CSP113-SOFTWARE ENGINEERING 2

Quality Assurance Plan Outline

A. **INTRODUCTION**

- **Description:** Overview of the QA plan, project context, and quality goals.

Example:

- *This plan outlines the quality assurance processes, standards, and responsibilities for the e-commerce mobile application to ensure reliability, performance, and user satisfaction.*

B. **OBJECTIVES OF QUALITY ASSURANCE**

- **Description:** Defines what the QA plan aims to achieve.

Example:

- *Ensure software meets customer requirements.*
- *Prevent defects through continuous testing and reviews.*
- *Maintain compliance with industry standards (e.g., ISO/IEC 25010).*

C. **SCOPE OF QUALITY ASSURANCE**

- **Description:** Specifies which aspects of quality are covered.

Example:

- *Functional correctness, usability, performance, reliability, maintainability, and security.*

D. **QUALITY STANDARDS AND METRICS**

- **Description:** Defines the standards and measurable indicators of quality.

Example:

- **Standards:** *IEEE 829 for test documentation, ISO 9001 for process quality.*
- **Metrics:**
  - *Defect density (defects per KLOC).*
  - *Test coverage (% of requirements tested).*
  - *Mean time to failure (MTTF).*

E. **QUALITY ASSURANCE ACTIVITIES**

- **Description:** Lists the QA methods and activities to ensure quality.

Example:

- **Process QA:** *Code reviews, inspections, pair programming.*
- **Product QA:** *Unit, integration, system, and acceptance testing.*
- **Automation QA:** *Automated regression testing using Selenium/JUnit.*

F. **ROLES AND RESPONSIBILITIES**

- **Description:** Defines the QA team's responsibilities.

Example:

- **QA Manager:** *Oversees QA process.*
- **Test Engineers:** *Write and execute test cases.*
- **Developers:** *Perform unit testing and fix defects.*
- **End-users/Clients:** *Participate in User Acceptance Testing (UAT).*

G. **TEST PLAN REFERENCE**

- **Description:** Links the QA plan to the detailed test plan.

Example:

- *QA activities will follow the test plan (document TP-ECOM-01), covering functional, usability, performance, and security testing.*

H. **DEFECT MANAGEMENT**

- **Description:** Defines how defects will be logged, tracked, and resolved.

Example:

- **Tools:** *Jira, Bugzilla.*
- **Severity Levels***: Critical, Major, Minor.*
- **Process***: Detect → Log → Assign → Fix → Retest → Close.*

I. **QUALITY CONTROL (QC) PROCEDURES**

- **Description:** Activities to verify outputs meet quality requirements.

Example:

- *Review code against coding standards.*
- *Conduct peer reviews and walkthroughs.*
- *Perform regression testing before each release.*

J. **Configuration Management**

- **Description:** Ensures all versions and changes are controlled.

Example:

- *GitHub/GitLab will be used for version control with a branching strategy and code review policies.*

K. **TOOLS AND TECHNIQUES**

- **Description:** Lists QA tools to be used.

Example:

- **Static Analysis:** *SonarQube.*
- **Automated Testing***: Selenium, JUnit, Postman.*
- **CI/CD Integration***: Jenkins.*

L. **RISK MANAGEMENT IN QA**

- **Description:** Identifies risks to software quality and mitigation strategies.

Example:

- **Risk:** *Incomplete test coverage → Mitigation: Use test coverage reports.*
- **Risk:** *Limited testing devices → Mitigation: Use cloud-based test environments.*

M. **MONITORING AND REPORTING**

- **Description:** Defines how QA progress and quality metrics will be tracked and reported.

Example:

- *Weekly QA reports including test execution status, defect trends, and quality metrics.*

N. **APPROVAL AND SIGN-OFF**

- **Description:** Defines who approves the QA plan and test results.

Example:

- *Project Manager, QA Lead, and Client Representative.*