EVOLUTIONARY


Software Concept
Preliminary Requirement Analysis
Design of Architecture and system core
Deliver Final Version
Develop a version
Incorparate Customer feedback
Deliver the version
Elicit Customer Feedback

Evolutionary Model:

The Evolutionary Model is an iterative and incremental development approach where the software is developed, refined, and improved in multiple cycles. It involves continuous feedback, adaptation, and enhancements based on changing requirements, allowing the system to evolve over time.

Mapping project to Evolutionary Model

❖ Requirements Analysis:

- ➢ *Identify features such as User Registration and Authentication, Role-based Access Control, QR-Based Complaint Registration & Tracking, and Mess Menu & Meeting Management.*

Working:

- ➢ *Define high-level requirements for the identified features.*

❖ Design of Architecture and System Core:

- ➢ *Identify the core module (e.g., User Registration and Authentication) and develop the system architecture around it.*
- ➢ *Prioritize features based on their criticality and impact on the system.*

Working:

- ➢ *Establish a preliminary architecture outlining the interactions between different components.*
- ➢ *Define the core functionalities of the system, focusing on the identified core module.*

❖ *Initial Version: User Management and Authentication:*

Features:

- ➢ *Develop the core functionalities of User Registration and Authentication.*
- ➢ *Implement Role-based Access Control to restrict functionalities based on user roles.*

Working:

- ➢ *Code the user registration module with two-factor authentication.*
- ➢ *Integrate authentication mechanisms and role assignment.*
- ➢ *Deploying the initial version to alpha testers and stakeholders for feedback.*

Feedback :

- *Gather user feedback on registration, authentication, and role-based access, focusing on usability, security, and effectiveness of permissions.*

❖ First Increment: Mess Information Display

Features:

- *Implement features related to displaying the mess menu and operational timings.*

Working:

- *Develop modules for showing daily or weekly mess menus, including breakfast, lunch, and dinner.*
- *Display operational timings for each meal.*

Feedback:

- *Collect feedback on the menu display, addressing any issues related to accuracy and accessibility.*

❖ Second Increment: Complaint Portal

Features:

- *Implement the QR-Based Complaint Registration and Tracking system.*

Working:

- *Code the functionality for users to register complaints by scanning QR codes in the mess.*
- *Develop a system for users to track the status and resolution progress of their registered complaints using unique reference id.*

Feedback:

- *Gather feedback on QR-based complaint registration for user convenience. Assess the effectiveness and user-friendliness of the complaint tracking system.*

❖ Third Increment: Mess Menu & Meeting Management:

Features:

- ➢ *Implement features related to displaying upcoming Mess Meeting details and identifying the responsible faculty in charge and also edit access to the menu.*

Working:

- ➢ *Develop modules for displaying meeting information entered by mess committee members*
- ➢ *Implement role-based control for menu editing access.*

Feedback:

- ➢ *Gather info on mess committee members' menu-editing privileges and confirm faculty-in-charge's ability to schedule timely mess meetings.*

❖ Fourth Increment: System Administration:

Features:

- ➢ *Implement System Configuration, Settings, Log, and Audit Trails for administrators.*

Working:

- ➢ *Allow administrators to configure system settings and customize mess-related parameters.*
- ➢ *Implement logging and audit trails for system activities.*

❖ Fifth Increment: Performance Optimization and Security Enhancements:

Features:

- ➢ *Address performance requirements, safety requirements, and security enhancements.*

Working:

- ➢ *Optimize system responsiveness, especially for features like viewing the mess menu by students.*

- ➢ *Enhance data safety by implementing security protocols and backup measures.*

❖ Sixth Increment: Future Implementations:

Features:

- ➢ *Integrate future enhancements like Transaction Integration, Automated Notifications, and Feedback Form/Food Review.*

Working:

- ➢ *Develop and integrate transaction capabilities for seamless payments within the system.*
- ➢ *Implement a notification system for upcoming events, announcements, and feedback requests.*

❖ Deployment of each new Version

- ➢ *For each increment, the software is deployed, and made available to alpha testers for feedback and performance review. The feedback is then utilized to refine and tune elements in the new incremental version.*

❖ Refinement and Completion:

Features:

- ➢ *Continue refining and enhancing the system based on feedback.*
- ➢ *Ensure all specified features are implemented, and the system is robust.*

Working:

- ➢ *Conduct final testing and quality assurance.*
- ➢ *Consider the Mess Management System complete and ready to be used in production, ready for deployment, and continuous improvement.*

Justifying Evolutionary Model for Mess Management System:

➔ **Dynamic Nature of Mess Management:**

The Evolutionary Model accommodates the dynamic nature of mess management by allowing ongoing adjustments based on real-time feedback and evolving needs, addressing changes in menu preferences, dietary requirements, and operational procedures.

➔ **Continuous Stakeholder Involvement:**

Involving various stakeholders, including students, faculty, and mess committee members, the Evolutionary Model encourages continuous feedback. This ensures the system aligns with practical user needs, enhancing stakeholder satisfaction and system effectiveness.

➔ **Adaptability to Evolving Requirements:**

Suited for projects with evolving or unclear requirements, the Evolutionary Model allows adaptability to changes in institutional policies and student preferences, ensuring the Mess Management System remains responsive to evolving needs.

➔ **Feedback-Driven Refinement:**

Facilitating continuous refinement based on user input, the Evolutionary Model ensures that the Mess Management System aligns with user expectations. This feedback-driven approach supports ongoing improvements in system functionality.

➔ **Reduced Planning Overhead:**

Requiring less upfront planning, the Evolutionary Model is advantageous in scenarios where detailed requirements may be unclear initially. This flexibility allows the development team to respond promptly to emerging needs, reducing planning overhead and promoting agility in development.