# Mekelle University - YouTube

# MEKELLE UNIVERSITY

# ETHIOPIAN INSTITUTE OF TECHNOLOGY - MEKELLE

# SCHOOL OF COMPUTING

# DEPARTMENT OF SOFTWARE ENGINEERING

# Subject: SOFTWARE TESTING

Title: SOFTWARE REQUIREMENT SPECIFICATION

**Amare Birhane Assefa -----------------------------------------------------------ID No: URG/171995/12**

Submitted to: mesele

Submission day: 2/2/2025

Table of Contents

[**Software Testing Plan** 2](#_Toc187608608)

[1. TESTING PLANNING AND CONTROL 2](#_Toc187608609)

[1.1 TEST PLANNING 2](#_Toc187608610)

[1.1.1 OBJECTIVE 3](#_Toc187608612)

[1.1.2 COMPONENTS OF A TEST PLAN 3](#_Toc187608613)

[1.1.3 TOOLS FOR TEST PLANNING 3](#_Toc187608614)

[1.2 TEST CONTROL 3](#_Toc187608615)

[1.2.1 OBJECTIVES 3](#_Toc187608616)

[1.2.2 ACTIVITIES 3](#_Toc187608617)

[1.2.3 TOOLS FOR TEST CONTROL 3](#_Toc187608618)

[1.2.4 DELIVERABLES 4](#_Toc187608619)

[2. TESTING ANALYSIS AND DESIGN 4](#_Toc187608620)

[2.1 TEST ANALYSIS 4](#_Toc187608621)

[2.1.1 OBJECTIVES 4](#_Toc187608622)

[2.1.2 ACTIVITIES 4](#_Toc187608623)

[2.2 TEST DESIGN 4](#_Toc187608624)

[2.2.1 OBJECTIVES 4](#_Toc187608625)

[2.2.2 ACTIVITIES 4](#_Toc187608626)

[2.2.3 DELIVERABLES 5](#_Toc187608627)

[2.2.4 TOOLS FOR ANALYSIS AND DESIGN 5](#_Toc187608628)

**Software Testing Plan**

# 1. TESTING PLANNING AND CONTROL

## 1.1 TEST PLANNING

A **Testing Plan** is a formal document that outlines the strategy, scope, resources, and schedule for testing activities in a software development project. It serves as a roadmap for the testing process and provides a clear framework for ensuring the quality of the software product.

### 1.1.1 OBJECTIVE

🡪 facilitate regression testing – to check the new added not affect existing functionality

🡪 integration testing

* database connection
* - endpoints testing using postman

🡪 Security Testing

🡪 performance testing

🡪payment testing

🡪functional testing

### 1.1.2 COMPONENTS OF A TEST PLAN

* **Scope of Testing**: Defines what will be tested (features, functionalities) and what will not be included.
* **Test Objectives**: Specifies what the testing aims to achieve (e.g., verifying functionality, performance).
* **Testing Strategy**: Outlines the overall approach (e.g., manual vs. automated testing).
* **Resources**: Identifies the team members involved in testing, their roles, and responsibilities.
* **Schedule**: Provides a timeline for testing activities, including milestones and deadlines.
* **Risk Management**: Analyzes potential risks in the testing process and proposes mitigation strategies.

### 1.1.3 TOOLS FOR TEST PLANNING

* **Test Management Tools**: Tools like Jira, TestRail, or Zephyr help organize test cases, track progress, and manage documentation.
* **Collaboration Tools**: Tools like Confluence or Slack facilitate communication among team members.
* **Requirement Management Tools**: Tools like IBM DOORS or Helix RM help manage and trace requirements.

## 1.2 TEST CONTROL

### 1.2.1 OBJECTIVES

Identify key controls

Assess assurance

Control of Test Execution

Improve risk management

Review and Adjust

Ensure quality

Quality Assurance

Communication Plan

### 1.2.2 ACTIVITIES

* **Progress Monitoring**: Regularly check the status of testing activities against the plan.
* **Issue Management**: Identify and address issues that arise during testing, adjusting the plan as necessary.
* **Test Reporting**: Create reports that summarize testing activities, outcomes, and outstanding issues.

### 1.2.3 TOOLS FOR TEST CONTROL

* **Dashboards**: Tools like Jira or Azure DevOps provide visual representations of testing progress.
* **Reporting Tools**: Tools like TestRail or Excel can be used to generate detailed test reports.

### 1.2.4 DELIVERABLES

* **Test Plan Document**: A comprehensive document that outlines the test strategy, scope, and schedule.
* **Test Progress Reports**: Regular updates that highlight the status of testing activities, issues encountered, and resolutions.

# 2. TESTING ANALYSIS AND DESIGN

## 2.1 TEST ANALYSIS

### 2.1.1 OBJECTIVES

Understand Requirements

**Identify Testable Criteria**

Design Test Cases

Identify Test Data

Establish Traceability

### 2.1.2 ACTIVITIES

* **Requirement Analysis**: Review software requirements to ensure clarity and completeness.
* **Risk Assessment**: Identify high-risk areas that require more rigorous testing.
* **Test Identification**: Determine the types of tests needed (functional, performance, security).

## 2.2 TEST DESIGN

### 2.2.1 OBJECTIVES

Test design aims to create detailed test cases and scripts that will effectively validate the software against its requirements.

### 2.2.2 ACTIVITIES

* **Test Case Design**: Develop test cases that specify inputs, execution conditions, and expected results.
* **Test Data Preparation**: Identify and prepare the necessary test data for executing test cases.
* **Test Environment Setup**: Configure the testing environment, ensuring it mimics the production environment.

### 2.2.3 DELIVERABLES

* **Test Cases**: Documented test cases that outline the testing approach for each requirement.
* **Test Data**: Prepared datasets that will be used during testing.
* **Test Environment Documentation**: Details about the configuration and setup of the testing environment.

### 2.2.4 TOOLS FOR ANALYSIS AND DESIGN

* **Test Case Management Tools**: Tools like TestRail or QTest help in documenting and managing test cases.
* **Requirement Management Tools**: Tools like Jama Connect or Helix RM assist in tracking requirements and their corresponding test cases.
* **Automation Tools**: Tools like Selenium or JUnit can help automate the execution of test cases.