«File Converter» Project

Test Plan SAMPLE **Project Documentation**

**Background**

**Purpose**

**Scope Audience File**

Estimations, schedule, strategy, and metrics are needed to organize the testing process efficiently.

To organize the testing process effective and efficient during the whole project period.

Testing process description, metrics, schedule, resources. Management staff, QA team, project team.

02 03 - Test Plan Sample.docx

Contents

[1. Project scope and main goals......................................................................3](#_page_20_0)

[2. Requirements to be tested...........................................................................3](#_page_20_0)

[3. Requirements NOT to be tested..................................................................3](#_page_20_0)

[4. Test strategy and approach .........................................................................3](#_page_20_0)

[4.1. General approach...................................................................................3](#_page_20_0)

[4.2. Functional testing levels.........................................................................3](#_page_20_0)

[5. Criteria..........................................................................................................4](#_page_29_0)

[6. Resources....................................................................................................4](#_page_29_0)

[7. Schedule ......................................................................................................4](#_page_29_0)

[8. Roles and responsibilities ............................................................................4](#_page_29_0)

[9. Risk evaluation.............................................................................................4](#_page_29_0)

[10. Documentation ..........................................................................................5](#_page_35_0)

[11. Metrics.......................................................................................................5](#_page_35_0)

1. Project scope and main goals

Develop a tool to catalog audio files, find duplicates of audio files and find corrupted audio files. Tool has to be able to create a list of all audio files in user’s possession along with duplication cross-reference. List has to be viewable via web-browser and editable in spreadsheet software. Tool should not fail during its working process.

2. Requirements to be tested

See referenced sections in “File Converter Requirements.docx”:  UR-1.\*: smoke test.

 UR-2.\*: smoke test, critical path test.  UR-3.\*: critical path test.

 BR-1.\*: smoke test, critical path test.

BR-2.\*: smoke test, critical path test.

BR-3.\*: critical path test.

BR-4.\*: critical path test.

 QA-1.\*: smoke test, critical path test.

 QA-2.\*: smoke test.

 QA-3.\*: critical path test.

L-3.\*: critical path test.

 L-4.\*: smoke test.

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

3. Requirements NOT to be tested

See referenced sections in “File Converter Requirements.docx”:  SC-1: the application is a console one by design.

 SC-2, L-1, L-2: the application is developed with proper Java and JRE version.

4. Test strategy and approach 4.1. General approach

The application is to be configured once by an experienced specialist and later used by end users, for whom only two operations are available – confirm creation of catalog of local audio files and forming a list of these audio files that will be shown via web-browser or spreadsheet software. Therefore, issues of usability, security, etc. not explored during testing.

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.

Due to the team cross-functionality, a significant contribution to quality improvement can be expected from the code review combined with manual testing using the white box method. Unit-testing will not be applied due to time limitations.

5. Criteria

 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](#_page_35_0) [percentage”](#_page_35_0) metric) if 100% of critical and major bugs are fixed (see [“Overall defects](#_page_35_0) [fixed percentage”](#_page_35_0) metric). Final requirements coverage by tests (see [“Requirements](#_page_49_0) [coverage by tests”](#_page_49_0) metric) should be at least 80%.

 Testing start criteria: new build.

 Testing pause criteria: critical path test must begin only after 100% success of test-cases on the smoke test (see [“Test cases success percentage”](#_page_35_0)); test process may be paused is with at least 25% test-cases executed there is at least 50% failure rate (see [“Stop-factor”](#_page_49_0) metric).

 Testing resumption criteria: more than 50% of bugs found during the previous iteration are fixed (see [“Ongoing defects fixed percentage”](#_page_35_0) metric).

 Testing finish criteria: more than 80% planned for the current iteration test cases are executed (see [“Test-cases execution percentage”](#_page_49_0)).

6. Resources

 Software: four virtual machines (two with Windows 10 Ent x64, two with Linux Ubuntu 18 LTS x64), two Java licenses (latest JRE version available).

 Hardware: two standard workstations (8GB RAM, i7 3GHz).  Personnel:

o Two developers with testing experience (100% workload during all project time). Roles: team lead, senior developer.

o Two testers with Java knowledge (100% workload during all project time). Role: tester.

 Time: 15 working days (120 work hours).

 Finances: according to the approved budget.

7. Schedule

 01.07 – requirements testing and finalizing.

 03.07 – test-cases and scripts for automated testing creation.

 04.07-11.07 – main testing stage (test-cases execution, defect reports creation).  12.07 – testing finalization, reporting.

8. Roles and responsibilities

 Senior developer: participation in requirements testing and code review.

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

9. Risk evaluation

Parsing (low probability): Complexity of accurate parsing of some audio formats may cause some unexpected errors to occur. Therefore, all supported audio formats have to be used during testing (mp3, flac, wav, ogg, wma).

 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 (low probability): the customer has indicated a deadline of 15.07, therefore time isn’t a critical resource. It is recommended to do our best to complete the project by 12.07 so that three days remains available for any unexpected issues.

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

10. Documentation

 Requirements. Responsible person – tester, deadline – 01.07.

 Test cases and defect reports. Responsible – tester, creation period – 03.07-05.07.  Test result report. Responsible person – tester, deadline – 12.07.

11. Metrics

 Test cases success percentage: 𝑇𝑆𝑃 = 𝑇𝑆𝑢𝑐𝑐𝑒𝑠𝑠 ∙ 100%, where

𝑇

𝑇𝑜𝑡𝑎𝑙

𝑇𝑆𝑃 – percentage of successfully passed test cases, 𝑇𝑆𝑢𝑐𝑐𝑒𝑠𝑠 – quantity of successfully passed test cases, 𝑇𝑇𝑜𝑡𝑎𝑙 – total quantity of executed test cases.

Minimally acceptable borders:

o Beginning project phase: 10%. o Main project phase: 40%.

o Final project phase: 80%.

 Overall defects fixed percentage: 𝐷𝐿𝑒𝑣𝑒𝑙 = 𝐷𝐿𝐶𝑒𝑙𝑜𝑣𝑒𝑠𝑒𝑙𝑑 ∙ 100%, where

𝐿𝑒𝑣𝑒𝑙

𝐷

𝐹𝑜𝑢𝑛𝑑

𝐹𝑇𝑃

𝐷𝐿𝑒𝑣𝑒𝑙 – overall defects fixation percentage by 𝐿𝑒𝑣𝑒𝑙 during all project lifetime, 𝐷𝐶𝑙𝑜𝑠𝑒𝑑 – quantity of defects of 𝐿𝑒𝑣𝑒𝑙 fixed during all project lifetime,

𝐹𝑇𝑃

𝐿𝑒𝑣𝑒𝑙

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

𝐹𝑜𝑢𝑛

Minimally acceptable borders:

Defect severity

Minor Medium Major Critical

Project phase

Beginning 10%

Main 15%

Final 20%

40% 50% 80%

50% 75% 90%

60% 100% 100%

 Ongoing defects fixed percentage: 𝐷𝐿𝑒𝑣𝑒𝑙 = 𝐷𝐿𝐶𝑒𝑙𝑜𝑣𝑒𝑠𝑒𝑙𝑑 ∙ 100%, where

𝐿𝑒𝑣𝑒𝑙

𝐷

𝐹𝑜𝑢𝑛𝑑

𝐹𝐶𝑃

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

𝐹𝐶𝑃

𝐿𝑒𝑣𝑒𝑙

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

𝐹𝑜𝑢𝑛

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%

 Stop-factor:

𝑌𝑒𝑠,𝑇𝐸 ≥ 25% && 𝑇𝑆𝑃 < 50% 𝑁𝑜,𝑇𝐸 < 25% || 𝑇𝑆𝑃 ≥ 50%

𝑆 = { , where

𝑆 – decision to pause the testing process, 𝑇𝐸 – current 𝑇𝐸 value,

𝑇𝑆𝑃 – current 𝑇𝑆𝑃 value.

 Test-cases execution percentage: 𝑇𝐸 = 𝑇𝐸𝑥𝑒𝑐𝑢𝑡𝑒𝑑 ∙ 100%, where

𝑇

𝑃𝑙𝑎𝑛𝑛𝑒𝑑

𝑇𝐸 – test-cases execution percentage, 𝑇𝐸𝑥𝑒𝑐𝑢𝑡𝑒𝑑 – quantity of executed test-cases,

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

Levels (borders):

o Minimal: 80%.

o Desired: 95%-100%.

 Requirements coverage by tests: 𝑅𝐶 = 𝑅𝐶𝑜𝑣𝑒𝑟𝑒𝑑 ∙ 100%, where

𝑅

𝑇𝑜𝑡𝑎𝑙

𝑅𝐶 – requirements coverage by tests (percentage),

𝑅𝐶𝑜𝑣𝑒𝑟𝑒𝑑 – quantity of requirements covered with test-cases,

𝑅𝑇𝑜𝑡𝑎𝑙 – overall quantity of requirements.

Minimally acceptable borders:

o Beginning project phase: 40%. o Main project phase: 60%.

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