

Name of the Assignment: Applying SDLC Models in Real-World Context which is a local college cafeteria wants to modernize its food ordering system.

Why Agile is the best choice for a cafeteria ordering app: In software engineering, there are several Software Development Life Cycle (SDLC) models, including Waterfall, Iterative, and Agile. While each has its merits, the Agile model is the most appropriate choice for developing a college cafeteria's online ordering app.

The Waterfall model is a traditional approach where each project phase must be completed before the next one can begin. This model works best when all requirements are known and unlikely to change, and there is no need for flexibility or backtracking. However, there may arise new ideas, features throughout the development process. The Waterfall model's rigid, sequential structure makes it difficult and costly to accommodate such changes.

The Agile model is built on the principles of flexibility and changes are always welcome. It breaks down the project into some small sprints, which allows for changes to be incorporated at any stage. This flexibility is crucial for a real-world application that will likely need ongoing updates and improvements to meet user needs. Therefore, Agile is the appropriate model to ensure the project's success and adaptability over time.

Justification of Agile model:

The Agile model is an appropriate choice for this project for the following reasons:

Reduced Risk: In agile sprints process it is easy to find any bugs or issues which reduce the risk of the project.

Early and Continuous Delivery: The agile model there will be released first version of the project within few days where have the some important features. After completing each part of the project there will be introduced a new version of the project.

Customer Collaboration: Agile emphasizes close collaboration with the cafeteria manager and students. After releasing any version of the project if they find any issues then they can inform us and developer can solve this problem in the next version.

1. Planning:

The planning phase involves defining the initial scope and objectives of the project.

Project Objective: To develop a user-friendly Online Cafeteria Ordering App that improves efficiency, reduces waiting times, and enhances sales tracking for the college cafeteria.

Scope: The project will include an online ordering system for students and a real-time order management dashboard for cafeteria staff. Besides there students can see which food are

available now and which are finished and based on this information they can order there foods.

2. Requirement Analysis

This phase identifies and documents all the requirements for the app.

Functional Requirements:

- i. Students must be able to view available food items and their prices.
- ii. Students can place an order from classroom or library and pickup there foods when it will be prepared.
- iii. Each student order will be unique because there information will be saved when they register first.
- iv. Cafeteria staff must be able to view incoming orders in real time.
- v. The manager needs to view daily sales reports.

Non-functional Requirements:

- i. The app should be user-friendly and easily understandable UI design so that students can find all of those services or features without any confusing.
- ii. The system should be secure to protect user data and payment information.
- iii. The system performance will be very good that students will be satisfied in online food ordering services.
- iv. The system reliability will be very good that means students will get their food after finishing pick-up time.

3. Design

The design phase involves creating the system architecture and user interface.

i. System architecture: In our system architecture there have Role based access control system where have three four role such as student, teacher, staff and admin. Besides there have different table into database for registration and food ordering system.

ii. User Interface: The user interface will be very easy that means when students login his account then they can adds items to a cart, and submits their order. The system then processes the order and give them pickup time, and also displays those ordering information on the cafeteria staff's dashboard so that they can prepare those foods.

4. Implementation

The development team will write the code for the app based on the design specifications.

Technologies: After completing the design phase then the developers will start that project. The full projects will divide into some small scrum and different scrum will make different developer teams such as front-end team will the front-end parts of that project scrum and backend team will start the back-end parts of that project scrum and testing team will test the front-end and back-end part after completing. Front-end team can use different types of language and framework like React or Vue.js, tailwind CSS and so on. Back-end team can use different types of language and framework like Node.js with Express.js or PHP with Laravel and SQL or NO-SQL for database design.

5. Testing

This phase ensures the app functions are made as expectation of requirement and are free from defects.

Test Cases:

Verify the testing team all the functional and non-functional requirements will be worked properly or not.

- i. Confirm that a student can add and remove items from their cart.
- ii. Ensure an order is successfully placed and the pick-up time is generated properly or not.
- iii. Validate that the cafeteria staff can see new orders appear on their dashboard in real time.
- iv. Test that the daily sales reports are stored into database properly or not.

6. Deployment & Maintenance

After completing the testing phase the project will be deployed by DevOps team.

Deployment: The app can be deployed to a cloud platform like Amazon Web Services (AWS) or Google Cloud Platform (GCP) or Microsoft azure platform (MAP). Those cloud service 24 hour running our project and give more secure.

Maintenance: Post-deployment, the team will monitor the app for bugs, performance issues, and security vulnerabilities. If they will face any issues then the developer must be fixed those issues and will give new version. If the college cafeteria want in future add new feature and there will come also new feature.

Summary:

This assignment proposes using the Agile Software Development Life Cycle (SDLC) model to develop a modern online ordering system for a college cafeteria. In this assignment I explain each of the part with a wonderful and easy understanding format. Here I cover all of the phase of the SDLC model and why Agile is best model and explain all of the phase of the agile model. The app will allow students to view menus, order food for pickup, and will provide staff with a real-time order dashboard and sales reports.