September 9, 2025

SWAPNAJIT MOHANTY

QA Specialist

Toronto, ON, Canada

Test PlAn

CIVICTECHTO WEBSITE MIGRATION

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description of Change |
| V1 | 09-09-2025 | Swapnajit Mohanty | Initial draft for submission |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Document History**

Contents

[1. Objective 3](#_Toc209288117)

[2. Project Overview 3](#_Toc209288118)

[3. Features to be Tested 3](#_Toc209288119)

[4. Features Not to be Tested 4](#_Toc209288120)

[5. Test Strategy 4](#_Toc209288121)

[5.1 Test Approach 4](#_Toc209288122)

[5.2 Testing Tools 5](#_Toc209288123)

[5.3 Defect Management 5](#_Toc209288124)

[6. Test Environment 6](#_Toc209288125)

[7. Test Data 6](#_Toc209288126)

[8. Test Dependencies 6](#_Toc209288127)

[9. Deliverables 6](#_Toc209288128)

[10. Entry & Exit Criteria 7](#_Toc209288129)

[11. Risks & Mitigation 7](#_Toc209288130)

[12. Assumptions 7](#_Toc209288131)

# 1. Objective

The objective of this Test Plan is to define the testing strategy, scope, approach, resources, timelines, and responsibilities to ensure that the CivicTechTO website migration from WordPress to GitHub Pages meets quality standards. It provides a structured approach to validate functionality, content, user experience, accessibility, performance, and deployment integrity.

# 2. Project Overview

* **Project Name:** CivicTechTO Website Migration
* **Project Description:** CivicTechTO is a Toronto-based volunteer community of technologists, designers, and civic innovators who work on open-source projects that benefit the city and its residents. The project involves migrating the existing WordPress website to a static site hosted on GitHub Pages using Ruby and Jekyll.
* **Technology Stack:** Ruby, Jekyll (static site generator), GitHub Pages (deployment), Bundler for dependency management.
* **Goal:**
  + Successfully migrate existing content and functionality to the new static site.
  + Validate stability, usability, and correctness post-migration.
  + Ensure navigation, events, projects, resources, theme switching, and external integrations (Slack, Guild, social media) work as intended
* **Target Users:** Volunteers, civic innovators, tech enthusiasts, general public, potential partners/non-profits

# 3. Features to be Tested

* Core pages, menus, buttons, and navigation flows
* Migrated Content (text, images, media)
* User Experience and Accessibility
* Deployment and CI/CD Pipeline validation
* External integrations (Slack, Guild, social media links)

# 4. Features Not to be Tested

* Backend database validations (site is static).
* Slack/Guild/third-party platform internal workflows beyond simple redirects.

# 5. Test Strategy

## 5.1 Test Approach

The test approach defines the scope and direction of the test effort.

**Step -1: Test Design & Preparation**

1. Review migration scope and identify test scenarios.
2. Prepare test cases, test data, and checklists.
3. Set up test environments (local, staging, GitHub Pages).

**Step -2: Test Execution**

The testing will be carried out in the following sequential order, covering all relevant **testing types:**

1. **Migration Testing**

* Confirm all content has been successfully migrated from WordPress to GitHub Pages.
* Validate build process and dependencies using Bundler and GitHub Actions.
* Verify site renders correctly in the local and staging environment before functional testing begins.

1. **Sanity Testing**

* Quickly validate core flows.

1. **Functional and Integration Testing**

* Verify functionality of all website features including forms, events, and project pages.
* Check integrations with external systems for proper redirects and accessibility.
* Validate that navigation flows are intuitive and error-free.

1. **UI/UX Testing**

* Check responsiveness across devices (desktop, tablet, mobile).
* Validate theme switching (Auto/Light/Dark).

1. **Performance Testing**

* Measure page load speed.
* Perform light stress/load testing to identify potential bottlenecks
* Verify Caching/ performance of static assets.

1. **Accessibility Testing**

* Validate WCAG compliance: alt text, colour contrast, ARIA roles.
* Ensure assistive technologies (screen readers) can interact with site elements.

1. **Security Testing**

* Validate HTTPS implementation and secure external link handling.

1. **Compatibility Testing**

* Browsers: Chrome, Edge, Firefox.
* OS/Device: Windows

1. **Regression & Continuous Verification**

* Re-run key test scenarios after bug fixes or minor updates.
* Verify no new issues have been introduced post-deployment.

1. **Stage & Production Validation**

* Deployment/ Pipeline Testing: verify GitHub Actions builds successfully (if used).
* Confirm site availability and proper functioning after deployment.

**Step -3: Management & Reporting**

1. Log defects in GitHub Issues or excel, track through closure.
2. Share execution summary and results.

**Step -4: Post-Test Analysis / Closure**

1. Verify exit criteria.
2. Document lessons learned for future migrations.

## 5.2 Testing Tools

* Link Checker: broken-link-checker (npm package)
* Performance: Apache JMeter
* Accessibility: Axe, Wave, CCA, NVDA
* Defect Tracking: GitHub Issues/ MS Excel
* Gen AI: Gemini/ChatGPT, GPT4ALL (To create/optimize test scenarios)

## 5.3 Defect Management

* All defects will be logged in GitHub Issues/ Excel with priority and severity.
* Defects will be tracked until closure and verified after fixes.
* High/critical defects must be resolved before production deployment.

# 6. Test Environment

* **Local:** Run site using bundle exec jekyll serve.
* **Staging:** Preview branch deployments (if enabled in GitHub Pages).
* **Production:** Live civictech.ca site on GitHub Pages.

**Browsers:** Latest Chrome, Firefox, Edge  
**Devices:** Desktop. Mobile, Tab

# 7. Test Data

* Migrated content (projects, events, resources).
* Sample inputs for forms (RSVP, Guild signup).
* External links for verification.

# 8. Test Dependencies

* Migration of content completed
* CI/CD pipeline – GitHub Actions workflow configured and functional
* Access to local, staging and production environments
* The third-party integrations (e.g., Slack, Guild) must be operational and accessible for verification.

# 9. Deliverables

* Test Plan
* Test Cases
* Test Execution Report
* Bug Reports (in GitHub Issues/ Excel)

# 10. Entry & Exit Criteria

* **Entry:**
  + Development/Test environment set up.
  + Migration content available.
  + Build succeeds locally.
* **Exit:**
  + All functional test cases passed.
  + No high/critical bugs open.
  + Site verified on staging + production.

# 11. Risks & Mitigation

|  |  |  |  |
| --- | --- | --- | --- |
| No | Risk | Impact | Mitigation |
| 1. | Missing content during migration | High | Cross-verify with original site, maintain migration checklist |
| 2. | Broken links | Medium | Run automated link checker, validate redirects |
| 3. | Build failures on GitHub Pages | High | Validate dependencies and CI/CD before release |

# 12. Assumptions

* All static content has been exported from WordPress before testing begins.
* The scope of testing is limited to public-facing functionality and content. Internal workflows of third-party platforms (Slack, Guild) are not included.
* Any issues found during testing can be reproduced and resolved within the project timeline, subject to environment stability and availability of required resources.