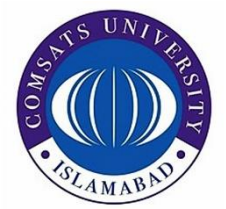
SOFTWARE TESTING

THE ROLE OF A TESTER IS TO BREAK THE SOFTWARE

Prepared For :
SIR MUKHTAIR ZAMIN

Prepared By :
BASIT IQBAL
FA21-BSE-050



Comsats University Islamabad Abbottabad Campus

INTRODUCTION

GROUP MEMBERS:

- BASIT IQBAL (FA21-BSE-050)

PROJECT NAME:

- GENETIC ALGORITHM

COURSE NAME:

- SOFTWARE TESTING

SUBMITTED TO:

- MUKHTAIR ZAMIN

UNIVERSITY:

- COMSATS UNIVERSITY ISLAMABAD ABBOTTABAD CAMPUS

SUBMISSION DATE:

- 09TH MAY 2024

ASSIGNMENT NUMBER:

- ASSIGNMENT 01

ASSIGNMENT NO: 01

Question:

What is the Test Plan. Explain the Test Plan Templates Evolution?

Answer:

Test Plan:
Test Plan is a formal document that act as a blueprint for the complete testing process, and it also ensures the quality and reliability of the Software.

Importance:

Test Plan is considered an important element because it guides the tester to perform testing in the most efficient way and maintain the log of the testing process. Also, it outlines the approach or strategy that would be used to test the Software and also to ensure the coverage of all the requirements and systematic evaluation of the software quality, ultimately decreasing the chances of risks and defects. And at last, empowering the reliability of the Software Product.

Evolutions:

Years	Templates
1983 to 2002	IEEE 829 Standard Template
2002 onward	ISTQB Template
Mid 2000's	Agile Test Plan Template
After 2010	Customized Templates
Ongoing / Today	Modern Templates

1. **IEEE 829 Standard Test Plan Template:**

Date: 1983

Overview:

It Provided a structured format for "Test Documentation".

Purpose of Development:

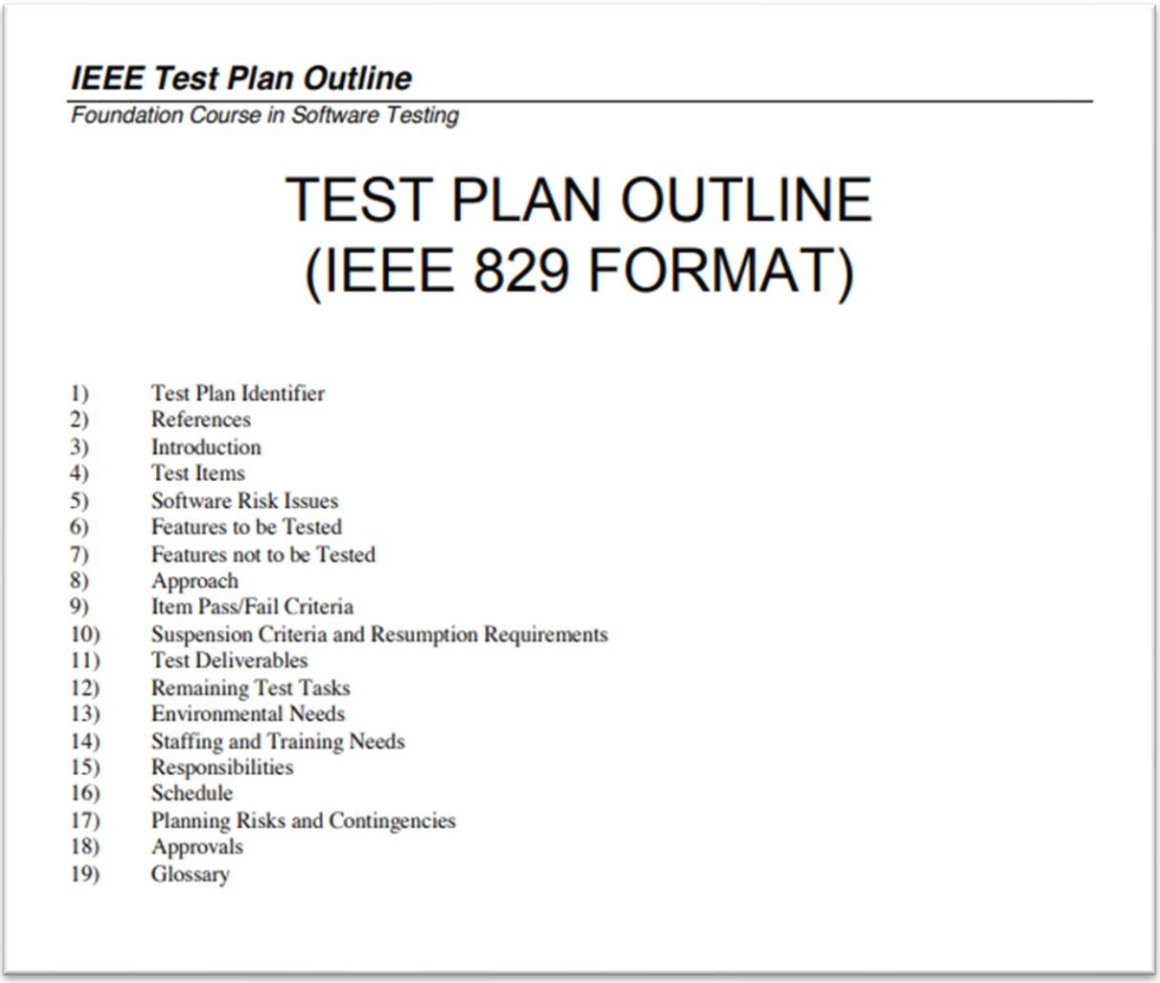
Issue:

When the IEEE 829 standard was not developed, there was a lack of standardized documentation practices in Software Testing. Test Plans varied widely in format, content, and level of detail, which lead to inconsistencies and inefficiencies.

Solution:

IEEE 829 provided a structured and standardized format for documenting test plans, ensuring consistency across different projects and organizations. This standardized approach improved communication and traceability in testing activities.

Template:



IEEE Test Plan Outline

Foundation Course in Software Testing

TEST PLAN OUTLINE
(IEEE 829 FORMAT)

1) Test Plan Identifier
2) References
3) Introduction
4) Test Items
5) Software Risk Issues
6) Features to be Tested
7) Features not to be Tested
8) Approach
9) Item Pass/Fail Criteria
10) Suspension Criteria and Resumption Requirements
11) Test Deliverables
12) Remaining Test Tasks
13) Environmental Needs
14) Staffing and Training Needs
15) Responsibilities
16) Schedule
17) Planning Risks and Contingencies
18) Approvals
19) Glossary

←----------→

2. ISTQB Test Plan Template:

Date: 2002

Overview:

The “International Software Testing Qualifications Board” (ISTQB) introduced a test plan template as a part of its certification syllabus.

Purpose of Development:

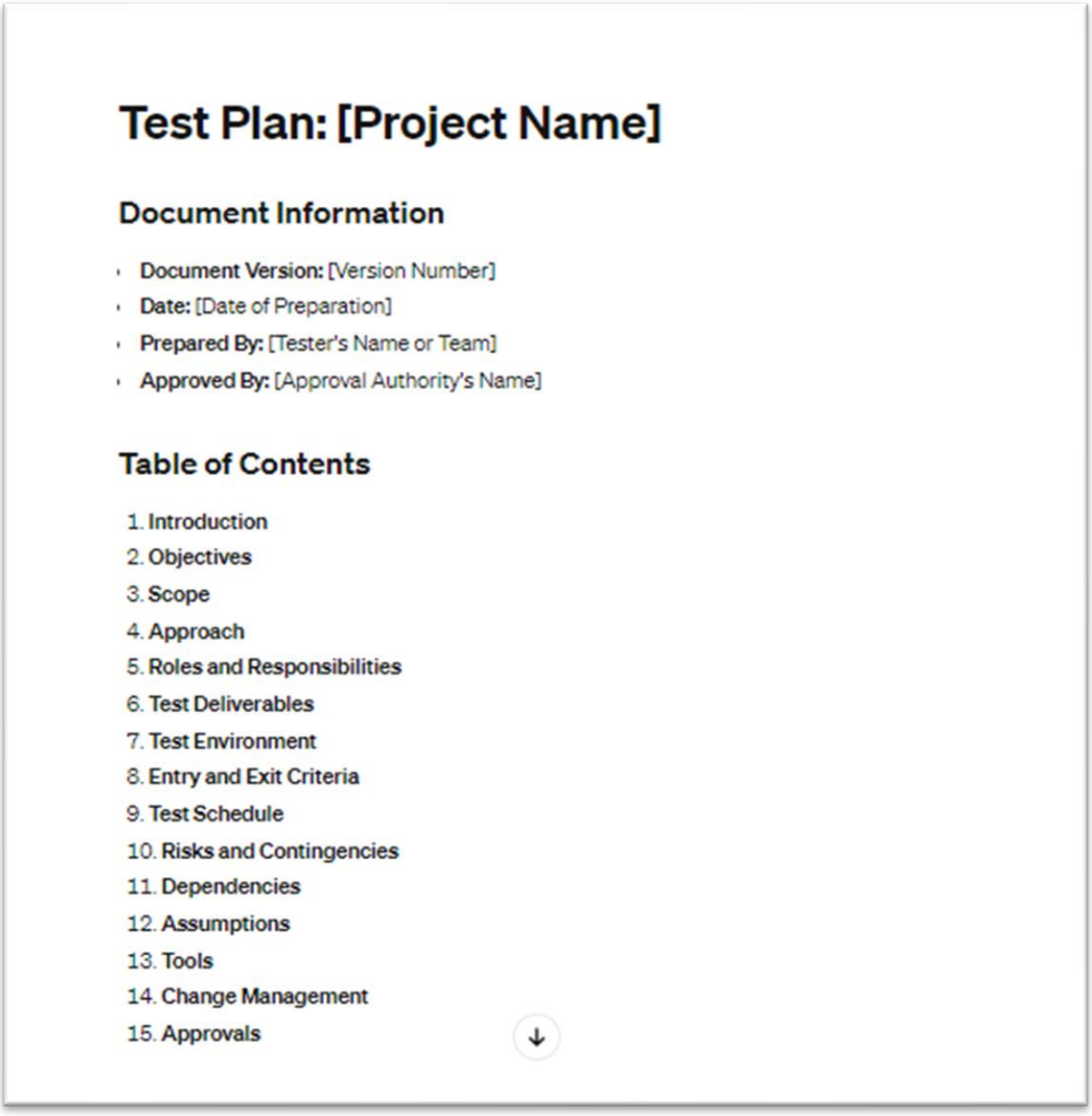
Issue:

Testing terminology and practices differed across regions and organizations, leading to confusion and miscommunication among testing professionals.

Solution:

The ISTQB test plan template established a common vocabulary and framework for test planning. It ensured a global understanding of Testing concepts and best practices, enhancing collaboration and knowledge sharing in the testing community.

Template:



Test Plan: [Project Name]

Document Information

- **Document Version:** [Version Number]
- **Date:** [Date of Preparation]
- **Prepared By:** [Tester's Name or Team]
- **Approved By:** [Approval Authority's Name]

Table of Contents

1. Introduction
2. Objectives
3. Scope
4. Approach
5. Roles and Responsibilities
6. Test Deliverables
7. Test Environment
8. Entry and Exit Criteria
9. Test Schedule
10. Risks and Contingencies
11. Dependencies
12. Assumptions
13. Tools
14. Change Management
15. Approvals

3. Agile Test Plan Template:

Date: Mid 2000's

Overview:

As in Mid 2000's agile methodologies got popular so the testing also evolved to be more lightweight and iterative.

Purpose of Development:

Issue:

Traditional Test Plans were heavily documented for Agile Development environments, where the requirements and priorities change frequently. Also, Agile development opposes heavy documentation and promotes less documentation approach.

Solution:

Agile test plan templates are lightweight, adaptable, and focused on essential testing activities. They align with Agile principles of flexibility and responsiveness to change, enabling testing to keep pace with rapid development iterations.

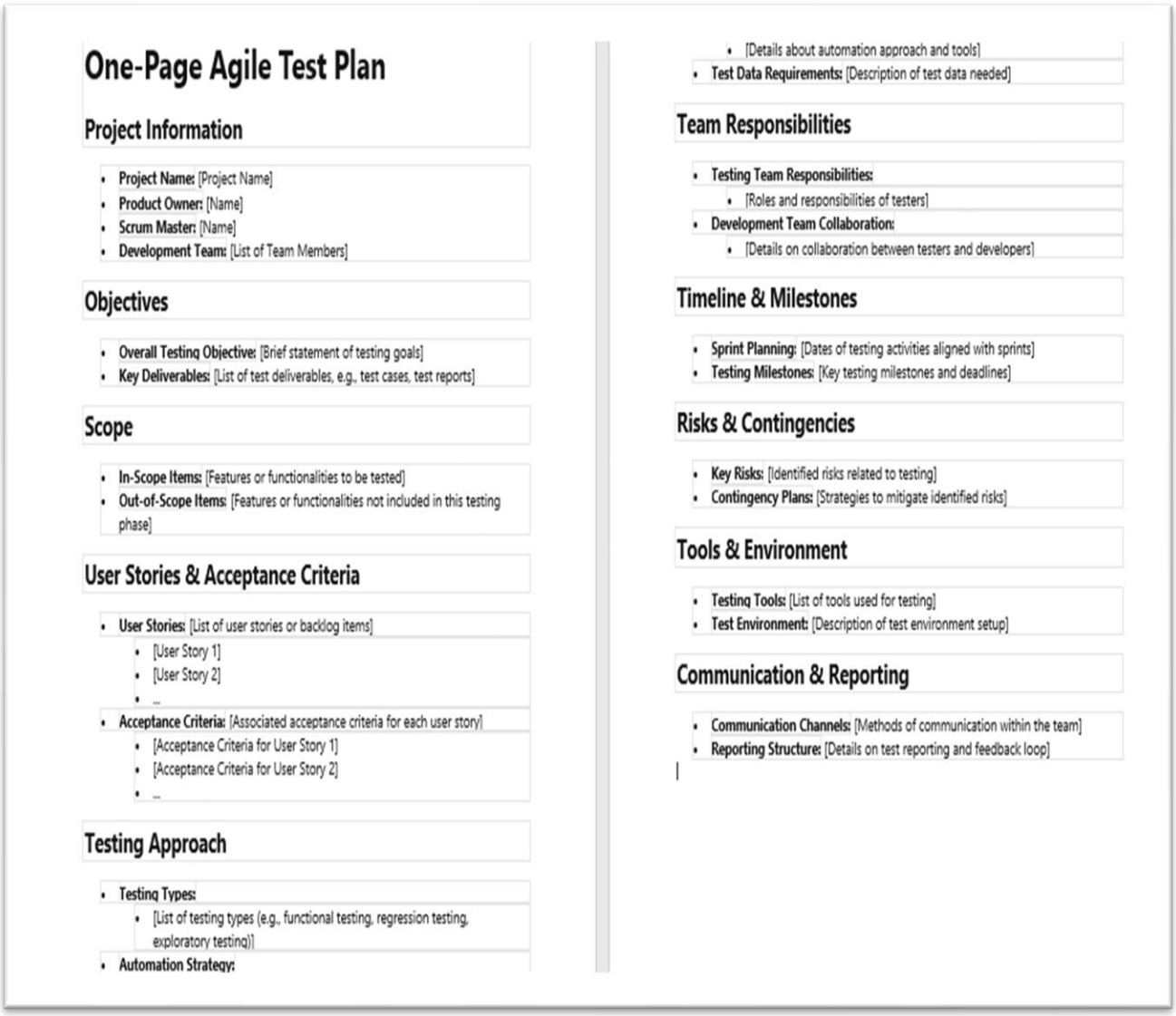
Templates:

There are various templates of Agile Test Plan:

i. One-Page Agile Test Plan.
ii. Test Strategy in Agile.
iii. User Story-Based Test Plan.
iv. Agile Test Pyramid.
v. Exploratory Test Planning.
vi. Continuous Testing Plan. Etc

And I have gathered some templates and pasted below.

1. One Page Agile Test Plan:



One-Page Agile Test Plan

Project Information

- **Project Name:** [Project Name]
- **Product Owner:** [Name]
- **Scrum Master:** [Name]
- **Development Team:** [List of Team Members]

Objectives

- **Overall Testing Objective:** [Brief statement of testing goals]
- **Key Deliverables:** [List of test deliverables, e.g., test cases, test reports]

Scope

- **In-Scope Items:** [Features or functionalities to be tested]
- **Out-of-Scope Items:** [Features or functionalities not included in this testing phase]

User Stories & Acceptance Criteria

- **User Stories:** [List of user stories or backlog items]
  - [User Story 1]
  - [User Story 2]
  - ...
- **Acceptance Criteria:** [Associated acceptance criteria for each user story]
  - [Acceptance Criteria for User Story 1]
  - [Acceptance Criteria for User Story 2]
  - ...

Testing Approach

- **Testing Types:**
  - [List of testing types (e.g., functional testing, regression testing, exploratory testing)]
- **Automation Strategy:**
  - [Details about automation approach and tools]
- **Test Data Requirements:** [Description of test data needed]

Team Responsibilities

- **Testing Team Responsibilities:**
  - [Roles and responsibilities of testers]
- **Development Team Collaboration:**
  - [Details on collaboration between testers and developers]

Timeline & Milestones

- **Sprint Planning:** [Dates of testing activities aligned with sprints]
- **Testing Milestones:** [Key testing milestones and deadlines]

Risks & Contingencies

- **Key Risks:** [Identified risks related to testing]
- **Contingency Plans:** [Strategies to mitigate identified risks]

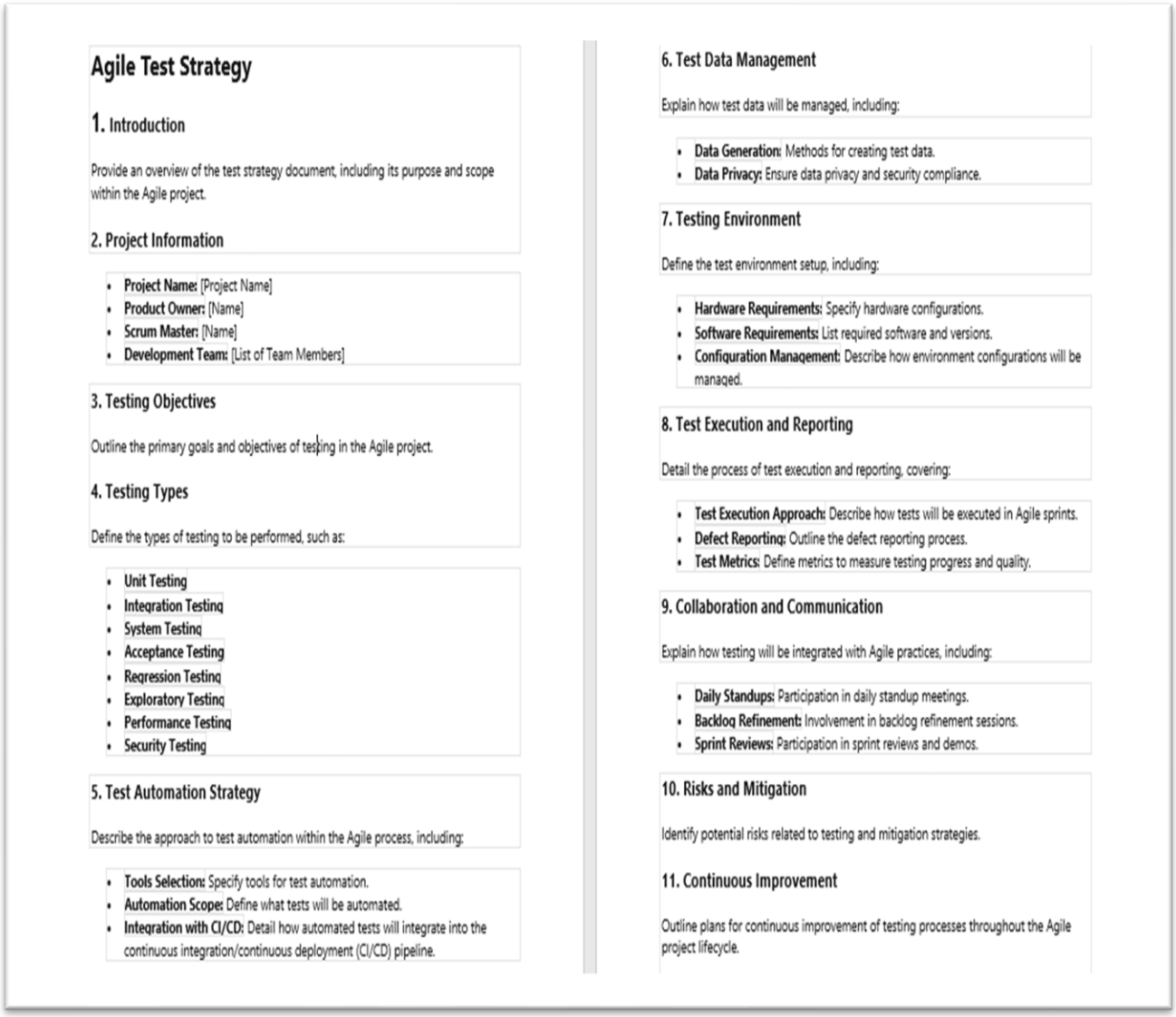
Tools & Environment

- **Testing Tools:** [List of tools used for testing]
- **Test Environment:** [Description of test environment setup]

Communication & Reporting

- **Communication Channels:** [Methods of communication within the team]
- **Reporting Structure:** [Details on test reporting and feedback loop]

2. Agile Test Plan Strategy:



Agile Test Strategy

1. Introduction

Provide an overview of the test strategy document, including its purpose and scope within the Agile project.

2. Project Information

- **Project Name:** [Project Name]
- **Product Owner:** [Name]
- **Scrum Master:** [Name]
- **Development Team:** [List of Team Members]

3. Testing Objectives

Outline the primary goals and objectives of testing in the Agile project.

4. Testing Types

Define the types of testing to be performed, such as:

- **Unit Testing**
- **Integration Testing**
- **System Testing**
- **Acceptance Testing**
- **Regression Testing**
- **Exploratory Testing**
- **Performance Testing**
- **Security Testing**

5. Test Automation Strategy

Describe the approach to test automation within the Agile process, including:

- **Tools Selection:** Specify tools for test automation.
- **Automation Scope:** Define what tests will be automated.
- **Integration with CI/CD:** Detail how automated tests will integrate into the continuous integration/continuous deployment (CI/CD) pipeline.

6. Test Data Management

Explain how test data will be managed, including:

- **Data Generation:** Methods for creating test data.
- **Data Privacy:** Ensure data privacy and security compliance.

7. Testing Environment

Define the test environment setup, including:

- **Hardware Requirements:** Specify hardware configurations.
- **Software Requirements:** List required software and versions.
- **Configuration Management:** Describe how environment configurations will be managed.

8. Test Execution and Reporting

Detail the process of test execution and reporting, covering:

- **Test Execution Approach:** Describe how tests will be executed in Agile sprints.
- **Defect Reporting:** Outline the defect reporting process.
- **Test Metrics:** Define metrics to measure testing progress and quality.

9. Collaboration and Communication

Explain how testing will be integrated with Agile practices, including:

- **Daily Standups:** Participation in daily standup meetings.
- **Backlog Refinement:** Involvement in backlog refinement sessions.
- **Sprint Reviews:** Participation in sprint reviews and demos.

10. Risks and Mitigation

Identify potential risks related to testing and mitigation strategies.

11. Continuous Improvement

Outline plans for continuous improvement of testing processes throughout the Agile project lifecycle.

3. User-Story Based Test Plan:

User Story-Based Test Plan

1. Project Information

- **Project Name:** [Project Name]
- **Product Owner:** [Name]
- **Scrum Master:** [Name]
- **Development Team:** [List of Team Members]

2. Overview

Provide a brief overview of the test plan, focusing on the approach to testing user stories and meeting acceptance criteria.

3. User Stories to be Tested

List the user stories or backlog items that will be tested in this phase.

User Story ID	User Story Description	Acceptance Criteria
US-001	[Description of User Story 1]	[Acceptance Criteria for User Story 1]
US-002	[Description of User Story 2]	[Acceptance Criteria for User Story 2]
...	...	...

4. Testing Approach

Describe the testing approach for each user story, including:

- **Testing Types:** Specify the types of testing (e.g., functional testing, regression testing).
- **Test Techniques:** Outline the testing techniques to be used (e.g., boundary value analysis, equivalence partitioning).
- **Automation Strategy:** Define the automation strategy for user story testing.

5. Test Scenarios

Outline the test scenarios derived from acceptance criteria for each user story.

User Story ID	Test Scenario Description	Test Steps	Expected Result
US-001	[Test Scenario for User Story 1]	[Steps to Execute the Test]	[Expected Outcome]
US-002	[Test Scenario for User Story 2]	[Steps to Execute the Test]	[Expected Outcome]
...	...	...	...

6. Test Data Requirements

Specify the test data needed to execute the test scenarios for each user story.

4. **Customized Organizational Templates:**

Date: Ongoing

Overview:

Almost all the organizations develop their own test plan templates tailored to their specific needs and processes.

Purpose of Development:

Issue:

Generic test plan templates might not be able to address the unique requirements, processes, and constraints of individual organizations or projects.

Solution:

Customized organizational templates allow companies to tailor test plans to their specific needs, in corporation industry standards, compliances requirements, and internal best practices. This customization improves efficiency and effectiveness in testing activities.

FAANG companies:

Almost all the FAANG companies are known for their emphasis on automation. Their test plans likely incorporate a high degree of automated testing throughout the development lifecycle.

All the FAANG companies including Microsoft, Google, and Amazon have their own custom Test Plan Template, so that they can tackle the problem in their own specific ways.

None of the templates of the FAANG companies are available because they are obviously not available publicly.

5. **Modern and DevOps-oriented Test Plans:**

Date: 2010s

Overview:

After the rise of DevOps and continuous testing, test plans have become more integrated into automated pipelines.

Purpose of Development:

Issue:

Traditional test plans were not designed to support continuous integration and continuous delivery (CI/CD) practices, leading to bottlenecks and delays in testing.

Solution:
Modern test plans for DevOps emphasize automation, scalability, and integration with CI/CD pipelines. They facilitate continuous testing throughout the software development lifecycle, ensuring rapid feedback and quality assurance in fast-paced delivery environments.

6. Industry Trends:

Shifted towards Automation:

More shifted toward automation testing within test plans to increase the efficiency and test coverage.

Integration with CI/CD Pipelines:

Test plans are integrated into continuous integration and delivery pipelines to support rapid and reliable software releases.

Focus on Traceability and Reporting:

Test plans include provisions for traceability matrices and detailed reporting to ensure transparency and accountability in testing activities.

Summary:

As I have explained the evolution in detail, now I would like to summarize the whole discussion. The evolution of test plans in software testing has progressed from early documentation standards to current focus on agility, automation , and integration with modern dev practices, Initially introduce with the IEEE 829 standard in the 1980s to address the lack of standardized docs, test plans have since evolve to include standardized templates such as mentioned in the ISTQB. With the introduction of Agile methodologies, test plans adapted to become more lightweight and flexible, focusing on iterative development and delivery. Customization became important as organizations tailored test plans to meet their specific needs, incorporating industry regulations tailored test plans to meet their specific needs, incorporating industry regulations tailored regulations and internal processes. In recent years, the integration of test plans with DevOps practices and continuous testing has become prominent, emphasizing automation, scalability, and rapid feedback. The evolution of test plans has not stopped and never will as change is inevitable and so the test plan will keep on evolving with the industry practices.

Conclusion:

Hence from all the above discussion, we can conclude the Test Plan templates were developed to keep everyone on same page and were done manually but today in the era of AI. They are mostly automated.

←----------→ *The End.....!*