Functional Test Plan

General information	
Customer	
Developer	
Created by (Author)	(QATestLab)
Preparation date	
Version	0.3
Status	For confirmation

Revision History

Version	Description	Author	Date	Approved by	
				Author	Date
0.1	Creating				
0.2	Updating				
0.3	Updating				
0.4	Updating				

Related Documents

Link	Document name
	FUNCTIONAL SPECIFICATIONS Version 1.0

Content

1. Introduction

2. Purpose of the document

This document describes a test strategy for the project "" and approaches, which the test team will use to verify the quality of the product prior to release. The document also lists the different resources that are needed for a successful testing of the project.

3. Objective

The purpose of the test strategy is to formalize the testing process, plans and approaches to testing, interfacing process with the development team and the project team to achieve the high quality of the software product. The strategy takes into account the specifics of the functionality of the project ""

4. Description of the project

“” project is developing by company _ on request of company _. External testing services are provided by QATestLab company.

“” is the system ….

5. Scope of work

6. The components and functions to be tested

№	Components/Applications name	Functions	Link
1	*Back end*	***Back end:*** *Log in*	*FUNCTIONAL SPECIFICATIONS Version 1.0*

		Forgot password ***Admin:*** *Log in* *Forgot password* *Create User account* *List Editor account* *Log Out*	*(including the next versions)*
2	*The website including the web player*		

7. The components and functions not to be tested

№	Components/Applications name	Functions	Comment
1	*Program code*		*Unit testing- testing the source code*
2	*The 3d party system*		
3	*Web services without web interface*		*Testing requests to the server directly, without web interface.*

8. Quality criteria

The delivered product must work in accordance with the requirements and the functional specification listed in sections “Scope of Work “and “Related documents”.

The delivered product must not contain any known defects with critical and high priority in the final version, and no more than 30 bugs in all other priorities.

9. The decisive factors of the project success

- Compliance with schedule for developing and approving specifications for the development of parts of the product.
- Compliance with the schedule and the completion of development and testing of all functionality in time.
- The application should not include known defects with critical and high priority at the time of the final version.
- Functional requirements are not changed at the last moment.

10. Limitations, assumptions and risks

- The late submission of information, delays in document approval by the Customer.
- Changes in the requirements during product development.
- Ambiguous requirements can increase the risk of bugs appearance in the system.
- The narrow time frame increases the risk of bugs appearance at the time of the transfer system in live. If the timing of development phase is not met, it will directly affect the timing of testing.

11. The risks of the project

Risk Id	Risk description	Probability (High / Medium /Low)	Influence (High / Average / Low)	Effects on Cost / Schedule / Quality
RI.1	The late submission of information, delays in document approval by the Customer.	Medium	High	Schedule
RI.2	Incorrect or incomplete stated requirements.	High	High	Cost, Schedule
RI.3	Changes in the requirements during development.	High	High	Cost, Schedule
RI.4	Problems with the delivery of the product into production because of the unavailability of servers.	High	Medium	Schedule
RI.5	Problems of integration with internal systems of the Customer due to the absence or unavailability of appropriate interfaces and test environment.	Medium	High	Cost, Schedule, Quality
RI.6	Errors in the 3d party providers of the software.	Low	High	Cost, Schedule
RI.7	The narrow time frames. If the time frame of development phase is not met, it will directly affect the timing of testing.	Medium	High	Cost, Schedule, Quality

12. Plan to reduce the risks

Risk Id	Actions to reduce the risk
RI.1	Compliance with the rules of planning and organizing meetings. Timely information about the unavailability of employees (including due to vacation, illness, etc.). The schedule of meetings and the provision of necessary information in advance.
RI.2	Splitting development into short iterations. Frequent demonstrations of new functionality.
RI.3	Fixing the basic list of requirements in the contract. A dedicated Product Owner from the Customer. Frequent demonstrations of new functionality.
RI.4	Getting further details on installing the product from the Customer's IT department as soon as possible.

RI.5	Communicating with the customer, who are responsible for the provision and operation of appropriate interfaces in order to agree in advance with him all the features and performance bottlenecks. Try to solve them before going to live. Install beta version of the product and to test the product in order to identify the problems associated with integration.
RI.6	The provision of an initial stage of development for defining and studying the architecture.
RI.7	Follow the development schedule. Timely notification of potential problems or shifts in the schedule.

13. Assumptions

All functional requirements are not yet defined in detail. Estimates made on the basis of how the QATestLab sees the system at the time of the analysis requirements. Estimates may change (increase or decrease) depending on the appearance of new requirements for the system.

14. Resources

15. The team of external testing

Company	Name	Role	Contact Information
QATestLab		Program Manager	Skype: E-mail:
QATestLab		QA Lead	Skype: Email:

16. Tools for testing

№	Tool	Comment
1	Mantis	For inserting bugs and issues
2	TestRail	For managing and writing test cases

17. Deliverables

18. Testing Documentation and Reports

№	Title	Responsible person	Frequency (delivery time)	Delivery method
1	Test Plan	QA Lead	One time before testing	e-mail
2	Scenario of functional, UI and acceptance testing	QA Lead	Upon receipt of the final version of specification	e-mail

3	Bug reports	QA Lead QA Team	After bug detection	Insert into Mantis
4	Reports on the results of testing	QA Lead QA Team	After every test / deliveries	e-mail
5	Acceptance testing report	QA Lead	At the end of the acceptance tests	e-mail

19. Responsibilities

Program Manager:

- Managing the whole testing process.
- Providing all the needed resources for the testing activities.
- Validation of the documentation.
- Validation of reviews according to the test plan.

QA Lead:

- Gathering and learning the requirements.
- Planning the entrusted testing works.
- Monitoring the entrusted testing activities, making sure that the works are performed according to the plan.
- Reporting about the progress, percentage of executed tests, number and quality of the found errors.

Test Engineer:

- Writing test cases in compliance with the test plan.
- Performing test cases and finding errors.
- Logging found errors into the approved bug tracking system.

Development team:

- Provides all the artifacts listed in this document.
- Is responsible for the timely update of status of defects in Mantis.

The Project Coordinator is responsible for the completeness of the information transmitted on the functionality developed within the new release.

The Product Owner is responsible for providing information and requirements on time, and for coordinating the product launch.

All project participants follow the test strategy.

Conducting a voice meeting with the contact person from the development team, from the testing team, the Project Coordinator and Product Owner, to take stock of the development and testing of the deliverables, the event is desirable, but not mandatory.

20. Handling exceptional situations

Section is intended to address situations where there is a violation of an agreed process. Exceptions are: urgent product release, reducing the time of testing, etc.

- Project Coordinator with Product Owner shall decide on the functionality that will be included in the iteration.
- Project Coordinator with Product Owner makes a decision about the changes in the business process.
- As a result of changes in the business process testing team change the test plan.
- All deviations from the process must be documented in Mantis.
- If the blocking or critical issue is found, which will not allow the version to go live, QA Lead should report a bug and notify Development team via Skype or email.
- If the issue cannot be solved quickly, QA Lead should write an email with description of the issue to the Project Coordinator and Product Owner.
- Project Coordinator with Product Owner make a decision if product can go/or not go live.

21. Strategy of testing

22. Testing phases

Main stages of work of the testing team:

1. Development of testing strategies
2. Coordination and approval of testing strategies
3. Receiving of the final version of the specification
4. The study of the specification.
5. Writing test cases for the new functionality.
6. Confirming test cases with the Customer.
7. Testing new functionality, executing test cases.
8. Writing bug reports (including inserting bug/issues into bug-tracking system – podio.com).
9. Regression testing (including retest fixed defects).
10. Selecting test cases with high priority for the acceptance test.
11. Executing of the acceptance test cases.
12. Writing test report.

23. Acceptance criteria

1. Specification received and confirmed
2. The release manager and project leader received an access to the version from the developers, followed by Release notes (with a list of new functionality).
3. Release manager assigns the task to the test team.

Test team can partially or completely suspend work in this version, if the following occurs:

1. There is an error in functionality, which does not allow continuing testing.
2. There is a serious problem that prevents the continuation of testing (non-working or damaged test environment, force majeure, such as turning off the Internet or electricity).

3. The developers have not corrected the problem that blocked the testing.

24. Testing Methods

Manual functional testing is regarded as the primary method of testing applications.

25. Test Types

26. Functional testing.

The purpose of the test is to verify the functionality of all features of the program are working properly.

27. UI testing

GUI testing is the process of ensuring proper functionality of the graphical user interface (GUI) for a given application and making sure it conforms to its written specifications.

28. Mobile testing

Mobile application testing is a process by which application software developed for hand held mobile devices is tested for its functionality, usability and consistency.

29. Regression testing

The purpose of regression testing is to verify, that the defects, which was corrected earlier, are not appeared again and adding the new functionality and fixing defects do not cause deterioration in the quality of the product.

30. Smoke testing

Smoke Testing is performed after software build to ascertain that the critical functionalities of the program is working fine. It is executed "before" any detailed functional or regression tests are executed on the software build.

31. Compatibility testing

The purpose of compatibility testing is to verify how well software works in different combinations of hardware/software/operating system/network environment.

32. Defect tracking and documentation

The tools described in **Tools for testing** will be used to account for the defects and documentation. Metrics and statistics will be included in the reports on the results of testing.

The priority in this project is used to display the seriousness of the impact of the defect on the functionality of the application.

Definition of defects priority:

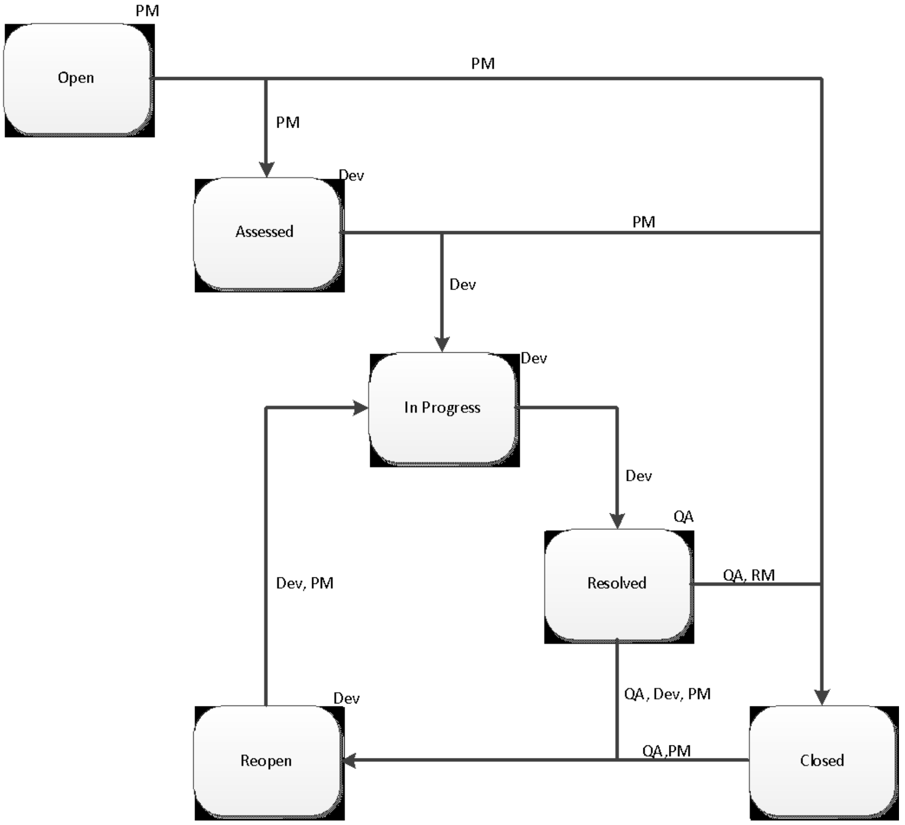
1. Critical – If need to urgently fix the defect. Application component or module is completely denied or unavailable.
2. High - an error that requires a binding decision to release functional. Incorrect operation of the main parts of functional violation of basic logic functionality. The incorrectness of design that prevents the normal operation of the basic functional parts.

3. Major – necessarily to fix before release. Incorrect operation of the main functional parts. The incorrectness of design that prevents the normal operation with the basic functional parts.
4. Minor – errors, that do not impede the work of the functional, but cause inconvenience for the user.
5. Trivial – the problem does not affect to the quality of the product.

All bugs should be added in the bug tracking system. All bugs should contain the following information:

- Summary
- Issue/bug type (Functional, UI, Usability, Integration, data show, data process)
- Steps to reproduce
- Actual result
- Expected result
- Screenshot (for visual bags)
- System log (if necessary)
- Priority

33. The life cycle of the bug



34. Criteria of the testing is complete

- Smoke testing achieved
- All test scenarios from the Test Plan have been executed.
- All found bugs listed in the bug tracker.
- All bugs have a priority.
- All fixed bugs are retested.

35. Acceptance Plan

36. Criteria for beginning of acceptance testing

The main criteria for the initiation of the process of acceptance testing:

1. All the involved resources are ready.

2. All the acceptance test scenarios have been developed and approved.

3. All the required artifacts, including the release candidate version available before the start of the process.

37. The process of acceptance testing

The main stages of acceptance testing:

1. Executing acceptance testing scenarios by QATestLab team.

2. Storing the results of passing the acceptance test scenarios in a bug-tracking system Mantis, in the form of bug reports.

3. Fixing defects and providing a new product version by development team.

4. Start a new test cycle, during which the bug fixes will be tested and acceptance scenarios will be executed again.

The results of the acceptance tests:

1. QATestLab team writes a report on the results of the acceptance test.

2. The representative of the "[redacted]' announces completion of the acceptance testing, after the last test cycle, and based on the results of the final acceptance test.

38. Schedule of acceptance testing

Acceptance testing will begin on the next business day after receiving the release candidate product versions.

39. Troubleshooting and corrective action

All the problems detected during acceptance testing shall be recorded in bug-tracking system. If the parties cannot agree on the priority of the defects, representative of the “[redacted]” makes the decision.

40. Risks of the acceptance testing

1. The recent changes in the product may influence the process of acceptance testing;

2. During the acceptance test, the major defects found in the specifications, will lead to changes in the software.

41. Product acceptance criteria

Product acceptance criteria
[AC_001] Completeness of artifacts
[AC_002] The number of defects below the limit
[AC_003] Successful completion of test scenarios

[AC_001] Completeness of artifacts

Description: All of the artifacts listed in paragraph 5 shall be provided by QaTestLab.

Tasks:

1. Compare the list of artifacts that are listed in this document and provided after the fact.

2. Create a report on the acceptance testing.

[AC_002] The number of defects below the limit

Description:

The number of defects shall not exceed the limit specified below:

Priority	Acceptable number of defects	
Critical	0	
High	0	
Normal	5	For discussion [2]
Low	10	For discussion [2]
Trivial	15	For discussion [2]

The number of defects that do not affect the yield of the product to live.

[2] The maximum number of defects.

Tasks:

1. Complete acceptance test scenarios

2. Record all defects found in Mantis

3. Calculate the number of critical defects

4. Compare the number of defects found with the table above

5. Record results in report on acceptance testing

[AC_003] Successful completion of test scenarios

Tasks:

1. Analyze the results of passing the test scenarios from the TestRail

2. Record results in a report on acceptance testing

42. The matrix of product acceptance

Acceptance criteria	Related artifact	The effect on the result of the acceptance of the product
[AC_001] Completeness of artifacts	Release Candidate build is not provided	**Acceptance failed**
	The rest of the artifacts	Conditional acceptance
[AC_002] The number of defects below the limit	Critical	**Acceptance failed**
	High	Conditional acceptance if workarounds have been provided
	Normal	**Acceptance failed**
	Minor	Conditional acceptance
	Trivial	Conditional acceptance
[AC_003] Successful completion of test scenarios		Conditional acceptance

Conditional Acceptance means that the formal acceptance will be completed, with condition of correction of defects by development team.