

**Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Национальный исследовательский университет ИТМО»  
(Университет ИТМО)**

**Факультет Инфокоммуникационных технологий (ИКТ)**

**Образовательная программа Мобильные и сетевые технологии**

**Дисциплина: Проектная документация**

**ОТЧЕТ**

**по практической работе 6**

**«Разработка программы и методики испытаний по проекту»**

**Выполнил:**

Рейнгеверц Вадим Алексеевич

**Группа:**

К34401

**Проверила:**

Валитова Юлия Олеговна

Санкт-Петербург  
2024

# Введение

Разработка программы и методики испытаний результатов проекта

## Содержание

### Задание 1

Изучить:

- [ГОСТ 19.301-79](#) ЕСПД. Программа и методика испытаний. Требования к содержанию и оформлению
- [ГОСТ Р 59792-2021](#). Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды испытаний автоматизированных систем.

---

### Задание 2

Выбрать ГОСТ, который будет использоваться в качестве руководства при разработке **программы и методики испытаний** результатов проекта

- Обосновать выбор

Выбран **ГОСТ 19.301-79**, так как он относится именно к программной документации (в отличие от ГОСТ Р 59792-2021, который описывает автоматизированные системы) и потому, что в остальной части документации проекта уже использовались ГОСТы 19-ой серии.

### Задание 3

На основании каких документов будет разрабатываться “Программа и методика испытаний”?

На основании документа о техническом задании для проекта "Разработка Electron приложения для чтения электронных книг «Bookord»"

## Задание 4

Разработать **программу и методику испытаний** результатов проекта, по которому разрабатывается проектная документация

### Объект испытаний

Объектом испытаний является "Разработка Electron приложения для чтения электронных книг «Bookord»".

Сокращенное наименование приложения: «Bookord», Приложение.

### Цель испытаний

Цель проведения испытаний

Целью проведения испытаний является проверка соответствия «Bookord» требованиям технического задания.

Задачами испытаний является последовательное выполнение этапов проверки согласно настоящей Программе и методике испытаний.

### Требования к программе

Требования, подлежащие проверке во время испытаний и заданные в техническом задании на программу

### Назначение разработки

Предоставление пользователю возможности чтения и хранения электронных книг на настольных компьютерах Windows, MacOS, Linux.

### Требования к функциональным характеристикам

Приложение позволяет пользователю добавлять электронные книги, просматривать в добавленные книги в библиотеке приложения, видеть метаданные книг (обложка, название, автор, год...), открывать книги.

В открытой книге текст разбивается на страницы и пользователь может производить навигацию как листанием страниц, так и нажимая на элементы оглавления.

Для каждой книги пользователь может добавлять заметки, закладки, выделения текста. Приложение запоминает какая последняя страница книги была открыта пользователем.

Пользователь может персонализировать визуальную часть приложения и отображения книг: менять цвет фона и текста, менять шрифт и размер текста, модифицировать отступы в книге.

Electron приложение должно поддерживать Windows 10-11, macOS 12-14, Ubuntu 18.04, и распространяться в виде установочных файлов для советующих ОС.

## Требования к программной документации

- Состав программной документации, предъявляемой на испытания
- Специальные требования (если они заданы в техническом задании на программу)
- Инструкции по воспроизведению установочных файлов для всех поддерживаемых ОС из исходного кода.
- Руководство пользователя

## Состав и порядок испытаний

### Технические и программные средства, используемые во время испытаний

- Компьютер с операционной системой macOS 12+
  - С установленной JavaScript средой Node.js
- Компьютер с операционной системой Ubuntu 18+
  - С установленной JavaScript средой Node.js
  - С установленными пакетами `fakeroot`, `dpkg` и `snappcraft`
- Компьютер с операционной системой Windows 10+
  - С установленной JavaScript средой Node.js
  - С установленным Windows 10 SDK

## Состав комиссии

- Заказчик
- Специалисты со стороны заказчика
- Представители со стороны команды разработки

## Порядок проведения испытаний

### Этап 1

**Продолжительность:** пол рабочего дня

1. Проверка наличия и готовности Git-репозитория с исходным кодом всех программных модулей и разделов Приложения, с историей изменений.
2. Проверка наличия и готовности инструкций по воспроизведению установочных файлов Приложения для всех поддерживаемых ОС из исходного кода.
3. Проверка наличия и готовности руководства пользователя.

**При успешном прохождении этапа:** переход к этапу 2

**При неуспешном прохождении этапа:** прекращение проведение испытаний, составление акта выявленных нарушений, принятие решение комиссией о новой дате проведения программы и методики испытаний

## Этап 2

Продолжительность: пол рабочего дня

1. Проверка воспроизводимости установочных файлов Приложения из исходного кода для операционных систем семейства Windows, macOS, Ubuntu.
2. Проверка успешности процесса установки Приложения для операционных систем семейства Windows, macOS, Ubuntu.

**При успешном прохождении этапа:** переход к этапу 3

**При неуспешном прохождении этапа:** прекращение проведение испытаний, составление акта выявленных нарушений, принятие решение комиссией о новой дате проведения программы и методики испытаний (начиная с этапа 2)

## Этап 3

Продолжительность: 3 рабочих дня

1. Проверка соответствия установленного Приложения требованиям.
2. Проведение **пользовательского приемочного тестирования (UAT)**, для операционных систем семейства Windows, macOS, Ubuntu

**При успешном прохождении этапа:** подписание акта приемки выполненных работ

**При неуспешном прохождении этапа:** прекращение проведение испытаний, составление акта выявленных нарушений, принятие решение комиссией о новой дате проведения программы и методики испытаний (начиная с этапа 3)

## Методы испытаний

## Описания используемых методов испытаний

- Их следует располагать согласно последовательности требований

### 1. Проверка Git-репозитория.

1. Проверка наличия приватного репозитория по ссылке на GitHub или GitLab
2. Проверка наличия исходного кода в репозитории для всех модулей Приложения
3. Проверка наличия семантических коммитов датируемых с начала разработки приложения

### 2. Проверка инструкций по воспроизведению установочных файлов

1. Проверка наличия инструкций для воспроизведения установочных файлов для операционных систем семейства Windows, macOS, Ubuntu

### 3. Проверка руководства пользователя

1. Проверка наличия руководства пользователя
2. Проверка руководства пользователя на содержание следующих разделов:
  - Добавление книг
  - Просмотр добавленных книг в библиотеке приложения
  - Просмотр метаданных книг
  - Открытие книги
  - Навигация по книге через листания страниц и элементов оглавления
  - Перевод выделенного текста книги
  - Добавление выделения текста в книгу
  - Добавление заметки текста в книгу
  - Добавление закладки в книгу
  - Запоминание последней открытой страницы книги
  - Кастомизация шрифта книг

### 4. Воспроизведение установочных файлов по инструкциям

1. Воспроизведение установочных файлов Приложения на компьютере с операционной системой Windows 10
2. Воспроизведение установочных файлов Приложения на компьютере с операционной системой macOS 12
3. Воспроизведение установочных файлов Приложения на компьютере с операционной системой Ubuntu 18

### 5. Установка Приложения используя полученные установочные файлы:

1. Установка Приложения на компьютере с операционной системой Windows 10
2. Установка Приложения на компьютере с операционной системой macOS 12
3. Установка Приложения на компьютере с операционной системой Ubuntu 18
6. Проверка соответствия установленного Приложения требованиям
  1. Открытие Приложения на компьютере с операционной системой Windows 10
  2. Последовательная, ручная проверка приложение на наличие функциональности, описанной в требованиях к программе
7. Проведение **UAT** с участием двух "конечных пользователей" для каждой из операционных систем поддерживаемых Приложением, на основе 3-х тест-кейсов:

## ☰ Test Case №1

### 📋 Status (tag)

Pass/Fail/Blocked

## Feature

User trades stocks

Представление "Оболочка приложения" выполняет функции нативной панели заголовка (**Title Bar**)

## Scenario

User requests a sell before close of trading

Пользователь запускает приложение

## Given

The **given** part describes the state of the world before you begin the behavior you're specifying in this scenario. It can be thought as the pre-conditions to the test

Исходя из того, что у пользователя устройство с операционной системой macOS

## When

The **when** section is that behavior that you're specifying

Когда пользователь запускает приложение

## Then

Finally the **then** section describes the changes you expect due to the specified behavior

То тогда, группа кнопок **Minimize**, **Maximize** и **Close** находится в левой части **Title Bar'a**



## ☰ Test Case №2

### 📄 Status (tag)

Pass/Fail/Blocked

## Feature

User trades stocks

Представление "Библиотека" позволяет добавлять книги

## Scenario

User requests a sell before close of trading

Пользователь попытался добавить файл (книгу) с неподдерживаемым расширением

## Given

The **given** part describes the state of the world before you begin the behavior you're specifying in this scenario. It can be thought as the pre-conditions to the test

Исходя из того, что у пользователя нет добавленных книг

## When

The **when** section is that behavior that you're specifying

Когда пользователь перетаскивает файл с расширением `.jpg` в экран приложения, но еще не отпустил кнопку

## Then

Finally the **then** section describes the changes you expect due to the specified behavior

То тогда, дропзона меняет цвет на красный и выводит сообщение о том что этот файл не поддерживается

## ☰ Test Case №3

### 📋 Status (tag)

Pass/Fail/Blocked

## Feature

User trades stocks

Представление "Чтение" *сохраняет прогресс чтения*

## Scenario

User requests a sell before close of trading

Пользователь открывает книгу, которую уже читал

## Given

The **given** part describes the state of the world before you begin the behavior you're specifying in this scenario. It can be thought as the pre-conditions to the test

Исходя из того что пользователь долистал книгу "TestBook.epub" до главы 2, и закрыл ее

## When

The **when** section is that behavior that you're specifying

Когда пользователь открывает ее снова

## Then

Finally the **then** section describes the changes you expect due to the specified behavior

То тогда, книга "TestBook.epub" открывается сразу на главе 2