**«Audio Cataloger» Project**

**Test Plan**

|  |  |
| --- | --- |
| **Background** | Estimations, schedule, strategy, and metrics are needed to organize the testing process efficiently. |
| **Purpose** | To organize the testing process effective and efficient during the whole project period. |
| **Scope** | Testing process description, metrics, schedule, resources. |
| **Audience** | Management staff, QA team, project team. |
| **File** | Test Plan AudioCataloger Project.docx |

Contents

1. Project scope and main goals .............................................................................................3

2. Requirements to be tested ..................................................................................................3

3. Requirements NOT to be tested .........................................................................................3

4. Test strategy and approach ................................................................................................3

4.1. General approach ...........................................................................................................3

4.2. Functional testing levels ..................................................................................................3

5. Criteria ..............................................................................................................................4

6. Resources ..........................................................................................................................4

7. Schedule ............................................................................................................................5

8. Roles and responsibilities ....................................................................................................5

9. Risk evaluation ...................................................................................................................5

10. Documentation ................................................................................................................5

11. Metrics .............................................................................................................................5

1.Project scope and main goals

Failure resistant automated audio cataloger with duplications cross-reference and corrupted audio files elimination viewable via web-browser and editable in spreadsheet software.

2.Requirements to be tested:

- UR-1.\*: smoke test

- UR-2.\*:smoke test, critical path test

- UR-3.\*:smoke test

- BR-1.:smoke test, critical path test

- BR-2.:smoke test

- BR-3.:smoke test, critical path test

- BR-4.:smoke test, critical path test

- QA-\*:smoke test, critical path test

- L-4.:smoke test

- DS-\*:smoke test, critical path test

3.Requirements NOT to be tested:

- SC-1:the application is a console one by design.

- SC-2, L-1, L-2:the application is developed with the proper JAVA/JRE version.

- L-3: no implementation is needed

- QA-1.1:the testing is done in point BR\_1.

- SC-3:the testing is done in point L-4.

Test strategy and approach

4.1. General approach.

The application is considered to be in a console only, so issues of usability, security are not explored.

4.2.Functional testing levels.

- Smoke test: automated with batch files under Windows and Linux.

- Critical path test: executed manually.

- Extended test: not executed as the probability of defects detection on this level is negligibly small.

The team consists of two developers and two testers who are expected to assume

cross-functionalities. So, to contribute significantly to the quality improvement the followings

events will be fulfilled:

- code review

- automated testing (smoke tests)

- manual testing (critical path tests)

- white-box testing (unit testing should be done due to the requirement of the high reliability of the application during its working process).

5.Criteria

5.1.Acceptance criteria:100% success of test cases on smoke test level and 90% success of test cases on critical path test level (see ["Test cases success percentage"](#Success_percentage) metric) if 100% of critical and major bugs are fixed (see ["Overall defects fixed percentage"](#Overall_defects_fixed) metric). Final requirements coverage by tests (see ["Requirements coverage by tests"](#Requirements_coverage) metric) is to be >= 80%.

5.2.Testing start criteria: new build.

5.3.Testing pause criteria: critical path test must begin only after 100% success of test cases on the smoke test (see ["Test cases success percentage"](#Success_percentage) metric); test process may be paused when >=25% of the test-cases executed result in >=50% of tests failed (see ["Stop-factor"](#Stop_factor) metric).

5.4.Testing resumption criteria: >50% of bugs found during the previous iteration are fixed (see ["Ongoing defects fixed percentage"](#Ongoing_defects_fixed) metric).

5.5.Testing finish criteria: >=80% of test cases planned for the current iteration are executed (see "Test-cases execution percentage").

6.Resources:

6.1.Software: eight virtual machines (four with Windows 10 Ent x64, four with Linux Ubuntu 18 LTS x64), eight JAVA licenses (minimal JRE version 8.0.60).

6.2.Hardware: four standard workstations (8GB RAM, i7 3GHz).

6.3.Personnel:

- One senior developer with testing experience (100% workload during all project time).

Roles: team lead, senior developer.

- One middle developer (100% workload during all project time).

Roles: middle developer, build engineer.

- One tester with JAVA/JRE knowledge (100% workload during all project time).

Role: tester.

- One tester (manual testing)

Role: tester.

- Time: 15 working days (8h\*15 = 120 workhours).

- Finances: according to the approved budget

7. Schedule

- 05.08 - 07.08 – requirements testing, gathering additional info, finalizing.

- 08.08 – 09.08, 12.08 - test-cases and scripts for automated testing creation.

- 13.08 - 16.08, 19.08 – main testing stage (test-cases execution, defect reports creation).

- 20.08 – 22.08 - testing finalization, reporting.

- 23.08 - one day for unexpected issues.

8. Roles and responsibilities

- Senior developer: participation in requirements testing and code review, team-leading.

- Middle developer: coding

- Tester 1: participation in requirements testing,

documentation creation, test-cases execution, participation in code-review.

- Tester 2: participating in documentation creation, test-cases execution, task results reporting.

9. Risk evaluation

- Personnel (low probability): if any team member is inaccessible, we can contact the representatives of the “Cataloger” project to get a temporary replacement (the commitment from the “Cataloger” PM John Smith was received).

- Time (high probability): the customer has indicated a deadline of 26.08, therefore time is a critical resource. It is recommended to do our best to complete the project by 22.08 so that one day (23.08) remains available for any unexpected issues.

- Other risks: no other specific risks have been identified.

10. Documentation

- Requirements. Responsible – testers, developers, deadline – 07.08.

- Test cases and defect reports. Responsible – testers, deadline - 19.08.

- Test result report. Responsible person – tester, deadline – 22.08.

11. Metrics

1.1.Test cases success percentage:

𝑇𝑆𝑃 = \* 100%, where

𝑇𝑆𝑃 – percentage of successfully passed test cases,

𝑇𝑆𝑢𝑐𝑐𝑒𝑠𝑠 – quantity of successfully passed test cases,

𝑇𝑇𝑜𝑡𝑎𝑙 – total quantity of executed test cases.

Minimally acceptable borders:

- Beginning project phase: 10%.

- Main project phase: 40%.

- Final project phase: 80%.

1.2.Overall defects fixed percentage:

= \*100%, where

– overall defects fixation percentage by 𝐿𝑒𝑣𝑒𝑙 during all project lifetime,

– quantity of defects of 𝐿𝑒𝑣𝑒𝑙 fixed during all project lifetime,

– quantity of defects of 𝐿𝑒𝑣𝑒𝑙 found during all project lifetime.

Minimally acceptable borders:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Defect severity | | | |
| Minor | Medium | Major | Critical |
| Project phase | Beginning | 10% | 40% | 50% | 80% |
| Main | 15% | 50% | 75% | 90% |
| Final | 20% | 60% | 100% | 100% |

1.3.Ongoing defects fixed percentage:

= \*100%, where

– defects fixation percentage by 𝐿𝑒𝑣𝑒𝑙 (defects found in the previous build and fixed in the current build),

– quantity of defects of 𝐿𝑒𝑣𝑒𝑙 fixed in the current build,

– quantity of defects of 𝐿𝑒𝑣𝑒𝑙 found in the previous build.

Minimally acceptable borders:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Defect severity | | | |
| Minor | Medium | Major | Critical |
| Project phase | Beginning | 60% | 60% | 60% | 60% |
| Main | 65% | 70% | 85% | 90% |
| Final | 70% | 80% | 95% | 100% |

1.4.Stop-factor:

S= , where

𝑆 – decision to pause the testing process,

𝑇𝐸 – current 𝑇𝐸 value,

𝑇𝑆𝑃 – current 𝑇𝑆𝑃 value.

1.5.Test-cases execution percentage:

𝑇𝐸 = \*100%, where

𝑇𝐸 – test-cases execution percentage,

𝑇𝐸𝑥𝑒𝑐𝑢𝑡𝑒𝑑 – quantity of executed test-cases,

𝑇𝑃𝑙𝑎𝑛𝑛𝑒𝑑 – quantity of planned (to execution) test-cases.

Levels (borders):

- Minimal: 80%.

- Desired: 95%-100%.

1.6.Requirements coverage by tests:

𝑅𝐶 = \*100%, where

𝑅𝐶 – requirements coverage by tests (percentage),

𝑅𝐶𝑜𝑣𝑒𝑟𝑒𝑑 – quantity of requirements covered with test-cases,

𝑅𝑇𝑜𝑡𝑎𝑙 – overall quantity of requirements.

Minimally acceptable borders:

- Beginning project phase: 40%.

- Main project phase: 60%.

- Final project phase: 80% (90%+ recommended).