

# CSCI 4210: Introduction to Software Engineering



University of New Orleans  
Department of Computer Science



## Capstone Project: Touchless Kiosk System

---

Dev-Cycle 1:  
MVP Development & Integration Readiness

---

## Cycle Overview

The MVP Development Cycle focuses on creating the Minimum Viable Product (MVP) by implementing isolated components and ensuring they are ready for integration. This phase will build essential features that form the foundation of a functioning system, aligning with Agile principles to iterate on a working product. Pods will complete tasks independently while adhering to guidelines for a unified codebase.

## Key Objectives

- **Component Isolation:** Develop system components independently to prepare them for seamless integration.
- **Establish Core Functionality:** Build essential features needed to create a functional MVP for user testing and feedback.
- **Integration Readiness:** Ensure that all components are compatible and ready for system-wide integration.

# Cycle Tasks

## Part 1: Core Feature Development

**Objective:** Implement the core functionalities identified by each pod in their requirements document. This includes building isolated components for each primary feature, without interdependency on other components, allowing for independent testing and refinement.

### 1. Develop Core Systems:

- Implement core system features as identified in the requirements, such as user management, scheduling, or data storage.
- Develop necessary APIs for these core systems, ensuring secure data handling and access control.
- Establish data persistence mechanisms for any required data, considering security and scalability.

### 2. Build Primary User Interfaces:

- Design and develop user interfaces that support the core functionalities, as specified by user stories and stakeholder needs.
- Implement basic UI components and layouts that align with the system requirements.
- Set up back-end support to connect UI components with core functionality.

### 3. Integrate Input/Output Devices (if applicable):

- Set up necessary hardware integrations (e.g., gesture controls, touchless sensors) if they are essential to the MVP.
- Develop preliminary data handling or processing for device input to support system interactions.

## Part 2: Integration Preparation

**Objective:** Prepare each component for integration by following coding standards and API conventions established in Cycle 0. This includes defining API documentation, conducting initial tests, and ensuring compatibility across components.

### 1. Establish API Conventions:

- Define REST API endpoints and request/response formats.
- Document API specifications for user authentication, dashboard, and gesture control.

### 2. Conduct Unit Tests for Isolated Components:

- Write unit tests to validate each core function.
- Ensure test coverage includes edge cases and error handling.

### 3. Implement Error Logging and Basic Security:

- Set up logging for error tracking and debugging.
- Apply basic security measures (e.g., input validation, authentication checks).

## Part 3: Collaborative Code Review and Refinement

**Objective:** Ensure code quality and compatibility through peer review sessions. This allows team members to align on best practices, catch potential issues early, and make refinements as needed.

### 1. Peer Review Sessions:

- Schedule code review sessions within each pod.
- Focus on adherence to standards, readability, and efficiency.

### 2. Refine Code Based on Feedback:

- Make improvements based on peer feedback.
- Document any changes to API or system functionality.

## Deliverables

- **Feature-Ready Components:** Code for each core feature with unit tests, API documentation, and error handling.
- **API Documentation:** Comprehensive documentation outlining each API endpoint, parameters, responses, and expected behavior.
- **Testing and Review Summary:** Summary report of unit tests, error logs, and peer review findings with noted improvements and adjustments.

## Submission Guidelines

All submissions and task progress will be tracked via the shared GitHub Projects Kanban Board. Each pod is responsible for updating the board to reflect the current status of tasks during the MVP Development cycle, ensuring collaborative visibility of progress.

- **GitHub Projects Kanban Board:** Tasks should be organized into columns such as "To Do," "In Progress," and "Completed" for clear tracking. Each task should be labeled with relevant tags and assigned to team members as appropriate.
- **Core Feature Implementations:** Code for each core feature must be integrated into the shared GitHub repository.
- **API Documentation:** Submit API documentation as a markdown file, clearly defining each endpoint and function.
- **Testing and Review Summary:** Each pod will submit a report summarizing unit tests, peer review outcomes, and any refinements made, included as a markdown file within the repository.

# Grading Rubric for Dev-Cycle 1

This rubric evaluates key aspects of MVP development, assessing functionality, integration readiness, documentation, and collaboration. It is designed to support the objectives of Cycle 1 while allowing students flexibility in how they organize their projects.

Category	Description	Points
<b>Core Feature Implementation</b>	Complete, functional code for each core feature or module, meeting requirements. Demonstrate effective component isolation, allowing each feature to be tested independently.	25%
<b>Testing and Integration Readiness</b>	Conduct thorough unit testing on each component, covering edge cases and error handling. Ensure components align with API conventions and are prepared for integration with other systems.	25%
<b>Documentation Quality</b>	Clear, thorough API documentation, including each endpoint, parameters, responses, and intended functionality. Summarize testing and review processes. Ensure each team member's contributions are evident in the documentation.	20%
<b>Peer Code Review</b>	Conduct peer code reviews within each pod. Submit a brief summary of findings, improvements made, and unresolved issues identified during reviews.	15%
<b>KanBan Board Management</b>	Maintain consistent updates to the GitHub Projects Kanban Board, with tasks organized by status (To Do, In Progress, Completed). Demonstrate active tracking of progress, labeling tasks, and assigning them to team members.	15%
<b>Total</b>		<b>100%</b>

## Detailed Breakdown

- **Core Feature Implementation (25%):** Features should meet functional requirements based on user stories and stakeholder needs. Each feature should operate independently, reflecting component isolation principles.
- **Testing and Integration Readiness (25%):** Unit tests should cover primary use cases, edge cases, and error handling. Components should be compatible with API conventions to facilitate future integration.
- **Documentation Quality (20%):** Each core feature must be accompanied by API documentation detailing its use and expected outputs. Include a Testing and Review Summary report detailing testing efforts and any code refinement based on peer reviews.
- **Peer Code Review (15%):** Conduct peer reviews, focusing on code readability, adherence to standards, and efficiency. Summarize the findings and document code changes based on feedback, listing unresolved issues if present.
- **GitHub Project Board Management (15%):** Maintain the project board with clear labeling of tasks by cycle and priority. Ensure all members participate in updating the board to reflect progress, assignments, and completion status.

## Appendix: Agile Development Roadmap

This Agile Development Roadmap provides an outline of each development cycle in the project, highlighting the objectives and activities central to each phase.

0. **Discovery and Planning Cycle** – This phase centers on requirements gathering, stakeholder analysis, and initial roadmap creation. It aligns with Agile’s discovery phases, where understanding user and system needs is essential for setting a clear development path.
1. **MVP Development and Integration Readiness Cycle** – This cycle emphasizes initial development, focusing on isolated component building and preparing for integration. The goal is to establish a minimal product foundation that can operate independently but is ready for further integration.
2. **User Feedback and Refinement Cycle** – After the MVP is complete, this phase collects user feedback, allowing for testing insights to guide iterative improvements. This cycle ensures the product evolves to meet real user needs effectively.
3. **Feature Expansion and Optimization Cycle** – Building on the MVP, this phase introduces high-priority features and optimizations. It expands the product’s functionality and enhances performance, creating a more robust and refined system.
4. **Deployment and Postmortem Analysis Cycle** – The final phase focuses on deploying the application, delivering a structured presentation, and conducting a post-mortem to reflect on the project’s journey and lessons learned.

Each cycle represents an integral stage in Agile development, from planning to deployment, ensuring continuous alignment with user needs and system goals.