

**Test Case Manager – приложение за управление на тестови сценарии**

INFB01 Дипломиране

*Име:* **Борислав Ивов Вапцаров** *Научен ръководител:*

*Фак. №:* **F55118**

*Специалност:* Информатика

*Модул:* Приложна информатика

*Департамент:* Информатика

**22.05.2014, София**

Съдържание

[Уводна част 3](#_Toc510184470)

[Обзорна част 5](#_Toc510184471)

# Уводна част

Като инженер „**автоматичен контрол качеството“** на софтуера в ежедневната си работа се сблъсквам с разнообразие от проблеми. Част от работата ми е да създавам ръчни (manual) тест кейсове и да ги използвам като основа за създаване на автоматични тестове, които кореспондират помежду си.

Първата стъпка от създаването на ръчни тест кейсове води до проблема, че имаме нужда от място, на което те да бъдат съхранявани, интуитивно и лесно да се поддържат и като цяло да се направи процеса удобен и лесен за изпълнение, както и да има възможност няколко инженери по контрола на качеството на софтуера да го използват едновременно (в един екип има повече от един такъв инженер).

Всяка свързана информация трябва да бъде записана и съхранявана по определен начин, а за тест кейсове се използват най-често Microsoft Word или Excel, но когато има функционалности с по-голям брой тест кейсове, възможностите на гореспонеатите методи са неефективни и са трудни за поддържане; не позволяват едновременно изпълнение, четене и писане

Има различни подходи към извършването на всичко гореопоменато. В повечето случаи информацията, която трябва да се съхрани, обнови и евентуално да се изтрие при нужда, включва следното: идентификационен номер (в поредност) и име на теста; стъпки за създаването на теста; очакван резултат за всяка стъпка или за теста като цяло; среда или специфики на браузъра; дата на създаване и промяна; статус (преминал успешно/провален/блокиран/пропуснат/неизпълнен); проблеми; свързани автоматични сценарии и техните характеристики; други добавки, и прочие.

Съществуващото решение за справянето с тази информация е специализирано за целта приложение, чиято пазарна стойност е висока и то невинаги дава желаният резултат. На настоящето ми работно място изпитваме нужда от такова приложение, но тези, които са на пазара са или твърде скъпи, или не са с желаното качество, или и двете. Поради тази причина реших да създам приложение, което отговаря на изискванията, които имаме и същевременно е безплатно и позволява надграждане и променяне в случай на промяна на изискванията ни.

Заради времеотнемащата настояща процедура, основна цел и съответно резултат от дипломната работа ще бъде WPF приложение, наречено “**Test Case Manager**”, което ще бъде от тип „клиент - сървър с база от данни. Това приложение ще позволи интуитивен и лесен начин на създаване на йерархия от тест кейсове за нуждите на съответния софтуерен продукт, изпълнение на ръчните тест кейсове през отделен прозорец и процедура наречена „тест рън“, както и отбелязването на

несъответствия от очакваният резултат, под формата на „бъгове“, които ще имат възможност да се регистрират в GitHub напълно автоматично.

Регистрираните бъгове ще съдържат в себе си основните параметри на тестовия сценарии, които е бил изпълнен: заглавие, приоритет, тестови стъпки, кога и кой е изпълнил теста. Приложението, ще предоставя както копиране в клипбоард-а на връзка към новосъздаденият бъг, така и наличието на директна препратка, която може да се достъпи с клик на мишката.

За да бъде реализиран проекта, трябва да решим следните задачи:

* Да постигнем лесно и интуитвно управление на множество от тестови сценарии, свързани с конкретни функционали области и свръзаните с тях проекти.
* Да предоставим възможност за едновременна работа и изпълнение на тестови сценарии от няколко души едновременно чрез „*оптимистично заключване*“ (*optimistic concurrency / locking*), тоест ще приложим стратегия, където четем даден запис от базата и когато искаме да го променим, ще проверим дали наново прочетената версия, отговаря на тази, която сме прочели в самото начало, ако да – извършваме промяната, ако ли не – то започваме от самото начало.
* Съхранение на информация за това как да се направи тестването, да се планират тестови дейности и да се дава справка за статуса на дейностите на контрола по качеството.
* Да предоставят начин за изпълнение, чрез пускането на избрани от потребителя тестове в тест рън и запазването на информацията от него.
* Лесно маркиране на преминалите тест кейсове, на провалените, блокираните и тези, преминали успешно настоящия test run.
* Да маркира кои тестове минават през съответния тест рън.
* Да позволява един тест да участва в множество тест рънове, както и един тест рън да има множество тестове.
* Интеграция с ***GitHub*** системата за проблеми, където само с няколко клика да може да създава или логва бъг спрямо изпълнен тест.
* Aвтентикация.
* Статистика, свързана с това колко теста са минали, колко са се провалили при миналия test run, колко от тестовете са автоматизирани и прочие.
* Система, чрез която администратора ще може да определя кои потребители имат права да добавят други потребители, променят или трият такива.
* Лесно навигационно меню.
* Възможност за промяна на цветовете на основният прозорец (контури, цвят при селекция, основен фонов цвят с преливане).

# Обзорна част

Обзорна част: Обяснение за какво става въпрос в тезата, какво ще се демонстрира. Описват се известни решения, конкурентни на предлаганото. Ако в заданието е включен избор на технология, описват се възможните технологии с техните предимства и недостатъци. Ако технологията е предварително избрана, представят се съображенията за този избор. Тук е мястото, където описват границите на разработката. Тук може да каже ясно и точно за какво смята да говори и за какво смята да не говори. Също така да се обоснове защо е включил/изключил дадено нещо от тезата си. Обем на обзорната част: до 10 стр.

Инструментите за управление на тестовете се използват за съхраняване на информация как тестовете трябва да бъдат изпълнени, планиране на дейностите по тестване и докладване за състоянието на дейностите по осигуряване на качеството. Инструментите дават различни подходи за тестване и по този начин имат различни групи характеристики. Обикновено те се използват за поддържане и планиране на ръчно тестване, изпълнение или събиране на данни за изпълнение от автоматизирани тестове, управление на множество среди и въвеждане на информация за открити дефекти, като тенденцията през последните години е да има отделна система за регистриране и следене на дефекти/бъгове. Инструментите за управление на тестовете предлагат перспективи за рационализиране на процеса на тестване и позволяват бърз достъп до анализ на данни, инструменти за съвместна работа и лесна комуникация в множество екипи по проекта. Много от инструментите за управление на тестове включват възможности за управление на изискванията, макар че все по-рядко срещаме тази възможност, тъй като има специализирани програми, които се занимават само с това.

Инструментите за управление на тестове дават възможност на екипите да консолидират и структурират процеса на тестване, като използват един инструмент за управление на тестовете, вместо да инсталират няколко приложения, предназначени да управляват само една стъпка от процеса. Инструментите за управление на тестовете може да предоставят на екипите възможност да управляват тестовата среда, автоматизираните тестове, дефекти и задачи по проекта, макар че подобни приложения се разработват години от големи софтуерни компании. Някои приложения включват разширени табла за управление и подробно проследяване на ключови показатели, което позволява лесно проследяване на прогреса и управлението на програмни грешки. Улеснение е за мениджърите да виждат графики спрямо как се движи работата, колко теста за изпълнени, колко са минали и колко са открили дефекти в тестовата среда.

Инструмент за управление на тестове, който включва всичкo необходимо за управление на тестовия процес, може да спести на тестери нервите и проблемите по инсталирането на отделни приложения, необходими за процеса на тестване. Те могат да бъдат реализирани с минимална програмна способност, което позволява лесен монтаж и мониторинг на тестовия процес в множество групи и/или проекти. След инсталирането, екипите имат незабавен достъп до потребителски интерфейс и могат веднага да започнат да работят и да записват тестови сценарии. Тези типове приложения са предназначени да опростят процеса на управление на тестовете с вградени високи нива на автоматизация и проследяване, но все пак не изискват усъвършенствани умения за програмиране или знания, които да бъдат приложени. Те са полезни за екипи, които управляват разнообразни тестови сценарии и за по-големи екипи, които се нуждаят от всеобхватно приложение за управление на проекти, макар последната възможност да се предлага единствено и само при платени продукти.

При администриране на тестови случаи потребителите могат да получат достъп до разнообразни „прозорци“ за управление, за да получат незабавен достъп до данните, което прави процеса на тестване ефикасен и точен. Типът на използваният „прозорец“се определя от обхвата на проекта и от информацията и данните, които трябва да бъдат извлечени по време на процеса на тестване. Данните могат да бъдат споделяни и достъпни в множество екипи на проекта, което позволява ефективна комуникация и сътрудничество през целия процес на тестване. Обикновено налични са йерархични структури в няколко нива, например: проект, който може да съдържа области, всяка област може да съдържа функционални или нефункционални ръчни тестове. Те от своя страна могат да се включват в колекция, която може да бъде изпълнявана многократно от тестъри или други роли в екипа. Добрият инструмент за управление на тестовете трябва да позволява промяна или триене на всеки един обект от йерархичното ниво, както и регистриране на дефекти, които са породени от разлика между очакваният резултат от тестването и това, което се вижда от изпълняващият тестовата дейност. Задължително всеки тест трябва да съдържа в себе си стъпки, като всяка стъпка трябва да описва какво трябва да се направи и съответно какво трябва да се случи като очакван резултат.

Като резултат от задачите, трябва да получим приложение, което може директно да се използва от софтуерната индустрия. Проектът цели да олесни и помогне на инженерите по контрол на качеството на софтуера и на бизнес анализаторите да вършат по-добре работата си, използвайки разработеното приложение – **Test Case Manager**, което ще е просто и функционално.

На пазара има много инструменти, които до някаква степен предоставят на потребителите всички гореспоменати функционалности, естествено срещу крупна сума

пари. От безплатните алтернативи, на пръсти се броят тези, които дори могат да се инсталират коректно, а какво остава да се използват по предназначение.

По-надолу ще разгледаме някои от най-известните инструменти за управление на тестове на пазара, техните плюсове и минуси, цена, разработка и човеко-часа нужни за направата им. Ще направим съпоставка между тях и приложението, което разработваме.

Първият инструмент, който ще разгледаме е **TestRail**, разработен е от **Gurock Software** - основана през 2004 г. и с офиси във Франкфурт, Дъблин, Остин и Хюстън. Имат над 900+ софтуерни инжинери и се фокусират единствено и само върху изграждането и поддържането на мощни инструменти с красиви интерфейси, които помагат на софтуерните екипи по целия свят да доставят надежден софтуер. Gurock е компания на Idera, Inc. Idera, Inc. е компания-майка на световните брандове за продуктивност на софтуер за B2B, чиито решения позволяват на техническите потребители да правят повече с по-малко усилия и съответно по-бързо. Маркерите на Idera, Inc. обхващат три отдела - инструменти за бази данни, инструменти за разработчици и инструменти за управление на тестове. Тези три отдела са евангелизирани от милиони членове на общността и повече от 50 000 клиенти по целия свят, включително някои от най-големите медицински центрове в света и технологични компании.

Само от описанието на компанията разбираме, че **TestRail** е технически изпипан инструмент; ще предостави голяма гама от възможности за управление на тестовете, логване на бъгове и изкарване на статистики на потребителите, като освен функционалност, предоставя помощ в реално време, естествено срещу съответното заплащане, което не е малко.

**TestRail** помага за управлението и проследяването на усилията си за тестване на софтуер и да организирането на отдела за осигуряване на качеството. Неговият интуитивен уеб базиран потребителски интерфейс улеснява създаването на тестови сценарии, управлението на тестовете и координирането на целия процес на тестване. Добавя проекти с табла за управление и отчети за дейността. Позволява получаването на информация в реално време за напредъка в тестване, добавя филтри и известия по имейл.

Ще изброим някои от основните функционалности, които **TestRail** предоставя:

* Уеб базирано приложение с модерен дизайн
* Изпълнение на тестове и проследяване на тестовите резултати
* Метрики и репорти свързани с процеса по тестване
* Интеграция с основите системи за проследяване на бъгове, като **JIRA**
* Персонализиране списъци със задачи, изпращане на писма
* Гранулярни позволения, роли и потребители
* Гъвката и адаптивна среда

XRay, TestRail, Zephyr