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

УДК 004.4.124

Зайцев
Евгений Дмитриевич

Автоматизация тестирования web-приложений на .Net

АВТОРЕФЕРАТ
на соискание академической степени
магистра технических наук

по специальности 1-40 81 01 – Информатика и технологии разработки программного обеспечения

Научный руководитель
Таборовец В.В.
кандидат технических наук, доцент

Минск 2019

КРАТКОЕ ВВЕДЕНИЕ

В современном мире всё больше внимания уделяется автоматизации различных процессов с целью перекладывания рутинных задач на компьютеры, уменьшения денежных затрат на персонал, исключения фактора человеческой ошибки.

Ярким примером рутинной задачи является тестирование. Если разработчики пишут 10 модулей для приложения в неделю, то за первые 5 рабочих дней тестировщику надо протестировать 10 модулей. За следующие 5 дней – уже 20 модулей, потому что необходимо убедиться, что, во-первых, новые 10 модулей работают правильно, а во-вторых, что они не сломали ничего из существующего функционала. В самом начале у тестировщика не так много работы, но с каждой неделей её объём растёт, и в какой-то момент он приходит к руководству и говорит, что не справляется с объёмом работы. В таком случае можно либо нанять дополнительных тестировщиков, либо автоматизировать тестирование на проекте. Как раз об автоматизации тестирования и пойдёт речь в данной статье. Действительно ли она избавляет от всех проблем на проекте и должна быть применена повсеместно?

Считается, что автоматическое тестирование — это безусловное благо. Если разработчики и/или автоматизаторы тестирования пишут тесты — это хорошо, лучшее качество достигается большим количеством тестов. Однако в реальном мире их чаще не пишут, а все тестирование проводится вручную. Не стоит списывать такое положение дел на некомпетентность, глупость или банальную лень разработчиков.

На наличие АТ на проекте влияет большое количество факторов. Продолжительность проекта, бюджет, желание заказчика, возможности команды, которая занимается реализацией, грамотная архитектура приложения, наличие дополнительных мощностей и квалифицированных сотрудников, методология разработки, которой следуют в данной команде или на всём проекте, и так далее.

Например, для коротких проектов (примерно полгода или меньше) не имеет смысла разворачивать инфраструктуру, искать специальных людей и вкладываться в автоматизацию в принципе, потому что к моменту, когда она, как предполагается, окупится, проект уже давно будет отдан заказчику. В этом случае выгоднее бросить все силы на функциональное тестирование.

В продолжительных проектах (2 года и более), особенно в случае поддержки, а не разработки с нуля, чем раньше будет принято решение о внедрении автоматизации и само её внедрение, тем быстрее она окупится, а также освободит от большого пласта работ функциональных тестировщиках, позволяя им сосредоточиться на новом функционале, получении навыков автоматизации или разработки и повышении квалификации.

Также эффективность автоматизации зависит от методологии разработки, которая используется на проекте. Например, Waterfall вообще не предназначен

для автоматизации, потому что тестирование проводится лишь один раз и после окончания разработки, что противоречит смыслу автоматизации тестирования.

Ну и не стоит забывать, что многое решает то, какое приложение разрабатывается и как будет выполняться тестирование. Если планируется тестировать web-приложение, имея при этом чёткие структурированные шаги – почему бы не рассмотреть Selenium Web Driver. Если необходимо работать с desktop-приложением – на помощь придёт Coded UI. При необходимости использовать BDD, чтобы люди, далёкие от разработки могли понять, что происходит в тесте и принять участие в разработке, – TestComplete, Cucumber или опять Coded UI окажутся эффективными. А если разрабатывается мобильное приложение, то Appium будет хорошим выбором.

Это и многое другое будет более подробно рассмотрено далее в работе, выделены основные преимущества и недостатки различных инструментов, сферы их применения и влияние на конечный продукт.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Цель и задачи исследования

В современном мире существует большое количество фреймворков для автоматизации. Некоторые из них удачны, некоторые не очень. Но в большинстве своём они подходят для узкоспециализированных задач и имеют средства, которые можно и нужно использовать в других проектах, однако сами фреймворки для этого не предназначены.

Целью диссертационной работы является разработка программного продукта, впитавшего в себя лучшее из различных фреймворков, позволяющего автоматизировать широкий спектр web-приложений с максимальным комфортом.

Для достижения поставленной цели необходимо решить следующие задачи:

1 рассмотреть различные фреймворки для автоматизации тестирования и выделить их основные достоинства и недостатки;

2 продумать необходимость добавления дополнительного функционала для удобства взаимодействия с фреймворком и интеграцией его в существующий проект;

3 разработать архитектуру фреймворка для автоматизации на основе полученных в предыдущих пунктах результатах;

4 реализовать фреймворк для автоматизации тестирования web-приложения;

5 провести тестирование полученной системы и определить, насколько она соответствует поставленным требованиям.

Объектом исследования являются проекты по длительной разработке или поддержке web-приложений с продолжительным сроком тестирования и относительно стабильной структурой.

Предметом исследования является контроль качества проектов путём тестирования, как функционального, так и автоматизированного, а также возможность частичной замены первого последним с минимальными ресурсными затратами.

Основной *гипотезой*, положенной в основу диссертационной работы, является возможность создания фреймворка для автоматизации тестирования web-приложений, который будет удобнее существующих решений, используя их достоинства и нивелируя недостатки.

Связь работы с приоритетными направлениями научных исследований и запросами реального сектора экономики

Появление первых компьютеров повлекло за собой и появление их непременных спутников - компьютерных программ (программных продуктов).

Программирование, как и любая человеческая деятельность, каковым оно является, практически невозможно без ошибок. И если на ранних этапах развития процесс отладки программ, в силу их достаточной примитивности, не представлял большой проблемы, то в настоящее время картина происходящего резко изменилась.

В процессе разработки программные продукты подвергаются изменениям. Как бы хорошо они ни были написаны первоначально, в них вносятся изменения, обусловленные необходимостью исправления существующих ошибок, выявленных в процессе их выполнения, или же желанием внести в программу дополнительные изменения. Расширение областей применения старых программ приводит к появлению новых функциональных требований, не учтенных изначально.

Всё вышесказанное привело к тому, что контроль над изменениями приобрел статус критического фактора для сохранения полезности программ, т. е. фактора, ставшего по степени значимости доминирующим, что не могло не привлечь внимания специалистов. По этой причине появились тестирование ПО и люди, которые им занимаются. А так как мы вошли в технологическую эру, то люди стараются автоматизировать рутинные процессы, в которые частично можно записать ручное тестирование. Например, регрессионное тестирование или BVT тесты.

Личный вклад соискателя

Результаты, приведенные в диссертации, получены соискателем лично. Вклад научного руководителя В. В. Таборовца, заключается в формулировке целей и задач исследования.

Апробация результатов диссертации

Основные положения диссертационной работы докладывались и обсуждались на X Республиканской студенческой научно-практической конференции “Адаптация малого и среднего бизнеса к особенностям инновационной экономики” (Минск, Беларусь, 23 февраля 2018 г.); XVIII научно-технической конференции студентов, аспирантов и молодых специалистов “Новые информационные технологии в телекоммуникациях и почтовой связи” (Минск, Беларусь, 16-17 мая 2018 г.); I технической конференции для студентов “EPAM Student Insider” (Минск, Беларусь, 10 февраля 2019 г.); XIX научно-технической конференции студентов, аспирантов и молодых специалистов “Новые информационные технологии в телекоммуникациях и почтовой связи” (Минск, Беларусь, 14-15 мая 2019 г.). Также были опубликованы в научном журнале “Проблемы инфокоммуникаций” (Минск, Беларусь, 2018).

Опубликованность результатов диссертации

По теме диссертации опубликованы 4 печатные работы, из них 1 статья в научном журнале (входит в список ВАК), 3 работы в сборниках трудов и материалов научно-практической и научно-технических конференций.

Структура и объём диссертации

Диссертация состоит из введения, общей характеристики работы, четырёх глав, заключения, списка использованных источников, списка публикаций автора и приложений. В первой главе был проанализирован рынок и рассмотрены основные его представители и их особенности. Вторая глава посвящена выдвинутым к фреймворку требованиям и выбору технологий, сделанному на их основе. В третьей главе рассказано о реализации фреймворка и инфраструктуры, рассмотрена их конфигурация и представлено обоснование сделанного выбора. В четвёртой главе представлены практические результаты и произведено их сравнение с результатами, полученными при помощи существующего фреймворка. Общий объём работы составляет 85 страниц, из которых 57 страниц основного текста, 28 рисунков, 1 таблица, список использованных источников на 5 страницах и 11 приложений на 22 страницах.

ОСНОВНОЕ СОДЕРЖАНИЕ

Во **введении** определена область и указаны основные направления исследования, показана актуальность темы диссертационной работы, дана краткая характеристика исследуемых вопросов, обозначена практическая ценность работы.

В **первой главе** рассмотрена автоматизация в целом, и существующие решения, представленные на рынке, в частности, определены их основные достоинства и недостатки, а также сферы их применения. Для этого были выбраны шесть наиболее популярных инструментов для автоматизации тестирования web-, desktop- и мобильных приложений.

Для каждого из них рассмотрены ситуации, в которых они используются, порог вхождения, задачи, с которыми они хорошо справляются, а также их лицензия и стоимость.

В данном разделе неформально рассматривались основные существующие конкуренты разрабатываемого продукта и выяснялись те их свойства, которые можно было перекрыть, сделав лучше, доступнее, понятнее.

Вторая глава посвящена требованиям, которые были изначально выдвинуты заказчиком и руководством, на основе которых и выбирались инструменты, с которыми необходимо было работать. В качестве требований были язык программирования, среда исполнения тестов, необходимость интеграции с TFS (nUnit, выбранный в качестве среды исполнения тестов, не имеет встроенной интеграции с)

Вторая глава посвящена разработке архитектуры ПО и алгоритмов для систем вибрационного контроля, обеспечивающих непрерывную регистрацию и определение амплитудно-фазовых параметров на базе вычислительной машины общего назначения с применением модуля АЦП, подключаемого на стандартную шину, и универсальной многозадачной ОС семейства Windows.

Обеспечение функционирования КСВК в режиме реального времени зависит от организации взаимодействия системного ПО и прикладного ПО КСВК. Операционная система Windows не удовлетворяет требованиям, предъявляемым к системам реального времени, по следующим причинам: недетерминированное время реакции на прерывание; малое количество приоритетов; возможность инверсии приоритетов; неопределенность времени реакции на системные вызовы Win32 API; особенности управления памятью в ОС.

Для обеспечения функционирования КСВК в режиме реального времени на базе ОС Windows предложена совокупность следующих решений [1].

Своевременное выполнение критически важной при вводе вибросигналов процедуры передачи данных между аппаратной и программной подсистемами

достигается путем специальной организации передачи данных по каналу прямого доступа к памяти (ПДП).

Недостаточное количество приоритетов ОС Windows и возможность их инверсии решается путем минимизации количества процессов, функционирующих в системе, и установкой для программных потоков КСВК приоритетов реального времени.

Непрерывный ввод данных от АЦП в режиме реального времени обеспечивается с помощью специальной организации процедуры считывания данных. Для этого в оперативной памяти создается буфер, организованный по кольцевой схеме, разделенный на два равных блока, и обеспечивается формирование флага готовности при полном заполнении данными одного из блоков буфера.

Предложен алгоритм взаимодействия процедур чтения данных из кольцевого буфера и их обработки/отображения [2]. Для этого в соответствии с частотой дискретизации модуля аналого-цифрового ввода вычисляется время готовности блока данных размером в половину буфера. На этот временной интервал программируется работа таймера. Срабатывание таймера приводит к созданию нового потока чтения данных с приоритетом реального времени. Поток чтения данных проводит постоянный опрос флага готовности данных. Установка флага готовности запускает процедуру копирования данных из кольцевого буфера, по окончании которой поток завершается.

Диаграмма переходов состояний при функционировании КСВК, построенного в соответствии с предложенной схемой взаимодействия, представлена на рисунке 1.

Завершение потока чтения данных освобождает процессор для решения других задач, в частности, выполнения потока обработки/отображения данных, который работает до тех пор, пока по событию от таймера не будет инициирован очередной поток считывания данных из кольцевого буфера.

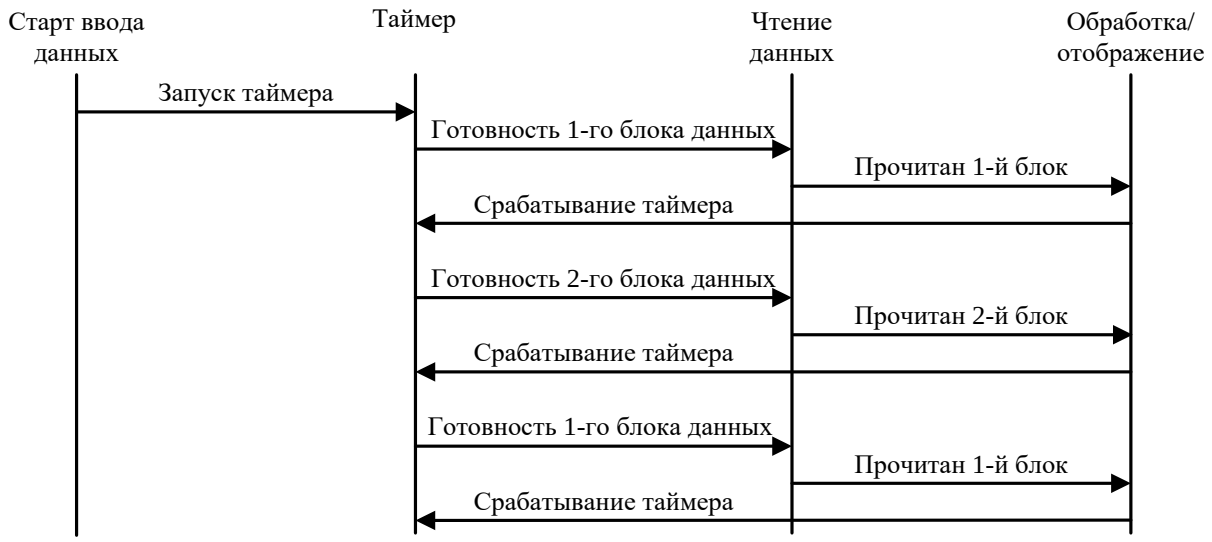

Рисунок 1 – Диаграмма переходов состояний при функционировании КСВК

В **третьей главе** предложены методы формирования диагностических признаков и определения информативно значимых параметров для СППР по оценке ТС сложных механизмов на основе вейвлет-анализа и спектрального анализа, а так же алгоритм кластерного анализа для принятия решения по отнесению исследуемого объекта к определенной группе вибрационного состояния с формированием начального положения ядер кластеров на основе экспертных и экспериментальных оценок.

Сущность метода формирования диагностических признаков для системы поддержки принятия решений по оценке технического состояния механизмов с вращательным движением на основе вейвлет-анализа временных реализаций вибрационных сигналов состоит в применении нормализованных вейвлетов для обнаружения изменений сигнала в определенной частотной полосе [3]. В процессе работы КСВК в режиме реального времени вибрационные сигналы непрерывно преобразуются в цифровую форму и подвергаются вейвлет-преобразованию. В качестве вейвлетов для анализа вибрационных сигналов предлагается применять гауссовы вейвлеты и вейвлет Морле. Дискретное вейвлет-преобразование реализуется выражением

$$C(N,m)=\sum_{n=0}^{N} s(n+m)\Psi\left(\frac{8(n-m)-4N}{N}\right), \qquad (1)$$

где N – ширина вейвлета,

m – временной сдвиг, изменяющийся от 0 до (L – N), L – число дискретных отсчетов в анализируемой временной реализации исследуемого сигнала,

Ψ – вейвлет-функция.

Исходными данными для алгоритма кластерного анализа, решающего задачу по отнесению исследуемого объекта к определенной группе вибрационного состояния, являются вибрационные сигналы, полученные для группы однотипных объектов и исследуемого объекта.

На первом шаге алгоритма на основе спектрального анализа, полосового спектрального анализа, количественных оценок формы сигнала, вейвлет-анализа выбирается ряд параметров, наиболее полно характеризующих состояние объекта. На втором шаге строится характеристический вектор. Координатами вектора являются значения выбранных на первом шаге параметров. Третий шаг заключается в кластеризации полученных характеристических векторов.

Для реализации процедуры кластеризации используется итерационный метод динамических ядер. Начальное положение ядер определяется одним из следующих способов [3].

1. В качестве ядер на основе экспертных или экспериментальных оценок выбираются определенные характеристические векторы.

2. При условии первоначальной неопределенности ядра формируются путем обработки экспериментально полученных для группы однотипных объек-

тов характеристических векторов, заключающейся в определении наилучшего, наихудшего и среднего значений для каждой координаты вектора.

Результатом кластеризации является отнесение характеристических векторов к одному из сформированных кластеров. Каждому из полученных кластеров ставится в соответствие определенное техническое состояние входящих в него объектов. Таким образом, отнесение исследуемого объекта или системы к какому-либо из кластеров позволяет сопоставить его с одной из возможных категорий технического состояния.

В **четвертой главе** рассмотрена практическая реализация ПО для многоканальной системы вибрационного контроля, представлены результаты экспериментального исследования разработанной системы. Система построена по модульно-функциональному принципу, когда основные функциональные действия реализуются отдельными классами. Диаграмма классов ПО автоматизированной системы вибрационного контроля и ППР приведена на рисунке 2.

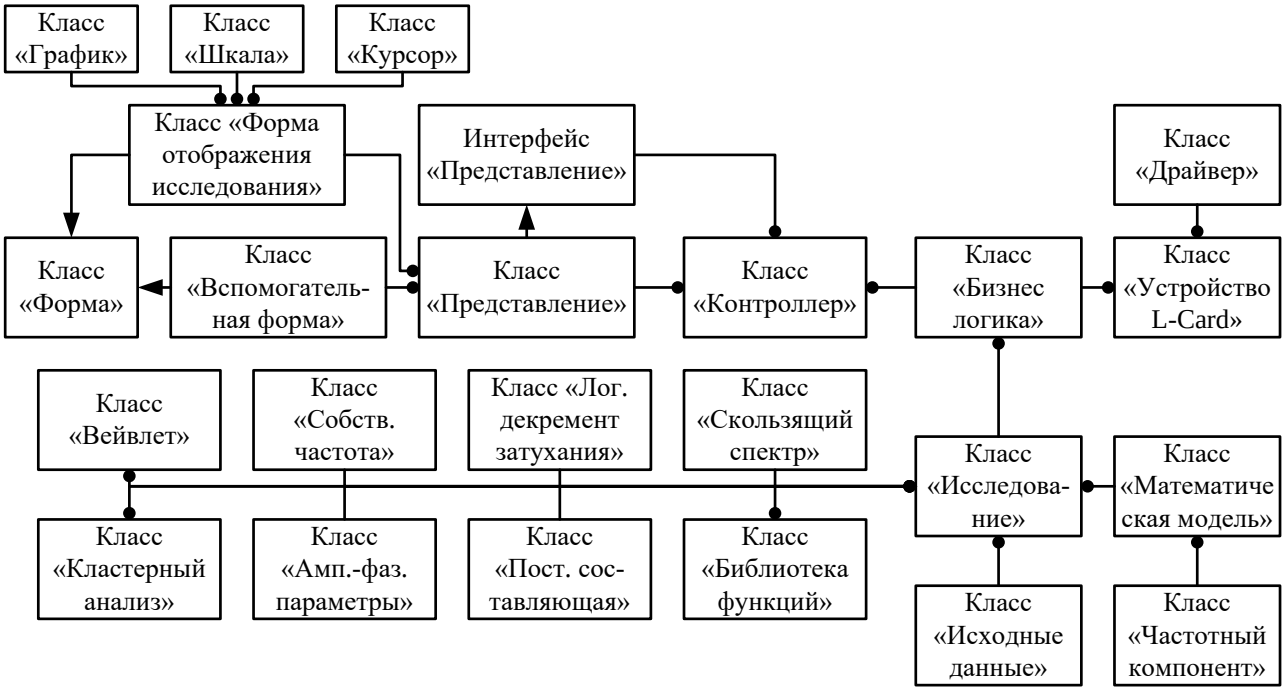

Рисунок 2 – Диаграмма классов программного обеспечения автоматизированной системы вибрационного контроля и ППР

Экспериментальные исследования разработанной программной системы автоматизированного вибрационного контроля и ППР выполнены на базе опытного образца измерительно-вычислительного комплекса, решающего задачи вибрационного контроля и ППР по оценке технического состояния механизмов с вращательным движением и остаточной устойчивости и жесткости зданий и сооружений. Проведены эксплуатационные испытания комплекса и метрологическая аттестация в РУП «Белорусский государственный институт метрологии».

ЗАКЛЮЧЕНИЕ

Основные научные результаты диссертации

1 Рассмотрены существующие предложения на рынке инструментов для автоматизации тестирования, выделены их основные достоинства и недостатки, а также сегмент, в котором их использование максимально эффективно и обосновано.

2 Рассмотрены различные инструменты, которые могут использоваться для создания собственного решения, на основе полученных требований были выбраны наиболее подходящие из них. Также рассмотрены проблемы, которые могут возникнуть при выборе конкретного инструмента, и пути их решения.

3 Реализован фреймворк для автоматизации тестирования, который можно использовать для внедрения автоматического тестирования на проект и который включает в себя дополнительный функционал, который позволит более удобно им пользоваться, минимизировать дублирование кода, а также повысить читаемость, поддерживаемость и расширяемость решения.

4 Выстроена инфраструктура для запуска, хранения и обработки тестов и их результатов, которая позволяет минимизировать время, необходимое на поиск неисправностей, основанное на тестовых результатах, получить дополнительную статистику по тестовым запускам и оценить общее состояние приложения.

5 Было проведено сравнение полученных результатов с данными предыдущих решений по автоматизации тестирования на проектах, для которых данный тестовый фреймворк изначально создавался. Скорость работы тестов была увеличена практически в 3 раза, а стабильность – почти в 5 раз.

Рекомендации по практическому использованию результатов

1.Полученный фреймворк можно использовать как основу для построения автоматизированного тестирования на проекте, тестирование которого полностью или в подавляющем большинстве случае затрагивает только web-приложения, их API, работу с БД.

2.Полученное решение в некоторой мере применимо для тестирования мобильных версий сайта или приложений (если дополнительно внедрить Appium, например), но для тестирования desktop-приложений созданный фреймворк будет неэффективным.

3.Полученные инфраструктура и фреймворк сделаны максимально независимыми, так что при необходимости их можно разделить, оставив, например, только инфраструктуру, настроенную на работу уже с существующим фреймворком.

4.Полученное решение можно использовать для обучения стажёров или функциональных тестировщиков автоматизированному тестированию и программированию, прививая им хорошие практики написания кода, такие как модульность, использование Page Object Model паттерна и др.

СПИСОК ОПУБЛИКОВАННЫХ РАБОТ

1-А.Зайцев, Е. Д. К Вопросу об эффективности автоматизации тестирования web-, desktop- и мобильных приложений / Е. Д. Зайцев, Д. М. Зайцев // “Проблемы инфокоммуникаций”. – Минск, Белорусская государственная академия связи. – 2018. – №2 (8). – С. 56-63.

2-А.Зайцев, Е. Д. Применение облачных технологий в учебном процессе / Е. Д. Зайцев // Адаптация малого и среднего бизнеса к особенностям инновационной экономики: Сборник материалов X Республиканской студенческой научно-практической конференции (Минск, 23 февраля 2018) / Институт предпринимательской деятельности – Минск: Ковчег, 2018. – С. 236-238.

3-А.Зайцев, Е. Д. Применение ARDUINO как пример развития социальных и образовательных навыков у молодёжи в информационном обществе / Е. Д. Зайцев, Д. М. Зайцев // Новые информационные технологии в телекоммуникациях и почтовой связи: материалы XVIII научно-технической конференции студентов, аспирантов и молодых специалистов, 16.05-17.05.2018 г., Минск, Республика Беларусь / редколлегия: А. О. Зеневич [и др.]. – Минск: Белорусская государственная академия связи, 2018. – С. 92-93.

4-А.Зайцев, Е. Д. Философия информационного общества / Е. Д. Зайцев, Д. М. Зайцев // Новые информационные технологии в телекоммуникациях и почтовой связи: материалы XVIII научно-технической конференции студентов, аспирантов и молодых специалистов, 14.05-15.05.2019 г., Минск, Республика Беларусь / редколлегия: А. О. Зеневич [и др.]. – Минск: Белорусская государственная академия связи, 2019. – С. 104-106.