

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

УДК 004.05

Жданко
Татьяна Владимировна

**МЕТОДЫ И СРЕДСТВА АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ ВЕБ-
ПРИЛОЖЕНИЙ**

АВТОРЕФЕРАТ

на соискание степени магистра информатики и вычислительной техники
по специальности 1-40 81 01 – Информатика и технологии разработки
программного обеспечения

Научный руководитель
Матвейчук Наталья Михайловна
кандидат физико-математических
наук

Минск 2018

ВВЕДЕНИЕ

Большинство программных продуктов, выпускаемых сегодня, являются веб-ориентированными приложениями, рассчитанными на работу в интернет-браузере. Разработка веб-приложения подразумевает под собой решение множества задач из совершенно разных областей, таких как: работа с большими объемами данных, вычислительные системы, сетевая безопасность, дизайн и прочее. Тестирование является важной и неотъемлемой частью разработки программного обеспечения. Возрастающее количество исходного кода, а также возрастающее число разработчиков, вызывает все большее количество ошибок в разрабатываемом приложении. Одной из основных функций тестирования является обнаружение этих ошибок как можно раньше. Скорость, с которой можно идентифицировать эти ошибки, полностью зависит от подхода тестирования, количества тестируемых, их опыта, а также инструментов, используемых в процессе тестирования. Инструменты автоматизированного тестирования позволяют проводить тестирование проще и более эффективно.

Под автоматизированным тестированием подразумевается процесс тестирования ПО, при котором основные функции и шаги теста, такие как запуск, инициализация, выполнение, анализ и выдача результата, производятся автоматически с помощью инструментов для автоматизированного тестирования. Автоматизированное тестирование обладает множеством достоинств, связанных главным образом с высокой скоростью выполнения тестов и возможностью выполнять однотипные тесты снова и снова.

Практика показала, что наиболее эффективным способом тестирования большого количества программ в ограниченные сроки является создание тестовых фреймворков, предоставляющих стандартный набор операций, позволяющих без адаптации к программе тестировать приложение.

Тестовый фреймворк представляет собой набор классов и методов, реализованных таким образом, что при подключения такого проекта в качестве библиотеки к другому проекту появляется возможность использовать уже готовую логику, не прибегая к реализации необходимой функциональности с нуля. Это дает существенный выигрыш во времени, а следовательно – и в финансовом плане этот метод окажется менее затратным. Но время – не единственный плюс использования готовых библиотек для тестирования. Такой подход позволяет избежать всевозможной путаницы с оформлением и написанием тестов, ведь фреймворк представляет собой основу – «каркас», на котором будет сконструирован весь тестовый подход.

Предоставляя разработчику или ручному тестировщику фреймворк, вы предоставляете архитектуру написания тестов и их синтаксис. Это значительно упрощает работу и позволяет повысить качество тестового продукта, позволяет быстро анализировать полученные в результате тестирования данные и находить места, в которых были выявлены ошибки.

Стоит также отметить, что фреймворк для тестирования веб-приложений зачастую включает в себя, кроме непосредственно самих тестовых методов, и различные другие методы как, например, сбор статистики, формирование отчетов, анализ полученных результатов, построение графиков, диаграмм и зависимостей, предоставление возможных путей устранения ошибок.

Библиотека БГУИР

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы исследования

Эффективная автоматизация тестирования является одной из наиболее важных составляющих быстрой и непрерывной поставки высококачественного программного обеспечения. Всё больше компаний заинтересовано в снижении времени вывода продукта на рынок, поэтому имеется устойчивая тенденция к ограничению времени тестирования – ведь тесты, особенно регрессионные, часто занимают много времени.

Использование фреймворков для автоматизированного тестирования приложений является эффективным способом для экономии временных и материальных ресурсов на контроль за качеством приложения, поэтому в настоящее время практически ни одно из веб-приложений не обходится без использования специальной тестовой библиотеки.

Разработанный фреймворк позволит проводить тестирование веб-приложений с помощью инструментов автоматизированного тестирования. Использование автоматизированных тестов даст возможность: быстро предоставлять разработчикам отчет о состоянии продукта; получить потенциально бесконечное число прогонов тестов; сократить время на проведение тестирования программного обеспечения; минимизировать влияние человеческого фактора на процесс тестирования.

Таким образом, в современных условиях актуальными являются применение инструментов автоматизированного тестирования, и разработка фреймворка, позволяющего осуществить автоматизированное тестирование различных веб-приложений.

Степень разработанности проблемы

Тестирование представляет собой комплексный процесс, который занимает значительную часть времени разработки программного обеспечения. Но в итоге, мы не можем быть уверены, что разработанный продукт достаточно протестирован, так как даже простое приложение содержит большое количество различных комбинаций значений и условий, которые делают его полное тестирование практически невозможным.

Тестирование не может гарантировать, что нет никаких дефектов в приложении, а только может показать наличие различных ошибок в программном обеспечении.

С ростом количества выполненных тестов растет количество обнаруженных ошибок. В тоже время растет количество ресурсов, необходимых для тестирования. Одним из путей увеличения количества выполняемых тестов почти с тем же количеством ресурсов, является использование инструментов и технологий автоматизированного тестирования.

Процесс автоматизации тестирования не заканчивается после получения результатов, наоборот, полученные результаты – это основа для нового цикла тестирования. В результате такого многократного повторения определенного набора шагов удастся обнаружить множество ошибок, в том числе и имеющих непостоянное воспроизведение.

Цель и задачи исследования

Целью данной работы является оптимизация процесса тестирования программного обеспечения, путем создания фреймворка, позволяющего осуществлять автоматизированное тестирование различных веб-приложений.

Поставленная цель работы определяет следующие основные задачи:

- 1 Изучить порядок автоматизированного тестирования веб-приложений.
- 2 На основании анализа методов и средств автоматизации тестирования веб-приложений, сформировать подход для разработки фреймворка
- 3 Определить требования к разрабатываемому фреймворку. Разработать фреймворк, позволяющий осуществить автоматизированное тестирование различных веб-приложений.

Область исследования

Содержание диссертационной работы соответствует образовательному стандарту высшего образования второй ступени (магистратуры) специальности 1-40 81 01 «Информатика и технологии разработки программного обеспечения».

Теоретическая и методологическая основа исследования

В основу диссертации легли исследования зарубежных и отечественных ученых в области контроля качества в рамках разработки программного обеспечения, принципов автоматизации тестирования веб-приложений.

При решении поставленных задач использованы методы и средства автоматизации тестирования веб-приложений, методы объектно-

ориентированного программирования.

В качестве инструментальных средств использовались объектно-ориентированный язык программирования *Java*, фреймворки *Cucumber* и *JUnit*.

Информационная база исследования сформирована на основе данных, опубликованных в журналах технического направления, а также архивов статистических данных компании EPAMSystems.

Научная новизна

Научная новизна и значимость диссертации заключается в разработке фреймворка, позволяющего осуществлять автоматизированное тестирование различных веб-приложений

Теоретическая значимость диссертации заключается в описании процесса создания фреймворка для автоматизированного тестирования веб-приложений.

Практическая значимость результатов исследования заключается в разработанном фреймворке, который позволит сократить время на проведение тестирования программного обеспечения; быстро предоставлять разработчикам отчет о состоянии продукта; получить потенциально бесконечное число прогонов тестов; минимизировать влияние человеческого фактора на процесс тестирования.

Основные положения, выносимые на защиту

1 Обоснование рациональности автоматизации тестирования программных продуктов. Методы и средства автоматизации тестирования веб-приложений.

2 Алгоритм написания тестовых сценариев с использованием инструмента программной поддержки BDD: *Cucumber* фреймворк и язык *Gherkin*. Процесс создания фреймворка для автоматизации тестирования веб-приложений.

3 Фреймворк для тестирования веб-приложений, позволяющий автоматизировать процесс тестирования и повысить эффективность контроля качества программного продукта.

Апробация диссертации и информация об использовании ее результатов

Основные теоретические результаты и законченные этапы

диссертационной работы отражены в материалах 53-й научной конференции аспирантов, магистрантов и студентов БГУИР, URL: <https://libeldoc.bsuir.by/handle/123456789/13147>, а также в научном журнале «Студенческий форум» выпуск №21 в разделе «Технический науки», URL: <https://nauchforum.ru/journal/stud/21/29539>.

Публикации

Изложенные в диссертации основные положения и выводы опубликованы в 2 печатных работах, представленные в виде двух докладов:

- 1 Автоматизация тестирования с применением BDD подхода.
- 2 Подходы к автоматизации тестирования веб-приложений.

Общий объем публикаций по теме диссертации составляет 9 страниц.

Структура и объем работы

Диссертация состоит из введения, общей характеристики работы, трех глав с краткими выводами по каждой главе, заключения, библиографического списка и приложений.

Во введении рассмотрены современное состояние проблемы организации процесса тестирования веб-приложений, автоматизация тестирования как одна из наиболее важных составляющих быстрой и непрерывной поставки высококачественного программного обеспечения. Определены основные цели и задачи исследований.

В первой главе рассмотрено понятие контроля качества в рамках разработки программного обеспечения. Факторы качества. Методы контроля качества программного обеспечения. Рассмотрены основные причины возникновения ошибок в разрабатываемом приложении, оценка стоимости ошибок. Раскрыты принципы автоматизации тестирования веб-приложений. Проведен сравнительный анализ ручного и автоматизированного подхода к тестированию. Выявлены преимущества и недостатки автоматизации тестирования. Приведено обоснование рациональности автоматизации тестирования программных продуктов.

Во второй главе были рассмотрены зависимость тестирования от методологии разработки программного продукта. Проведен анализ методологий и средств автоматизированного тестирования веб-приложений. Приведено обоснование выбора методологии BDD для тестирования веб-приложений. Рассмотрен пример написания тестовых сценариев с использованием инструмента программной поддержки BDD – Cucumber.

Описан основной процесс создания фреймворка для автоматизированного тестирования веб-приложений. Также была разработана функциональная модель процесса тестирования веб-приложения.

В данной главе обоснована рациональность создания фреймворка для тестирования веб-приложений. Приведен пример использования, создаваемого фреймворка.

Третья глава посвящена постановке задачи, а также рассмотрению методов ее решения. Были сформулированы требования к разрабатываемому фреймворку. Чтобы сделать процесс разработки понятным, были построены диаграммы последовательности системы, диаграмма вариантов использования, диаграмма состояния объекта и обобщенный алгоритм работы приложения. В заключение, было подробно расписано руководство по развертыванию и запуску тестов.

В приложении представлены публикации и графический материал в виде презентации.

Общий объем диссертационной работы составляет 78 страниц. Из них 58 страниц основного текста, 39 иллюстраций, библиографический список из 30 наименований, список собственных публикаций соискателя из 2 наименований, 3 приложения.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении рассмотрены современное состояние проблемы организации процесса тестирования веб-приложений, автоматизация тестирования как одна из наиболее важных составляющих быстрой и непрерывной поставки высококачественного программного обеспечения. Определены основные цели и задачи исследований.

В общей характеристике работы показана актуальность проводимых исследований, степень разработанности проблемы, сформулированы цель и задачи диссертации, обозначена область исследований, научная (теоретическая и практическая) значимость исследований, а также апробация работы.

В первой главе рассмотрены контроль качества в рамках разработки программного обеспечения понятия, а также автоматизация тестирования как наиболее эффективный способ контроля качества программного продукта.

Качество программного средства – совокупность свойств программного средства, которые обуславливают его пригодность удовлетворять заданные или подразумеваемые потребности в соответствии с его назначением [ГОСТ 28806–90 «Качество программных средств. Термины и определения»]. Показатели качества программного продукта представлены на рисунке 1.



Рисунок 1 – Показатели качества программного обеспечения

Наиболее эффективным способом контроля качества программного продукта является осуществление процесса тестирования разрабатываемой системы по протяжении всего жизненного цикла разработки.

Ряд компаний провел исследование оценки стоимости ошибок, возникающих на разных этапах создания программ. Каждая фирма действовала независимо, тем не менее результаты получены примерно одинаковые: если стоимость усилий, необходимых для обнаружения и устранения ошибок на стадии написания кода, принять за единицу, то

стоимость выявления и устранения ошибки на стадии выработки требований будет в 5-10 раз меньше, а стоимость обнаружения и устранения ошибки на стадии сопровождения – в 20 раз больше (рисунок 2).



Рисунок 2 - Оценка стоимости ошибок на разных этапах создания ПО

В зависимости от того, где и когда при работе над проектом разработки программного приложения был обнаружен дефект, цена его может разниться в 50—100 раз. Причина состоит в том, что для его исправления придется затратить средства на некоторые (или все) нижеперечисленные действия:

- повторная спецификация;
- повторное проектирование;
- повторное кодирование;
- повторное тестирование;
- внесение исправлений;
- создание документации.

Таким образом, чем раньше будут обнаружены ошибки, тем меньше они будут стоить.

Одним из путей увеличения количества выполняемых тестов, на разных этапах разработки, почти с тем же количеством ресурсов, является использование инструментов и технологий автоматизированного тестирования.

Автоматизация тестирования – это сложный процесс, представляющий собой непрерывный цикл контроля за качеством программного продукта с помощью специальных программных средств. Этот процесс не заканчивается после получения результатов, наоборот, полученные результаты – это основа для нового цикла тестирования. В процессе такого многократного повторения определенного набора шагов удастся обнаружить множество ошибок, в том числе и имеющих непостоянное воспроизведение. Процесс автоматизации тестирования представлен на рисунке 3.

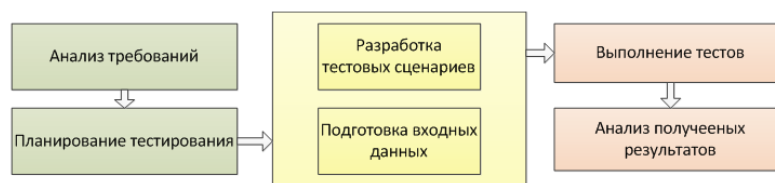


Рисунок 3 – Процесс автоматизации тестирования программного продукта

Основными преимуществами автоматизации тестирования являются: повторяемость, быстрое выполнение, меньшие затраты на поддержку, автоматически рассылаемые и сохраняемые отчеты, выполнение без вмешательства.

Для того, чтобы избежать неоправданного использования автоматизированных тестов, стоит обращать внимание на следующие моменты, которые влекут за собой необходимость постоянно обновлять тесты, а, следовательно, исключают возможность использования автоматизации на проекте: плохая тестируемость кода; большое количества связей между модулями; входные данные не вынесены в переменные значения. Диаграмма, показывающая соотношение использования рабочего времени для тестирования «плохо тестируемых» программ представлена на рисунке 4.

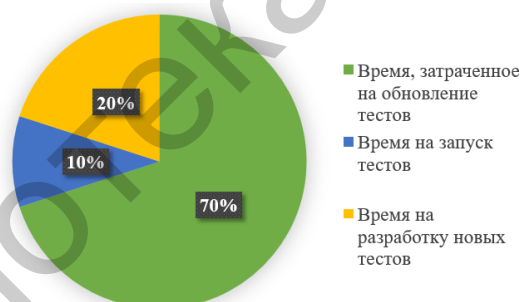


Рисунок 4 – Соотношение использования рабочего времени для тестирования «плохо тестируемых» программ

При условии, что программный продукт разработан так, что может легко поддерживать написание тестов, время на обновление последних существенно снижается, что позволяет использовать время для усовершенствования рабочих процессов, что является залогом эффективного тестирования (рисунок 5).

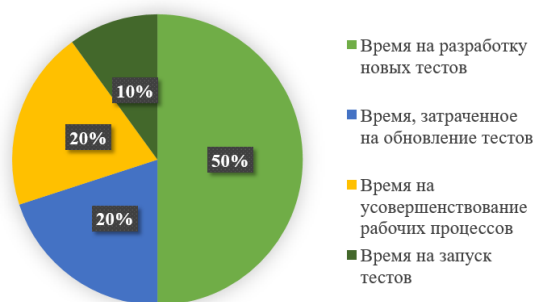


Рисунок 5 – Диаграмма, показывающая соотношение использования рабочего времени для тестирования «хорошо тестируемых» программ

Важно также учитывать архитектуру разрабатываемого программного продукта – если система состоит из однотипных модулей, автоматизация тестирования в таком случае будет более, чем оправдана.

Во второй главе рассмотрена зависимость тестирования от методологии разработки программного продукта. Рисунок 6 показывает какой процент составляют успешные проекты от общего числа проектов, разрабатывающихся с использованием той или иной методологии.

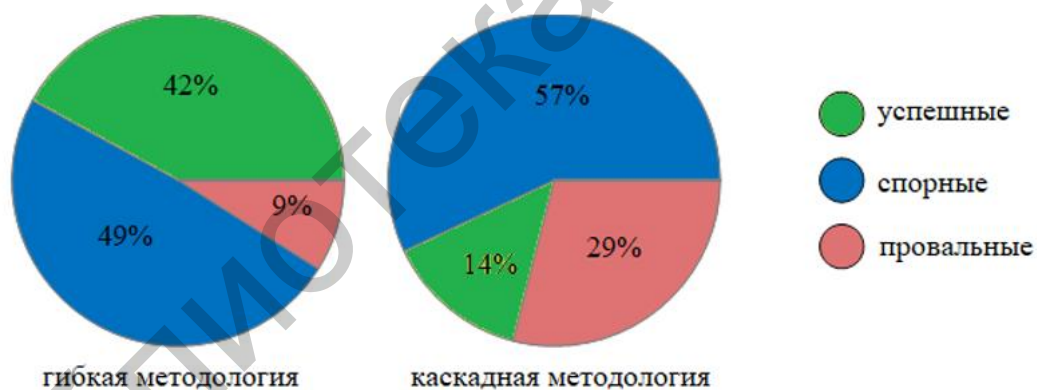


Рисунок 6 – Диаграммы, отображающие успешность проектов для гибкой и каскадной методологий разработки

Был проведен анализ существующих методологий и средств автоматизированного тестирования веб-приложений.

Тестирование, управляемое данными (Data-Driven Testing) представляет собой такой методологии к тестированию, при котором тестовые данные хранятся отдельно от тест-кейсов, например, в файле или базе данных. Такое разделение делает тесты логически более простыми. Методология DDT представлена на рисунке 7.



Рисунок 7 – Методология Data-Driven Testing

Keyword Driven Testing (Тесты, управляемые ключевыми словами) – это подход, в котором используются ключевые слова, описывающие набор действий, необходимых для выполнения определенного шага тестового сценария. Т.е. каждый шаг теста, такой как открытие или закрытие браузера, щелчок мышью, нажатие клавиши и т.д., описывается ключевым словом, таким как «открыть» (openbrowser), «нажать» (click).

Методика разработки через тестирование (TDD) заключается в организации автоматического тестирования разрабатываемых приложений путем написания модульных, интеграционных и функциональных тестов, определяющих требования к коду непосредственно перед написанием этого самого кода. Сначала пишется тест, который проверяет корректность работы еще ненаписанного программного кода. После этого разработчик пишет код, который выполняет действия, требуемые для прохождения теста. После того, как тест успешно пройден, по необходимости осуществляется доработка написанного кода, причём под контролем прохождения тестов.

Приведено обоснование выбора методологии BDD для тестирования веб-приложений.

BDD представляет собой расширение подхода TDD к разработке и тестированию, при котором особое внимание уделяется поведению системы/модуля в терминах бизнеса (заказчика). Если сравнивать BDD и TDD подходы, то для последнего характерны тесты, покрывающие непосредственно код программы, что не совсем сочетается с необходимостью проверять устойчивость и корректность работы веб-приложения при работе с пользователем. Веб-приложению необходимо симулировать работу реального пользователя, т.е. выделить максимально возможное количество ресурсов на тестирование именно пользовательского интерфейса, безопасности и надежности.

При тестировании программы с использованием TDD подхода, часть ресурсов будет задействовано для покрытия тестами той функциональности,

которая никогда не будет использована пользователем. А если учитывать большие масштабы современных проектов, то можно легко понять, что количество задействованных в таком тестировании ресурсов будет весьма велико (рисунок 8).

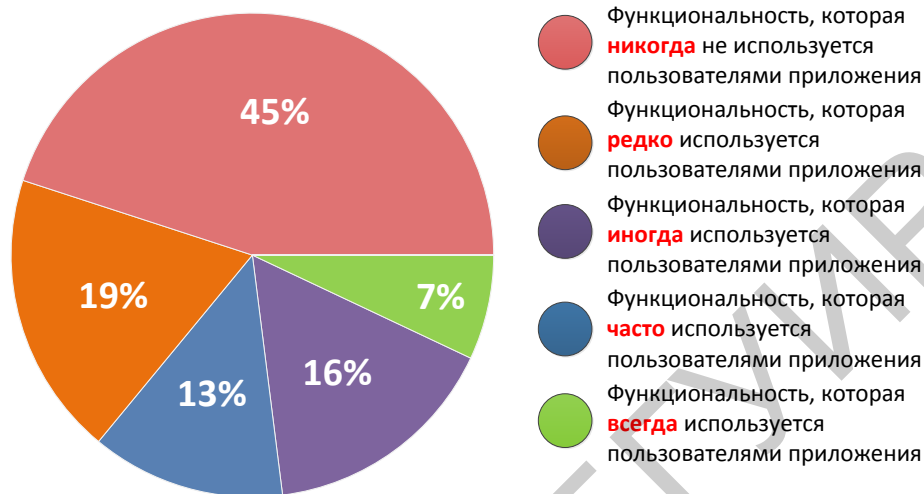


Рисунок 8 – Диаграмма использования функциональности веб-приложений пользователями

С помощью BDD тестировать систему может не только сам программист, пишущий код, но и любой другой человек, не разбирающийся в деталях реализации, но хорошо знающий систему с точки зрения пользователя. Для новичков BDD-скрипты – самый простой и естественный способ ознакомиться с документацией проекта.

В данной главе рассмотрено написание тестовых сценариев с использованием инструмента программной поддержки BDD: Cucumber фреймворк и язык Gherkin. Тесты на Cucumber состоят из двух файлов: тестовый сценарий и непосредственно его реализация на выбранном языке программирования. Это позволяет отделить сценарий от реализации, что дает возможность писать тестовые сценарии на читабельном – абсолютно понятном любому человеку языке (рисунок 9).

```
21 Scenario: Verify that the specified text exists on the page
22
23 Given I am on the main application page
24 When I click on a link with the URL '/page'
25 When I switch to a new window
26 Then the text 'text' exists
27
28
```

Рисунок 9 – Пример тестового сценария с использованием фреймворка Cucumber и языка написания тестов Gherkin

Также здесь описан основной процесс создания фреймворка для автоматизированного тестирования веб-приложений. Была разработана функциональная модель процесса тестирования веб-приложения.

Изложено обоснование рациональности создания фреймворка для тестирования веб-приложений. Пример использования, создаваемого фреймворка.

Третья глава посвящена постановке задачи, а также рассмотрению методов ее решения. Были сформулированы следующие задачи:

- подробно изучить процессы контроля качества в рамках разработки и эксплуатации программного продукта;
- разработать стратегию создания фреймворка;
- подготовить необходимые для разработки ресурсы;
- спроектировать и разработать программный продукт для тестирования веб-приложений;
- создать необходимую сопроводительную документацию и оформление созданного фреймворка;
- протестировать веб-приложение с помощью разработанной программы с целью продемонстрировать результат тестирования веб-приложения по средствам разработанного фреймворка.

Были сформированы требования к разрабатываемому фреймворку:

- 1 Использование операционных систем семейств Windows.
- 2 Для разработки фреймворка использовать язык программирования Java.
- 3 Паттерн для реализации функциональности – одиночка (singleton).
- 4 Использовать фреймворк Cucumber для разработки сценариев.
- 5 Автоматическое обновление модулей фреймворка с помощью сборщика проектов.

Чтобы сделать процесс разработки понятным, были построены диаграммы последовательности системы, диаграмма вариантов использования, диаграмма состояния объекта и обобщенный алгоритм работы приложения. Приведено обоснование оригинальных решений по использованию технических и программных средств.

В заключение, было подробно расписано руководство по развертыванию и запуску тестов.

ЗАКЛЮЧЕНИЕ

Основные научные результаты диссертации

По итогам проведенной работы можно сделать выводы:

1 Рассмотрены понятия контроля качества в рамках разработки программного обеспечения, факторы качества, методы контроля качества. Рассмотрено тестирование, как наиболее эффективный метод контроля качества программных продуктов. Проведен сравнительный анализ ручного и автоматизированного подхода к тестированию. Выявлены преимущества и недостатки автоматизации тестирования. Дано обоснование рациональности автоматизации тестирования программных продуктов. Изучен порядок автоматизированного тестирования веб-приложений.

2 Рассмотрены зависимость тестирования от методологии разработки программного продукта. Проведен анализ методов и средств автоматизированного тестирования веб-приложений, на основании которого, был сформирован подход для разработки фреймворка. Приведено обоснование выбора методологии BDD для тестирования веб-приложений.

3 Определены требования к разрабатываемой системе. Разработан фреймворк, на основе BDD методологии с применением инструмента Cucumber, позволяющий осуществлять автоматизированное тестирование различных веб-приложений.

СПИСОК ПУБЛИКАЦИЙ СОИСКАТЕЛЯ

1 – А. Жданко, Т. В. Автоматизация тестирования с применением BDD подхода / Т. В. Жданко // Компьютерные системы и сети: материалы 53-й научной конференции аспирантов, магистрантов и студентов (Минск, 2 – 6 мая 2017 г.). – Минск: БГУИР, 2017. – С. 207 – 208.

2 – А. Жданко, Т.В. Подходы к автоматизации тестирования веб-приложений // Студенческий форум: электрон. научн. журн. 2017. № 21(21) URL: <https://nauchforum.ru/journal/stud/21/29539>

РЭЗЮМЭ

Жданко Таццяна Уладзіміраўна

Метады і сродкі аўтаматызацыі тэставання вэб-прыкладанняў

Ключавыя словы: тэставанне, аўтаматызаванае тэставанне, вэб-прыкладанне, фрэймворк.

Мэтапрацы: аптымізацыя працэсу тэставання праграмнага забеспячэння, шляхам стварэння фрэймворка, які дазваляе ажыццяўляць аўтаматызаванае тэставанне розных вэб-прыкладанняў.

Атрыманыя вынікі і іхнавізна: разгледжаны паняцці кантролю якасці ў рамках распрацоўкі праграмнага забеспячэння; фактары якасці; метады кантролю якасці; тэставанне, як найбольш эфектыўны метада кантролю якасці праграмных прадуктаў. Выяўлены цяжкасці тэставання. Абгрунтавана рацыянальнасць аўтаматызацыі тэставання праграмных прадуктаў. Праведзены параўнальны аналіз ручнога і аўтаматызаванага падыходу да тэставання. Раскрытыя прынцыпы аўтаматызацыі тэставання вэб-прыкладанняў. Выяўлены перавагі і недахопы аўтаматызацыі тэставання. Разгледжана залежнасць тэставання ад метадалогіі распрацоўкі праграмнага прадукту. Праведзены аналіз метадаў і сродкаў аўтаматызаванага тэставання вэб-прыкладанняў. Дадзена абгрунтаванне выбару метадалогіі BDD для тэставання вэб-прыкладанняў. Апісаны асноўны працэс стварэння фрэймворка для аўтаматызаванага тэставання вэб-прыкладанняў. Распрацавана функцыянальная мадэль працэсу тэставання вэб-прыкладанняў. Распрацаваны фрэймворк, які дазваляе ажыццяўляць аўтаматызаванае тэставанне розных вэб-прыкладанняў.

Вобласць ужывання: тэставанне.

РЕЗЮМЕ

Жданко Татьяна Владимировна

Методы и средства автоматизации тестирования веб-приложений

Ключевые слова:тестирование, автоматизированное тестирование, веб-приложение, фреймворк.

Цель работы:оптимизация процесса тестирования программного обеспечения, путем создания фреймворка, позволяющего осуществлять автоматизированное тестирование различных веб-приложений.

Полученные результаты и их новизна:рассмотрены понятия контроля качества в рамках разработки программного обеспечения; факторы качества; методы контроля качества; тестирование, как наиболее эффективный метод контроля качества программных продуктов.Выявлены трудности тестирования. Обоснована рациональность автоматизации тестирования программных продуктов. Проведен сравнительный анализ ручного и автоматизированного подхода к тестированию. Раскрыты принципы автоматизации тестирования веб-приложений. Выявлены преимущества и недостатки автоматизации тестирования. Рассмотрена зависимость тестирования от методологии разработки программного продукта. Проведен анализ методов и средств автоматизированного тестирования веб-приложений. Дано обоснование выбора методологии BDD для тестирования веб-приложений. Описан основной процесс создания фреймворка для автоматизированного тестирования веб-приложений. Разработана функциональная модель процесса тестирования веб-приложения.Разработан фреймворк, который позволяет осуществлять автоматизированное тестирование различных веб-приложений.

Область применения:тестирование.

SUMMARY

ZhdankoTatsianaUladzimirayna

Methods and tools for automating testing of web applications

Keywords:testing, automated testing, web application, framework.

The object of study:optimization of the software testing process,by creating a framework that allows to execute automated testing of various web applications.

The results and novelty:the quality control definitions are considered as part of software development; quality control methods; testing, as the most effective method of software products quality control. The difficulties of testing are revealed. The rationality of software products test automation is justified. The comparative analysis of the manual and automated testing approach is carried out. Test automation principles of web-applications are defined. Automation testing advantages and disadvantages are revealed.The dependence of testing on the methodology of software development is considered.The analysis of methods and tools for automated testing of web applications is carried out. The rationale of choosing the BDD methodology for web applications testing is given. The main process of test automation framework creation for of web applications is described. The functional model of web application testing process has been developed. The framework that allows to execute automated testing of various web applications has been developed.

Sphere of application:testing.