**Оглавление**

[Введение 3](#_Toc140708726)

[1. Общая характеристика предприятия 4](#_Toc140708727)

[2. Программа IT Warm-up от компании Epam 5](#_Toc140708728)

[2.1 Цели и задачи 5](#_Toc140708729)

[2.2 Курс Functional Testing 6](#_Toc140708730)

[Заключение 14](#_Toc140708731)

[Приложение А 15](#_Toc140708732)

[Приложение Б 18](#_Toc140708733)

[Приложение B 21](#_Toc140708734)

Введение

Современному специалисту в сфере информационных технологий важно получать и закреплять теоретические знания, полученные в университете. Наиболее эффективно процесс обучения данным навыкам проходит в условиях реального предприятия. Так в период с 26 июня 2023 года по 23 июля 2023 года местом прохождения практики является компания ИООО «ЭПАМ Системз».

Компания «ЭПАМ Системз» разработала студентам курс для прохождения практики. В данном курсе в течение пяти дней проводили лекции на различные тематики, знакомство с компанией, обзор профессий в IT и технологий для изучения.

В качестве индивидуального задания требовалось пройти один из курсов на платформе Learn. Из перечня был выбран курс «Functional Testing» так как он является официальным дополнением к ранее пройденному курсу «Automated Testing» и обеспечивает должный опыт в тестировании программного обеспечения.

Функциональное тестирование — это процесс обеспечения качества в рамках [цикла разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%B4%D0%B8%D0%B8_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F), необходимый для проверки реализуемости [функциональных](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%BF%D0%B5%D1%86%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F) [требований,](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B5%D0%B1%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%BC%D1%83_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8E) согласно [спецификации](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B5%D1%86%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F) тестируемого [программного обеспечения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5). Функциональное тестирование проводится для [оценки соответствия](https://ru.wikipedia.org/wiki/%D0%9E%D1%86%D0%B5%D0%BD%D0%BA%D0%B0_%D1%81%D0%BE%D0%BE%D1%82%D0%B2%D0%B5%D1%82%D1%81%D1%82%D0%B2%D0%B8%D1%8F) [системы](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) или [компонента](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) заданным функциональным требованиям.

В рамках курса рассматриваются следующие темы:

-Введение в функциональное тестирование программного обеспечения

-Планирование тестирования

-Тестирование требований

-Тестовые примеры и сценарии тестирования

-Отчеты о дефектах

-Отчет о результатах тестирования

\*Основы автоматизации тестирования

Для проверки усвоения знаний студенту требуется для каждой темы прослушать лекции, пройти тестирование и сделать задание, на основе этого будет подсчитывается прогресс по окончанию курса.

Таким образом, производственная практика, имея своей целью углубление знаний студентов, является важнейшим этапом процесса обучения. Прохождение практики позволяет получить полноценное и качественное высшее образование с перспективой дальнейшего трудоустройства по специальности.

# 1. Общая характеристика предприятия

ИООО «ЭПАМ Системз» это – американская ИТ-компания с белорусскими корнями, основанная в 1993 году. Производитель заказного [программного обеспечения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), специалист по консалтингу, резидент [Белорусского парка высоких технологий](https://ru.wikipedia.org/wiki/%D0%91%D0%B5%D0%BB%D0%BE%D1%80%D1%83%D1%81%D1%81%D0%BA%D0%B8%D0%B9_%D0%BF%D0%B0%D1%80%D0%BA_%D0%B2%D1%8B%D1%81%D0%BE%D0%BA%D0%B8%D1%85_%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D0%B9)

Основным направлением деятельности компании является создание программного обеспечения, ИТ-консалтинг, интеграция приложений, портирование и миграция приложений, тестирование программного обеспечения, разработка цифровых стратегий.

Экспертиза ЕРАМ востребована в таких сферах, как автомобильная промышленность и производство, медиа и телекоммуникации, услуги в области деловой информации, медицинское ПО, естественные науки, энергетика, туризм, и других. Разработанные EPAM цифровые платформы для различных организаций, сфер образования и здравоохранения делают опыт взаимодействия с процессами и сервисами более комфортным, объединяют людей̆ и помогают улучшить их жизнь. В рейтинги «Fortune 500» и «Forbes Global 2000» включены 62 из 100 крупных клиентов компании.

За время существования «ЭПАМ Системз» в компании сложилась собственная корпоративная культура. К ее отличительным особенностям можно отнести демократизм и открытость в общении, свободный обмен знаниями, стимулирование инициативы и ответственности сотрудников.

# 2. Программа IT Warm-up от компании Epam

## **2.1 Цели и задачи**

Компания «ЭПАМ Системз» разработала для студентов курс для прохождения практики. В данном курсе в течение пяти дней проводили лекции на различные тематики, знакомство с компанией, обзор профессий в IT, Технологический институт EPIC, Контроль версий с помощью GIT, Soft skills и день английского языка, Front-End, Automated Testing, Functional Testing, С# & .NET, Data & Analytics, Cloud & DevOps. Пример лекций представлен на рисунке 2.1. Как задание для самостоятельного выполнения было зарегистрироваться на портале [training.by](https://training.by/#!/Home?lang=ru&City=37,38,39,40,41,42,50) и пройти тест на уровень английского.

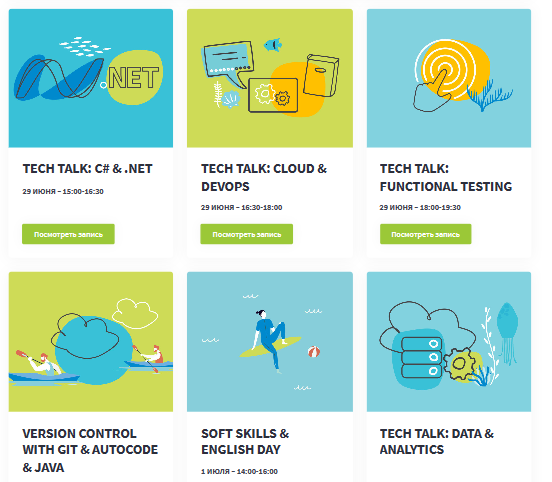


Рисунок 2.1 − Лекции, представленные компанией Epam

В качестве индивидуального задания требовалось пройти один из курсов на платформе Learn (https://learn.epam.com): Front-End, Automated Testing, Functional Testing, С# & .NET, Data & Analytics, Cloud & DevOps, Java. Из этого перечня был выбран курс «Functional Testing» так как он является официальным дополнением к ранее пройденному курсу «Automated Testing» и обеспечивает должный опыт в тестировании программного обеспечения.

## **2.2 Курс Functional Testing**

Функциональное тестирование — это процесс обеспечения качества в рамках [цикла разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%B4%D0%B8%D0%B8_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F), необходимый для проверки реализуемости [функциональных](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%BF%D0%B5%D1%86%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F) [требований,](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B5%D0%B1%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%BC%D1%83_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8E) согласно [спецификации](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B5%D1%86%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F) тестируемого [программного обеспечения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5). Функциональное тестирование проводится для [оценки соответствия](https://ru.wikipedia.org/wiki/%D0%9E%D1%86%D0%B5%D0%BD%D0%BA%D0%B0_%D1%81%D0%BE%D0%BE%D1%82%D0%B2%D0%B5%D1%82%D1%81%D1%82%D0%B2%D0%B8%D1%8F) [системы](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) или [компонента](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) заданным функциональным требованиям.

В рамках курса рассматриваются следующие темы:

1. Введение в функциональное тестирование программного обеспечения
2. Планирование тестирования
3. Тестирование требований
4. Тестовые примеры и сценарии тестирования
5. Отчеты о дефектах
6. Отчет о результатах тестирования
7. Основы автоматизации тестирования

Для проверки усвоения знаний студенту требуется для каждой темы прослушать лекции, пройти тестирование и сделать задание, на основе этого будет подсчитывается прогресс по окончанию курса (представлен на рисунке 2.2).

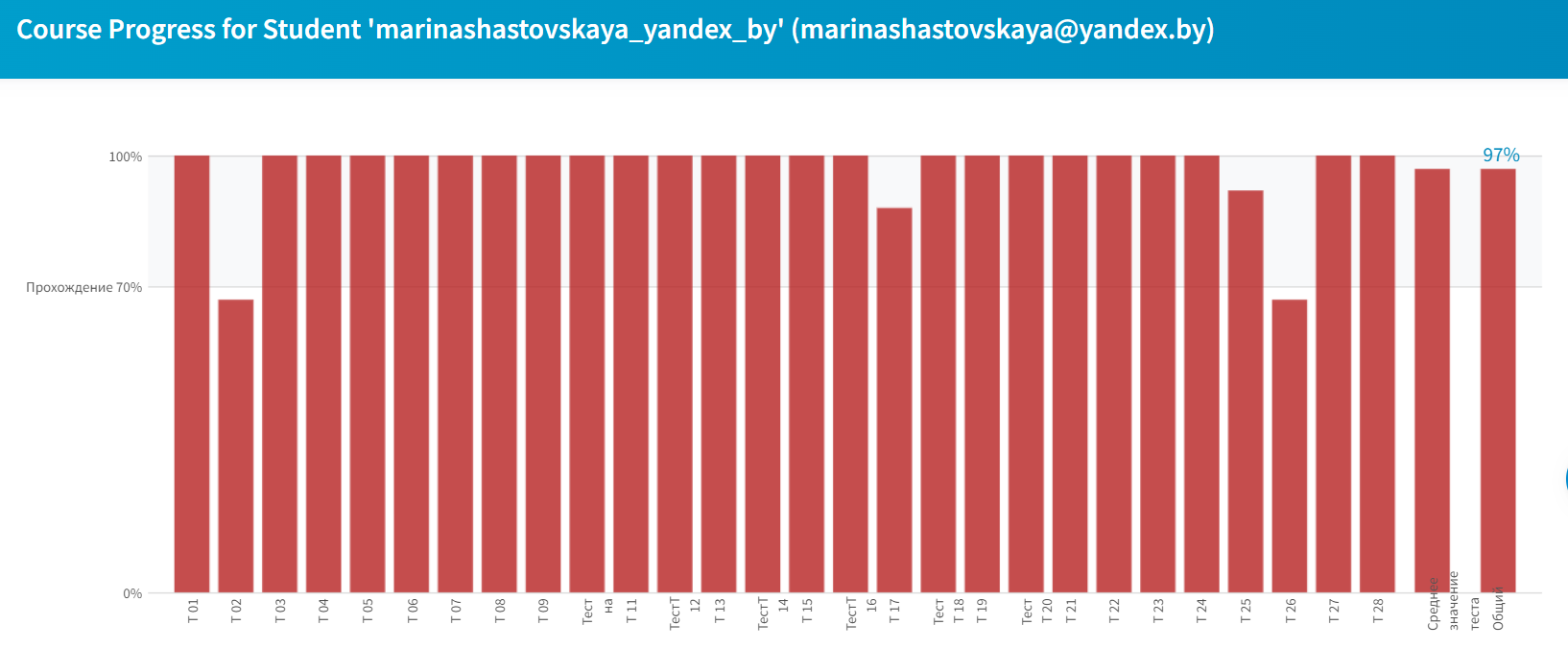


Рисунок 2.2 − Прогресс прохождения курса Functional Testing

Задания в курсе представлены после прослушивания и прохождения тестирования.

В первом задании необходимо было написать собственный план тестирования для «Audio Cataloger» Project, предполагая, что у вас есть команда из двух разработчиков и двух тестировщиков и 15 рабочих дней для завершения [проекта](https://elearn.epam.com/assets/courseware/v1/4686e9e50115f50b3b5328cf0a4e4339/asset-v1:EPAM+STI+RU+type@asset+block/Audio_Cataloger_Requirements.pdf).

План тестирования — это документ, в котором описываются шаги, необходимые для выполнения необходимого тестирования. В нем также указано, кто в вашей организации будет отвечать за каждую задачу, какие функции продукта тестируются и когда проверка должна быть завершена. План тестирования представлен в Приложении А.

Во втором задании нужно было проведите анализ требований для «Audio Cataloger» Project. Анализ требований — часть процесса [разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F), включающая в себя сбор [требований к программному обеспечению (ПО)](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B5%D0%B1%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%BC%D1%83_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8E), их систематизацию, выявление взаимосвязей, а также документирование. Он представлен в Приложении Б.

В третьем задании нужно было создать контрольные списки и тестовые примеры для «Audio Cataloger» Project. В уроке представлен шаблон для оформления задания в Exele.

Smoke test (Дымовой тест)- в [тестировании программного обеспечения](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F) означает минимальный набор тестов на явные [ошибки](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%B0%D1%8F_%D0%BE%D1%88%D0%B8%D0%B1%D0%BA%D0%B0). Такой тест обычно выполняется программистом; не проходившую этот дымовой тест программу не имеет смысла отдавать на более глубокое тестирование.

На рисунке 2.3 представлен чек-лист для Дымового теста.

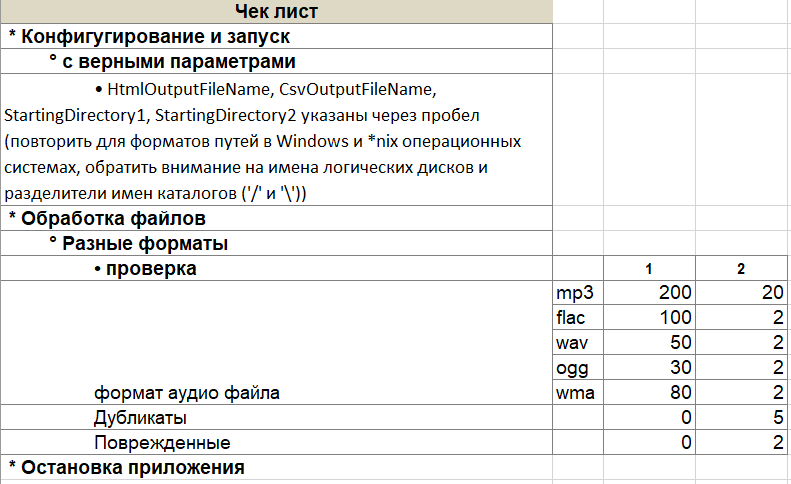


Рисунок 2.3 − чек-лист для Дымового теста

Дымовой тест легче автоматизировать, чем более глубокое и интеллектуальное тестирование. Автоматизация снижает количество ручного труда и поэтому позволяет проводить эти тесты чаще. Чем чаще выполняются тесты, тем раньше становится известно о проблемах, выявляемых этими тестами. Чем раньше становится известно о проблеме, тем легче её устранить. Автоматизация тестирования часто выполняется с помощью средств [непрерывной интеграции](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%BF%D1%80%D0%B5%D1%80%D1%8B%D0%B2%D0%BD%D0%B0%D1%8F_%D0%B8%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B0%D1%86%D0%B8%D1%8F).

На рисунке 2.4 представлен один из трёх составленных тестовых примеров для Дымового теста.

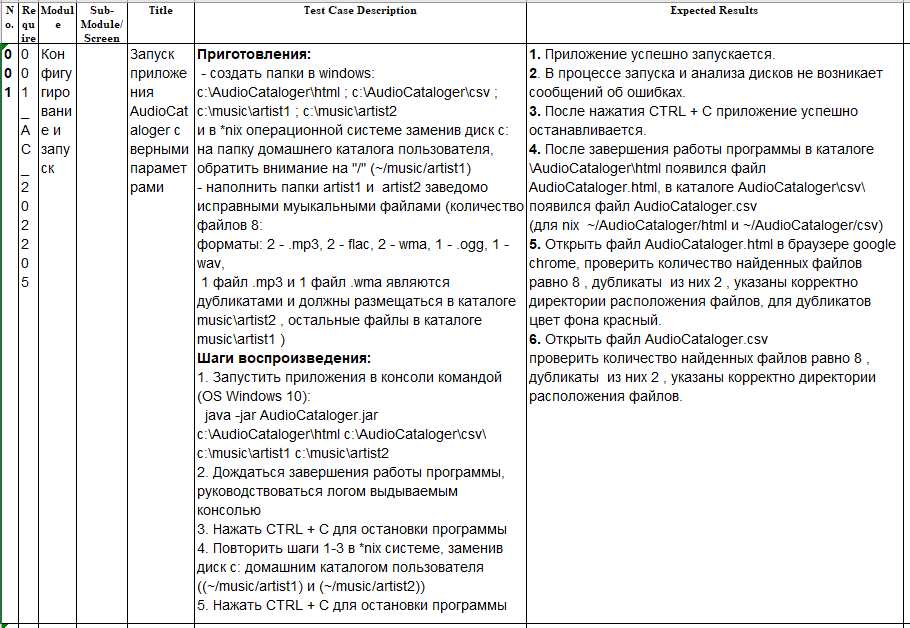


Рисунок 2.4 − Пример Дымового теста

Тестирование критического пути (critical path test): основной тип тестовых испытаний, во время которого основные элементы и функции приложения проверяются на предмет правильности работы при стандартном их использовании. В общем случае, тестирование критического пути охватывает основные сценарии использования разрабатываемого приложения.

Тест критического пути является одним из самых распространенных видов функционального тестирования. Чек-лист для тестирования критического пути представлен на рисунке 2.5.

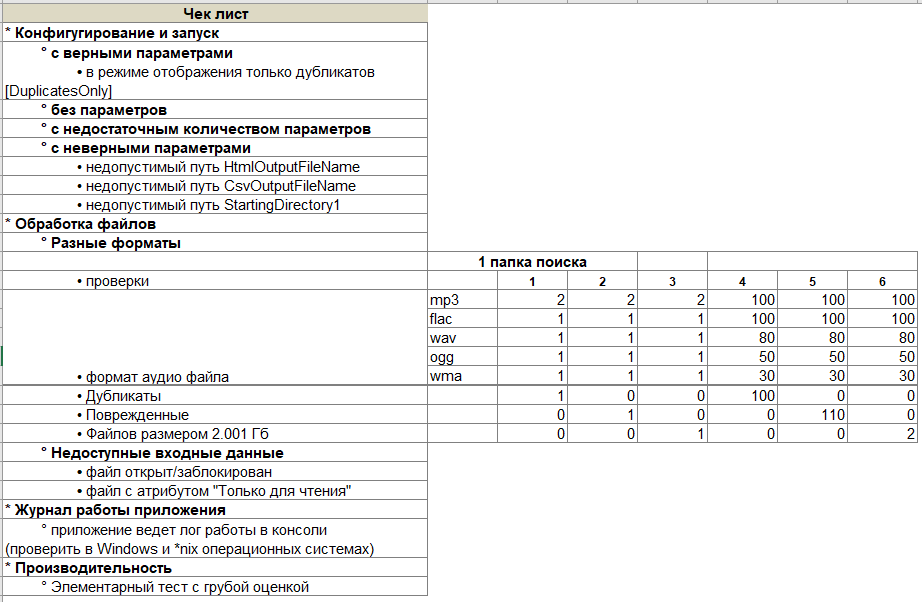


Рисунок 2.5 − чек-лист тестирования критического пути

На рисунке 2.6 представлен один из пятнадцати составленных тестовых примеров для тестирования критического пути.

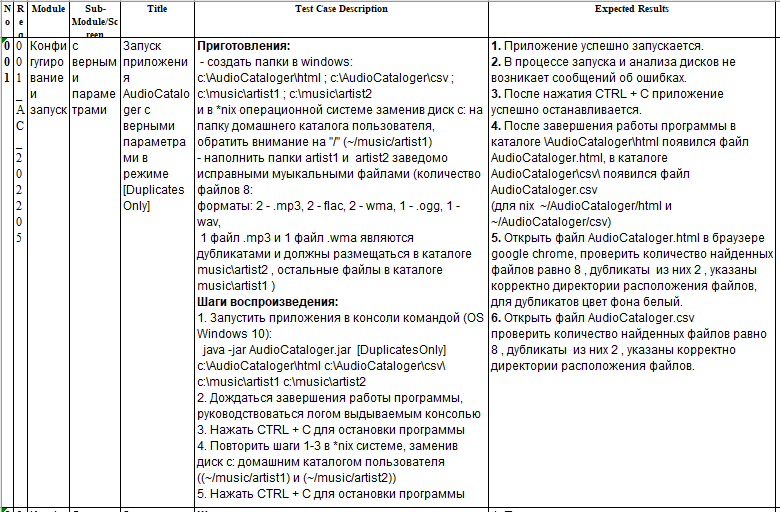


Рисунок 2.6 − Пример тестирования критического пути

В четвёртом задании нужно было создать отчёт по обнаруженным дефектам в «Audio Cataloger» Project. Пример заполненного отчёта для одного дефекта представлен на рисунке 2.7.

В результате тестирования «Audio Cataloger» Project было найдено 7 дефектов:

- приложение AudioCataloger не запускается в режиме DuplicatesOnly

- в отчетности приложения AudioCataloger некорректно определяются дубликаты музыкальных файлов

- ошибка работы приложения AudioCataloger при проверке музыкальных файлов формата .wma

- некорректная обработка приложением AudioCataloger файлов некорректной внутренней структурой

- некорректная обработка приложением AudioCataloger файлов некорректной внутренней структурой

- отчет приложения AudioCataloger в формате csv открывается в MS Excel без разбивки на колонки

- некорректно отображаются русские символы в отчете csv открытом MS Excel приложения AudioCataloger

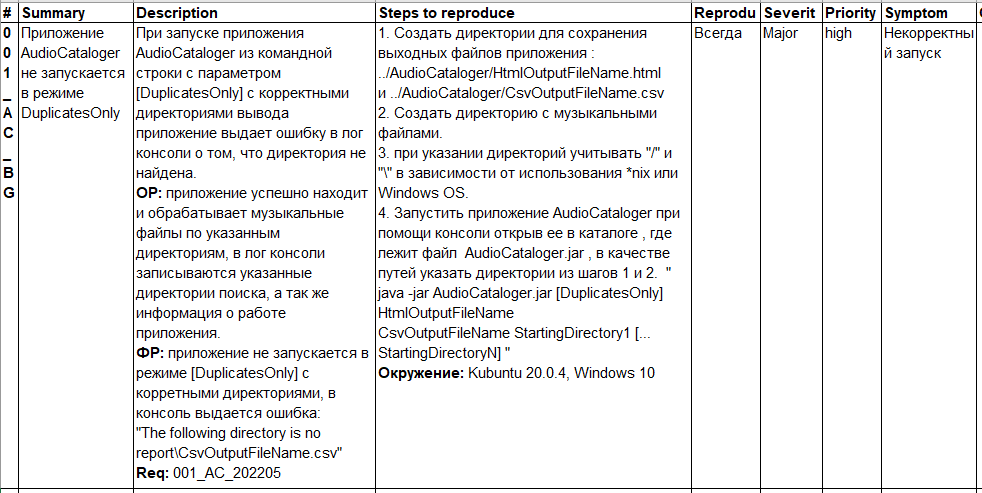


Рисунок 2.7 − Пример отчёта для одного дефекта

Для заполнения отчёта важна краткость и чёткость, поэтому используется таблица свойств, представленная на рисунке 2.8.

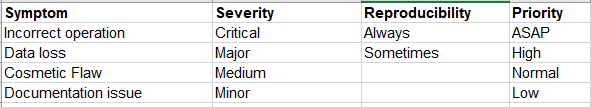


Рисунок 2.8 − Таблица свойств дефектов

Пятое задание основано на использовании программы Selenium IDE и составлении контрольного списка для тестирования приложения калькулятор (рисунок 2.9).

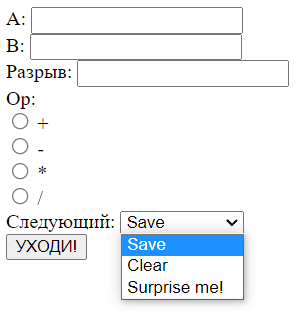


Рисунок 2.8 − Приложение калькулятор

По условию задания необходимо перед тестированием создать контрольный список. Контрольный список/лист проверок – это список проверок, которые помогают тестировщику тестировать приложение или отдельные функции. Основная цель чеклиста состоит в том, чтобы вы не забыли проверить всё, что планировали. Контрольный список представлен в ПРИЛОЖЕНИИ В.

Следуя пунктам чеклиста, нужно было протестировать калькулятор в программе Selenium IDE.

Selenium IDE — плагин для браузера Google для записи действий тестировщика и их воспроизведение для тестирования. IDE генерирует код для Selenium RC или Selenium WebDriver, который повторяет записанные действия пользователей.

Поток управления Selenium IDE поставляется с командами, которые позволяют добавлять условную логику и циклирование в тесты. Это позволяет выполнять команды (или набор команд) только при выполнении определенных условий в вашем приложении или выполнять команды повторно на основе заранее определенных критериев.

Условия в приложении проверяются с помощью выражений JavaScript. А условное ветвление позволяет изменять поведение в тесте, зацикливание выполняет итерации по заданному набору команд. В программе есть проверка синтаксиса. Если вы не уверены в правильности синтаксиса вашего потока управления, можно запустить тест, чтобы убедиться. Среда IDE обнаружит ошибки в синтаксисе потока управления и вызовет конкретную команду, которая неверна или отсутствует.

Интерфейс программы представлен на рисунке 2.9.

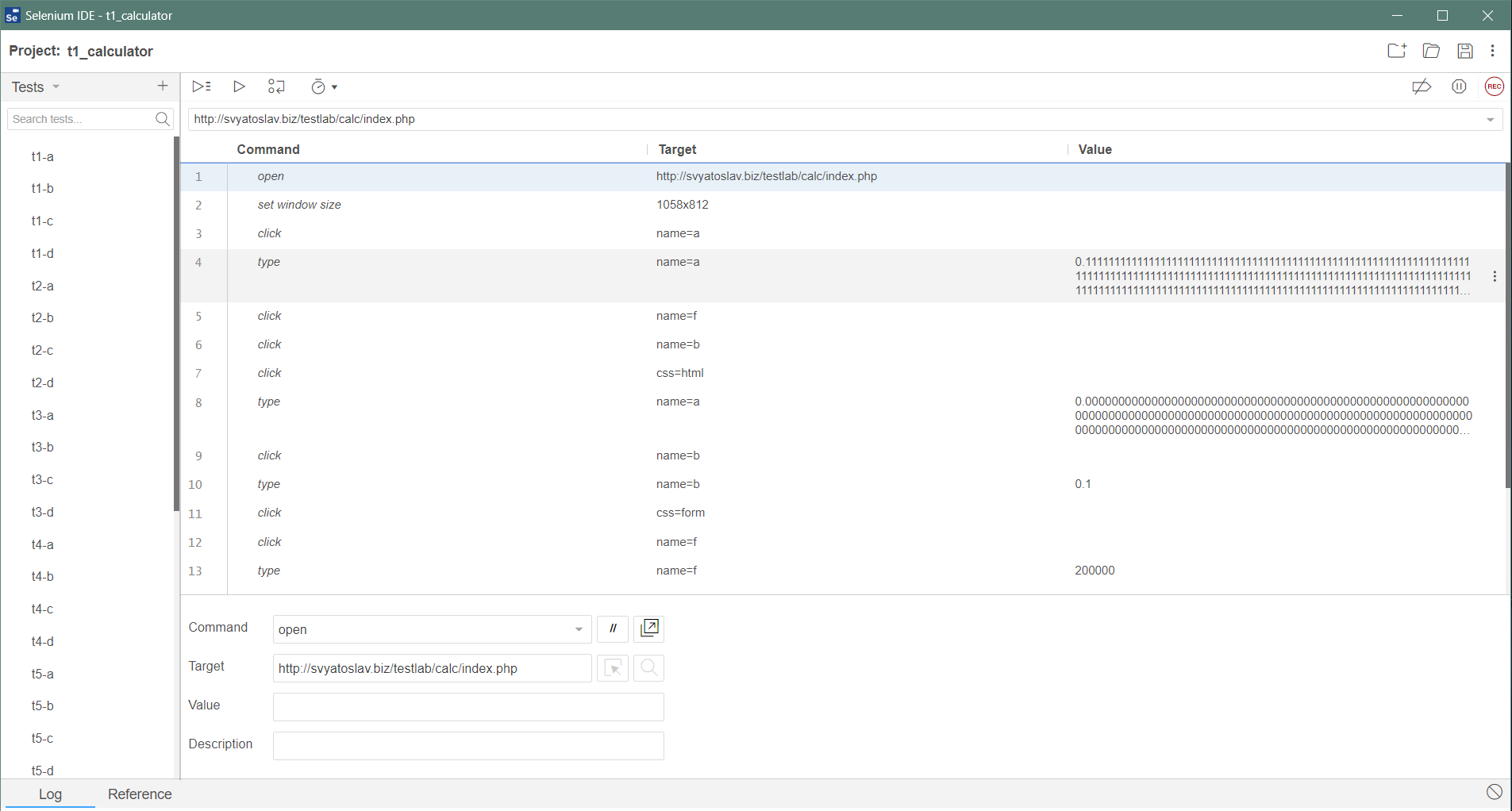


Рисунок 2.9 − Интерфейс программы Selenium IDE

Тест или набор тестов можно экспортировать на свой компьютер. В  настоящее время поддерживается экспорт на следующие языки и тестовые платформы: C # NUnit, Java JUnit, JavaScript Mocha, Python pytest.

Курс Functional Testing можно считать пройденным если результат превышает 70%. Исходя из итогового подсчёта, результат 97% и курс является пройденным (рисунок 2.10).

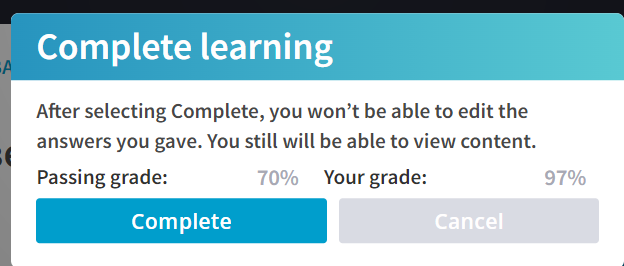


Рисунок 2.10 − Результат прохождения курса Functional Testing

Заключение

В результате прохождения технологической практики в компании «ЭПАМ Системз» были получены и подкреплены теоретические знания прослушиванием лекций, выполнением практических заданий и тестов.

Компания разработала лекции на различные тематики. Исходя из них студент должен был определится с индивидуальным курсом.

Был изучен курс Functional Testing. В нём рассматривались темы: введение в функциональное тестирование программного обеспечения, планирование тестирования, тестирование требований, тестовые примеры и сценарии тестирования, отчеты о дефектах, отчет о результатах тестирования, основы автоматизации тестирования.

Для подкрепления практических знаний был представлен проект «Audio Cataloger». Для него составлен собственный план тестирования, сделан анализ требований, написаны контрольные списки, тестовые примеры, отчёт по обнаруженным дефектам. Для изучения автоматизированного тестирования было выделено отдельное приложение. Для калькулятора требовалось написать контрольный список и сделать тесты в программе Selenium IDE.

В соответствии с полученным результатом прохождения курса, 97%, можно сделать вывод, курс пройден и знания были усвоены.

**Приложение А**

**План тестирования**

**1.Суть проекта и основные цели**

Быстрый и простой инструмент для создания списка всех имеющихся у пользователя аудиофайлов вместе с перекрестными ссылками на дубликаты. Полученный список должен быть доступен для просмотра в веб-браузере для быстрого просмотра и редактирования в программе для работы с электронными таблицами для тщательного просмотра и обработки.

**2. Требования к тестированию**

- UR-1.\*: дымовой тест.

- УР-2.\*: дымовой тест, тест критического пути.

- UR-3.\*: тест критического пути.

- BR-1, QA-2.\*: дымовой тест, тест критического пути.

- L-4, L-5: дымовой тест.

- DS-\*: дымовой тест, тест критического пути.

**3. Требования, НЕ подлежащие тестированию**

- СК-1: Приложение должно быть консольным.

- SC-2: Приложение должно быть разработано с использованием Java и распространяться как JARcontainer.

- SC-3: Приложение должно быть мультиплатформенным (с учетомЛ-4 [).](#page4)

- QA-1.1: эта характеристика производительности находится на нижней границе типичной производительности операций для таких приложений.

- L-3: реализация не требуется.

**4. Стратегия и подход к тестированию**

**4.1. Общий подход**

Приложение настраивается один раз опытным специалистом и в дальнейшем используется конечными пользователями, для которых доступна только одна операция – помещение файла во входную директорию. Поэтому вопросы юзабилити, безопасности и т.д. не исследовались при тестировании.

**4.2. Уровни функционального тестирования**

-Дымовой тест: автоматизирован с помощью пакетных файлов под Windows и Linux. Тест

-критического пути: выполняется вручную.

-Расширенный тест: не проводится, так как вероятность обнаружения дефектов на этом уровне пренебрежимо мала.

Ввиду кросс-функциональности команды значительный вклад в улучшение качества можно ожидать от проверки кода в сочетании с ручным тестированием по методу белого ящика. Модульное тестирование не будет применяться из-за крайних ограничений по времени.

**5. Критерии**

-Критерии приемки: 100% успешность тестовых случаев на уровне дымового теста и 90% успешность тестовых случаев на уровне критического пути тестирования (см.Процент успешных [тестовых случаев "метрика"), если](#page5) [исправлены](#page5) 100% критических и серьезных ошибок (см. "Фиксированный процент общих [дефектов метрика).](#page5) [Окончательное покрытие](#page5) требований тестами (см.Покрытие требований тестами метрика) [должно быть не](#page6) [менее 80%. Критерии](#page6) начала тестирования: новая сборка.

-Критерии приостановки тестирования: тест критического пути должен начинаться только после 100% успеха тесткейсов на дымовом тесте (см.Процент [успешных тестовых случаев ”); процесс](#page5) тестирования может быть приостановлен, если выполнено не менее 25 % тестовых случаев, частота отказов составляет не менее 50 % (см. Стоп-[фактор метрика).](#page6)

-Критерии возобновления тестирования: исправлено более 50% ошибок, обнаруженных на предыдущей итерации (см.[Фиксированный процент текущих дефектов](#page5) метрика).

-Критерий завершения тестирования: выполнено более 80% запланированных на текущую итерацию тест-кейсов [(см.Процент выполнения тест-кейсов](#page6) »).

**6. Ресурсы**

-Программное обеспечение: четыре виртуальные машины (две с Windows 10 Ent x64, две с Linux Ubuntu 18 LTS x64), две лицензии Java (доступна последняя версия).

-Аппаратное обеспечение: две стандартные рабочие станции (8GB RAM, i73GHz). Персонал:

-Один старший разработчик с опытом тестирования (100% загруженность в течение всего времени проекта). Роли: тимлид, старший разработчик.

-Один тестировщик со знанием Java и HTML (100% загруженность за все время проекта). Роль: тестировщик.

- Время: одна рабочая неделя (40 рабочих часов).

- Финансы: согласно утвержденному бюджету.

**7. Расписание**

- 25.05 – тестирование и доработка требований.

- 26.05 – тест-кейсы и скрипты для автоматизированного создания тестирования.

- 27.05-28.05 – основной этап тестирования (выполнение тест-кейсов, создание отчетов о дефектах).

- 29.05 – завершение тестирования, отчет.

**8. Роли и обязанности**

- Старший разработчик: участие в тестировании требований и ревью кода. Тестировщик:

- создание документации, выполнение тест-кейсов, участие в code-review.

**9. Оценка риска**

- Персонал (низкая вероятность): если какой-либо член команды недоступен, мы можем связаться с представителями проекта «Каталогизатор» для получения временной замены (получено обязательство от PM «Каталогизатора» Джона Смита).

- Время (высокая вероятность): заказчик указал крайний срок 01.06, поэтому время является критическим ресурсом. Мы рекомендуем сделать все возможное, чтобы завершить проект к

- 28.05, чтобы один день (29.05) оставался доступным для любых непредвиденных проблем. Другие риски: никаких других специфических рисков не выявлено.

* 1. **10. Документация**

-Требования. Ответственное лицо – тестер, срок – 25.05.

-Тестовые случаи и отчеты о дефектах. Ответственный – тестер, период создания – 26.05-28.05.

- Отчет о результатах тестирования. Ответственное лицо – тестер, срок – 29.05.

* 1. **11. Метрики**

- Процент успешности тестовых случаев:

- процент успешно пройденных тест-кейсов,

- количество успешно пройденных тест-кейсов,

- общее количество выполненных тестовых случаев.

**Минимально допустимые границы:**

- Начальная фаза проекта: 10%. Фаза

- основного проекта: 40%.

- Заключительный этап проекта: 80%.

**Минимально допустимые границы:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **Серьёзность дефекта** | | | |
| **Незначительный** | **Середина** | **Главный** | **Критический** |
| **Проект фаза** | **Начало** | **60%** | **60%** | **60%** | **60%** |
| **Основной** | **65%** | **70%** | **85%** | **90%** |
| **Финал** | **70%** | **80%** | **95%** | **100%** |

- решение приостановить процесс,

- тестирования,

- текущий ценить,

- текущий ценить.

**Процент выполнения тест-кейсов:**

- процент выполнения тест-кейсов,

- количество выполненных тест-кейсов,

- количество запланированных (к выполнению) тест-кейсов.

**Уровни (границы):**

- Минимум: 80%.

- Желаемый: 95%-100%.

**Покрытие требований тестами:**

- покрытие требований тестами (в процентах),

- количество требований, покрываемых тест-кейсами,

- общее количество требований.

**Минимально допустимые границы:**

- Начальная фаза проекта: 40%. Фаза

- основного проекта: 60%.

- Заключительный этап проекта: 80% (рекомендуется 90%+)

**Приложение Б**

**Анализ требований**

**1. Объем проекта**

**-**Разработка инструмента для:

-Каталог аудиофайлов.

-Поиск дубликатов аудиофайлов.

-Найдите поврежденные аудиофайлы.

**2. Основные цели**

-Предоставить Заказчику быстрый и простой инструмент для создания списка всех имеющихся у него аудиофайлов вместе с перекрестными ссылками на дубликаты.

- Полученный список должен быть доступен для просмотра в веб-браузере для быстрого просмотра и редактирования в программе для работы с электронными таблицами для тщательного просмотра и обработки.

- Инструмент не должен выходить из строя (по любой причине) в процессе работы (в отличие от многих конкурирующих инструментов).

**3. Критерии достижения основных целей**

**-**Инструмент должен быть консольным (для простоты), поддерживать интеллектуальные алгоритмы сравнения (для хорошей производительности) и следующие аудиоформаты[: см.БР-](#page4)1 . Поддержка вывода HTML (для браузера) и CSV (для редактора электронных таблиц).

-Для обеспечения надежности приложения необходимо внедрить глубокое тестирование на наличие негативных и критических ситуаций.

**4. Риски**

- Сложность точного парсинга некоторых аудиоформатов.

- Сложность (или невозможность) определения кодировки неанглийских тегов в файлах.

**5. Характеристики системы**

**-**СК-1: Приложение должно быть консольным.

-SC-2: Приложение должно быть разработано с использованием Java и распространяться как JARcontainer.

-SC-3: Приложение должно быть мультиплатформенным (с учетомЛ-4 [).](#page4)

**6.Требования пользователя**

-UR-1: Запуск и остановка приложения. о

-UR-1.1: Запуск приложения должен выполняться следующей консольной командой: «java -jar AudioCataloger.jar [DuplicatesOnly] HtmlOutputFileName CsvOutputFileName StartingDirectory1 [... StartingDirectoryN]»

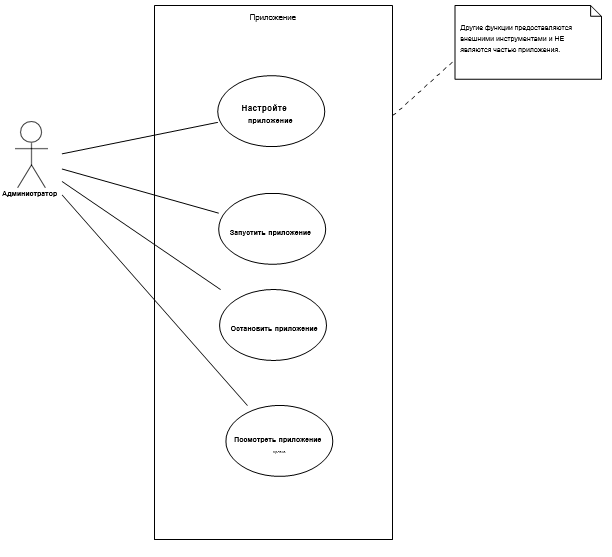
-UR-1.2: Остановку приложения (завершение работы) следует выполнять, применяя Ctrl+C к окну консоли, в котором находится запущенное приложение.

UR-2: Конфигурация приложения.

UR-2.1: Единственная доступная конфигурация — через параметры командной строки (см.ДС[-2 ).](#page5)

UR-2.2: Целевая кодировка для выходных текстовых сообщений — UTF8.

UR-3: Журнал приложений.



**7. Бизнес-правила**

-BR-1: Поддерживаемые форматы: mp3, flac, wav, ogg,

-wma. BR-2: Выходные форматы — HTML и CSV.

-BR-3: в режиме DuplicatesOnly цвет фона для дубликатов должен быть белым в выводе

-HTML. В обычном режиме фон для дубликатов должен быть красным.

-BR-4: Любое имя каталога или файла в выводе консоли должно быть полностью определенным,нормализованным.

**8. Атрибуты качества**

-QA-1: Устойчивость к входным данным

-ОК-1.1: См.БР-[1](#page4) требования к формату входных файлов. ОК-1.2:

-см.ДС-5.2 для [требований](#page5) к размеру входного файла.

-ОК-1.3: См.ДС-[5.3](#page5) подробнее о реакции приложения на неправильный формат входного файла.

-QA-2: Обработка исключений: ни при каких обстоятельствах приложение не должно падать с необработанным исключением. Независимо от того, насколько поврежден аудиофайл, приложение должно либо извлечь необходимые данные, либо заменить данные предопределенными заглушками в выходных данных.

-QA-3: Если указано несколько начальных каталогов, приложение должно проанализировать набор на предмет вложенности и/или дублирования, чтобы просканировать каждый реальный каталог только один раз.

**9. Ограничения**

-L-1: Приложение должно быть разработано с использованием Java как наиболее удобной кроссплатформенной среды.

-L-2: См.ДС[-1](#page4) сведения о версии JRE и конфигурации.

-L-3: Процесс установки и настройки JRE выходит за рамки этого проекта и поэтому НЕ описан ни в какой документации по продукту/проекту.

-L-4: Мультиплатформенные возможности приложения следующие: оно должно работать с Windows и Linux при условии наличия соответствующей версии Java (см.ДС-1.1 ) [работает](#page4) там.

10. Подробные характеристики

-DS-1: Ява

-DS-1.1: Минимальная версия JRE — 8.0.60.

-DS-1.2: Приложение должно работать только со стандартной JRE, т.е. без каких-либо дополнительных специфических библиотек и/или инструментов.

-DS-2: параметры командной строки

-DS-2.1: В процессе запуска приложение получает следующие параметры командной строки:

-[DuplicatesOnly] – необязательный параметр, указывающий, что в выводе должны отображаться только дубликаты аудиофайлов;

-HtmlOutputFileName – обязательный параметр, указывает на файл для вывода HTML;

-CsvOutputFileName – обязательный параметр, указывает на файл для вывода в формате CSV;

DS-2.2: Если какой-либо обязательный параметр командной строки опущен, приложение должно завершить работу, отображая стандартное сообщение об использовании (см.ДС-[3.1 ).](#page5)

DS-2.3: Любое количество параметров командной строки после StartingDirectory1 следует интерпретировать как набор каталогов для сканирования (см. [также](#page4)КА-3 ).

DS-2.4: Если значение любого параметра командной строки неверно, приложение должно завершить работу, отображая стандартное сообщение об использовании (см[.ДС-3.1 ) и](#page5) неправильное имя параметра, значение и правильное сообщение об ошибке ([см.ДС-3.2](#page5) ).

DS-3: Сообщения

DS-3.1: Сообщение об использовании.

DS-3.2: Сообщения об ошибках

ДС-4: журнал

DS-4.1: Приложение должно отображать свою текущую активность в консоли. Файлы журналов не нужны.

DS-4.2: Формат журнала консоли зависит от разработчиков.

DS-4.3: [Необязательно] Приложение должно перечислить и описать список заданных параметров командной строки в журнале.

DS-5: Формат и размер файла

DS-5.1: Приложение должно обрабатывать файлы следующих форматов: см.БР-1 . DS-5.2:Приложение должно обрабатывать файлы размером до 2 ГБ (включительно).

**Приложение B**

**Контрольный список**

1. Выполнить все четыре операции с целыми положительными числами

А. Прибавление

Б. Вычитание

С. Умножение

Д. Деление

1. Выполнить все четыре операции с целыми отрицательными числами

А. Прибавление

Б. Вычитание

С. Умножение

Д. Деление

1. Выполнить все четыре операции с дробными числами

А. Прибавление

Б. Вычитание

С. Умножение

Д. Деление

1. Выполнить все четыре операции с одним целым числом и со вторым дробным

А. Прибавление

Б. Вычитание

С. Умножение

Д. Деление

1. Выполнить все четыре операции с одним положительным числом и со вторым отрицательным

А. Прибавление

Б. Вычитание

С. Умножение

Д. Деление

1. Выполнить все четыре операции с целым положительным числом и текстовым сообщением “Деление на ноль”

А. Прибавление

Б. Вычитание

С. Умножение

Д. Деление

1. Разделите любое число на ноль и проверьте результат.
2. Выполнить операции

А. “Save”: все поля сохраняют свои значения.

Б. “Clear”: все поля возвращаются в состояние по умолчанию.

С. “Surprise me”: все поля (за исключением “Следующего”) получают случайные значения.

1. Выполнить операцию деления четного числа на нечётное. Проверьте правильность округления дробных разрядов
2. Проверить результат выполнения любой операции с очень большим числом.
3. Проверить результат выполнения любой операции с очень маленьким числом.
4. Выполнить операцию с машинным нулём (1.0f).
5. Выполнить операцию с специальным символом (%).
6. Выполнить операцию с буквой (X).