|  |  |
| --- | --- |
|  | «File Converter» Project |
| Test Plan SAMPLE  **Project Documentation** |
| **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** | 02 03 - Test Plan Sample.docx |

Contents

1. [Project scope and main goals 3](#_bookmark0)
2. [Requirements to be tested 3](#_bookmark1)
3. [Requirements NOT to be tested 3](#_bookmark2)
4. [Test strategy and approach 3](#_bookmark3)
   1. [General approach 3](#_bookmark4)
   2. [Functional testing levels 3](#_bookmark5)
5. [Criteria 4](#_bookmark6)
6. [Resources 4](#_bookmark7)
7. [Schedule 4](#_bookmark8)
8. [Roles and responsibilities 4](#_bookmark9)
9. [Risk evaluation 4](#_bookmark10)
10. [Documentation 5](#_bookmark11)
11. [Metrics 5](#_bookmark12)

# Project scope and main goals

Develop a simple and efficient audio file management tool with advanced features. That includes the development of a comprehensive system capable of cataloging audio files, identifying duplicates, detecting corrupted files, and facilitating the creation of organized lists for users. The tool will be designed to offer a seamless experience, ensuring the user can easily access, review, and process their audio files.

# Requirements to be tested

* UR-1.\*: smoke test.
* UR-2.\*: smoke test, critical path test.
* UR-3.\*: smoke test, critical path test.

# 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 using Java in JRE and distributed as a JARcontainer.
* L-3: no implementation required.
* SC-3, L-4: the application works with Windows and Linux.

# Test strategy and approach

## General approach

The testing approach for the Audio Cataloger tool will focus primarily on functional testing due to the critical nature of the application's functionality. The tool is designed to be configured once and utilized by end users for a specific task - cataloging audio files, finding duplicates, and detecting corrupted files. Given the simplicity of user interactions, the emphasis will be on ensuring the tool performs its core functionalities accurately and reliably. Usability, security, and other non-functional aspects will not be explored in testing.

## Functional testing levels

* Smoke test: automated with batch files under both Windows and Linux environments.
* 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 extreme time limitations.

# 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](#_bookmark13) [percentage](#_bookmark13)” metric) if 100% of critical and major bugs are fixed (see “[Overall defects](#_bookmark14) [fixed percentage](#_bookmark14)” metric). Final requirements coverage by tests (see “[Requirements](#_bookmark18) [coverage by tests](#_bookmark18)” 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](#_bookmark13)”); test process may be paused is with at least 25% test-cases executed there is at least 50% failure rate (see “[Stop-factor](#_bookmark16)” metric).
* Testing resumption criteria: more than 50% of bugs found during the previous iteration are fixed (see “[Ongoing defects fixed percentage](#_bookmark15)” metric).
* Testing finish criteria: more than 80% planned for the current iteration test cases are executed (see “[Test-cases execution percentage](#_bookmark17)”).

# Resources

* Software: four virtual machines (two with Windows 10 Ent x64, two with Linux Debian 18 LTS x64), Java Development Kit (JDK) 8.0.60 or higher (for development and testing).
* Hardware: two standard workstations (8GB RAM, i7 3GHz).
* Personnel:
  + One senior developer with testing experience (70% workload during all project time). Roles: team lead, senior developer.
  + Two testers with Java knowledge (100% workload during all project time). Role: tester.
  + One developer with Java knowledge (30% workload during all project time). Role: junior developer.
* Time: 15 workdays or 3 workweeks (120 work hours).
* Finances: according to the approved budget.

# Schedule

* 23.10 – requirements testing and finalizing.
* 26.10 – test-cases and scripts for automated testing creation.
* 31.10-7.11 – main testing stage (test-cases execution, defect reports creation).
* 10.11 – testing finalization, reporting.

# Roles and responsibilities

* Senior developer: participation in requirements testing and code review.
* Tester: documentation creation, test-cases execution, participation in code-review.
* Junior developer: participation in code-review and help with development.

# Risk evaluation

* Complexity of accurate parsing of some audio formats *(*Probability: Moderate): If parsing complexities persist, consider limiting support to widely used audio formats initially and gradually expanding compatibility through software updates.
* Complexity (or Impossibility) to detect encoding for non-English tags in files (Probability: Low): If automatic detection fails consistently, provide clear user instructions on how to manually configure character encoding for non-English tags. Offer user support through documentation and customer service channels.
* Other risks: no other specific risks have been identified.

# Documentation

* Requirements. Responsible person – team lead, deadline – 23.10.
* Test cases and defect reports. Responsible – tester, creation period – 31.10-7.11.
* Test result report. Responsible person – tester, deadline – 10.11.

# Metrics

* Test cases success percentage:

𝑇𝑆𝑃 = 𝑇𝑆𝑢𝑐𝑐𝑒𝑠𝑠 ∙ 100%, where

𝑇𝑜𝑡𝑎𝑙

𝑇

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%.
* Overall defects fixed percentage:

𝐷𝐹𝑇𝑃 = 𝐷 𝐶𝑙𝑜𝑠𝑒𝑑 ∙ 100%, where

𝐿𝑒𝑣𝑒𝑙

𝐿𝑒𝑣𝑒𝑙

𝐷

𝐹𝑜𝑢𝑛𝑑

𝐿𝑒𝑣𝑒𝑙

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

𝐿𝑒𝑣𝑒𝑙

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

𝐿𝑒𝑣𝑒𝑙

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

𝐿𝑒𝑣𝑒𝑙

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% |

* Ongoing defects fixed percentage:

𝐷𝐹𝐶𝑃 = 𝐷 𝐶𝑙𝑜𝑠𝑒𝑑 ∙ 100%, where

𝐿𝑒𝑣𝑒𝑙

𝐿𝑒𝑣𝑒𝑙

𝐷

𝐹𝑜𝑢𝑛𝑑

𝐿𝑒𝑣𝑒𝑙

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

𝐿𝑒𝑣𝑒𝑙

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

𝐿𝑒𝑣𝑒𝑙

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

𝐿𝑒𝑣𝑒𝑙

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:

𝑆 = {𝑌𝑒𝑠, 𝑇𝐸 ≥ 25% && 𝑇𝑆𝑃 < 50% , where

𝑁𝑜, 𝑇𝐸 < 25% || 𝑇𝑆𝑃 ≥ 50%

𝑆 – decision to pause the testing process,

𝑇𝐸 – current 𝑇𝐸 value,

𝑇𝑆𝑃 – current 𝑇𝑆𝑃 value.

* Test-cases execution percentage:

𝐸 𝑇𝐸𝑥𝑒𝑐𝑢𝑡𝑒𝑑

𝑇 = ∙ 100%, where

𝑇𝑃𝑙𝑎𝑛𝑛𝑒𝑑

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

𝑇𝑜𝑡𝑎𝑙

𝑅

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).