The purpose of this workflow document is to help us organize the work and define clear stages that need to be followed for the successful release of our product.

Software Engineering Workflow Document

1. Objective

The purpose of this document is to define the structured workflow for the software engineering process and lifecycle.

2. Overview of the Workflow Stages

The software engineering process is broken down into the 8 phases that are followed in a step-by-step manner:

1. Vision
2. Requirement
3. Design
4. Implementation
5. Testing
6. Release
7. Maintenance/Support
8. Upgrades/Updates

3. Workflow Stages Breakdown

3.1 Planning & Requirement Gathering

- Objective**:** Gather detailed project requirements from stakeholders and create a project plan.
- Tasks**:**
    - Meet with stakeholders to understand business goals and user needs.
    - Define functional and non-functional requirements. Mandatory and non-mandatory.
    - Establish tools.
    - Create a Software Requirements Specification (SRS) document.
    - Groom and prioritize features based on business value in the PBI.
- Deliverables**:**
    - Requirements Document (SRS)
    - Project Plan
    - PBI
    - Sprints

3.2 Design

- Objective**:** Create a blueprint for how the software will be built and organized.
- Tasks**:**
    - Break down requirements into sprint-able tasks.
    - Create system architecture, data models, and interface designs.
    - Choose appropriate technologies and frameworks.
    - Design APIs, database schemas, and user interfaces (UI/UX).
    - Review design with the development team and PO
- Deliverables**:**
    - System Architecture
    - Database Design
    - API Documentation
    - UI/UX Design Mockups

3.3 Development

- Objective**:** Write the code that implements the functionality and requirements as per the design.
- Tasks**:**
    - Set up development environment (version control, IDE, etc.).
    - Implement features

  - Break down tasks into smaller work units if needed
  - Follow coding standards and best practices
      - code reviews, tests
  - Write unit tests alongside code where applicable.
  - Perform regular code commits and push to GitHub
- Deliverables**:**
  - Developed Codebase
  - Documentation of comments and commits
  - Unit Tests

3.4 Testing

- Objective**:** Ensure that the software meets the specified requirements and is free of bugs.
- Tasks**:**
  - Unit Testing**:** Ensure individual components work as expected.
  - Integration Testing**:** Ensure multiple components interact correctly.
  - System Testing**:** Verify the overall system functionality.
  - Acceptance Testing**:** Ensure the system meets the business requirements.
  - Identify and fix defects
  - Automate tests where possible
  - Perform load and performance testing.

- Deliverables**:**
  - Test Plan
  - Test Cases & Results
  - Bug Reports and documentation

3.5 Deployment

- Objective**:** Deploy the software to the production environment.
- Tasks**:**
  - Prepare the deployment
  - Deploy code to staging environment and verify functionality.
  - Deploy to production environment.
  - Monitor and check for issues.
- Deliverables**:**
  - Deployment Checklist

3.6 Maintenance

- Objective**:** Ensure that the software continues to perform well and remain functional after deployment.
- Tasks:
  - Monitor the system for bugs and performance issues.
  - Provide patches and updates to fix defects
  - Gather review feedback for improvements.

    - Maintain proper documentation.

- Deliverables**:**
    - Issue Documentation
    - Software Updates and Patches
    - Feedback

4. Tools and Technologies

- Version Control: GitHub,
- Project Management: Jira
- Language: JS and C#
- Collaboration Tools: Discord, Microsoft word

5. Roles and Responsibilities

- Product Owner: Defines project vision, manages requirements, and prioritizes tasks.
- SCRUM Master: Acts as a facilitator and coach for the Agile methodology and ensures the team follows Scrum principles and practices by removing obstacles.
- Developers: Write and test the code according to design specifications.
- Stakeholders**:** Review and approve requirements and final product.

6. Workflow Diagram

A visual representation of the workflow:

[Planning & Requirement Gathering]

↓

[Design]

↓

[Development]

↓

[Testing]

↓

[Deployment]

↓

[Maintenance]

GitHub workflow