



# Тестирование программного обеспечения

---

+7 (913) 768 8364

Ул. Кутателадзе 4г, к.118

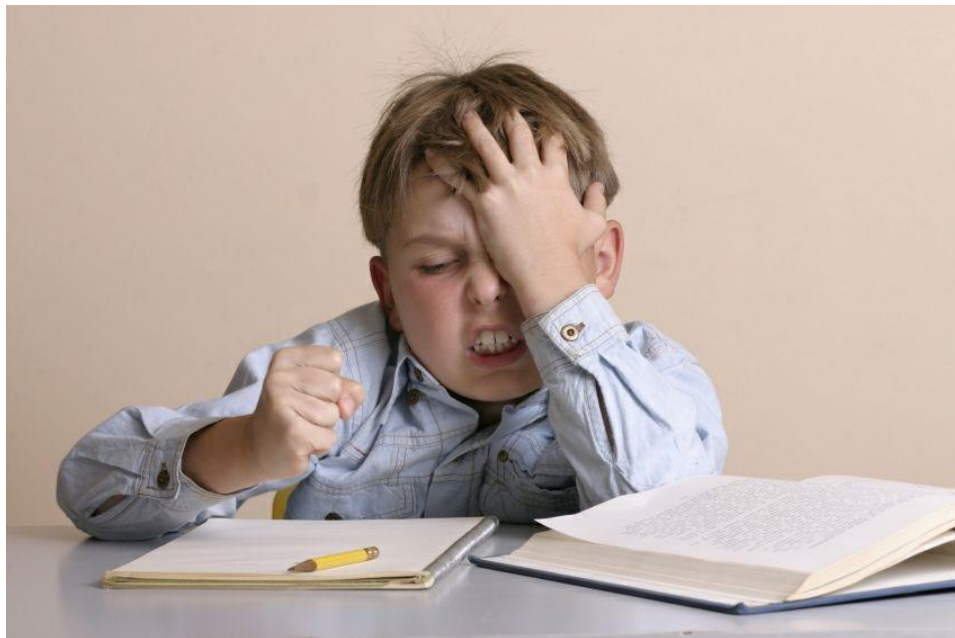
<https://academ-it-school.ru/>

# Вопросы



# Первое домашнее задание

- Как справились?
- Что изучали дополнительно?
- Есть ли вопросы?



# План занятия

- Степень важности функций приложения
- Классификация видов тестирования
- Тестовая документация
  - План тестирования
  - Чек-листы
  - Тест-кейсы
  - Наборы тест-кейсов
  - Матрица трассировки требований
  - Отчеты

# Глоссарий

- **Логи (logs), лог-файлы, журнал** – файлы, содержащие системную информацию работы сервера или компьютера, в которые заносятся определенные действия пользователя или программы.
- **Сборка (build)** – подготовленный для использования продукт. Может быть как одним файлом (например, установщик или мобильное приложение), так и набором файлов.

# Глоссарий

- **Интерфейс, пользовательский интерфейс, UI** – «внешняя» часть ПО; то, что видит и с чем взаимодействует пользователь
- **Фронт-энд** – функциональность и пользовательский интерфейс, работающие на клиентской части приложения или веб-сайта.
- **Бэк-энд** – программно-аппаратная часть сервиса, внутренняя логика программы, серверная часть приложения или веб-сайта

# Степень важности функций приложений

- Базовые функции, без которых существование приложения теряет смысл (самые важные функции — то, без чего приложение не будет работать, а также ключевые требования — ради чего приложение вообще создавалось)
- Функции, востребованные типичными пользователями в их повседневной работе
- Остальные функции (вся остальная заявленная функциональность; расширенные возможности; разнообразные «мелочи»; проблемы, которые не сильно повлияют на ценность приложения для конечного пользователя)

# Задание

Требуется протестировать консольное кроссплатформенное приложение, которое конвертирует текстовые файлы из кодировок WIN1251, CP866 и KOI8R в кодировку UTF8. Файлы могут иметь следующие разрешения: Plain Text (TXT), Hyper Text Markup Language Document (HTML) и Mark Down Document (MD). Приложение должно обрабатывать файлы размером до 50 МБ (включительно), игнорируя любой файл, размер которого превышает 50 МБ.

При запуске приложения оно получает из командной строки три параметра:

- SOURCE\_DIR — обязательный параметр, определяет путь к каталогу с файлами, которые необходимо обработать;
- DESTINATION\_DIR — обязательный параметр, определяет путь к каталогу, в который необходимо поместить обработанные файлы (этот каталог не может находиться внутри каталога SOURCE\_DIR или в его подкаталогах);
- LOG\_FILE\_NAME — необязательный параметр, определяет полное имя логфайла (по умолчанию лог-файл с именем «converter.log» размещается по тому же пути, по которому находится файл скрипта);

При указании недостаточного количества параметров командной строки приложение должно завершить работу, выдав сообщение об ошибке.

Остановка приложения производится нажатием Ctrl+C.



Функции, без которых существование приложения теряет смысл?

# Функции, без которых существование приложения теряет смысл

- Конфигурирование и запуск
  - С указанием корректных SOURCE\_DIR и DESTINATION\_DIR
- Обработка файлов
  - Форматы: TXT, HTML, MD
  - Кодировки: WIN1251, CP866 и KOI8R
  - Размер:  $\leq 50$  МБ
- Остановка
  - С помощью Ctrl+C

Функции, востребованные типичными пользователями в повседневной работе?

# Функции, востребованные типичными пользователями в повседневной работе

- Конфигурирование и запуск:
  - С верными параметрами:
    - Значения SOURCE\_DIR, DESTINATION\_DIR, LOG\_FILE\_NAME указаны и содержат пробелы и кириллические символы
    - Значение LOG\_FILE\_NAME не указано
  - Без параметров
  - С недостаточным количеством параметров
  - С неверными параметрами:
    - Недопустимый путь SOURCE\_DIR.
    - Недопустимый путь DESTINATION\_DIR.
    - Недопустимое имя LOG\_FILE\_NAME.
    - DESTINATION\_DIR находится внутри SOURCE\_DIR.
    - Значения DESTINATION\_DIR и SOURCE\_DIR совпадают.

# Функции, востребованные типичными пользователями в повседневной работе

- Обработка файлов:
  - Разные форматы, кодировки и размеры (в том числе недопустимые)
    - Значения SOURCE\_DIR, DESTINATION\_DIR, LOG\_FILE\_NAME указаны и содержат пробелы и кириллические символы
    - Значение LOG\_FILE\_NAME не указано
  - «Проблемные» входные файлы:
    - Нет прав доступа к файлу.
    - Файл открыт и заблокирован.
    - Файл с атрибутом «только для чтения»

# Остальные функции и особые сценарии

- Конфигурирование и запуск:
  - Запуск двух и более копий приложения с:
    - Одинаковыми параметрами SOURCE\_DIR, DESTINATION\_DIR, LOG\_FILE\_NAME.
    - Одинаковыми SOURCE\_DIR и LOG\_FILE\_NAME, но разными DESTINATION\_DIR.
    - Одинаковыми DESTINATION\_DIR и LOG\_FILE\_NAME, но разными SOURCE\_DIR.
    - ...
- Обработка файлов:
  - Файл верного формата, в котором текст представлен в двух и более поддерживаемых кодировках одновременно
  - ...
  - Размер входного файла:
    - > 50 мб
    - ...
- Остановка
  - ...

# Классификация видов тестирования

- По степени важности тестируемых функций
- По цели тестирования (связанные с изменениями)
- По объекту тестирования
- По доступу к коду
- По привлечению конечных пользователей
- По критерию позитивности сценариев
- По степени изолированности тестируемых компонентов (по уровню детализации)
- По степени автоматизированности тестирования
- По степени подготовки к тестированию
- По цели тестирования
- По запуску кода на исполнение

# По степени важности тестируемых функций

- Дымовое тестирование (smoke testing)
- Тестирование критического пути (critical path testing)
- Расширенное тестирование (extended testing)
- Приемочное тестирование (acceptance testing)



# Дымовое тестирование (smoke testing)

- Тип тестирования программного обеспечения, направленный на проверку работы наиболее важных, критических функций в системе. Результат этого тестирования используется для определения того, достаточно ли стабильна сборка, чтобы продолжить дальнейшее тестирование.

# Тестирование критического пути (critical path testing)

- При тестировании критического пути проверяются сценарии типичных пользователей в типичной деятельности

# Расширенное тестирование (extended testing)

- Исследование всей заявленной в требованиях функциональности (но при этом учитывается, какая функциональность является более важной, а какая — менее важной)
- Нетипичные, маловероятные случаи и сценарии использования функций и свойств приложения

# Приемочное тестирование (acceptance testing)

- Вид тестирования, проводимый на этапе сдачи продукта заказчику
  - Цель – определить готовность продукта перед его выпуском
  - Проверяется только основной функционал на соответствие требованиям
  - Может проводиться самим заказчиком

# По цели тестирования (связанные с изменениями)

- Дымовое тестирование (smoke testing)
  - Направлено вширь, для покрытия тестами как можно большего функционала в кратчайшие сроки
- Санитарное тестирование (sanity testing)
- Тестирование новой функции (new feature testing)
- Повторное тестирование (re-testing)
- Регрессионное тестирование (regression testing)

Что почитать:

<https://habr.com/ru/post/358142/>

# Санитарное тестирование (sanity testing)

- Краткая углубленная проверка определенной функциональности ПО, которая проводится с целью убедиться, что система работает должным образом
- Направлено вглубь проверяемой функции
- Как правило, проводится, когда исправлена какая-либо незначительная ошибка в системе или есть небольшое изменение в функциональности

# Тестирование новой функции (new feature testing)

- Тестирование новой функциональности, появившейся в конкретном выпуске (release) или сборке (build) программного продукта.

# Повторное тестирование (re-testing)

- Проводится для подтверждения исправления ошибки и корректности работы данного функционала



# Регрессионное тестирование (regression testing)

- Тестирование, проводимое с целью убедиться, что новые (добавленные) функции приложения и исправленные дефекты не оказали влияния на текущую, уже существующую функциональность, работавшую (и протестированную) ранее.
- Как правило, проводится перед выпуском билда в релиз

# По объекту тестирования

- Функциональное тестирование
- Нефункциональное тестирование
  - Тестирование интерфейса пользователя
  - Тестирование юзабилити
  - Тестирование локализации
  - Тестирование производительности
  - Тестирование безопасности
  - Тестирование совместимости

# Функциональное тестирование

- Тестирование отдельных функций
- Проводится с точки зрения пользователя
- Направлено на проверку способности ПО решать задачи, нужные пользователям

# Нефункциональное тестирование

- Тестирование качественных характеристик компонента или системы, которые могут быть измерены различными величинами, не относящимися к конкретной функции или действию пользователя

# Тестирование интерфейса пользователя (user interface testing, UI)

- Тестирование, при котором проверяются элементы интерфейса пользователя
- Проверки чаще всего оформляются в виде чек-листа

Что почитать:

<https://qaevolution.ru/testirovanie-po/vidy-testirovaniya-po/testirovanie-polzovatelskogo-interfejsa/>

# Тестирование юзабилити

- Призвано объективно оценить опыт пользователя, который будет работать с разрабатываемым интерфейсом (насколько пользователю удобно использовать продукт)
- При юзабилити-тестировании также проверяется интуитивность интерфейса
- Проверки чаще всего оформляются в виде чек-листа

Что почитать:

<https://prof-mk.ru/ux-ui-raznica/>



# Тестирование локализации

- Подразумевает проверку множества аспектов, связанных с адаптацией продукта для пользователей из других стран

Что почитать:

<https://software-testing.ru/library/testing/other-testing/3131-localization-testing>

# Тестирование производительности

- Тестирование, целью которого является определение работоспособности, стабильности, потребления ресурсов и других атрибутов качества приложения в условиях различных сценариев использования и нагрузок



# Тестирование безопасности

- Тестирование, цель которого заключается в том, чтобы проверить устойчивость приложения к внешнему доступу и ограничить условия для «кражи» — кражи данных, денег и информации
  - Целостность: ограничение круга пользователей, имеющих доступ к данным
  - Доступность: ресурсы должны быть доступны авторизованному пользователю, внутреннему объекту или устройству
  - Конфиденциальность: при каких условиях пользователь авторизован получить доступ к данным

# Тестирование совместимости

- Проверка того, как продукт взаимодействует с
  - «Железом» (аппаратным обеспечением, hardware)
  - ПО (браузерами/операционными системами) пользователей
- Тестирование с разными браузерами называется **кросс-браузерным** тестированием.
- Тестирование с разными ОС называется **кросс-платформенным** тестированием.

# По доступу к коду

- Черный ящик (black box testing)
- Белый ящик (white box testing)
- Серый ящик (grey box testing)

# Черный ящик (black box testing)

- Тестировщик не знает, как устроен продукт изнутри
- Идеи тестирования идут от образцов поведения пользователя:
  - Взяты из ТЗ
  - Найдены путем исследования продукта
  - Интуиция
  - Совет разработчика



# Белый ящик (white box testing)

- При этом подходе идеи для тестирования основываются на знании об устройстве и логике тестируемой части бэк-энда

# Серый ящик (grey box testing)

- Комбинация white box и black box подходов
- Внутреннее устройство программы известно лишь частично

# По привлечению конечных пользователей

- Альфа-тестирование (alpha testing)
  - Проводится внутри компании силами штатных тестировщиков. Имитация реального использования продукта
- Бета-тестирование (beta testing)
  - Использование почти готового продукта с привлечением ограниченного числа внешних пользователей
- Гамма-тестирование (gamma testing)
  - Исправление последних дефектов, идеи и улучшения. Часто происходит уже после выпуска

Что почитать:

<https://qatestlab.com/resources/knowledge-center/alpha-beta-gamma/>

# По критерию позитивности сценариев

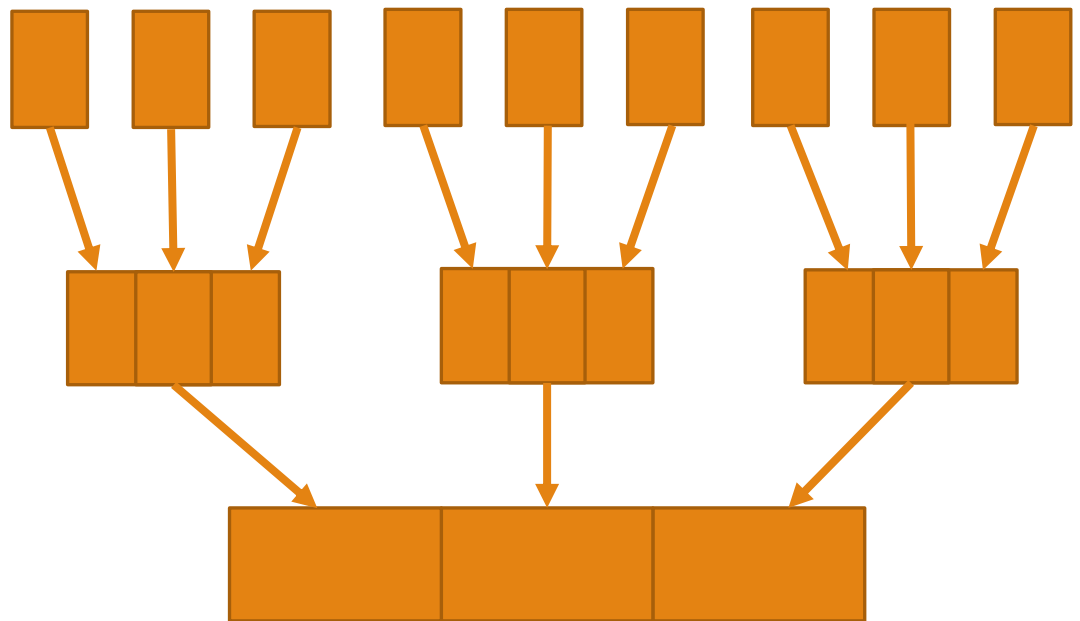
- **Позитивное тестирование (positive testing)**
  - Использует корректные данные и проверяет, что приложение правильно выполняет функции.
  - Направлено на проверку работы системы с теми типами данных, для которых она разрабатывалась.
  - Предполагает нормальное, «правильное» использование, работу системы.
- **Негативное тестирование (negative testing)** – тестирование, в рамках которого применяются сценарии, которые соответствуют внештатному поведению тестируемой системы.
  - Направлено на проверку устойчивости системы к различным воздействиям, валидации неверных данных, обработке исключительных ситуаций.
  - Проверяет ситуации, связанные с потенциальной ошибкой пользователя или потенциальным дефектом в системе.



# По степени изолированности тестируемых компонентов (по уровню детализации)

Компонентное (component testing) – тестирование на уровне логического компонента

Интеграционное (integration testing) – тестирование на уровне двух или больше компонентов



Системное (system testing) – проверка всей системы от начала до конца

# Пример

- Написать код, который бы находил полные имена и e-мейлы пользователей, потративших больше 1000\$ в онлайн-магазине с момента регистрации.
- Таким пользователям должен быть отправлен e-мейл с подарочным сертификатом, использование которого до 17 февраля включительно предоставит 5%-ю скидку на любую разовую покупку.

# Пример

1. Создание файла с полными именами, е-мейлами и номерами сертификатов
2. Рассылка пользователям е-мейлов
3. Правильное предоставление скидки вышеуказанным пользователям

# По степени автоматизированности тестирования

- Ручное тестирование (manual testing)
  - Исполнение тест-кейсов без помощи каких-либо программ, автоматизирующих работу
- Автоматизированное тестирование (automated testing)
  - Инструменты для помощи в черном и сером тестировании
  - Программы для регрессионного тестирования
- Смешанное/полуавтоматизированное тестирование

# По степени подготовки к тестированию

- Тестирование по тест-кейсам (scripted testing)
- Исследовательское тестирование (exploratory testing)
  - Одновременное изучение программного продукта, проектирование тестов и их выполнение
- Интуитивное тестирование (ad-hoc testing)
  - Выполняется без подготовки к тестам, без определения ожидаемых результатов, проектирования тестовых сценариев
  - Неформальное, импровизационное тестирование

Что почитать:

<https://training.qatestlab.com/blog/technical-articles/what-is-ad-hoc-testing/>

# По запуску кода на исполнение

- Динамическое (dynamic testing)
- Статическое (static testing):
  - Документы
  - Графические прототипы/эскизы
  - Код приложения
  - Настройки приложения
  - Тестовые данные

# Тест-кейс

- **Тест-кейс (test case)** — набор входных данных, условий выполнения и ожидаемых результатов, разработанный с целью проверки того или иного свойства или поведения программного средства.
- Под тест-кейсом также может пониматься соответствующий документ, представляющий формальную запись тест-кейса.

Что почитать:

<https://training.qatestlab.com/blog/technical-articles/test-case-topic/>

# Виды тест-кейсов

- **Высокоуровневый тест-кейс (high level test case)** — тест-кейс без конкретных входных данных и ожидаемых результатов
- **Низкоуровневый тест-кейс (low level test case)** — тест-кейс с конкретными входными данными и ожидаемыми результатами



# Глоссарий

- **Тестовое покрытие (test coverage)** – одна из метрик оценки качества тестирования, представляющая из себя плотность покрытия тестами требований либо исполняемого кода. Подробнее:  
<http://www.protesting.ru/testing/testcoverage.html>

# Цель написания тест-кейсов

- Структурировать и систематизировать подход к тестированию
- Вычислять метрики тестового покрытия (test coverage metrics) и принимать меры по его увеличению
- Отслеживать соответствие текущей ситуации плану
- Уточнить взаимопонимание между заказчиком, разработчиками и тестировщиками
- Хранить информацию для длительного использования и обмена опытом между сотрудниками и командами
- Проводить регрессионное тестирование и повторное тестирование
- Быстро вводить в курс дела нового сотрудника, недавно подключившегося к проекту

# Поля тест-кейса

Поле	Описание
Идентификатор (ID)	Должен быть уникальным (т.е. не повторяться)
Приоритет (Priority)	Пример: А – высокий В – средний С – низкий D – незначительный
Связанное требование (Related task)	Может быть ссылкой на документ или задачу в системе хранения тестов/багтрекере
Название (Summary)	[Модуль]. [Подмодуль]. Краткое описание
Предусловия (Preconditions)	Приготовления перед выполнением тест-кейса; описание окружения
Шаги воспроизведения (Steps to reproduce)	Список шагов для выполнения тест-кейса
Ожидаемый результат (Expected Result)	Может прописываться по конкретным шагам, а может быть общим для всего тест-кейса

# Поля тест-кейса

2.3

**Чат. Удаление сообщений. Удаление сообщения, содержащего вложения**

**Приоритет:** Высокий

**Связанное требование:** РК «Загрузка файлов», п. 1.1.4

**Предусловия:**

1. Подготовить файлы разрешенных к отправке форматов (jpeg, docx, pdf)
2. Проверить, что в чате пользователей Test111 и Test222 нет этих файлов в истории вложений

Шаги воспроизведения	Ожидаемый результат
1. Авторизоваться под пользователем Test111	
2. Зайти в чат с собеседником Test222	
3. Отправить сообщение с подготовленными вложениями	3. В чате в истории вложений обоих пользователей появились отправленные файлы
4. Удалить отправленное сообщение	4. Отправленное сообщение пропадает из чата, файлы пропадают из истории вложений

# Поля тест-кейса

Идентификатор (ID)	TEST-1453	
Приоритет (Priority)	С	
Связанное требование (Related task)	TEST-285	
Название (Summary)	Регистрация с некорректным e-mail	
Предусловия (Preconditions)	Открыть сайт <a href="http://www.test.com">www.test.com</a>	
Шаги воспроизведения (Steps to reproduce)	<ol style="list-style-type: none"><li>1. Нажать кнопку «Зарегистрироваться»</li><li>2. Ввести e-mail (см. ожидаемый результат) и корректный пароль</li><li>3. Нажать кнопку «Продолжить»</li></ol>	
Ожидаемый результат (Expected Result)	Значение	Результат
	w	Ошибка «E-mail указан некорректно»
	w@	Ошибка «E-mail указан некорректно»
	<a href="#">w'''n@a.com</a>	Ошибка «E-mail не может содержать спецсимволы»
	тест@a.com	Ошибка «E-mail не может содержать кириллицу»

# Заглавие (название)

- Информативность
- Хотя бы относительная уникальность (чтобы не путать разные тест-кейсы)
- Краткое и емкое, четко отражающее суть тест-кейса

Плохо	Хорошо
Тест 1	Запуск, одна копия, верные параметры
Тест 2	Запуск одной копии с неверными путями
Тест 78 (улучшенный)	Запуск, много копий, без конфликтов
Остановка	Остановка по Ctrl+C
Закрытие	Остановка закрытием консоли
...	...

# Наборы тест-кейсов

- Набор тест-кейсов (test case suite, test suite, test set) — совокупность тест-кейсов, выбранных с некоторой общей целью или по некоторому общему признаку.
- Иногда в такой совокупности результаты завершения одного тест-кейса становятся входным состоянием приложения для следующего тест-кейса.

# Наборы тест-кейсов

- Преимущества **свободных наборов**:
  - Тест-кейсы можно выполнять в любом удобном порядке, а также создавать «наборы внутри наборов».
  - Если какой-то тест-кейс завершился ошибкой, это не повлияет на возможность выполнения других тест-кейсов.
- Преимущества **последовательных наборов**:
  - Каждый следующий в наборе тест-кейс в качестве входного состояния приложения получает результат работы предыдущего тест-кейса, что позволяет сильно сократить количество шагов в отдельных тест-кейсах.
  - Длинные последовательности действий лучше имитируют работу реальных пользователей, чем отдельные «точечные» воздействия на приложение.



# Принципы построения наборов тест-кейсов

- На основе чек-листов
- По видам тестирования
  - По принципу проверки самых важных, менее важных и всех остальных функций приложения
- На основе разбиения приложения на модули и подмодули
- По принципу частоты обнаружения тест-кейсами дефектов в приложении

# Домашнее задание 2

- Задание 1
  - Требуется протестировать функции регистрации, авторизации, поиска и добавления товара в корзину для сайта <https://www.sportmaster.ru/>
  - Написать план тестирования
  - В план тестирования добавить тест-кейсы или наборы тест-кейсов
    - Если составляется набор тест-кейсов, то в описании набора указать принцип объединения
    - Учитывать приоритеты при ограниченном времени
    - Всего тест-кейсов должно быть 10 (не больше 10, не меньше 10)
    - Классифицировать каждый тест-кейс по всем видам тестирования
  - Написать краткий отчет по результатам тестирования (в свободной форме – какие выводы можно сделать)
- Задание 2
  - Составить чек-лист для тестирования UI/UX сайта <https://www.sportmaster.ru/>
    - 5 проверок UI
    - 5 проверок UX