Test Plan

(Version 1.0)

**Version info**

|  |  |  |  |
| --- | --- | --- | --- |
| Version number | Date | Edited by | Comments and changes |
| 1.0 | [Date] | V.Shevchenko | First version |

**Table of Contents**

1. INTRODUCTION
2. SCOPE
3. ASSUMPTIONS AND RISKS
   1. Assumptions
   2. Risks
4. TESTING STRATEGY
5. ENVIRONMENT
6. TEST DESIGN AND TESTING METHODS
7. TESTS EXIT CRITERIA
8. EXTERNAL LINKS

[All text to be edited is marked with blue and put into brackets. Text inside of brakets explains what we need to add on these sections]

1. INTRODUCTION

The Test Plan has been created to communicate the test approach to team members. It includes the objectives, scope, schedule, risks and approach. This document will clearly identify what the test deliverables will be and what is deemed in and out of scope.

The object of testing is a [Object name and description. It could be a webpage, mobile application, desktop application or just a test suite].

The testing is performed on [Overall list of features, modes or parts of obect to be tested e.g. front-end part, landscape mode, UI/UX, etc]. Main aim of testing is to ensure that product is working properly, performs correct search and inner functions of page are working as it is expected by users.

2.SCOPE

This part of test plan describes functionality that needs to be checked during testing period. At this point we are testing main functionality of website, functionality of links and accordance to designed layout on different devices or screen resolutions.

We must ensure that:

- [Detailed list of features and functions to be tested]

- [Overall expected behaviour description]

- [One item after another]

All tests must be performed on [description of environments to be tested on. Must be taken into consideration target marketing regiona and segment, software version, popularity of OS, browser, devices, hardware configuration etc]

Any changes to test scopes may be considered by team and added to this document. Please - update version info section in the header of this document.

3. ASSUMPTIONS AND RISKS

a. Assumptions

* [Assumptions example]
* [All thing that may influence n tests results]
* [Prebuild database, emulators, testing environments, etc]

b. Risks

* [All risks that can influence on process of testing]
* [E.g. impossibility to handle performance or security tests,]
* [Possible unavaliebility of testing environment or troubles with access]

4. TESTING STRATEGY

Product is [Current status of application: deployed and delivered to customers or in development only, version to be tested, state of current build]  
  
We need to start with Smoke tests to determine if system is available for deeper testing, if it has blockers or other critical bugs that can make any further testing impossible.

Functional testing should be performed according to test cases or user stories that were created before. We can perform ad-hoc testing to have an idea of functionality and to have base for creating test cases. We also can make some exploratory testing based on requirements mentioned in SCOPE section.

For non-functional methods of testing we are going to focus on [List of non-functional tests like Internationalization and Localization testing, UI testing (according to designed templates) and Usability testing, check for SQL, HTML or script injections]. We will not perform any tests connected with [List of tests not to be performed e.g. performance and data loading, installation testing].

All fixed bugs must be re-tested after fixing them. It will be good if we will have any possibility of Sanity testing of functions that were touched by a fixed bug. This is vital to make sure all bugs are fixed and fix is not affecting any connected features. Smoke and Regression testing must be performed for each build of a product that is installed on environment to make sure product is working properly, build is using all latest changes and no new bugs appeared.

User acceptance tests must be performed after successful passing of all previously mentioned tests. Main purpose is to be sure that system is ready for operation use. It must be performed by customers to make sure that it meets all their requirements. Need of Alpha and Beta testing stages must be considered with Product owner.

5. ENVIRONMENT

Server-side environment configuration must be as close as a production one.

Here is an example of a list of configurations for user-side environments. It can be changed according to regional or market demands. Test team need to keep in mind that we need to check different screen resolutions and different browsers. We may use PICT or similar tool for creating device configurations

Device: [Configurations of devices, we may mix OS, browsers or program versions, hardware configurations]

6. TEST DESIGN AND TEST METHODS

We are starting with ad-hoc testing to get acquainted with current product and it’s basic functions. In EXTERNAL LINKS section of this document you can find a checklist with functions to be tested. We may write test cases for all of these functions and it will definitely ease our work. But lack of time may prevent us from this.

We are using [Type of testing like black, grey or white box]. We are performing [Static testing if we are working with documentation and dynamic if we are working with set-up environment]. At this point we are performing [manual or automation tests. Tests can be split or we can give advice of what must be handled manually and what can be automated].

As we don’t have any “expected result” yet and if we have no time for preparing user stories or test cases we need to create a checklist. For an “expected result” we need to consider what we are waiting as a standard behaviour from this system. We may check other similar pages or elements to know how they should behave.

Let’s split product into sections:

* [Sections to be tested separately]

We will check each section in a separate way. [All additional parts to be tested e.g REST, etc]

[Section name]

[List of test cases, boundary values, limits, equeivalence classes – all detailed list of cases to be tested over this section]

[Section name]

[List of test cases, boundary values, limits, equeivalence classes – all detailed list of cases to be tested over this section]

[Section name]

[List of test cases, boundary values, limits, equeivalence classes – all detailed list of cases to be tested over this section]

7. TESTS EXIT CRITERIA

As we may never be sure that software is bug-free, our tests must continue until we have time for them. By the end of testing period we need to be sure that:

* all planned tests have been run (according to checklist)
* there are NO critical or high severity defects that are left
* all found bugs are fixed and retested

On lack of time team must determine list of functions priority and perform testing according to this list.

8. EXTERNAL LINKS

[[](https://docs.google.com/spreadsheets/d/e/2PACX-1vRyJAgRWBD7scPAGyV4oDyesYMg7sDUnWeLTEY6Z-mj9ZsCKSDUD3LC2-dcjhC3zYWjkpJG2P2f2mDE/pubhtml)List of external links for this test plan. Check lists, test cases, user-stories, etc]