Test Strategy Document

Title: Test Strategy

Date:	2026-02-16
Classification:	Confidential
Version:	1.0

1. Introduction

1.1. Purpose

The purpose of this test strategy document is to outline the approach and methodology for testing the VWO (Visual Website Optimizer) Digital Experience Optimization Platform. The goal is to ensure that the platform meets the required functional, performance, security, and scalability standards.

1.2. Scope

The scope of this test strategy includes testing of the VWO platform, focusing on functional, performance, security, and API aspects. The testing will cover the entire platform, including user interfaces, backend services, and integrations.

1.3. Objectives

The objectives of this test strategy are to:

• Ensure that the VWO platform meets the functional requirements outlined in the Product Requirements Document (PRD)

• Validate the performance and scalability of the platform under various loads and conditions

• Verify the security and compliance of the platform with respect to GDPR and other relevant regulations

• Develop and execute automated tests to ensure regression coverage and reduce manual testing effort

1.4. References

The following documents and resources will be used as references for this test strategy:

• Product Requirements Document (PRD) for VWO

• VWO platform architecture and design documents

• Relevant industry standards and regulations (e.g., GDPR, OWASP)

2. Project Overview

2.1. Project Description

The VWO Digital Experience Optimization Platform is an enterprise-grade solution designed to help businesses optimize their digital experiences. The platform provides features such as A/B testing, personalization, and analytics.

2.2. Key Stakeholders

The key stakeholders for this project include:

• Product owners and managers

• Development teams

• Quality assurance teams

• Operations and infrastructure teams

• Security and compliance teams

2.3. Application Architecture Overview

The VWO platform consists of a web-based user interface, backend services, and integrations with third-party services. The platform is built using a microservices architecture, with multiple services communicating with each other through APIs.

2.4. Key Business Flows

The key business flows for the VWO platform include:

• User onboarding and authentication

• Campaign creation and management

• A/B testing and experimentation

• Personalization and recommendation

• Analytics and reporting

3. Testing Approach

3.1. Overall Testing Philosophy

The overall testing philosophy for this project is to ensure that the VWO platform meets the required functional, performance, security, and scalability standards. The testing approach will be risk-based, with a focus on identifying and mitigating high-risk areas.

3.2. Testing Levels

3.2.1. Unit Testing

Unit testing will be performed at the code level, focusing on individual components and services. The goal is to ensure that each unit of code functions as expected and meets the required standards.

3.2.2. Integration Testing

Integration testing will be performed to validate the interactions between different components and services. The goal is to ensure that the platform functions as expected when different components and services are integrated.

3.2.3. System Testing

System testing will be performed to validate the entire platform, including user interfaces, backend services, and integrations. The goal is to ensure that the platform meets the required functional, performance, and security standards.

3.3. Testing Types

3.3.1. Functional Testing

Functional testing will be performed to validate that the platform meets the required functional standards. This will include testing of user interfaces, backend services, and integrations.

3.3.2. Regression Testing

Regression testing will be performed to ensure that changes to the platform do not introduce new defects or affect existing functionality.

3.3.3. Performance Testing

Performance testing will be performed to validate the performance and scalability of the platform under various loads and conditions.

3.3.4. Security Testing

Security testing will be performed to validate the security and compliance of the platform with respect to GDPR and other relevant regulations.

3.3.5. API Testing

API testing will be performed to validate the APIs used by the platform, including API security and performance.

3.3.6. Accessibility Testing

Accessibility testing will be performed to ensure that the platform is accessible to users with disabilities.

3.3.7. Compatibility Testing

Compatibility testing will be performed to ensure that the platform is compatible with different browsers, devices, and operating systems.

3.3.8. Disaster Recovery & Failover Testing

Disaster recovery and failover testing will be performed to ensure that the platform can recover from failures and disasters.

4. Automation

4.1. Automation Approach

The automation approach for this project will be to use a combination of commercial and open-source tools to automate testing. The goal is to automate as much testing as possible, while still maintaining a balance between automation and manual testing.

4.2. Automation Framework & Tools

The automation framework and tools used for this project will include:

• Selenium for web-based testing

• Appium for mobile-based testing

• JUnit and TestNG for unit testing

• Cucumber and SpecFlow for behavior-driven development (BDD)

4.3. Automation Coverage Targets

The automation coverage targets for this project will include:

• 80% of unit tests automated

• 70% of integration tests automated

• 60% of system tests automated

5. Environment and Data

5.1. Environment Topology

The environment topology for this project will include:

• Development environment

• Testing environment

• Staging environment

• Production environment

5.2. Test Data Management

Test data management will be performed using a combination of manual and automated approaches. The goal is to ensure that test data is accurate, complete, and secure.

5.3. Environment Management

Environment management will be performed using a combination of manual and automated approaches. The goal is to ensure that environments are properly configured, maintained, and secured.

6. Defect Management

6.1. Defect Lifecycle

The defect lifecycle for this project will include:

• Defect identification

• Defect reporting

• Defect prioritization

• Defect assignment

• Defect resolution

• Defect verification

6.2. Defect Severity & Priority Matrix

The defect severity and priority matrix for this project will include:

• Critical: defects that affect the functionality or security of the platform

• High: defects that affect the performance or usability of the platform

• Medium: defects that affect the appearance or user experience of the platform

• Low: defects that are cosmetic or minor

6.3. Defect SLA & Resolution Targets

The defect SLA and resolution targets for this project will include:

• Critical defects: resolved within 24 hours

• High defects: resolved within 3 days

• Medium defects: resolved within 5 days

• Low defects: resolved within 7 days

6.4. Defect Tracking & Reporting

Defect tracking and reporting will be performed using a combination of manual and automated approaches. The goal is to ensure that defects are properly tracked, reported, and resolved.

7. Risk Management

7.1. Risk Assessment Framework

The risk assessment framework for this project will include:

• Risk identification

• Risk analysis

• Risk prioritization

• Risk mitigation

7.2. Risk Register

The risk register for this project will include:

• Security risks

• Performance risks

• Scalability risks

• Compliance risks

7.3. Test Prioritization Based on Risk

Test prioritization will be based on risk, with high-risk areas receiving higher priority.

8. Entry and Exit Criteria

8.1. Entry Criteria

The entry criteria for this project will include:

• Completion of unit testing

• Completion of integration testing

• Completion of system testing

8.2. Exit Criteria

The exit criteria for this project will include:

• Completion of all testing activities

• Resolution of all critical and high defects

• Obtaining approval from stakeholders

9. Metrics and Reporting

9.1. Key Quality Metrics

The key quality metrics for this project will include:

• Defect density

• Test coverage

• Test efficiency

• Customer satisfaction

9.2. Reporting Cadence

Reporting will be performed on a regular basis, including:

• Daily reports

• Weekly reports

• Monthly reports

9.3. Test Management & Reporting Tools

Test management and reporting tools will include:

• JIRA

• TestRail

• Excel

11. Schedule

11.1. High-Level Test Schedule

The high-level test schedule for this project will include:

• Unit testing: 2 weeks

• Integration testing: 3 weeks

• System testing: 4 weeks

• Regression testing: 2 weeks

11.2. Key Milestones & Quality Gates

The key milestones and quality gates for this project will include:

• Completion of unit testing

• Completion of integration testing

• Completion of system testing

• Obtaining approval from stakeholders

12. Communication and Escalation

12.1. Communication Matrix

The communication matrix for this project will include:

• Daily stand-up meetings

• Weekly team meetings

• Monthly stakeholder meetings

12.2. Escalation Path

The escalation path for this project will include:

• Team lead

• Project manager

• Stakeholders

13. Assumptions, Constraints, and Dependencies

13.1. Glossary

The glossary for this project will include:

• VWO: Visual Website Optimizer

• PRD: Product Requirements Document

• GDPR: General Data Protection Regulation

13.2. Assumptions

The assumptions for this project will include:

• The VWO platform will be developed using a microservices architecture

• The platform will be deployed on a cloud-based infrastructure

• The platform will be used by a large number of users

13.3. Constraints

The constraints for this project will include:

• Time and budget constraints

• Resource constraints

• Technical constraints

13.4. Dependencies

The dependencies for this project will include:

• Development team

• Operations team

• Security team

• Stakeholders

Note: The above test strategy document is based on the provided JIRA project context and may require modifications based on the actual project requirements and constraints.