

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

УДК 004.05

Цегельник
Надежда Геннадьевна

**АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ ДЕСКТОПНЫХ ПРИЛОЖЕНИЙ
С ИСПОЛЬЗОВАНИЕМ TITANIUM**

АВТОРЕФЕРАТ

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

Научный руководитель
Поттосина Светлана Анатольевна
кандидат физико-математических
наук, доцент

Минск 2018

ВВЕДЕНИЕ

Усиление конкуренции между производителями программного обеспечения требует повышенного внимания к качеству продукции. Тестирование как инструмент контроля качества стало неотъемлемой частью процесса создания программного обеспечения. Разработка программного обеспечения является достаточно дорогостоящей деятельностью. Нарушения на любой стадии процесса разработки могут привести к нежелательным последствиям. Тестирования программного обеспечения выполняются для обеспечения соответствия программного обеспечения потребностям пользователей.

В настоящее время появилась тенденция автоматизации процесса тестирования программного обеспечения, что, бесспорно, положительно отражается на скорости и качестве тестирования, это позволяет снизить издержки компании и повысить качество продукта.

Практика показывает, что наиболее эффективным способом тестирования приложений в ограниченные сроки является создание тестовых фреймворков, которые представляют собой шаблонные методы и требуют минимальных дополнений для создания рабочей версии автоматических тестов. Это тестирование, которое использует программные средства для выполнения тестов и проверки результатов их выполнения, что помогает сократить время тестирования и упростить его процесс.

Стоит также отметить, что для непрерывного запуска тестов, в проектной деятельности используют различные CI (Continuous Integration) сервера, которые позволяют выполнять сборку новых версий продукта, непрерывно проходить итерации сборки, тестирования, выполняют сбор статистики тестирования, формирование отчетов, анализ полученных результатов, построение графиков, диаграмм и зависимостей.

Таким образом, можно заметить, что использование фреймворков для тестирования приложений является эффективным способом для экономии временных и материальных ресурсов на контроль за качеством приложения. Целью данной работы является оптимизация процесса тестирования путем его автоматизации с использованием технологии Titanium и средств непрерывной интеграции.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы исследования

В связи с быстрым развитием систем разработки программного обеспечения и сетевых технологий, а также усилением конкуренции между производителями программного обеспечения повышенное внимание уделяется качеству продукции. Тестирование, а в частности автоматизированное тестирование, стало наиболее эффективным способом контроля качества программных средств. Следовательно, тема автоматизации десктоп-приложений является актуальной.

На сегодняшний день существует ряд средств для организации автоматизации тестирования визуальной формы программных продуктов. Но в процессе исследования было выявлено недостаточное число технологий и средств для тестирования десктоп-приложений, позволяющих эффективно организовать процесс разработки тестового фреймворка.

Таким образом, автоматизация тестирования десктоп-приложений является актуальной на сегодняшний день.

Степень разработанности проблемы

Отсутствие отработанных методик организации процесса тестирования препятствует эффективному принятию решений в данном аспекте. Анализ практического опыта проведения ИТ-проектов выявил множество недостатков в выборе методологий тестирования, а также организации непрерывной интеграции тестирования. Существуют проблемы совместимости технологий, удобства использования, а также уровня сложности внедрения того или иного решения. Данные вопросы требуют детального и глубокого анализа, что и было сделано в данной работе.

В процессе исследования существующих средств для организации автоматизации визуального тестирования программных продуктов было выявлено, что сфера тестирования десктоп-приложений в данный момент слабо развита. Найден перечень технологий и средств для автоматизации тестирования десктоп-приложений. Однако, у них выявлены существенные недостатки, не позволяющие использовать их в данной работе. При разработке фреймворка была использована технология Titanium, позволяющая эффективно организовать процесс разработки тестового фреймворка.

Цель и задачи исследования

Целью данной работы является оптимизация процесса тестирования путем его автоматизации с использованием технологии Titanium и средств непрерывной интеграции.

Для достижения поставленной цели были сформулированы следующие задачи:

1 Дать обоснование рациональности внедрения автоматизированного тестирования десктоп-приложений, основанное на оценке эффективности. Описать процесс организации автоматизированного тестирования в рамках проектной деятельности.

2 Предложить инструментарий формирования аналитической части системы, основанный на анализе гибких методологий разработки, существующих технологий тестирования и возможностей непрерывной интеграции.

3 Разработать фреймворк для автоматизации тестирования десктоп-приложений на основе технологии Titanium с последующей интеграцией в системе непрерывного запуска тестов.

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 организации проектной деятельности, организации процесса тестирования, автоматизации тестирования. При решении поставленных задач использованы методы Agile для организации проектной деятельности, методы RИO для определения рациональности внедрения автоматизированного тестирования, подход BDD для тестирования, методы объектно-ориентированного программирования.

В качестве инструментальных средств использовались объектно-ориентированный язык программирования C#, фреймворк Titanium и сервер TFS.

Информационная база исследования сформирована на основе данных,

опубликованных в журналах технического направления, а также архивов статистических данных компании EPAM Systems.

Научная новизна

Научная новизна заключается в формировании аналитической части системы. Был произведен анализ гибких методологий разработки, существующих технологий тестирования и возможностей непрерывной интеграции.

Теоретическая значимость диссертации заключается в описании процесса организации автоматизированного тестирования в рамках проектной деятельности.

Практическая значимость результатов исследования заключается в разработанном подходе реализации тестового фреймворка, а также организации его непрерывной интеграции в условиях гибкой методологии разработки.

Основные положения, выносимые на защиту

1 Дано обоснование рациональности внедрения автоматизированного тестирования десктоп-приложений, основанное на оценке эффективности.

2 Предложен инструментарий формирования аналитической части системы, основанный на анализе гибких методологий разработки, существующих технологий тестирования и возможностей непрерывной интеграции.

3 Разработан фреймворк для автоматизации тестирования десктоп-приложений на основе технологии Titanium с последующей интеграцией в системе непрерывного запуска тестов.

Апробация диссертации и информация об использовании ее результатов

Основные теоретические результаты и законченные этапы диссертационной работы отражены в материалах 52-й научно-технической конференции аспирантов, магистрантов и студентов БГУИР, 53-й научно-технической конференции аспирантов, магистрантов и студентов БГУИР, студенческой международной научно-практической конференции «Молодежный научный форум: Технические и математические науки».

Публикации

Изложенные в диссертации основные положения и выводы опубликованы в 3 печатных работах, представленные в виде трех докладов на научных конференциях.

Общий объем публикаций по теме диссертации составляет 10 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 рассмотрено современное состояние проблемы организации процесса автоматизации тестирования десктоп-приложений, определены основные цели и задачи исследований.

В первой главе раскрыты понятия качества программного продукта и тестирования в контексте проверки качества. Выявлены положительные и отрицательные аспекты автоматизации тестирования, а также рассмотрены особенности тестирования десктоп-приложений.

Во второй главе были рассмотрены методологии итерационной разработки программного обеспечения, доказана эффективность использования данного подхода. Также проведен анализ эффективности и рациональности внедрения непрерывной интеграции в рамках разработки с использованием гибких методологий. Был сделан вывод, что этот подход позволяет обеспечить непрерывный процесс проверки программного продукта на соответствие требованиям. Внедрение непрерывной интеграции обеспечит постоянный и эффективный контроль качества даже на ранних стадиях разработки. Также была разработана функциональная модель организации процесса автоматизированного тестирования. В этой главе приступили к поиску необходимого подхода автоматизации тестирования, проанализировали существующие технологии, позволяющие создавать фреймворки для тестирования десктоп-приложений.

Третья глава посвящена постановке задачи, а также рассмотрению методов ее решения. Были сформированы требования к разрабатываемому фреймворку. Чтобы сделать процесс разработки понятным, были построены диаграммы последовательности системы, диаграмма вариантов использования и обобщенный алгоритм работы приложения. Приведен пример организации процесса автоматизированного тестирования, а также создания тестового

фреймворка. 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 72 страницы. Из них 61 страницы основного текста, 15 иллюстраций, библиографический список из 33 наименований, список собственных публикаций соискателя из 3 наименований, 3 приложения.

Библиотека БГУИР

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении рассмотрено современное состояние проблемы организации процесса автоматизации тестирования десктоп-приложений, определены основные цели и задачи исследований, определены основные направления исследований, а также дается обоснование актуальности темы диссертации.

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 тестирование как инструмент проверки качества программного обеспечения.

Качество программного обеспечения – это совокупность характеристик программного обеспечения, относящихся к его способности удовлетворять установленные и предполагаемые потребности. Программный продукт можно считать качественным, если он соответствует предъявляемым к нему требованиям.

Нарушения на любой стадии процесса разработки могут привести к таким нежелательным последствиям, как увеличение сроков разработки, возрастание стоимости программного продукта, а также всевозможным дефектам в работе программы.

Итак, следует начать с первого этапа жизненного цикла разработки – тестирование требований. Проектирование – проверку созданных прототипов. В течении этапа разработки важно провести модульное, интеграционное и системное тестирование. Помимо этого, применяется регрессионное тестирование – проверка новых версий продукта.

В настоящее время появилась тенденция автоматизации процесса тестирования программного обеспечения, что, бесспорно, положительно отражается на скорости и качестве тестирования, это позволяет снизить издержки компании и повысить качество продукта.

Автоматизированные тесты требуют меньше затрат на поддержку. В момент, когда автоматизированные тесты уже написаны, то на их поддержку тратится меньше времени, чем на ручное тестирование того же объема задач. Бесспорно, автоматизированные тесты работают быстрее человека. При этом происходит отличная экономия времени.

Также можно упомянуть, что тесты выполняются без вмешательства тестировщика. Во время выполнения тестов он может заниматься более полезными делами. Весомым преимуществом является то, что тесты могут быть запущены в нерабочее время, например, ночью.

Сравнение временных затрат при ручном и автоматизированном тестировании представлено на рисунке 1.

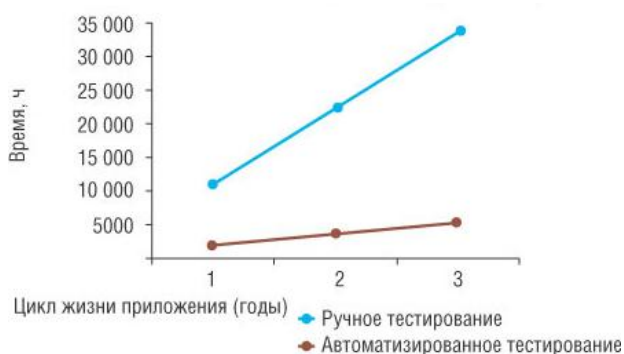


Рисунок 1 – Сравнение временных затрат при ручном и автоматизированном тестировании

Немаловажным аспектом является повторяемость. Плюсом являются также автоматически генерируемые отчеты о результатах тестирования, представляющие наглядную статистику выполнения тестов.

Стоит упомянуть, что не все кейсы могут быть подвержены автоматизации. Бывают ситуации, при которых некоторые тесты лучше оставить на ручное тестирование.

Серьезным вопросом для автоматизации служит необходимость обновления тестов с приходом новых версий продукта. Особенно актуален данный вопрос в контексте регрессионного тестирования. При каждом обновлении функционала программного обеспечения требуется доработка тестов. Поэтому для того, чтобы избежать неэффективного использования автоматизированных тестов, стоит обращать внимание на качество фреймворка тестирования и стратегии тестирования. График сравнения эффективности ручного и автоматизированного тестирования приведен на рисунке 2.

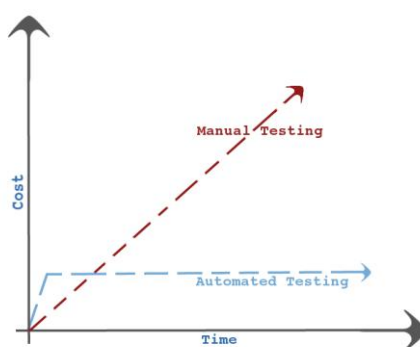


Рисунок 2 – График сравнения эффективности ручного и автоматизированного тестирования

Рассмотрим особенности тестирования десктоп-приложений.

Desktop приложения представляют собой программы, предназначенные для работы на отдельных рабочих станциях или персональных компьютерах, не требующие для своей работы доступа к интернету. Взаимодействие с пользователем осуществляется посредством стандартного интерфейса.

Особенностью данного типа приложений является зависимость от операционной системы, прямой доступ в локальной файловой системе, а также необходимость предварительной установки на каждый компьютер. Такие приложения предназначены для удовлетворения статических потребностей клиентов. В отличие от веб-приложений, к ним можно получить доступ только на том компьютере, на котором они установлены.

С точки зрения тестирования десктоп-приложений стоит обратить внимание на такие особенности:

- для их тестирования и запуска не требуется доступ к интернету;
- прежде, чем приступить к тестированию, необходимо предварительно установить приложение;
- зависимость от платформы;
- стандартный для определенной ОС интерфейс пользователя.

Для тестирования этих приложений существуют специфические особенности:

- тестирование установки;
- тестирование установки обновлений;
- деинсталляция.

При тестировании инсталляции необходимо проверить созданы ли нужные папки и ярлыки в системе при установке, также правильно ли размещены программы в файловой системе, протестировать установку приложения с правами администратора/ без прав администратора и т.д.

В процессе тестирования обновлений проверяется отсутствие повреждений пользовательских данных после установки обновлений, а также доступность созданных ранее файлов.

Тестирование деинсталляции позволяет проверить, удалены ли файлы и ярлыки после удаления программы, вместе с тем проверяются папки текущего пользователя системы на наличие остаточных файлов программы и т.д.

Во второй главе будет рассмотрен процесс тестирования десктоп-приложения на примере Microsoft Word. В процессе тестирования будут учтены все особенности автоматизации такого рода приложений.

Тестирование на проекте разворачивается в условиях Agile – гибкой методологии разработки. Это значит, что процесс разработки и тестирования выполняется в виде серии циклов, т.е. итеративно. Итерация представляет собой

полный жизненный цикл в миниатюре, в результате которого выполняется определенный прирост в функциональности. Согласно гибким методологиям, требования к продукту и их реализация формируются динамически в начале каждой итерации, что позволяет уменьшить риски (рисунок 3).

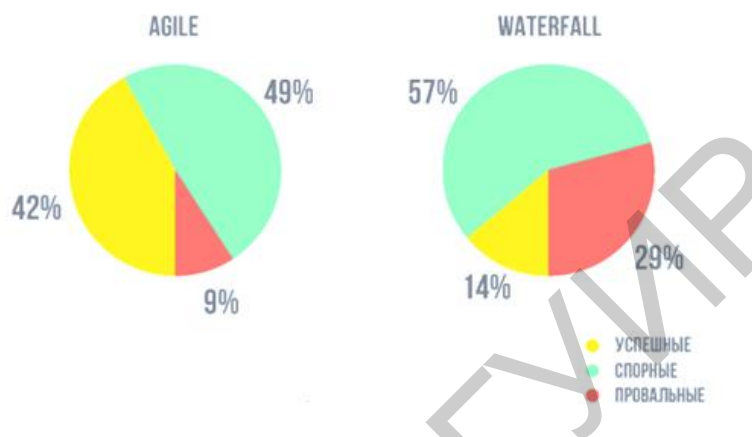


Рисунок 3 – Диаграмма успешности проектов с Agile и водопадной методологиями разработки

Применение методов непрерывной интеграции (CI) позволяет автоматизировать этапы создания, сборки и установки программного кода в рамках процесса разработки ПО. Одним из главных достоинств CI является то, что на протяжении всего процесса разработки на сервере всегда располагается самая свежая рабочая версия разрабатываемой системы. Это программный код такого качества, работу которого можно в любой момент времени продемонстрировать заказчику. Есть возможность показать готовые или готовящиеся компоненты ПО заказчику и получить от него обратную связь.

На всем протяжении разработки данный продукт подвергается непрерывному тестированию. Становится возможным найти ошибки и отклонения в работе различных компонентов еще на ранних стадиях разработки программного обеспечения. Очевидно, что чем раньше ошибка выявляется, тем легче и оперативнее она может быть исправлена. Как результат, риски увеличения сроков разработки снижаются, а качество программного кода повышается.

К тому же, по завершении разработки, программный продукт тестируется ручными и автоматизированными тестировщиками, проверяется, соответствует ли ПО требованиям заказчика в случаях, максимально приближенным к реальным бизнес-сценариям.

Получается, что сочетание практики непрерывной интеграции дает гарантию качества программного продукта, подтверждает, что все требования

заказчика были учтены и реализованы в полной мере, данный продукт не имеет дефектов в программном коде.

В процессе построения непрерывной интеграции можно выделить ряд основных этапов которые приведены на рисунке 4.

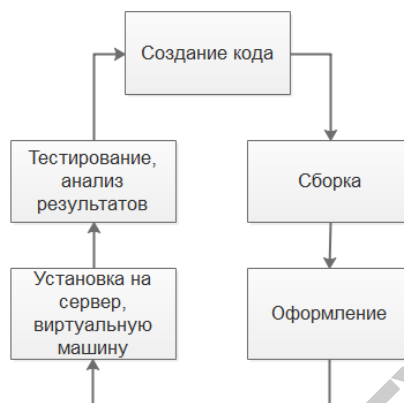


Рисунок 4 – Этапы методологии непрерывной интеграции

Первый этап – разработка исходного кода. Код программного продукта зачастую пишется командой различных специалистов. Для возможности совместной работы над одинаковыми модулями приложения были созданы такие решения, как системы контроля версий (Version Control System). В них код располагается в хранилище – репозитории.

Второй этап – сборка, компиляция и компоновка. В условиях многопоточности разработки логично применять систему планирования и сборки кода, позволяющую автоматизировать запуск задач по прогонке тестов, деплою приложения на сервер и т.д. Результаты такого прогона автоматически обрабатываются и показываются в единообразном виде.

Третий этап – оформление и подготовка пакета установки.

Следующий этап – установка на сервер. Наиболее эффективной считается «бесшумная» установка, которая предусматривает автоматическую установку на сервер или компьютер, не требуя дополнительного вмешательства человека.

Завершающий этап – тестирование и анализ результатов.

В рамках рассматриваемого проекта был выбран сервер TFS (Team Foundation Server). Он представляет собой продукт компании Microsoft и является комплексным решением, т.е. интегрирует систему контроля версий, сбора данных, составления отчетов, систему отслеживания проектных задач и статусов задач для корпоративной работы на проекте, по разработке ПО. Тем самым данное решение позволяет повысить эффективность процесса разработки.

Стоит сказать, что разрабатываемый тестовый фреймворк нацелен на работу через визуальную форму (интерфейс пользователя).

При разработке тестового фреймворка была выбрана методология DDT.

Методология тестирования, управляемого данными (DDT) применяется в автоматизации тестирования ПО, представляет собой тестирование, выполнение и верификация которого производится на основе данных, которые хранятся в БД или любых других источниках данных.

Обычно сравнивают эталонные данные с теми, что на выходе получает система из метода (функции, программы и т.п.).

Разрабатываться тестовый фреймворк будет с использованием *Titanium*, который является наследником Coded UI. *Titanium* вобрал в себя все преимущества родителя, а также в нем создана отличная структура и возможность писать тестовый фреймворк с применением паттерна PageObject, что упрощает тесты и делает их понятными. Он предоставляет возможности:

- Visual Studio in-box;
- поддержка Microsoft, интеграция в TFS;
- интеграция с MTM, что позволяет поддерживать DDT;
- возможность интеграции с .Net;
- удобный доступ к элементам и их свойствам;
- возможность всесторонней проверки в одном Assert методе;
- создание кастомных визуальных отчетов.

Недостатком является то, что он платный, доступен в таких версиях Visual Studio, как Premium и Enterprise.

В третьей главе в контексте организации процесса тестирования десктоп-приложения ставится задача создания фреймворка для тестирования приложения, позволяющего повысить эффективность контроля качества программного продукта. Также немаловажна организация непрерывного запуска тестов для детального и исчерпывающего анализа результатов.

Для выполнения поставленной цели были сформулированы следующие задачи:

1 Дать обоснование рациональности внедрения автоматизированного тестирования десктоп-приложений, основанное на оценке эффективности. Описать процесс организации автоматизированного тестирования в рамках проектной деятельности.

2 Предложить инструментарий формирования аналитической части системы, основанный на анализе гибких методологий разработки, существующих технологий тестирования и возможностей непрерывной интеграции.

3 Разработать фреймворк для автоматизации тестирования десктоп-

приложений на основе технологии Titanium с последующей интеграцией в системе непрерывного запуска тестов.

В связи с поставленными задачами и с целью корректной работы фреймворка, необходимо придерживаться следующих требований к системе:

- использование операционных систем семейств Windows;
- для разработки фреймворка использовать язык программирования C#;
- использовать фреймворк Titanium для разработки сценариев;
- реализовать решение с использованием паттерна PageObject;
- для запуска тестов, хранения кода и организации тестирования использовать TFS;
- хранение тест-кейсов, а также входных значений для них в MTM.

Для начала стоит рассказать про общую структуру фреймворка. При разработке тестовых фреймворков «хорошим тоном» считается использование особого подхода организации элементов в коде – применение Page Object. Это популярный в автоматизированном тестировании шаблон проектирования, который упрощает поддержку написанных тестов и уменьшает количество дублируемого кода. Описывается объектно-ориентированный класс, который выступает интерфейсом страницы тестируемого приложения. Методы данного класса используются в тесте при взаимодействии с элементами пользовательского интерфейса приложения. Большим преимуществом является то, что при изменении дизайна пользовательского интерфейса, нужно исправлять не сами тесты, а только лишь код внутри класса Page Object. Соответственно, все изменения, которые необходимо осуществить для реализации поддержки нового интерфейса, будут сосредоточены в одном месте.

В приложении имеется 3 основных модуля:

- MCWinFramework – содержит непосредственно описание страничек и форм приложения, а также методы манипуляции элементами странички;
- SharedCore – здесь хранятся различные утилиты и методы, не зависящие от интерфейса, странички;
- TestingScenarios – в данном проекте пишутся непосредственно тесты.

Методы для доступа к элементам десктоп-приложений предоставляет утилита Titanium. Элементы представляются как property в классе, а Titanium позволяет наглядно описать элемент по какому-либо свойству (Name, Id, ClassName).

В заключение было приведено руководство по развертыванию системы.

ЗАКЛЮЧЕНИЕ

Основные научные результаты диссертации

По итогам проведенной работы можно сделать выводы:

1 Дано обоснование рациональности внедрения автоматизированного тестирования десктоп-приложений, основанное на оценке эффективности. Описан процесс организации автоматизированного тестирования в рамках проектной деятельности.

2 Составлен инструментарий формирования аналитической части системы, основанный на анализе гибких методологий разработки, существующих технологий тестирования и возможностей непрерывной интеграции.

3 Разработан фреймворк для автоматизации тестирования десктоп-приложений на основе технологии Titanium с последующей интеграцией в системе непрерывного запуска тестов.

СПИСОК ПУБЛИКАЦИЙ СОИСКАТЕЛЯ

1 – А. Цегельник, Н.Г. Программная поддержка организации IT-проектов на основе гибких методологий/ Н.Г. Цегельник, С.А. Поттосина // Проблемы экономической информатики: материалы 52-й науч. конф. аспирантов, магистрантов и студентов, Минск, 25-30 апр. 2017 г. – Минск: БГУИР, 2016.

2 – А. Цегельник, Н.Г. Автоматизация тестирования десктоп-приложения с использованием Titanium / Н.Г. Цегельник, С.А. Поттосина // Проблемы информатики: материалы 53-й науч. конф. аспирантов, магистрантов и студентов, Минск, 25-30 апр. 2017 г. – Минск: БГУИР, 2017.

3 – А. Цегельник, Н.Г. Автоматизация тестирования пользовательского интерфейса десктоп приложений / Н.Г. Цегельник // Молодежный научный форум: Технические и математические науки: электр. сб. ст. по материалам LII студ. междунар. науч.-практ. конф. — М.: «МЦНО». — 2017 —№ 12(52).

Библиотека БГУИР

РЭЗЮМЭ

Цагельнік Надзея Генадзьеўна

Аўтаматызацыя тэсціравання дэсктоп-прыкладанняў з выкарыстаннем Titanium

Ключавыя словы: тэставанне, аўтамацізацыя тэставання, дэсктоп-прыкладанні.

Мэта працы: аптымізацыя працэсу тэставання шляхам яго аўтаматызацыі з выкарыстаннем тэхналогіі Titanium і сродкаў бесперапыннай інтэграцыі.

Атрыманыя вынікі і іх навізна: вивучаны асноўныя метады і тэхналогіі кантролю якасці праграмных прадуктаў. Даказана эфектыўнасць выкарыстання тэставання як інструмента кантролю якасці. Выяўлены станоўчыя і адмоўныя аспекты аўтаматызацыі тэставання, разгледжаны асаблівасці тэставання дэсктоп-прыкладанняў. Выраблены аналіз гнуткіх метадалогій распрацоўкі, існуючых тэхналогій тэставання і магчымасцяў бесперапыннай інтэграцыі. На падставе праведзенага аналізу пацверджана эфектыўнасць і рацыянальнасць ўкаранення бесперапыннай інтэграцыі ў працэсе распрацоўкі з выкарыстаннем гнуткіх метадалогій. Апісаны працэс арганізацыі аўтаматызаванага тэставання ў рамках праектнай дзейнасці. Распрацаваны падыход да рэалізацыі тэставага фреймворка з выкарыстаннем тэхналогіі Titanium, а таксама арганізацыі яго бесперапыннай інтэграцыі ва ўмовах гнуткай метадалогіі распрацоўкі.

Вобласць ужывання: тэставанне.

РЕЗЮМЕ

Цегельник Надежда Геннадьевна

Автоматизация тестирования десктоп-приложений с использованием Titanium

Ключевые слова: тестирование, автоматизированное тестирование, десктоп-приложение.

Цель работы: оптимизация процесса тестирования путем его автоматизации с использованием технологии Titanium и средств непрерывной интеграции.

Полученные результаты и их новизна: изучены основные методы и технологии контроля качества программных продуктов. Доказана эффективность использования тестирования как инструмента контроля качества. Выявлены положительные и отрицательные аспекты автоматизации тестирования, рассмотрены особенности тестирования десктоп-приложений. Произведен анализ гибких методологий разработки, существующих технологий тестирования и возможностей непрерывной интеграции. На основании произведенного анализа подтверждена эффективность и рациональность внедрения непрерывной интеграции в процессе разработки с использованием гибких методологий. Описан процесс организации автоматизированного тестирования в рамках проектной деятельности. Разработан подход к реализации тестового фреймворка с использованием технологии Titanium, а также организации его непрерывной интеграции в условиях гибкой методологии разработки.

Область применения: тестирование.

SUMMARY

Tsahelnik Nadzeya

Automation of desktop application testing using Titanium

Keywords: testing, automated testing, desktop application.

The object of study: optimization of the testing process by automating it using Titanium technology and continuous integration.

The results and novelty: The main methods and technologies for software products quality control were investigated. The effectiveness of using testing as a quality control tool has been proven. The positive and negative aspects of test automation was identified. Features of desktop application testing was considered. The analysis of flexible development methodologies, existing testing technologies and the possibilities of continuous integration was carried out. Based on the analysis, the effectiveness and rationality of implementing continuous integration in the development process using flexible methodologies was confirmed. The process of organization of automated testing within the framework of project activity is described.

An approach to implement a test framework using Titanium technology has been developed, also continuous integration has been configured.

Sphere of application: testing.