

# 2023 Agile Test Plan and Strategy Mind Map Template

## Test Strategy

- Project Details**
  - Enter your project name
  - Start Date - End Date
  - Launch Date
- Project Objectives**
  - 🎯 Quality Goals
  - 👥 Team Structure

## Manual Strategy

- Testing Types**
  - ✅ Functional Testing
  - ✅ System Testing
  - ✅ Integration Testing
  - ✅ Exploratory Testing
  - ✅ Risk-Based Testing
- Test Environments**
  - 1 DEV
  - 2 QA
  - 3 UAT
  - 4 PROD
- Testing Tools**
- Reporting**
- Risk Analysis**
  - Limitations
  - Risks
  - Assumptions
  - Constraints
- Definition of Done**
  - Feature List
    - Login
    - Register
    - Email App
  - Product Coverage Outline (Template)
    - Data
    - Structure
    - Function
    - Operations
    - Interface
    - Platform
    - Time
- Platforms**
  - WEB
    - 1 Chrome
    - 2 Safari
    - 3 Edge
    - 4 Firefox
  - MOBILE
    - iPhone
    - iPad
    - Android
- Automation Strategy**
  - Code Management
  - API Automation Framework
  - Front-End Automation Framework
  - Reporting: Allure Reporting Tool
  - Risk Analysis
  - Testing Types
  - Automation Execution

## Useful Links/References

- Reports**
  - Test Coverage Report
  - Test Summary Report
- Metrics/KPIs**
  - Test Coverage

# 2023 Agile Test Plan and Strategy Mind Map Template

Please duplicate this mind map in the mindmeister main menu to be able to edit.

## 1. Project Objectives

### 1.1. Quality Goals

1.1.1. Example: To achieve 100% compliance with the internal quality standards by the end of next quarter

1.1.2. Example: To increase customer satisfaction by 20% for the next two quarters

1.1.3. Example: To reduce product defects by 25% within the next 6 months

1.1.4. Example: To improve on-time delivery by 5% within the next 6 months

### 1.2. Team Structure

#### 1.2.1. Internal Team

##### 1.2.1.1. Senior Solutions Architect

1.2.1.1.1. <Insert Names>

##### 1.2.1.2. Front End Developer

##### 1.2.1.3. Back End Developer

1.2.1.3.1. <Insert Names>

##### 1.2.1.4. Program Manager

1.2.1.4.1. <Insert Names>

##### 1.2.1.5. Senior QA Specialist

1.2.1.5.1. <Insert Names>

## 2. Project Details

2.1. Enter your project name

2.2. Start Date - End Date

2.3. Launch Date

## 3. Test Strategy

### 3.1. Manual Strategy

#### 3.1.1. Testing Types

3.1.1.1. :white\_check\_mark: Functional Testing

3.1.1.1.1. :white\_check\_mark: Acceptance Testing

3.1.1.1.2. :white\_check\_mark: End-to-End Testing

3.1.1.1.3. :white\_check\_mark: UI Testing

3.1.1.2. :white\_check\_mark: System Testing

3.1.1.2.1. :white\_check\_mark: Sanity Testing

3.1.1.2.2. :white\_check\_mark: Smoke Testing

3.1.1.2.3. :white\_check\_mark: Regression Testing

3.1.1.3. :white\_check\_mark: Integration Testing

3.1.1.4. :white\_check\_mark: Exploratory Testing

3.1.1.5. :white\_check\_mark: Risk-Based Testing

3.1.2. Test Environments

3.1.2.1. :one: DEV

3.1.2.1.1. <Insert Links>

3.1.2.2. :two: :star: QA

3.1.2.2.1. Front End

3.1.2.2.1.1. <Insert Links>

3.1.2.2.2. Back End

3.1.2.2.2.1. <Insert Links>

3.1.2.3. :three: :star: UAT

3.1.2.3.1. <Insert Links>

3.1.2.4. :four: PROD

3.1.2.4.1. <Insert Links>

3.1.3. Testing Tools

3.1.3.1. Test Case Management

3.1.3.1.1. Testrail

3.1.3.2. Defect Tracking

3.1.3.2.1. JIRA

3.1.3.2.2. <Insert Link>

3.1.3.3. Web & Mobile Testing

3.1.3.3.1. Browserstack

3.1.3.3.2. <Insert Link>

### 3.1.3.4. Performance and Load Testing

#### 3.1.3.4.1. Loadster

#### 3.1.3.4.2. <Insert Link>

### 3.1.4. Reporting

#### 3.1.4.1. Test cases

#### 3.1.4.2. Test reports

#### 3.1.4.3. Release reports

### 3.1.5. Risk Analysis

#### 3.1.5.1. Limitations

##### 3.1.5.1.1. Eg: Time & QA Resources

#### 3.1.5.2. Risks

##### 3.1.5.2.1. Stability and creations of environments

#### 3.1.5.3. Assumptions

#### 3.1.5.4. Constraints

### 3.1.6. Definition of Done

#### 3.1.6.1. Feature List

##### 3.1.6.1.1. Login

##### 3.1.6.1.2. Register

##### 3.1.6.1.3. Email App

#### 3.1.6.2. Product Coverage Outline (Template)

**Link:** <https://www.mindmeister.com/1966583547>

Inspired by James Bach  
"Heuristic Test Strategy Model"

- Duplicate this mind map to begin.

##### 3.1.6.2.1. Data

Everything that the product processes.

###### 3.1.6.2.1.1. Input/Output

###### 3.1.6.2.1.2. Preset

### 3.1.6.2.2. Structure

\_Everything that comprises the physical product. \_

#### 3.1.6.2.2.1. Code Structures

The code structures that comprise the product, from executables to individual routines.

#### 3.1.6.2.2.2. Hardware Components

Any files other than multimedia or programs, like text files, sample data, or help files

##### 3.1.6.2.2.2.1. Examples

3.1.6.2.2.2.1.1. Text Files

3.1.6.2.2.2.1.2. Sample Data

3.1.6.2.2.2.1.3. Help Files

#### 3.1.6.2.2.3. Non-executable files

Any hardware component that is integral to the product.

#### 3.1.6.2.2.4. Infrastructure notes

##### 3.1.6.2.2.4.1. Web

3.1.6.2.2.4.1.1. Servers

3.1.6.2.2.4.1.1.1. Delivery 1

3.1.6.2.2.4.1.1.2. Delivery 2

3.1.6.2.2.4.1.2. Coveo

3.1.6.2.2.4.1.3. Databases

3.1.6.2.2.4.1.3.1. Web

3.1.6.2.2.4.1.3.2. Core

3.1.6.2.2.4.1.3.3. ActiveCommerce

### 3.1.6.2.3. Function

Everything that the product does.

#### 3.1.6.2.3.1. Application

#### 3.1.6.2.3.2. Calculation

3.1.6.2.3.3. Time-related

3.1.6.2.3.4. Testability

3.1.6.2.3.5. Interactions

3.1.6.2.3.6. Error Handling

3.1.6.2.3.7. Multimedia

3.1.6.2.3.8. Startup/Shutdown

3.1.6.2.3.9. Transformations

3.1.6.2.3.10. Security-related

#### 3.1.6.2.4. Operations

How the product will be used.

3.1.6.2.4.1. Users

3.1.6.2.4.2. Environment

3.1.6.2.4.3. Common Use

3.1.6.2.4.4. Extreme Use

3.1.6.2.4.5. Disfavored Use

#### 3.1.6.2.5. :star: Interface

\_Every conduit by which the product is accessed or expressed. \_

3.1.6.2.5.1. User Interfaces

Any element that mediates the exchange of data with the user (e.g. displays, buttons, fields, whether physical or virtual).

3.1.6.2.5.2. System Interfaces

3.1.6.2.5.3. API/SDK

3.1.6.2.5.4. Import/Export

#### 3.1.6.2.6. Platform

Everything on which the product depends (and that is outside your project)

3.1.6.2.6.1. External Hardware

3.1.6.2.6.2. External Software

3.1.6.2.6.3. Embedded Components

#### 3.1.6.2.6.4. Product Footprint

#### 3.1.6.2.7. Time

Any relationship between the product and time.

##### 3.1.6.2.7.1. Input/Output

##### 3.1.6.2.7.2. Fast/Slow

##### 3.1.6.2.7.3. Changing Rates

##### 3.1.6.2.7.4. Concurrency

#### 3.1.7. Platforms

##### 3.1.7.1. WEB

###### 3.1.7.1.1. :one: Chrome

###### 3.1.7.1.2. :two: Safari

###### 3.1.7.1.3. :three: Edge

###### 3.1.7.1.4. :four: Firefox

##### 3.1.7.2. MOBILE

###### 3.1.7.2.1. iPhone

###### 3.1.7.2.2. iPad

###### 3.1.7.2.3. Android

#### 3.2. Automation Strategy

##### 3.2.1. Code Management

###### 3.2.1.1. Eg: Gitlab

##### 3.2.2. API Automation Framework

###### 3.2.2.1. Eg: Data Driven Approach

##### 3.2.3. Front-End Automation Framework

###### 3.2.3.1. Web

###### 3.2.3.1.1. Eg: Webdriver.io, Cypress.io

###### 3.2.3.2. Mobile

###### 3.2.3.2.1. Eg: Appium

##### 3.2.4. Reporting

###### 3.2.4.1. Allure Reporting Tool

### 3.2.5. Risk Analysis

#### 3.2.5.1. Limitations

3.2.5.1.1. Eg: Time & QA Resources

#### 3.2.5.2. Risks

3.2.5.2.1. Stability and creations of environments

### 3.2.6. Testing Types

#### 3.2.6.1. :unlock: Non Functional Testing

3.2.6.1.1. :unlock: Performance Testing

3.2.6.1.1.1. :white\_check\_mark: Load Testing

#### 3.2.6.2. Functional Testing

3.2.6.2.1. Unit testing

3.2.6.2.2. :lock: Integration Testing

3.2.6.2.3. :lock: System Testing

3.2.6.2.3.1. Sanity Testing

3.2.6.2.3.2. Smoke Testing

3.2.6.2.3.3. Regression Testing

3.2.6.2.4. :lock: End-to-End Testing

#### 3.2.6.3. :white\_check\_mark: Integration Testing

3.2.6.3.1. API Testing

### 3.2.7. Automation Execution

3.2.7.1. Load Testing

3.2.7.2. Performance Testing

3.2.7.3. Integration Testing

## 4. Useful Links/References

## 5. Metrics/KPIs

5.1. Test Coverage

## 6. Reports

6.1. Test Coverage Report

6.2. Test Summary Report