Министерство образования и науки РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

Воронежский государственный технический университет

ПЛАНИРОВАНИЕ ТЕСТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Методические указания
к выполнению лабораторных работ по дисциплине "«Верификация и оценка качества программного обеспечения »
для студентов магистратуры направления 09.04.02
«Информационные системы и технологии»

Воронеж 2019

Содержание

Лабораторная №1 Запуск системы и знакомство Kiwi TCMS

Цель работы: изучить процесс запуска системы Kiwi TCMS и познакомиться с основными ее функциями

1. Теоретические сведения

Тестирование программного обеспечения (Software Testing) - проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом.

В более широком смысле, тестирование - это одна из техник контроля качества, включающая в себя активности по планированию работ (Test Management), проектированию тестов (Test Design), выполнению тестирования (Test Execution) и анализу полученных результатов (Test Analysis).

Основные цели тестирования:

- Обнаружение дефектов (раньше, чем их найдёт клиент);
- Повышение уверенности в уровне качества ПО;
- Предотвращение появления дефектов;

Команда разработчиков создала канонический источник информации о тестировании, который будет полезен как менеджерам, так и сотрудникам QE Associates.

Цели тестирования могут отличаться, в зависимости от этапа разработки ПО, на котором оно проводится. К примеру, на этапе кодирования целью тестирования будет вызов как можно большего количества сбоев в работе программы, что позволит локализовать и исправить дефекты. В то же время, при приемочном тестировании необходимо показать, что система работает правильно. В период сопровождения, тестирование в основном необходимо

для того, чтобы убедится в отсутствии новых багов, появившихся во время внесения изменений.

Семь принципов тестирования:

- Принцип 1 — Тестирование демонстрирует наличие дефектов (Testing shows presence of defects);
- Принцип 2 — Исчерпывающее тестирование недостижимо (Exhaustive testing is impossible);
- Принцип 3 — Раннее тестирование (Early testing);
- Принцип 4 — Скопление дефектов (Defects clustering);
- Принцип 5 — Парадокс пестицида (Pesticide paradox);
- Принцип 6 — Тестирование зависит от контекста (Testing is concept depending);
- Принцип 7 — Заблуждение об отсутствии ошибок (Absence-of-errors fallacy).

Чтобы успешно смоделировать рабочий процесс тестирования с помощью Kiwi TCMS, необходимо понимать, как внутренние структуры данных связаны друг с другом(рис.1).

План тестирования - это контейнерный объект высокого уровня, который используется для описания действий по тестированию. Наиболее важные скалярные свойства - это текст документа и тестируемый продукт. Наиболее важные агрегированные свойства - это список тестовых случаев и тестовых запусков (также называемых тестовыми выполнениями).

Тестовый план описывает, как выполнить конкретный сценарий. Этот объект используется для записи сценариев в базу данных.

Для каждой тестируемой сборки продукта необходимо создать артефакт Test Run. Test Run содержит набор объектов Test Execution, которые создаются тестировщиком по умолчанию, отметки времени начала / окончания и т. Д.

Объект Test Execution - это контейнер, который связывает фактический сценарий тестирования с его статусом выполнения, дополнительными

комментариями и ошибками, обнаруженными во время тестирования. Объект Test Execution имеет отношение 1 к 1 к Test Case, для которого он имеет статус. Таким образом, объект Test Execution представляет собой реальный тестовый пример, который будет / был выполнен для конкретной сборки продукта.

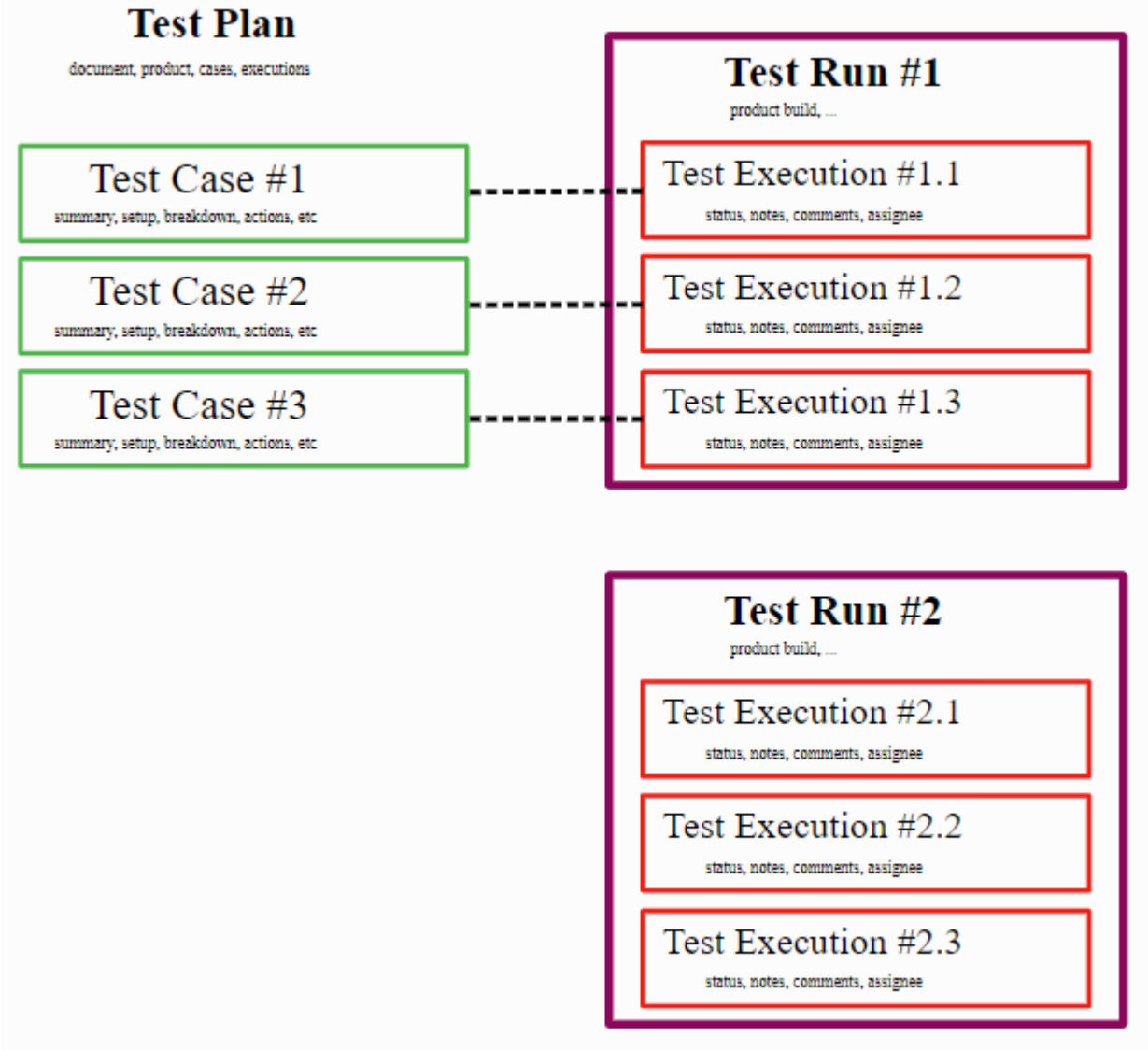

Рисунок 1 - Организация данных в Kiwi TCMS

Домашняя страница - это основной раздел TCMS. Он обеспечивает быстрый доступ к назначенным тестовым запускам и планам тестирования. В этом разделе объясняются функции, доступные с домашней страницы.

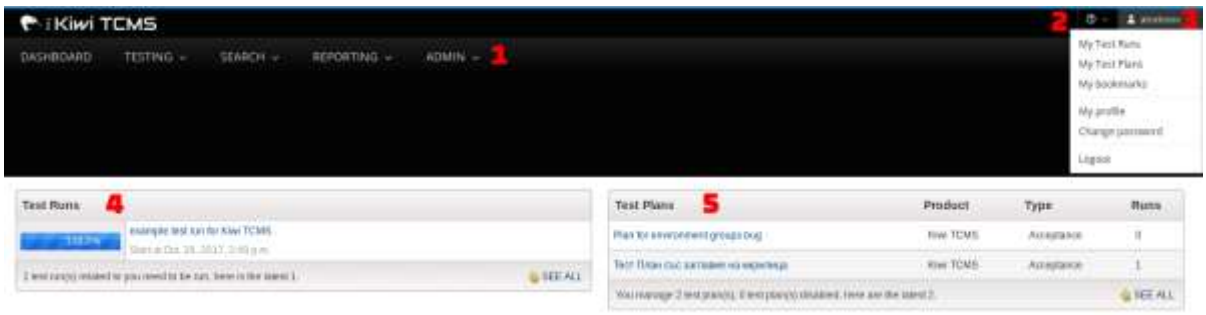

Рисунок 2 – Домашняя страница

1. Панель навигации
2. Меню помощи
3. Персональное меню
4. Панель управления тестовым запуском
5. Панель управления вашего плана тестирования

Панель навигации состоит из главного меню, меню справки и личного меню. Когда вы происходит щелчок по пункту меню, появляется подменю.

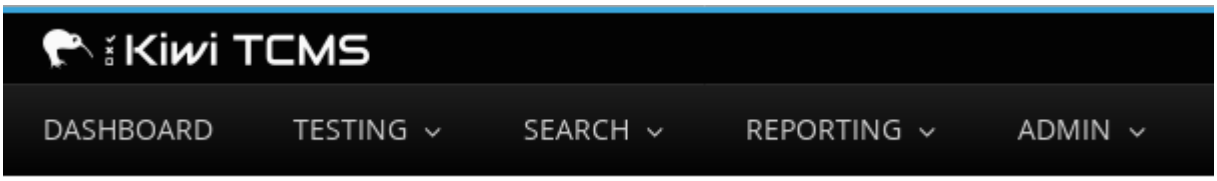

Рисунок 3 – Панель навигации

Главный экран содержит список тестовых запусков, связанных с пользователем, и графический дисплей, показывающий состояние их завершения. Чтобы получить доступ к пробному запуску, необходимо щелкнуть на его имя.

Главный экран также содержит список планов тестирования, связанных с пользователем. Чтобы получить доступ к плану тестирования, нужно щелкнуть на его имя.

В меню «Справка» есть дополнительные ссылки на страницы справки:

- Сообщить о проблеме с Kiwi TCMS
- Гид пользователя

- Руководство администратора
- Документ API службы XML-RPC
- Версия Kiwi TCMS

2. Пример решения поставленной задачи

Первым делом нужно зарегистрироваться в системе для дальнейшей работы.

Для этого:

1. Необходимо осуществить переход по ссылке: https://public.tenant.kiwitcms.org/accounts/login/
2. Откроется страница, изображенная на рис.4.

Рисунок 4 – Страница для авторизации пользователя

3. Так как ранее система не использовалась, нужно зарегистрироваться, нажав на кнопку Зарегистрировать аккаунт, в правой части страницы. Откроется страница с пустыми полями для заполения (рис.5).

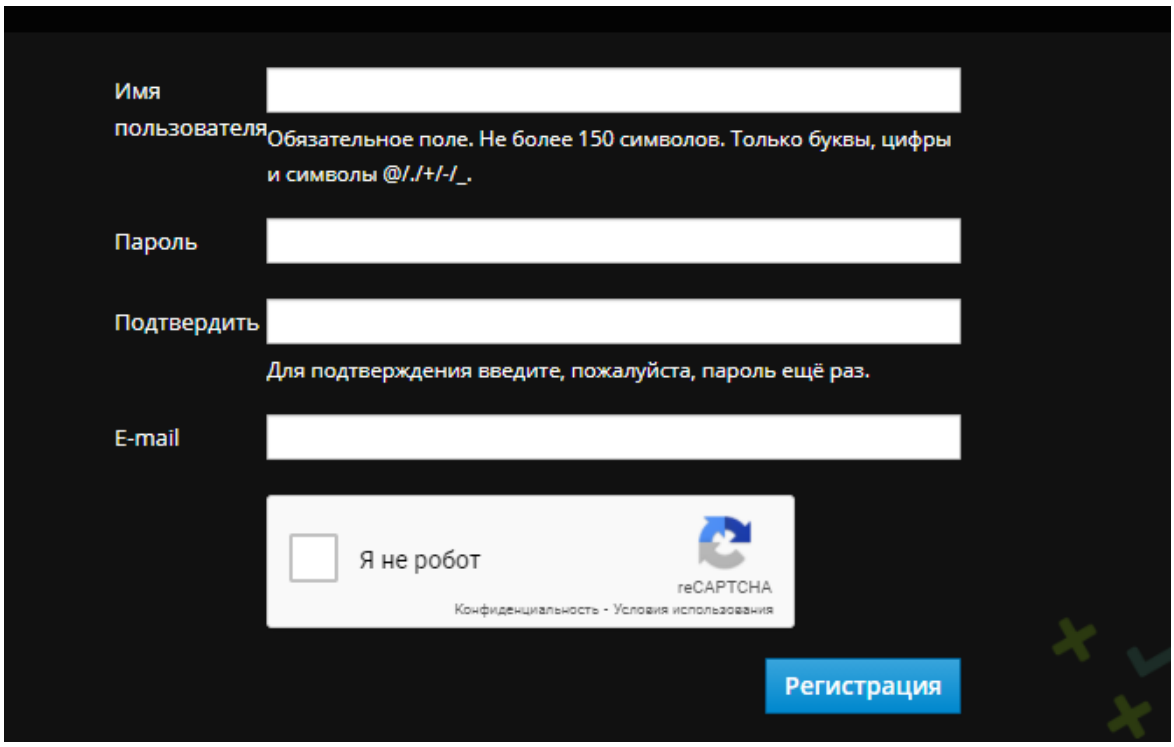

Рисунок 5 – Регистрация нового пользователя в системе

После заполнения всех полей на почту приходит письмо с подтверждением регистрации. Чтобы начать работать, нужно просто перейти по ссылке.

Лабораторная №2 Создание и редактирование плана тестирования

Цель работы: изучить теоретические сведения о плане тестирования, научиться работать с планом тестирования в системе Kiwi TCMS.

1. Теоретические сведения

Тест-план (Testplan, план тестирования) – это документ, описывающий весь объем работ по тестированию, начиная с описания тестируемых объектов, стратегии, расписания, критериев начала и окончания тестирования, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения.

Тест-план является важной составляющей любого грамотно-организованного процесса тестирования, так как содержит в себе всю необходимую информацию, описывающую данный процесс. Но в большинстве случаев, с которыми придется столкнуться, тест-план будет играть более формальную роль, но, все же, его присутствие имеет много преимуществ. Например:

- Возможность приоритезации задач по тестированию.
- Построение стратегии тестирование, согласованной со всей командой.
- Возможность вести учет всех требуемых ресурсов, как технических, так и человеческих.
- Планирование использования ресурсов на тестирование.
- Просчет рисков, возможных при проведении тестирования.

В зависимости от конкретизации описываемых задач, тест-план может иметь два уровня детализации: мастер тест-план и детальный тест-план.

Детальный тест-план содержит в себе задачи тестирования для каждой команды, для каждого релиза или итерации проекта. Создается детальный тест-план либо для декомпозированной части проекта, либо для небольших проектов. Он может состоят с:

- Перечень областей тестирования с приоритетами.
- Стратегия тестирования.
- Перечень возможных рисков.
- Перечень необходимых ресурсов.
- План выполнения проекта.

Мастер тест-план в свою очередь создается либо для организации процесса тестирования между несколькими командами, которые тестируют один проект, но имеют разные задачи, либо для проекта, который состоит из множества итераций, которые связывает какая-нибудь общая информация, повторение которой в каждом релизе занимает слишком много времени. В мастер тест-план входит:

- Общая информация о проекте (ссылки на документацию, баг-трекеры, и т.д.)
- Положения, описывающие процесс тестирования, заведения дефектов и т.д.
- Критерии готовности продукта к выпуску.

Для облегчения жизни тестировщикам, существуют несколько шаблонов тест-планов (IEEE, RUP).

2. Пример создания и работы с планом тестирования

План тестирования должен определять, какие функции продукта будут тестироваться и какова общая стратегия тестирования. Это документ высокого уровня, который не должен включать конкретные шаги тестирования. Рекомендуется использовать формат плана тестирования IEEE 829, но также можно вводить любой текст. Чтобы создать план тестирования:

1. В главном меню выберите тестирование - Новый план тестирования (рис.1).

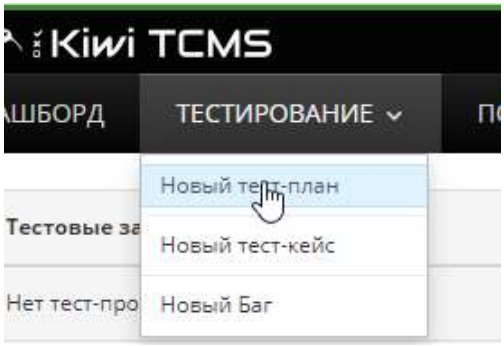

Рисунок 1 – Создание плана тестирования

На экране «Создать новый план тестирования» необходимо выполнить следующие действия:

- Ввести имя;
- Выбрать продукт;
- Выбрать версию;
- Выбрать тип плана тестирования;
- Ввести идентификатор родителя, это необязательно;
- В текстовом поле «План документа» ввести детали плана тестирования;
- Ввести ссылку на любую дополнительную информацию (например, страницу вики), это необязательно.

Далее нужно нажать на кнопку Сохранить (рис.2).

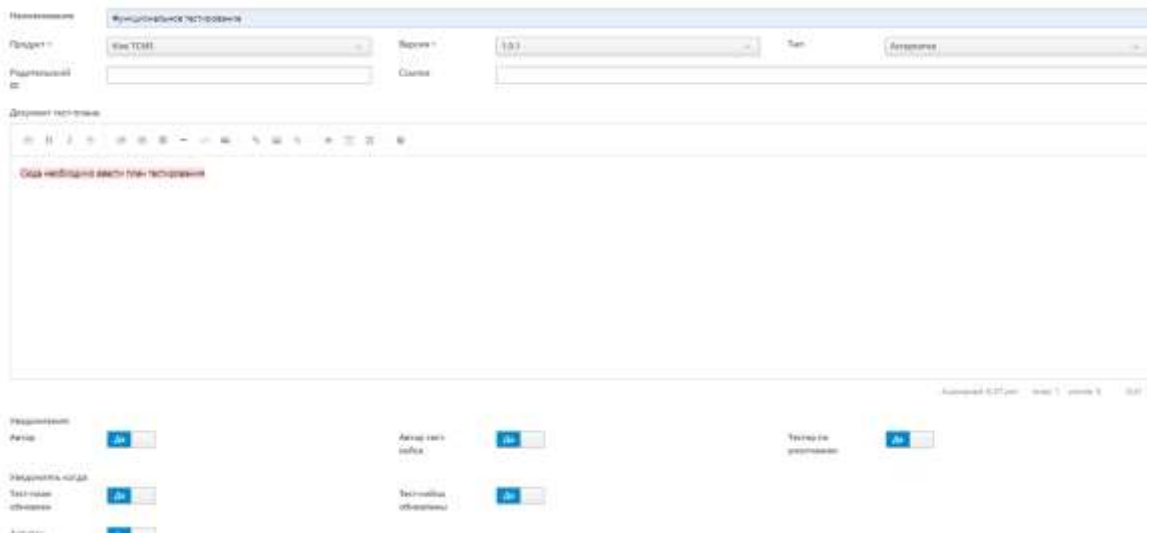

Рисунок 2 – Заполнение полей плана тестирования

Клонирование позволяет пользователю воспроизвести существующий план тестирования для нового продукта или новой версии текущего продукта. Чтобы клонировать план тестирования:

1. Необходимо открыть план тестирования, который будет клонирован.
2. В меню навигации объекта щелкнуть элемент «Клонировать».

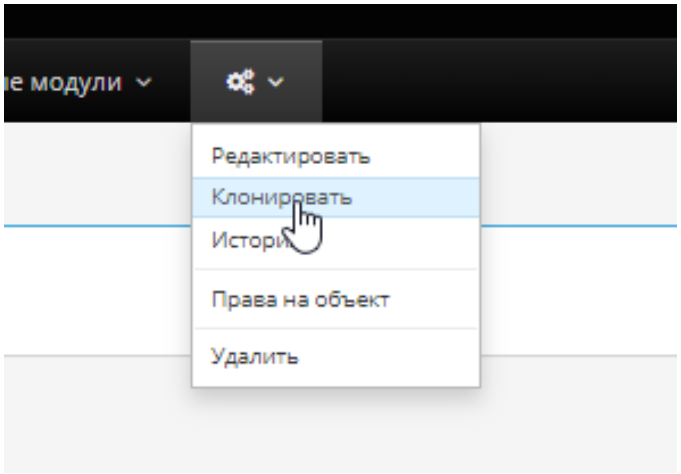

Рисунок 3 – Клонирование тест-плана

3. На экране клонирования тест-плана необходимо выполнить следующие действия:
    - Ввести имя.
    - В «Продукт» выбрать продукт.
    - В «Версия» выбрать версию.
    - Отметить Клонировать ТС, если нужно клонировать тестовые примеры, иначе они будут связаны только с новым ТС.
    - Отметить родительский ТР, если вы хотите создать отношение родитель-потомок к новому ТР.

Рисунок 4 – Заполнение дополнительных полей при клонировании

4. Нажать на кнопку «Клонировать», и будет создан новый план тестирования.

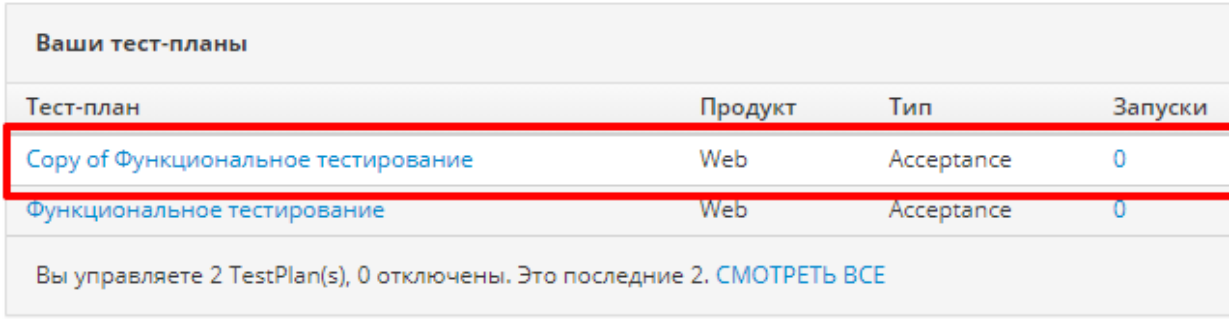

Рисунок 5 – Клонированный тест-план

Функция редактирования изменяет поля в плане тестирования. Это не меняет никаких тестовых случаев или тестовых прогонов, связанных с планом тестирования. Чтобы изменить план тестирования необходимо:

- Открыть план тестирования для редактирования.
- В меню навигации по объекту щелкнуть элемент «Редактировать».

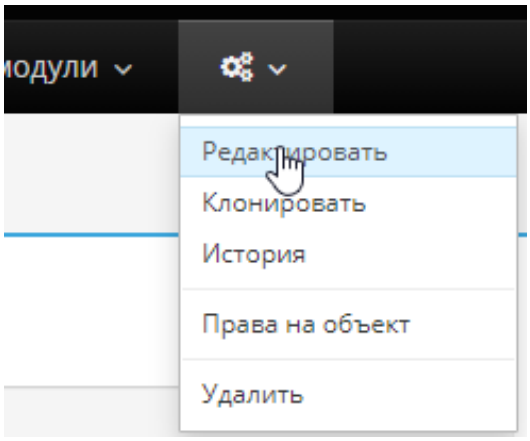

Рисунок 6 – Выбор элемента для редактирования тест-плана

Если необходимо, можно отредактировать поля(рис.7):

- Наименование
- Продукт
- Версия
- Тип
- Документ тест-плана

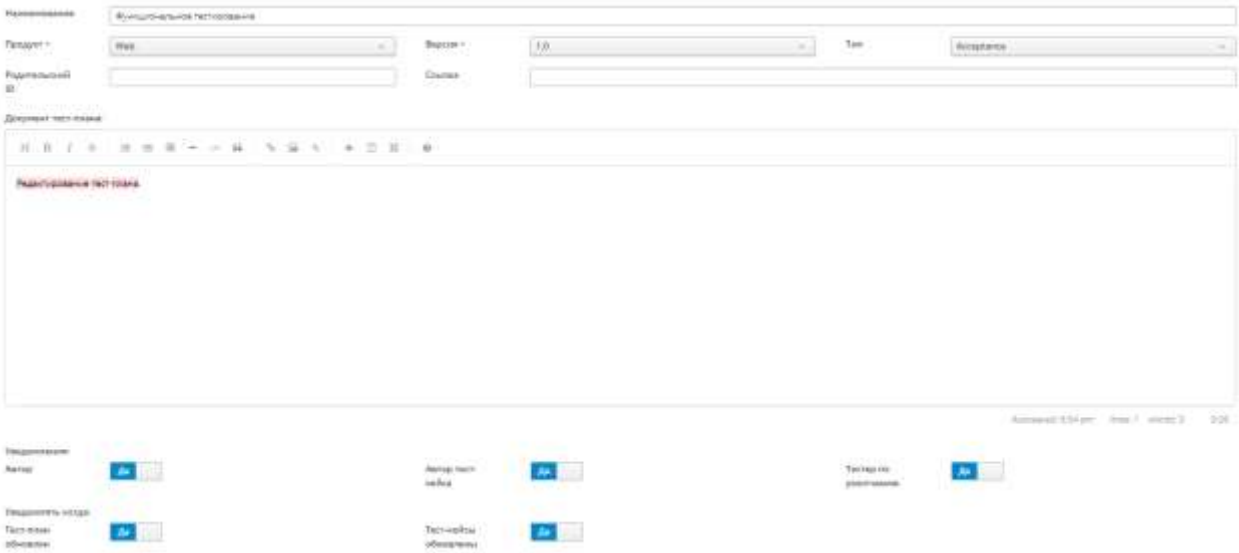

Рисунок 7 – Редактирование полей тест-плана

Лабораторная №3 Разработка и редактирование тестовых случаев

Цель работы: изучить теоретические сведения о тестовых случаях(тест-кейсах) и научиться создавать их в системе Kiwi TCMS.

1. Теоретические сведения

Тестовый случай (Test Case) - это артефакт, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части.

Тест кейсы разделяются по ожидаемому результату на позитивные и негативные:

- Позитивный тест кейс использует только корректные данные и проверяет, что приложение правильно выполнило вызываемую функцию.
- Негативный тест кейс оперирует как корректными, так и некорректными данными (минимум 1 некорректный параметр) и ставит целью проверку исключительных ситуаций (срабатывание валидаторов), а также проверяет, что вызываемая приложением функция не выполняется при срабатывании валидатора.

Структура данного артефакта имеет следующий вид:
Выполняемое действие (Action) – Ожидаемый результат (Expected result) – Фактический результат (Test result).

Каждый тест кейс должен иметь 3 части:

- PreConditions (Предусловия) - Список действий, которые приводят систему к состоянию пригодному для проведения основной проверки. Либо список условий, выполнение которых говорит о том, что система находится в пригодном для проведения основного теста состояния.
- Test Case Description (Описание тестового случая) - Список действий, переводящих систему из одного состояния в другое, для получения

результата, на основании которого можно сделать вывод о удовлетворении реализации, поставленным требованиям.

- PostConditions (Постусловия) - Список действий, переводящих систему в первоначальное состояние.

2. Пример создания тест-кейса в системе

Для создания тест-кейса в системе Kiwi TCMS необходимо выполнить следующие действия:

1. На вкладке Тестирование выбрать Новый тест-кейс.

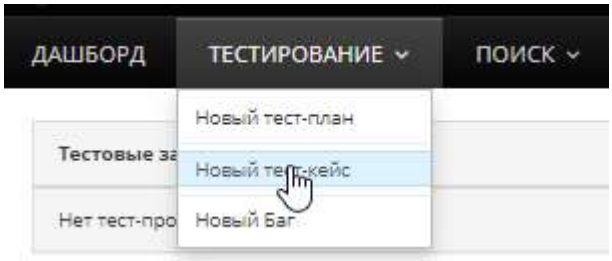

Рисунок 1 – Создание нового тест-кейса

2. На открывшейся странице заполнить все необходимые данные и нажать на кнопку Сохранить (рис.2).

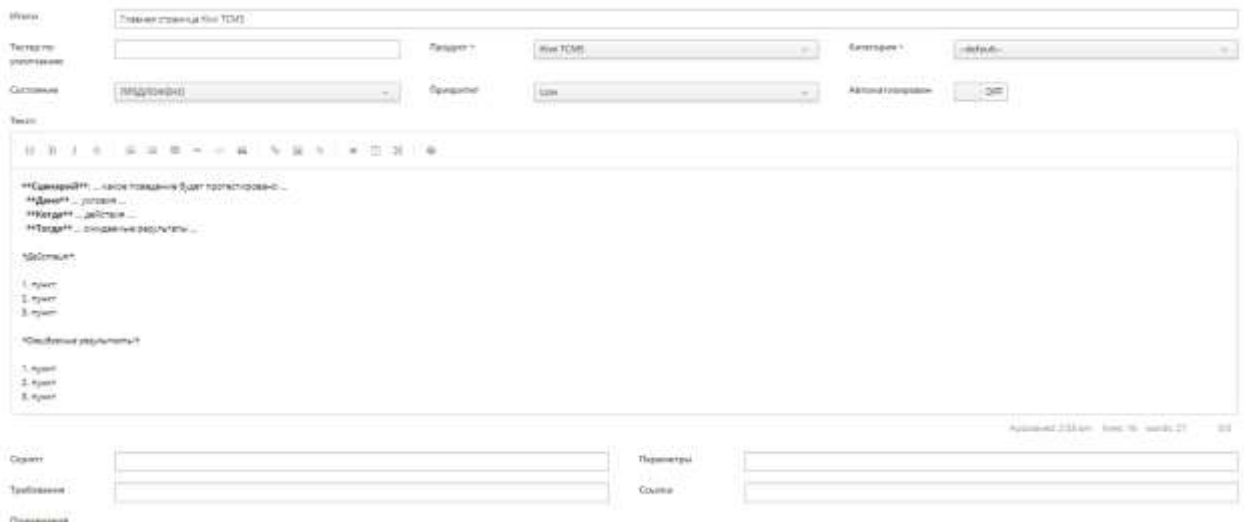

Рисунок 2 – Заполнение полей тест-кейса

При написании тест-кейса четкие инструкции по настройке помогают снизить вероятность сбоя из-за неправильной среды. Четкий набор действий с

измеримыми ожидаемыми результатами гарантирует, что тест-кейс дает согласованные результаты независимо от того, кто его выполняет. Должны быть предоставлены инструкции по поломке, чтобы гарантировать, что тестовая среда возвращается в исходное состояние.

Есть несколько способов связать тестовые наборы и планы тестирования:

- Добавить существующий план тестирования в тест-кейс;
- Добавить существующий тест-кейс в план тестирования.
- Создать тест-кейс из плана тестирования;

Чтобы связать план тестирования с существующим тест-кейсом необходимо:

1. Открыть существующий тест-кейс и найти карточку планов тестирования.
2. Ввести идентификатор или название плана в текстовое поле автозаполнения.
3. Выбрать подходящий план и нажать Enter или нажать кнопку +

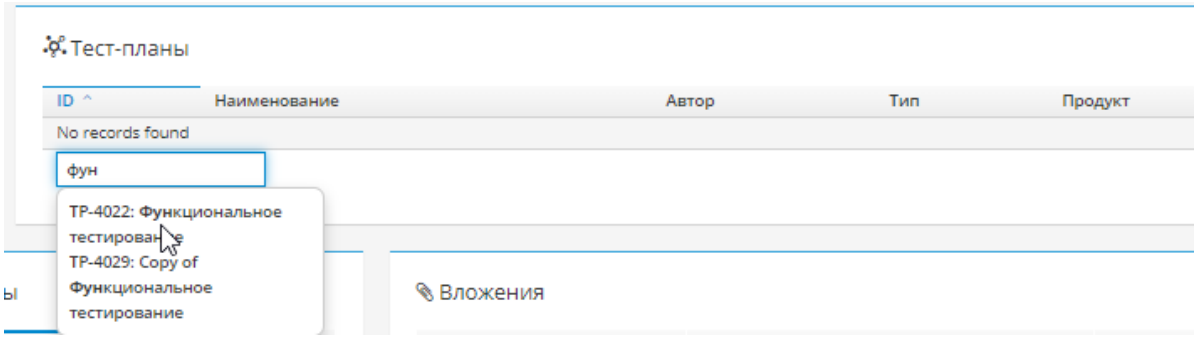

Рисунок 3 – Добавление существующего тест-плана в созданный тест-кейс

Существующие тестовые случаи могут быть добавлены в планы тестирования с помощью панели инструментов (рис.4), для этого необходимо:

1. Открыть существующий план тестирования.
2. Быстрый поиск по идентификатору плана или сводке с автозаполнением.

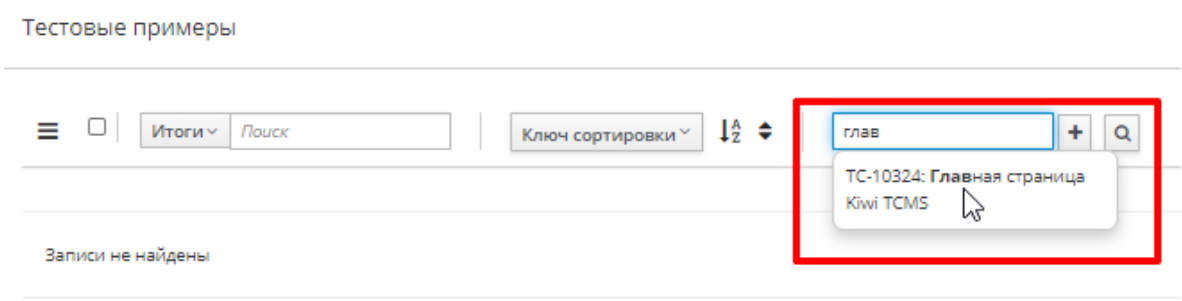

Рисунок 4 - Добавление существующего тест-кейса в готовый тест-план

Чтобы создать новый тестовый случай внутри документа плана тестирования необходимо:

1. Открыть существующий план тестирования, щелкнуть раскрывающееся меню и выбрать Новый тест-кейс(рис.5).

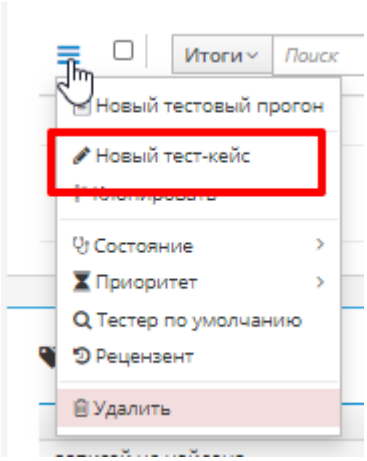

Рисунок 5 – Создание тест-кейса внутри тест-плана

Далее появится окно с полями для заполнения, изображенное на рис.2.

В Kiwi TCMS можно использовать рабочий процесс клонирования-редактирования, чтобы очень быстро создавать похожие тестовые случаи в рамках одного плана тестирования. Чтобы клонировать тест-кейс необходимо(рис.6):

1. Открыть тестовый случай, который нужно клонировать.
2. Щелкнуть Клонировать в меню навигации.
3. Обновить имя и нажать Сохранить.

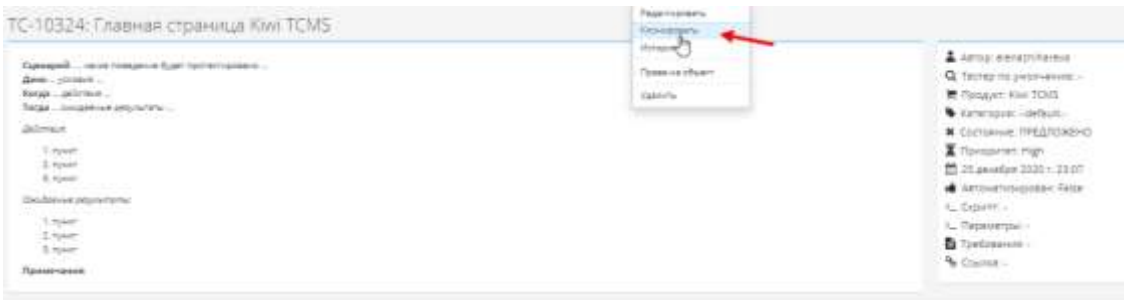

Рисунок 6 – Клонирование тест-кейса

Лабораторная №4 Создание, редактирование и выполнение тестовых прогонов

Цель работы: понять, что такое тестовый прогон и научиться создавать его в системе Kiwi TCMS.

1. Теоретические сведения

Как правило, тестовые прогоны представляют собой набор из нескольких наборов тестов с необходимой информацией, например, о том, кто обрабатывает задействованные тест-кейсы в любой конкретный момент. Именно здесь происходит фактическое тестирование, и после выполнения тестовых прогонов можно определить соответствующие результаты.

Тестовые прогоны представляют собой набор из нескольких наборов тестов с необходимой информацией, например, о том, кто обрабатывает задействованные тестовые случаи в любой конкретный момент. Именно здесь происходит фактическое тестирование, и после выполнения тестовых прогонов можно определить соответствующие результаты.

Простые тестовые прогоны содержат тестовые случаи, которые проверяют работу каждой единицы системы (определенных классов или небольших групп связанных функций) независимо от ее среды. Такие тестовые случаи не должны иметь внешних зависимостей. Они сосредоточены на одной, отделенной от всего, функциональности.

2. Пример создания тестового прогона

Тестовые прогоны создаются для определенного плана тестирования. В тестовый прогон могут быть добавлены только подтвержденные тестовые случаи (они же готовы к выполнению). Тестовый запуск можно назначить любому пользователю в Kiwi TCMS. Чтобы создать тестовый прогон необходимо:

1. Открыть план тестирования.

2. Выбрать тест-кейсы для выполнения (только подтвержденные). Если тест-кейс не подтвержден, то необходимо зайти в редактирование тест-кейса и сменить статус на Подтвержден.

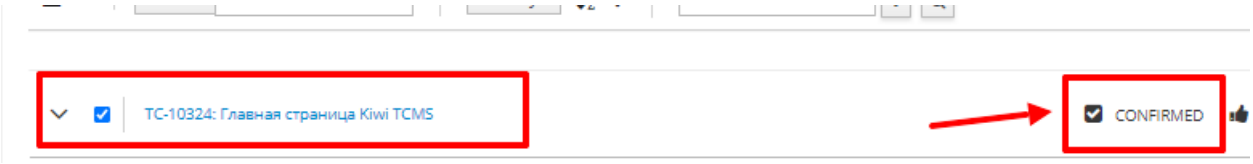

Рисунок 1 – Выбор тест-кейса в плане тестирования

3. В меню массового обновления нажать на Новый тестовый прогон (рис.2).

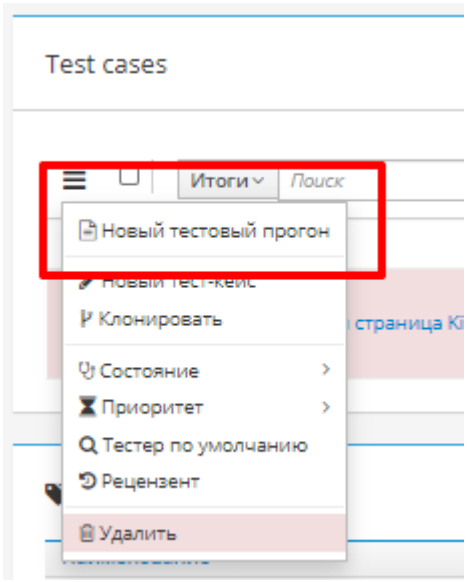

Рисунок 2 – Новый тестовый прогон

На странице создания тестового прогона необходимо отредактировать поля: Сборка, Менеджер и Тестер по умолчанию, дополнительно – Примечание (не обязательно)(рис.3).

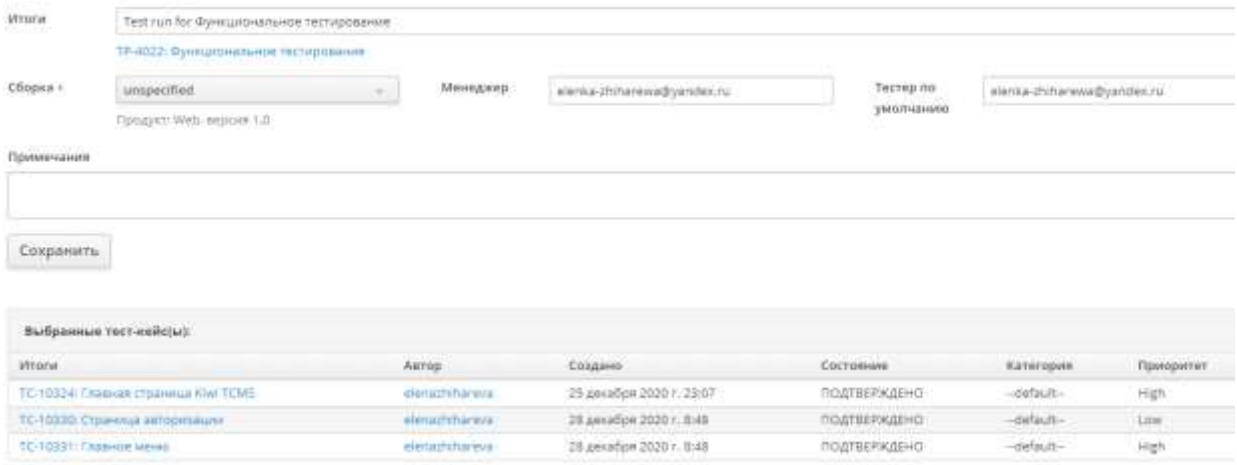

Рисунок 3 – Заполнение полей в тестовом прогоне

Подтвержденные тестовые случаи будут показаны внизу страницы выше. Далее, нужно нажать на кнопку Сохранить, иначе тестовый прогон не будет создан. Kiwi TCMS уведомляет тестировщика по умолчанию по электронной почте о том, что им назначен новый тестовый прогон.

Чтобы добавить тестовый случай к существующему прогону необходимо открыть Тестовый прогон и с помощью виджета для быстрого поиска найти тест-кейс и, нажав на значок плюсика возле поисковой строки, добавить (рис.4).

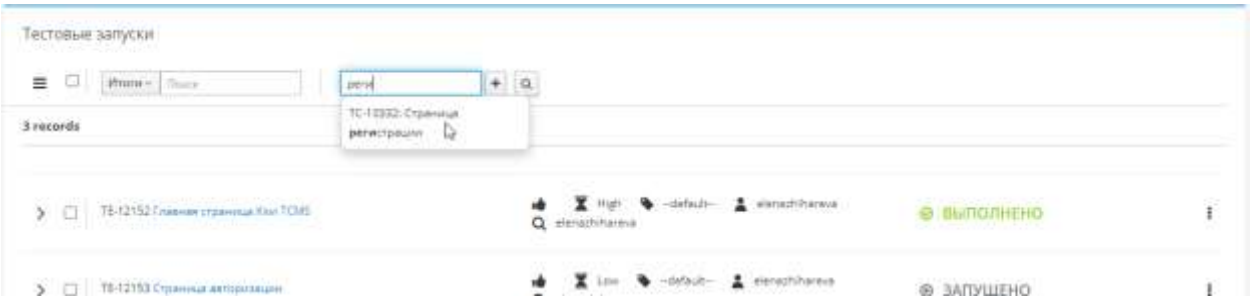

Рисунок 4 – Добавление тест-кейса в тестовый прогон

Подтвержденные тестовые примеры будут добавлены в тестовый запуск.

Состояние тестового прогона можно изменить с «Выполняется» на «Завершено», даже если все тестовые случаи не были выполнены.

Чтобы изменить статус тестового запуска:

1. Открыть тестовый прогон.

2. Переключить кнопку состояния в положение ВЫКЛ, после чего прогон автоматически будет завершен (рис.5).

3. Чтобы снова активировать тестовый прогон, необходимо снова нажать на переключатель в строке Статус.

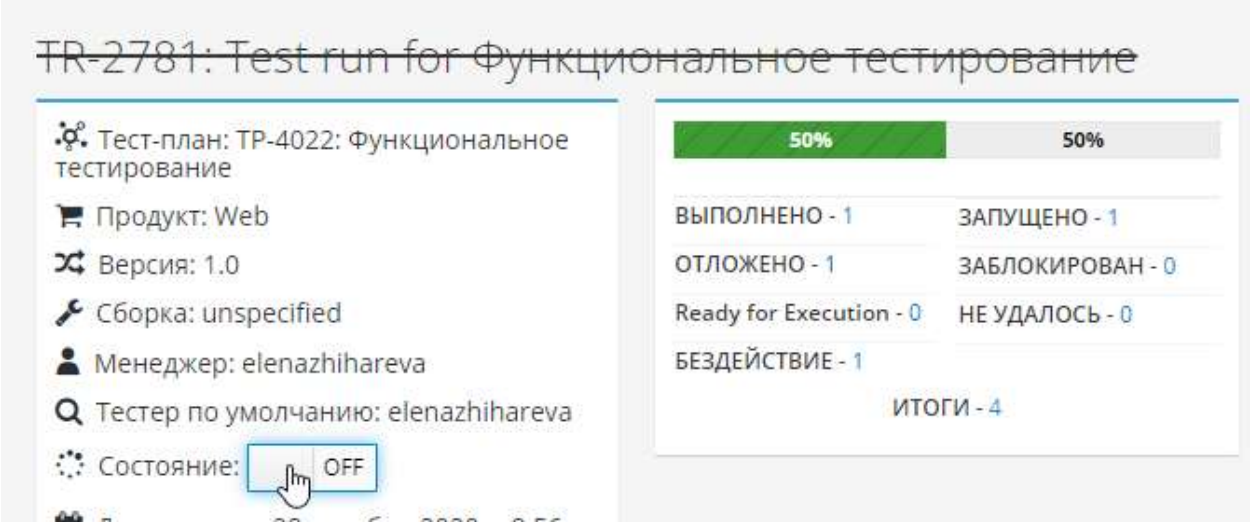

Рисунок 5 – Завершение тестового прогона

Тестовые прогоны можно выполнить в любое время. Тестировщики могут выполнять любой из тестовых случаев в рамках тестового прогона, независимо от порядка их появления. Тестировщики также могут добавлять комментарии для аннотирования выполнения тестов.

Чтобы установить статус тест-кейса в тестовом прогоне, необходимо выполнить шаги для воспроизведения и, когда тест-кейс будет пройден, можно добавить комментарий и выбрать из списка состояний статус(рис.6).

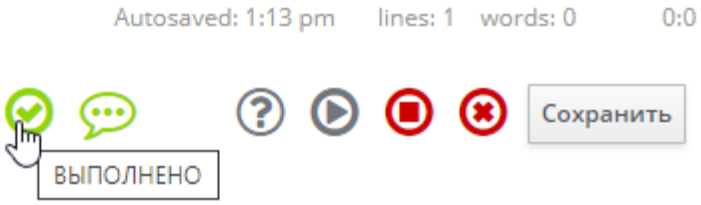

Рисунок 6 – Статус тест-кейса в тестовом прогоне

Функция редактирования изменяет поля в тестовом прогоне. Чтобы отредактировать тестовый прогон, нужно открыть его и нажать на шестерёнки в правом верхнем углу (алгоритм такой же как и редактирование тест-плана и тест-кейса), выбрать строку Редактировать. При необходимости отредактировать поля. Нажать на кнопку Сохранить.

Лабораторная №5 Автоматизированное тестирование веб-сайта

Цель работы: понять, что такое автоматизированное тестирование и ознакомиться с примером разработанного фреймворка для автоматизации веб-приложения.

1. Теоретические сведения

Как известно, тест-кейсы можно выполнять ручным способом, то есть когда это делает непосредственно сам человек, а также автоматизированным.

Автоматизированное тестирование программного обеспечения — часть процесса тестирования на этапе контроля качества в процессе разработки про граммного обеспечения. Оно использует программные средства для выполнения тестов и проверки результатов выполнения, что помогает сократить время тестирования и упростить его процесс.

Цель - сократить время тестирования и упростить его процесс.

Наилучший вариант использования автоматических тестов:

— регрессионное тестирование;

— юнит-тестирование

— тестирование API

— Тестирование GUI

Возможности:

- больше тестовое покрытие;
- сокращение времени и затрат;
- становятся доступными некоторые виды тестирования.

Для того, чтобы связать разработанный фреймворк с веб-сайтом необходим веб-драйвер.

Selenium WebDriver, или просто WebDriver – это драйвер браузера, то есть не имеющая пользовательского интерфейса программная библиотека, которая позволяет различным другим программам взаимодействовать с

браузером, управлять его поведением, получать от браузера какие-то данные и заставлять браузер выполнять какие-то команды.

2. Пример фреймворка для автоматизации тестирования

В данном пункте представлен пример работы настроенного тестового сценария, который проверяет страницу поиска на сайте «https://cchgeu.ru/». На следующих рисунках представлен последовательный процесс создания тестового сценария, его запуска, а также формирования результата с помощью системы генерации отчетов Allure 2.0.

В начале, создается новый Java проект, в первую очередь в файле pom.xml прописываются зависимости, с которыми будет проводиться работа в проекте(рис.7).

Рисунок 7 – Файл pom.xml

Далее, будет система будет открывать главную страницу, вводить конкретное значение в поле поиска и проверять количество найденных элементов на странице.

Поиск элементов происходит с помощью локаторов и некоторых методов.

By.xpath("//button[@class='cap']")

By.xpath("//input[@class='active']");

By.xpath("//i[@class='fas fa-search'] ");

Подсчет элементов происходит с помощью метода VgtuListingPage, который изображен на рис.8.

Рисунок 8 - метод RealtListingPage.

После того как конкретные классы и методы прописаны для определённого веб-приложения, можно запускать проверку.

Проверка осуществляется через Maven. В данном примере запускаются только первые четыре этапа жизни: clean, validate, compile, test (рис.9).

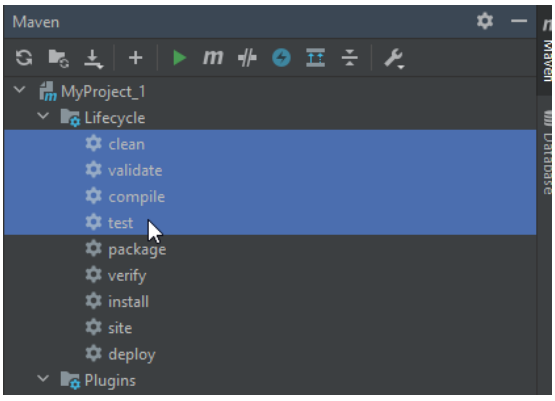

Рисунок 9 – Запуск системы

После запуска системы в браузере Google Chrome открывается сайт образовательного учреждения, затем система сама открывает необходимые элементы, которые были прописаны заранее, запускает тест и закрывает браузер.

В IntelliJ IDEA можно убедиться в том, что система сработала верно и тест пройден(рис.10).

```
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 12.288 s - in TestSuite
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  16.156 s
[INFO] Finished at: 2022-01-31T21:03:09+03:00
[INFO] ------------------------------------------------------------------------

Process finished with exit code 0
```

Рисунок 10 – Информация о запуске автотеста

Затем необходимо просмотреть отчет по тестированию через дополнительный плагин Allure, подключение которого было прописано в pom.xml. Данный плагин находится во вкладке Maven(рис.11).

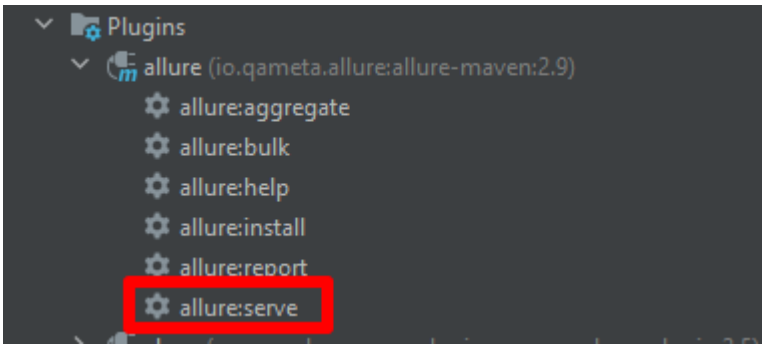

Рисунок 11 – Плагин Allure

После запуска allure:serve появляется ссылка на Allure, где можно посмотреть отчет(рис.12).

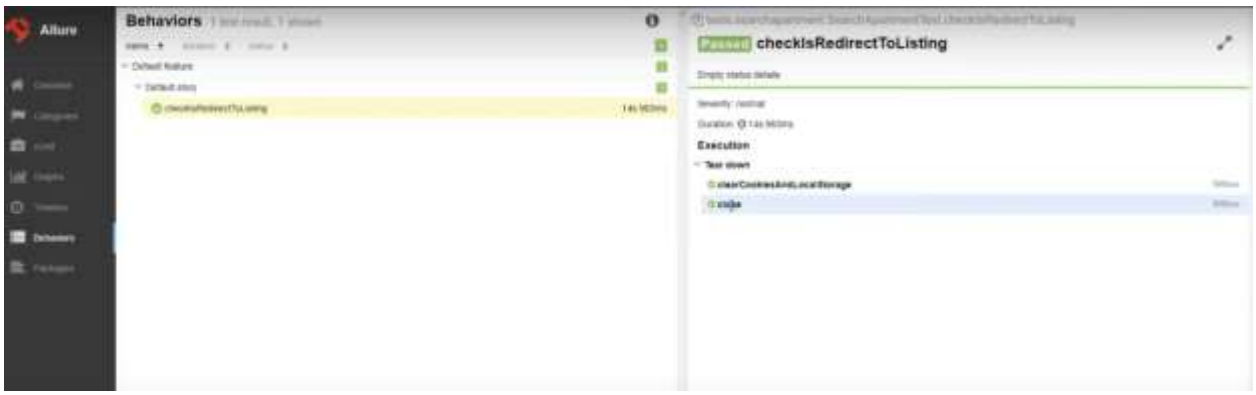

Рисунок 12 – Отчет о тестировании Allure

По данному отчету легко определить, что статус прохождения всего теста «Passed», то есть успешно пройден. В отчете также видно, что все шаги отмечены зелеными значками, то есть все шаги пройдены успешно. В случае падений теста, шаг, на котором тест завершился, и невыполненные шаги будут отмечены особыми цветами, а также будет другой статус у всего теста, например «Failed» (Провален) или «Broken» (Сломан).

Таким, образом, разработав только классы веб-страниц с нужными элементами, и написав тестовый сценарий, с помощью разработанного фреймворка получилось запустить тест сложного корпоративного портала, такого как «https://cchgeu.ru/».

Контрольное задание

Выбрать любую систему и протестировать ее, а конкретно:

1. Зарегистрироваться в системе
2. Составить тестовый план, внести в систему;
3. Придумать к выбранной системе тест-кейсы, добавить их в систему и привязать к тестовому плану (должно быть не менее 5);
4. Создать тестовый прогон;
5. Выполнить тест-кейсы;
6. Отметить статус прохождения тест-кейса в тестовом прогоне;
7. Завершить тестовый прогон.

Контрольные вопросы:

- Что такое тестирование(определение)?
- Принципы тестирования?
- Что такое тест-план?
- Что такое тестовый случай(тест-кейс)?
- Этапы составления тест-кейса?
- Зачем используют тестовый прогон?
- Что такое автоматизированное тестирование?
- Цели использования автоматизированного тестирования?