

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

TestRunner

-

**softwarový prostředek pro
správu testů**

Zadání

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 18. června 2014

Lukáš Tancer

Abstract

In this thesis I deal with the testing software products. This problem falls within the area, which is called software development process. Because this area is pretty huge, main focus is on testing inside Kerio technologies, Inc.

The main purpose of this work is to improve their internal program called TestRunner. At the moment TestRunner has a few bugs left from previous versions. Some functionality is no longer needed. Finally, some new functions are required.

The goal is to keep TestRunner vital for next few years.

Obsah

| | | |
|----------|---|----------|
| 1 | Úvod | 1 |
| 2 | Problematika testování | 2 |
| 2.1 | Testování obecně | 2 |
| 2.2 | Role testování ve vývoji Softwaru | 4 |
| 2.2.1 | Agilní metodiky | 4 |
| 2.2.2 | Scrum obecně | 6 |
| 2.2.3 | Testování ve scrumu | 8 |
| 2.3 | Typy testování | 10 |
| 2.3.1 | Manuální testování | 10 |
| 2.3.2 | Automatické testování | 11 |
| 2.4 | Metody testování | 12 |
| 2.4.1 | Black box testování | 12 |
| 2.4.2 | White box testování | 12 |
| 2.4.3 | Gray box testování | 12 |
| 2.5 | Typy testů | 13 |
| 2.5.1 | Unit testy | 13 |
| 2.5.2 | Funkční testy | 13 |
| 2.5.3 | Integrační testy | 13 |
| 2.5.4 | Systémové testy | 14 |
| 2.5.5 | Akceptační testy | 16 |
| 2.6 | Chyby | 18 |
| 2.6.1 | Závažnost | 18 |
| 2.6.2 | Priorita | 18 |

| | | |
|----------|--|-----------|
| 2.6.3 | Životní cyklus chyby | 19 |
| 3 | TestRunner | 20 |
| 3.1 | Obecně | 20 |
| 3.2 | Popis funkčnosti | 21 |
| 3.3 | Specifikace požadavků | 21 |
| 3.3.1 | Dashboard | 21 |
| 3.3.2 | Redukce práv | 21 |
| 3.3.3 | Závislost testovacích případů na platformě | 22 |
| 3.3.4 | Vyhledávání | 22 |
| 3.3.5 | Dvojjazyčnost | 22 |
| 3.3.6 | Proměnlivá velikost levého panelu | 22 |
| 3.3.7 | Bugfix | 23 |
| 3.4 | Aktuální programové prostředí | 24 |
| 3.5 | Přechod na nový operační systém | 25 |
| 3.6 | Analýza | 25 |
| 3.6.1 | Datová vrstva | 25 |
| 3.6.2 | Aplikační vrstva | 27 |
| 3.7 | Oprava chyb | 31 |
| 3.7.1 | Tlačítka Won't test a Reset | 31 |
| 3.7.2 | Manažer komentářů | 31 |
| 3.7.3 | Ostatní | 32 |
| 3.8 | Odebrání nadbytečných funkcí | 32 |
| 3.8.1 | Zrušení dvojjazyčnosti | 32 |
| 3.8.2 | Redukce uživatelských rolí | 33 |
| 3.9 | Nové funkce | 34 |
| 3.9.1 | DashBoard | 34 |
| 3.9.2 | Vyhledávání | 36 |
| 3.9.3 | Součty průměrných časů testů pro kapitoly | 36 |
| 3.9.4 | Proměnlivá velikost levého panelu | 37 |
| 3.9.5 | Ověřování přes LDAP | 37 |
| 3.9.6 | Vazba testovacích případů s platformou | 39 |
| 3.9.7 | Tooltipy | 41 |

| | | |
|----------|---------------------------------|-----------|
| 4 | Testy TestRunneru | 44 |
| 4.1 | Automatické testování | 44 |
| 4.2 | Manuální testování | 45 |
| 5 | Závěr | 47 |

1 Úvod

V této práci se budu zabývat testováním softwarových produktů. Protože tato oblast je poměrně rozsáhlá, zaměřím se hlavně na testování uvnitř společnosti Kerio Technologies s.r.o., pro kterou je tato práce primárně určena.

Hlavním cílem práce bude vylepšení jejich interního programu jménem TestRunner. V tuto chvíli zbývá v TestRunneru několik chyb z předchozích verzí. Některé funkce již nejsou nadále potřeba. Naopak vyvstala potřeba nových. Tyto úpravy prodlouží životnost TestRunneru o několik dalších let.

Jako první provedu přechod na novější operační systém a kompletní omlazení stávajícího softwaru. Dále provedu hloubkovou analýzu kódu. Drobné změny v kódu mi pomohou k lepšímu pochopení celého programu. Program následně vyčistím od přebytečných funkcí a některé části kódu inovuji. Nakonec implementuji nový autentizační systém a další požadované funkce.

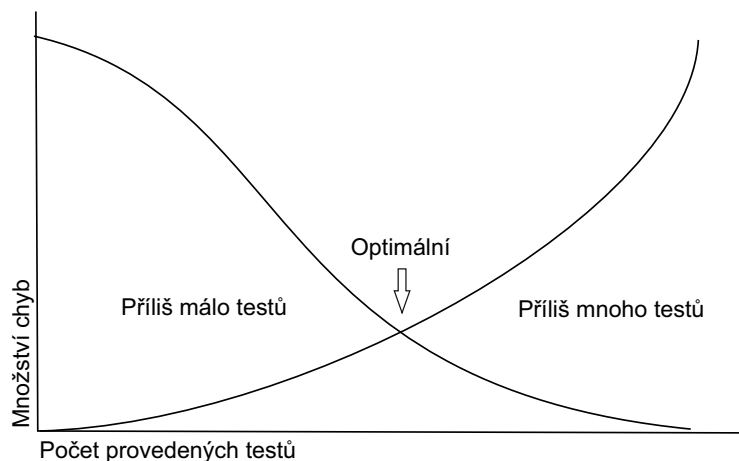
Před předáním program důkladně otestuji. V rámci předání budu změny prezentovat celému QA oddělení Kerio Technologies s.r.o.. Vzhledem k tomu, že rozsah změn je poměrně velký, program předám a nasadím ve dvou iteracích.

2 Problematika testování

2.1 Testování obecně

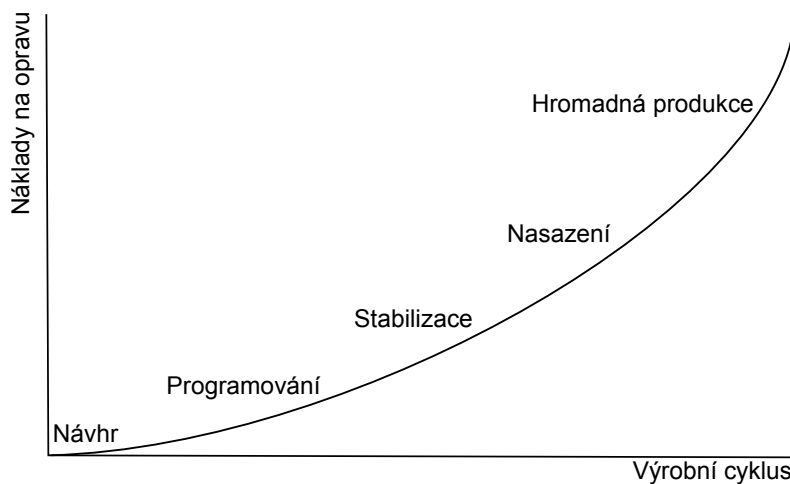
Hlavní myšlenkou testování je zjišťování kvality testovaného produktu. V našem případě je oním produktem software. Důvod, proč se zajímáme o testování, je jednoduchý. U softwaru nemůžeme nikdy dosáhnout perfektního vývojového cyklu. Abychom se mu mohli alespoň přiblížit, je bezpodmínečně nutné, aby byl jakýkoliv produkt důkladně testován.

Bohužel i sebelepší testovací strategie mám nemůže zaručit, že výsledný produkt bude bezchybný. Pokud ovšem testování věnujeme dostatek času, můžeme množství chyb výrazně omezit. Nyní nastává problém. Musíme si rozvrhnout testování tak, abychom našli co nejvíce chyb za co nejmenší množství času viz obr. 2.1.



Obrázek 2.1: Optimální hladina v softwarovém testování.[15]

Čím později se chyba projeví, tím obtížnější a nákladnější bývá její odstranění viz obr. 2.2. Z tohoto důvodu probíhá testování již ve velice raných etapách vývoje.



Obrázek 2.2: Náklady na opravu chyby v průběhu výrobního cyklu [16].

Produkt se z pravidla předkládá k testování s každou novou verzí. Ta může být například noční, hodinová či inkrementální. Proto je velké množství chyb odhaleno velice brzy. Bohužel je další velké množství chyb, které se projeví až v pozdější etapě vývoje. Je to způsobeno hlavně architektonickými nedostatky, lidskými chybami, množstvím kódu a jeho integrací do jednoho celku.

2.2 Role testování ve vývoji Softwaru

2.2.1 Agilní metodiky

Kerio Technologies s.r.o. za dobu své existence vyzkoušelo řadu metodik vývoje softwaru. Nejlépe se osvědčily agilní metodiky a ty nyní využívá k vývoji svých produktů. Oproti tradičním metodikám, nemá program pevně zadanou funkcionalitu, která se musí dodržet nehledě na zdroje a čas. Hlavní jsou čas a zdroje, od kterých se odvíjí, kolik funkcionality bude implementováno (viz obr. 2.3). Nehrozí tedy, že by v daném termínu nebyl program hotov, ale spíše nebude pokryta funkcionalita s nižší prioritou. Kerio vyvíjí software v iteracích s periodou přibližně půl roku.



Obrázek 2.3: Rozdíl v plánování mezi klasickým a agilním přístupem [13].

Agilní metodiky jsou založeny na iteracích. K pochopení, jak vlastně fungují, nám poslouží následující postup:

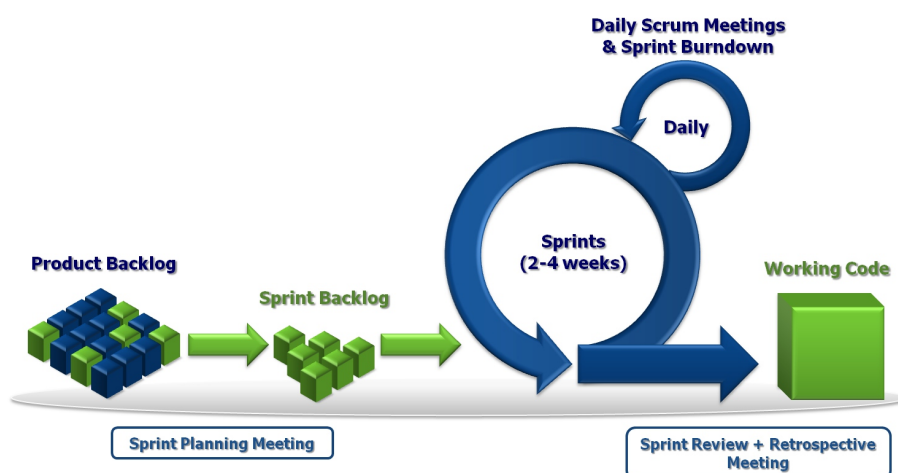
- Nultá iterace: Základ aplikace
 1. Analýza problému.
 2. Naprogramování základní kostry programu.
 3. Hlavním cílem je hotový kus aplikace, který lze demonstrovat.
- První až N-tá iterace: Vývoj aplikace.
 1. Výběr implementované funkcionality a analýza postupu.
 2. Implementace.
 3. Build, Testování, Předvedení.
- Vydání produktu.
- Při rozvoji či údržbě pokračujeme dále v iteracích stejně jako ve vývoji.

Po krátkém úvodu do agilních metodik si nyní pojd'me ukázat jejich hlavní výhody:

1. Je dobře známo, kolik práce tým zvládne v jedné iteraci udělat a díky tomu lze řídit rychlost týmu.
2. Možnost změny zadání za běhu.
3. Rychlé doručení produktu k zákazníkovi.
4. Po každé iteraci máme potenciálně vydatelný produkt.

2.2.2 Scrum obecně

Existuje mnoho agilních metodik, protože ale popis všech není cílem této práce, uvedu pouze metodiku Scrum, která je firmou Kerio Technologies s.r.o. aktivně využívána. Je to opět iterativní vývoj rozdělený do jednotlivých fází nazývaných sprinty. Ty trvají v průměru 2-4 týdny. Na každodenních schůzkách se aktualizuje postup na projektu, dokončené úkoly, následující úkoly a případné problémy. Nespornou výhodou scrumu je také schopnost samostatné organizace týmu. Manažerovi tedy výrazně ubude práce. A protože ke scrumu není potřeba žádný dodatečný trénink, je velice jednoduché s ním začít. Přibližná funkčnost scrumu viz obr. 2.4.



Obrázek 2.4: Jak funguje Scrum [10].

Abychom mohli se scrumem začít, je třeba v týmu určit následující role:

- Product owner - Zodpovídá za priority, za to, co se bude v příštím sprintu implementovat, a určuje implementační detaily.
- Scrum Master - Stará se o veškeré potřeby týmu a zároveň vývojáře řídí. Neměl by ovšem být programátor.
- Development Team - Jedná se o vývojáře a testery.
- Customer - Zákazník ovlivňuje priority a slouží jako prvotní impulz a zpětná vazba zároveň.

Dále budeme potřebovat zavést artefakty:

- Product backlog - Je to seznam veškerých úloh řazený podle priorit, který je pro produkt udržován. Má ho na starosti Product owner.
- Sprint backlog - Je to seznam veškerých úloh pro daný sprint. Úlohy se dále dělí na dílčí podúlohy. Je dobrou praxí, že tyto podúlohy mají rozsah 4-16 hodin[10]. Sprint backlog má na starosti vývojový tým a bývá často realizován jako scrumboard viz obr. 2.5.



Obrázek 2.5: Příklad scrumboardu v Kerio Technologies [16].

2.2.3 Testování ve scrumu

Vzhledem k povaze této práce nás bude ve scrumu nejvíce zajímat role testera. Ten je součástí vývojového týmu a má tyto povinnosti:

1. Účastní se všech schůzek týmu.
2. Spolupracuje na designu aplikace.
3. Vypracovává testy, kterými aplikace musí projít. Tyto testy budou nadále využívány i v následujících sprintech a zajišťují celkovou integritu aplikace.
4. Musí ověřit funkčnost všech funkcionalit implementovaných ve sprintu.
5. Testuje se vždy aktuální build.
6. Nahlašuje veškeré chyby včetně čísla buildu.

Z charakteru výčtu je vidět, že většina z těchto povinností je pouze organizační. Rozebereme pro nás zajímavý bod 3.

Nejdůležitější je stanovit si výstupní kritéria, za kterých můžeme test považovat za úspěšný. Ty vychází ze specifikací nebo požadavků na software. Na první pohled by se mohlo zdát výhodné vymezení konkrétního výstupu programu. Bohužel často existuje mnoho cest jak dosáhnout cíle a uživatel si může vybrat jinou než námi testovanou. Proto zavedeme testování, které toto zohlední a bude vytvářet různé cesty průchodem programu, tzv. testovací případy. Ty mohou být navrženy samotným testerem nebo na základě konkrétního příběhu uživatele. Další výhodou tohoto postupu je jeho jednoduchost. Není potřeba žádná rozsáhlá dokumentace ani údržba a navíc zvýšíme celkovou flexibilitu testování.

Další velice důležitou vlastností testu je jednoznačnost. U každého testu bychom měli být schopni určit, zdali test dopadl či nedopadl úspěšně. Bohužel nalezneme i případy, kde jsou požadavky na test z části nekompletní. Zde se vyplatí využít naše nejzkušenější testery. Ti dostanou za úkol program prozkoumat. Tím se program naučí, pak vytvoří testy a provedou je.

2.3 Typy testování

2.3.1 Manuální testování

Obecný popis

V případě manuálního testování jsou jednotlivé testovací scénáře vykonávané člověkem. Pro testera jsou časově náročné, zdlouhavé a únavné. S tím je spojena další nižší spolehlivost testů kvůli přítomnosti lidského faktoru. Další nevýhodou je nutnost velkých investic do lidských zdrojů, které bývají při opakovaných testech vyšší než investice do výpočetních zdrojů. Výhodou manuálního testování je mnohem vyšší flexibilita. Další nepopiratelnou výhodou je lidský cit pro vyhodnocení správného či špatného výsledku.

Hlavně v rané části vývoje, kdy se aplikace velice často mění, je manuální testování nejvhodnější.

V praxi

Každý z produktů vyvíjených Kerio Technologies má na stálo v týmu minimálně jednoho člena z oddělení QA. Jeho/Její starostí je testování aktuálně implementovaných funkcionalit. V okamžiku, kdy produkt dospěje do stádia stabilizace, bývají dočasně k testování přiděleni další členové QA.

2.3.2 Automatické testování

Obecný popis

Automatické testování je ve většině případů znatelně rychlejší než manuální. Nevyžaduje tolik investic do lidských zdrojů. To se projeví hlavně pokud máme mnoho různých prostředí, pro která je program vyvíjen. Automatické testy je zvládnou pokrýt všechny současně. Zároveň fungují vždy stejně, a je tedy eliminován lidský faktor (v případě, že je test napsán správně). Za nevýhodu automatických testů můžeme považovat časovou náročnost na vývoj a údržbu. Bohužel ani automatické testy nemohou testovat všechno, a tak je nelze používat jako jedinou metodu testování.

Automatické testy najdou nejčastěji využití v pozdějších etapách testování, kdy se funkcionalita programu příliš nemění. Důvodem je nižší režie na jejich vývoj a údržbu.

V praxi

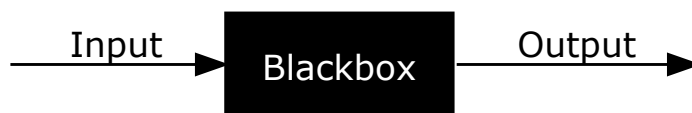
První linií testování je automatický překlad produktu každý den po půlnoci. Díky tomu jsou odhaleny kritické chyby bránící překladu. O infrastrukturu automatických testů a sdílené části testů se stará tým Autotest, který spadá pod tým QA. Samotný vývoj testů si následně produktové týmy zajišťují samy. Testy jsou automaticky spuštěny každý den po úspěšném překladu produktu a dále kdykoliv na požádání.

2.4 Metody testování

Existují různé metody testování softwaru. V této části ale uvedu pouze tři základní. Těmi jsou black, white a gray box testování [12].

2.4.1 Black box testování

Každý test je navržen bez znalosti jakýchkoliv vnitřních struktur programu. Ani tester nemá znalosti architektury či zdrojových kódů. Tomuto postupu se říká Black-box testování viz obr. 2.6.



Obrázek 2.6: Black-box testování [2].

2.4.2 White box testování

Testování je detailní průzkum vnitřní logiky a struktury zdrojových kódů. Aby mohl tester začít s tímto testováním, potřebuje znalost fungování architektury a zdrojových kódů. Průzkumem kódu poté zjišťuje, která část funguje nevhodně.

2.4.3 Gray box testování

Testování probíhá s omezenou znalostí architektury a zdrojových kódů. Oproti black box testování, kde tester mohl testovat pouze uživatelské rozhraní aplikace, u grey box testování je možné kontrolovat i návrhové dokumenty či databázi. Díky těmto znalostem může tester snáze připravovat data i testovací scénáře.

2.5 Typy testů

Vzhledem k tomu, že vývoj SW zpravidla probíhá v několika fázích, je zřejmé, že v jednotlivých fázích budeme provádět rozdílné typy testů. Jejich rozdělení lze dobře vystihnout takto:

2.5.1 Unit testy

Každý vývojář by si měl svou práci vždy sám otestovat. Už od samého začátku může eliminovat velké množství chyb použitím unit testů. Ty zahrnují nástroje, metodiky a činnosti, jejichž cílem je ověřování správné funkčnosti dílčích částí neboli jednotek zdrojového kódu. V rámci testování by se měl vývojář zaměřit i na to, jestli používá vhodné algoritmy, návrhové vzory, datové typy atd.

2.5.2 Funkční testy

Testy jsou prováděné na straně dodavatele. Aplikace nemusí být testována na plně integrovaném prostředí. Jsou vhodné pro testování jednotlivých funkčních částí, ale nezaručují funkci celku jako takového.

2.5.3 Integrační testy

Integrační testy zjišťují, zdali všechny součásti programu spolupracují tak, jak mají. Jsou zvláště užitečné pro modulární programování. Můžeme je dále rozdělit takto:

Regresní testy

Ověřují, zda se v důsledku změny neobjevila chyba tam, kde to předtím již bylo v pořádku. Jsou ideálním kandidátem pro automatizaci.

Smoke testy

Jedná se o krátký test, který slouží jako rychlé ověření, zda je vyvíjená aplikace připravena pro další fázi testování. Obvykle se provede jen jednoduché ověření, že všechny části aplikace jsou implementovány, nainstalovány a spuštěny. Jsou ideální pro testování základních funkcí softwaru. Protože se příliš často nemění, jsou ideální pro automatizaci.

2.5.4 Systémové testy

Během těchto testů je aplikace ověřována jako funkční celek. Tyto testy jsou používány v pozdějších fázích vývoje. Ověřují aplikaci z pohledu zákazníka. Většinou slouží jako výstupní kontrola softwaru.

Recovery testy

Účelem je otestovat, jak rychle a zda se vůbec produkt vzpamatuje po pádu systému, HW chybě či jiné neočekávané události.

Bezpečnostní testy

Odhalují, jak a zda se vůbec systém chrání před neautorizovaným přístupem. Další možné funkce jsou i nalezení nežádoucího kódu, bezpečnostních chyb nebo zranitelností.

Zátěžové testování

Testování nezahrnuje pouze funkčnost, ale také to jak se program bude chovat v extrémních situacích. K tomu nám pomůže zátěžové testování, kde budeme testovat stabilitu v následujících situacích:

- Zatížení velkým množstvím informací, ať už nárazově nebo dlouhodobě.
- Spouštění velkého množství kopií programu.

Výkonnostní testy

Při tomto testu systém odolává velkému počtu různých požadavků a sleduje se, jaká je jeho odezva, resp. jak je ovlivněn výkon aplikace, např. rychlost odpovědí na jednotlivé požadavky. Tím lze vysledovat, kterým částem systému je třeba věnovat větší pozornost, a těch pak provést příslušné optimalizace.

Instalační testy

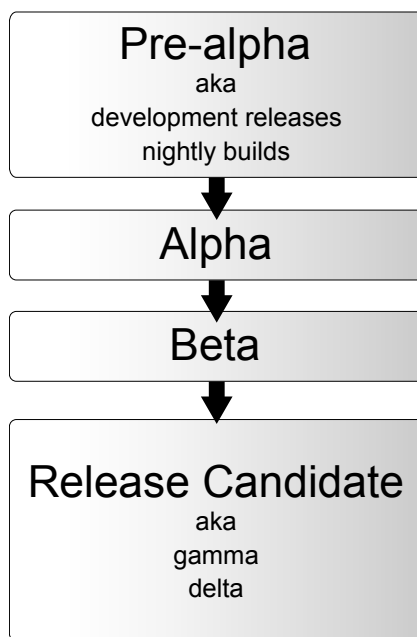
Tyto testy slouží ke kontrole hladkého průběhu instalace, odinstalace a přechodu na novější verzi softwaru.

2.5.5 Akceptační testy

Akceptační testy slouží k ověření funkčnosti před nasazením do ostrého provozu. Pokud aplikace úspěšně projde akceptačními testy je připravena k vydání. V této fázi byla již naprogramována veškerá funkcionalita a zbývá program doladit. Jakákoliv další vylepšení se už nedělají.

Z hlediska vývoje lze stabilizaci produktu rozdělit do fází viz obr. 2.7. Akceptační testy obvykle probíhají od bety.

Právě akceptačními testy se primárně zabývá aplikace TestRunner, které se budu dále věnovat v kapitole 3.



Obrázek 2.7: Stabilizace produktu [8].

Pre-alpha

Předtím, než může produkt postoupit do alpha verze, je třeba ověřit, zdali jsou opraveny všechny nahlášené chyby. Všechny testovací případy jsou sdruženy do skupin. Z těch jsou dále vytvářeny protokoly, kde je jednoznačně vidět, zdali produkt prošel či nikoliv. Pokud je produkt vyvíjen pro velké množství platforem, stává se kompletní testování nereálným. Můžeme proto testy na některých z nich omezit.

Alpha

V této fázi je produkt připraven k prvnímu použití, ovšem stále může být velice nestabilní. Pády aplikace, případně ztráta dat, nemusí být neobvyklé. Proto je využíván pouze zaměstnanci společnosti. Veřejnosti je zpravidla nepřístupný.

Beta

Prototyp finálního produktu. Ještě stále má mnoho chyb a může mít problémy s výkonností. Je přístupný omezenému okruhu obchodních partnerů nebo veřejnosti. Tím se z něj stává první verze produktu dostupná mimo vývojovou firmu. Nespornou výhodou je mnohem větší základna testerů. Ta je schopná najít to, co relativně malý testovací tým neodhalí.

Release candidate

Release candidate (RC) je beta verze, která má potenciál být finálním produktem, a pokud se neobjeví výrazná chyba, tak se jím také stane. RC už bývá přístupný široké veřejnosti. Často tak slouží jako reklamní produkt vábící potenciální zákazníky.

2.6 Chyby

Hlavním úkolem testera je hledat chyby a je tedy úspěšný pokud chybu nalezne. U chyby nalezené při vývoji softwaru hodnotíme závažnost a prioritu.

2.6.1 Závažnost

Typ závažnosti je definován testerem na základě důležitosti postižené funkcionality. Odráží, jak kritická je chyba pro celý produkt.

2.6.2 Priorita

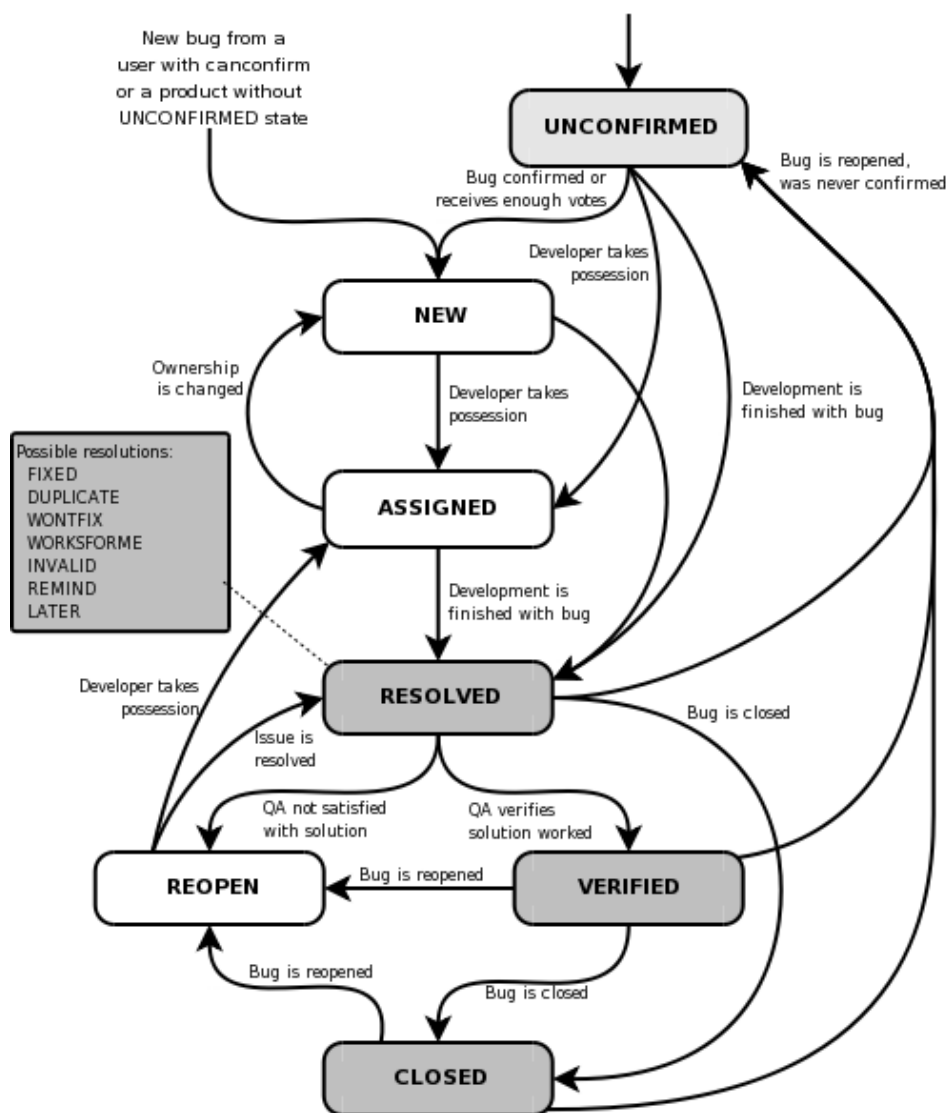
Priorita určuje, jak rychle má být chyba opravena. Je úzce spojena s plánováním řešení problémů. Často je založena na požadavcích a přáních zákazníků. Čím vyšší je priorita, tím rychlejší je oprava chyby.



Obrázek 2.8: Priorita vs závažnost chyby.

2.6.3 Životní cyklus chyby

Každá chyba prochází za svůj život různými stavy. Tyto stavy přehledně znázorňuje následující obrázek převzatý z dokumentace k Bugzille [3] ke kterému není potřeba žádný komentář.



Obrázek 2.9: Životní cyklus chyby podle Bugzilly [3].

3 TestRunner

3.1 Obecně

Všechny teoretické jsou nyní vysvětleny, pojďme se tedy podívat přímo na aplikaci TestRunner. TestRunner pokrývá základní potřeby testování v QA Kerio Technologies s.r.o.. Byl vytvořen pro tyto účely:

1. Správu testovacích případů a testů.
2. Poskytnout stavy a výsledky testů celé firmě.
3. Zefektivnit testovací proces.

První verze byla účelově vytvořena přímo firmou Kerio Technologies s.r.o. a to bez předešlých dokumentů. Později byl TestRunner upravován podle požadavků QA. Nejdříve za pomoci vlastních zaměstnanců a později i studentů ZČU. Od jeho vytvoření uběhlo již několik let a za tu dobu kód postupně rostl a stal se špatně udržovatelným. Zároveň z předchozích verzí:

1. Zůstaly chyby, které je nutné opravit.
2. Určité funkce nejsou nadále potřeba.
3. Jiné funkce je naopak potřeba implementovat.

Mým úkolem bude:

1. Implementace požadovaných funkcí.
2. Odstranění přebytečných funkcí.
3. Oprava chyb.

3.2 Popis funkčnosti

Základní popis funkcionality TestRunneru:

Pro čtenáře neznalého funkcionality TestRunneru doporučuji nejdříve důkladné prostudování uživatelské příručky uvedené v přílohách. Příručka byla vytvořena již dříve pro potřeby QA Kerio Technologies s.r.o.. a v rámci této práce prošla kompletní revizí.

3.3 Specifikace požadavků

V následujících bodech se budu blíže věnovat konkrétním požadavkům na mojí práci.

3.3.1 Dashboard

- Na úvodní stránce TestRunneru přidejte grafický přehled aktuálně běžících testů.
- V přehledu bude vidět, kolik práce je z každého testu hotovo a kolik zbývá.
- Z přehledu se bude možné kliknutím dostat na patřičný test.

3.3.2 Redukce práv

- Současný TestRunner má několik rolí: 'guest', 'tester', 'autotester', 'translator', 'project manager', 'administrator'.
- Cílem je zredukovat práva na: 'guest', 'tester', 'administrátor'.
- Role 'guest' bude jako nepřihlášený uživatel.

- Role 'tester' převezme veškerá práva pro přihlášené uživatele mimo administrátora.

3.3.3 Závislost testovacích případů na platformě

- Některé testovací případy nedávají na některých platformách smysl - měly by být svázány s konkrétní platformou.
- Vytvořit strom závislostí testovacích případů na platformách.

3.3.4 Vyhledávání

- Do pohledu 'Create or Edit Test Cases' dodělat stejné vyhledávání, jako už je hotové v pohledu 'Test Case Description'.
- Pokud to půjde s minimálními náklady, tak totéž i do pohledů 'Auto-Test Description' a 'Translate Test Case Description'.

3.3.5 Dvojjazyčnost

- Odstranění veškeré funkcionality česko-anglické dvojjazyčnosti, ponechána bude pouze angličtina.
- Aktuální popisy testů musí být zachovány a budou sloučeny do jednoho popisu.

3.3.6 Proměnlivá velikost levého panelu

- Levý panel by měl mít volitelnou velikost.
- Přidat možnost panel kompletně skrýt.

3.3.7 Bugfix

- Položka 'Title' (název TestCase) je funkčně omezena na cca 160 znaků, po překročení přestane fungovat navigační strom v pohledu 'Create or Edit Test Cases'.
- Není vidět, kdo nastavil 'Won't test'.
- Udělat součty průměrných časů na úrovni kapitol.
- Tlačítko 'Won't test' je vyšedlé, pokud už byl testovací případ testován. Tlačítko ale funguje.
- Pokud je testovací případ ve stavu 'Won't test', tak jsou vyšedlá tlačítka 'Send & Close' a 'Add bug'. V tomto případě tlačítka nefungují vůbec.
- V případě, že je některý testovací případ ve stavu 'Won't test', nemělo by to mít vliv na stav, který je pro celou kapitolu.
- Pokud je vyplněn pouze text v EN, není pak vidět v pohledu 'Test Case detail' popis žádný.
- Pokud nastavím 'Won't test', měl by se uložit také testerův komentář, např. proč to nechce testovat. To se neděje a komentář se neukládá.
- Manažer komentářů skáče na začátek stránky. Při 'Edit testcase' hlavní stránka přeskočí na začátek (a možná se znovu načte). Dává to smysl po úpravě kdy se může seznam komentářů změnit, ale ne když jdu teprve editovat komentář.
- Manažer komentářů se po úpravě komentářů vždy vynuluje do 'Show New', místo aby se obnovil a držel si nastavení.
- Zřejmě obrácená logika tlačítek nebo špatné pořadí akcí v manažeru komentářů, potvrzené 'Reject&Close' neudělá nic, odmítnuté to stejně provede.

3.4 Aktuální programové prostředí

Implementace TestRunneru běží na virtualizovaném linuxovém systému CentOS ve verzi 5.5. Virtualizaci zajišťuje komplexní řešení od firmy VMware, Inc. K samotnému běhu aplikace jsou pak nutné tyto součásti:

- Firebird ve verzi 1.5 [4] [11]
- PHP ve verzi 5.1.6 [9] [1]
- Plugin InterBase pro PHP ve verzi 15

Jako externí zdroje na jiných virtuálních strojích jsou využívány:

- Interní databáze Bugzilly [3], ke které jsem vzhledem k citlivé povaze informací neměl přístup.
- Autentizační server Kerberos (viz [6]), ke kterému jsem přístup také neměl.

Vzhledem k tomu, že všechny součásti byly v době psaní této práce již poměrně zastaralé, navrhl jsem kompletní upgrade systému. I přes firemní politiku "Co funguje, do toho se nesahá"mi byl upgrade schválen.

3.5 Přechod na nový operační systém

Nejdříve jsem systém zkoušel instalovat na svém notebooku. Protože k původnímu operačnímu systému CentOS nebyly výhrady, byl využit znovu ve verzi 6.4. Bohužel se mi díky tomuto systému dlouhou dobu nedařilo propojit PHP server (apache) s databází Firebird. Problém byl konkrétně ve funkci 'Security-Enhanced Linux', která neznala port, který databáze používala ke komunikaci. Bug byl sice nahlášen, ale opraven pro verzi 6.4 bohužel nebyl. Po odhalení této chyby zbývalo dočasně odpojit Kerberos. Na základě pozdější domluvy byl Kerberos odpojen natrvalo a místo něho využita autentizace přes LDAP (viz [7]). Takto vytvořený virtuální stroj byl poté kompletně přenesen do již existující infrastruktury oddělení QA.

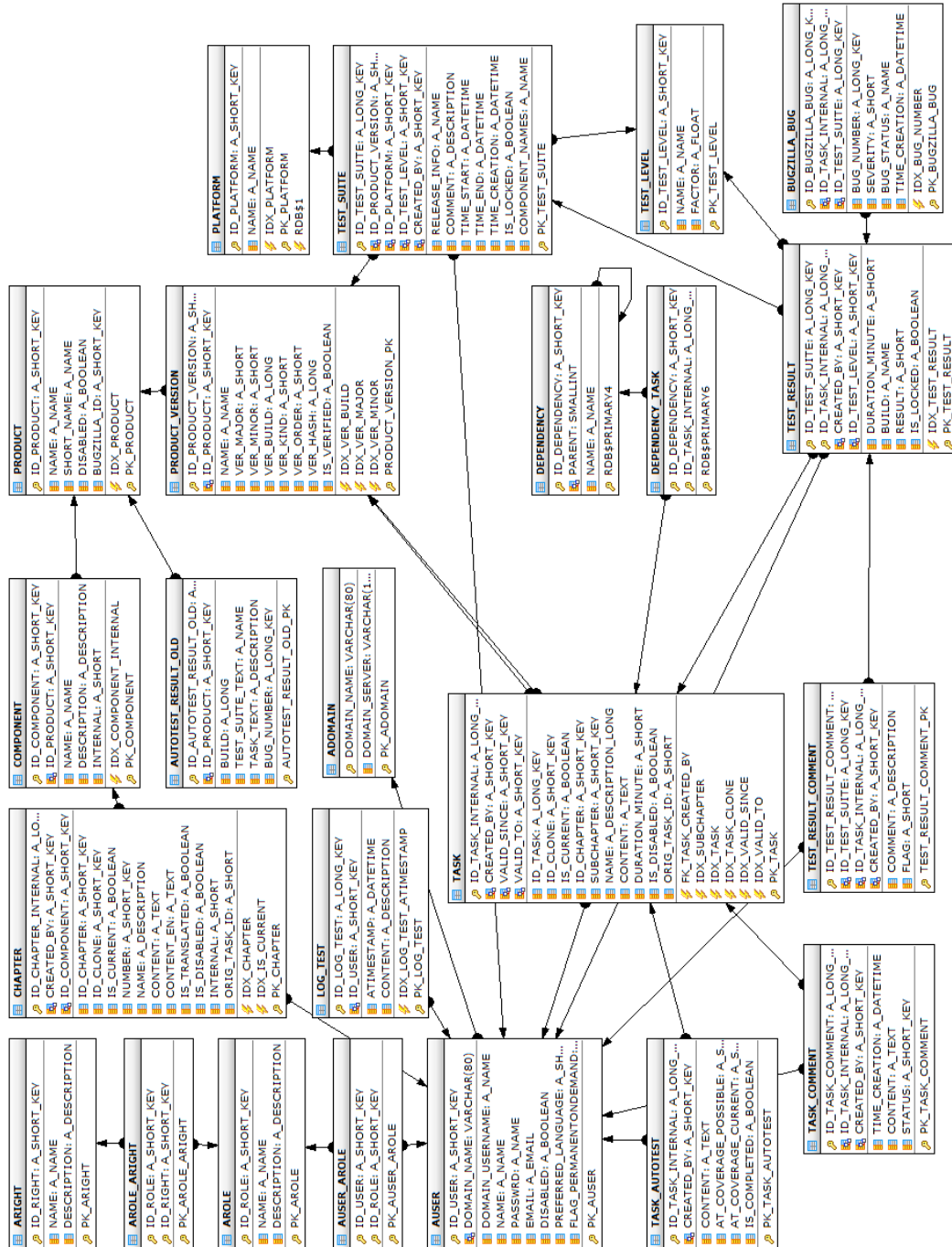
3.6 Analýza

Nejtěžší částí práce bylo proniknout do zdrojových kódů. Situaci ještě zhoršoval fakt, že na TestRunneru vždy s odstupem času pracoval někdo jiný. Tím se v projektu vystřídalo velké množství různých přístupů a řešení.

Program používá třívrstvou architekturu pro oddělení dat, aplikační logiky a grafického uživatelského rozhraní.

3.6.1 Datová vrstva

Pro orientaci v datové vrstvě, jsem vypracoval DB model viz obr. 3.1. Tento model detailně zachycuje všechny tabulky i vazby.



Obrázek 3.1: DB model.

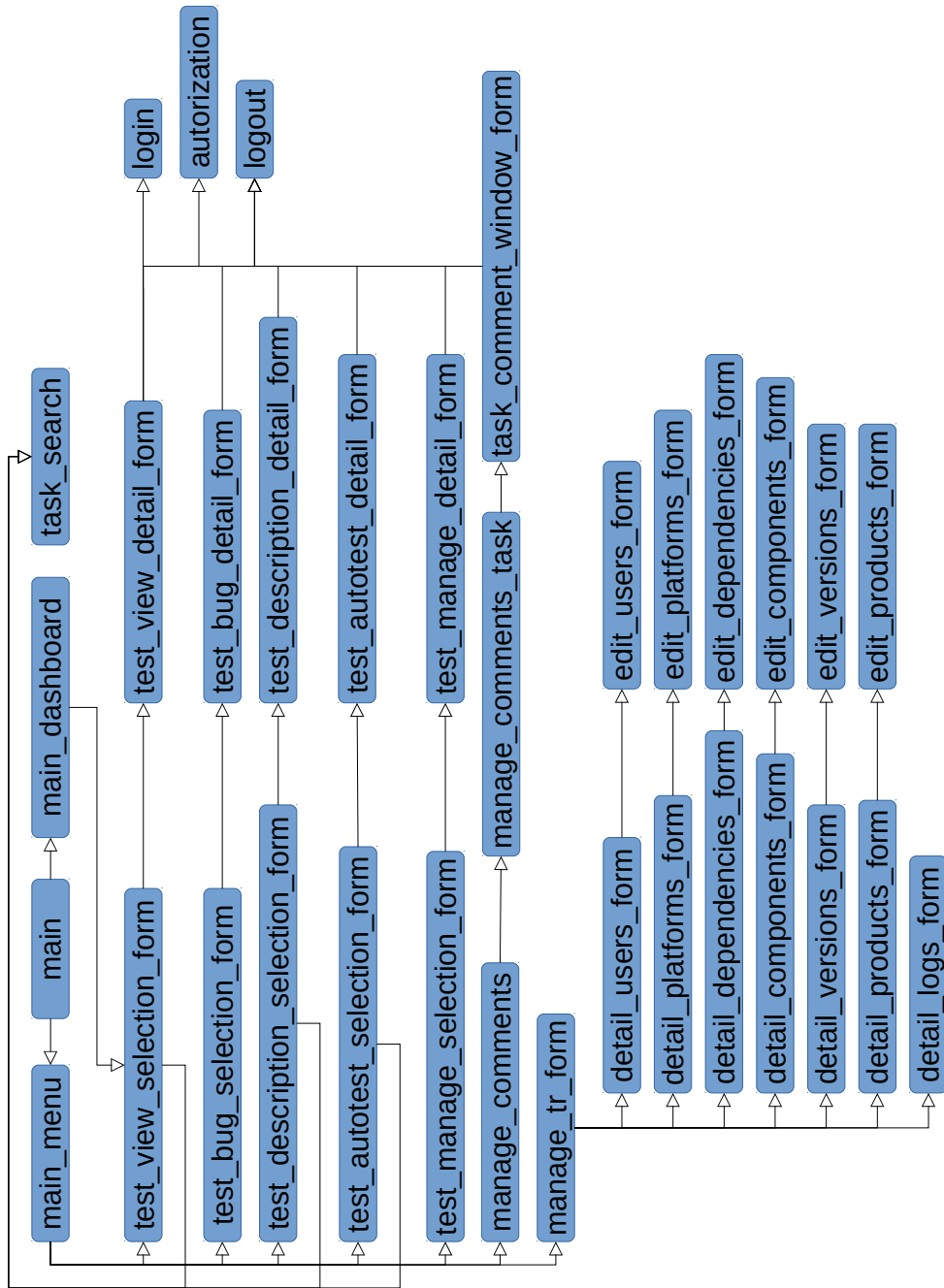
3.6.2 Aplikační vrstva

Aplikační vrstvu jsem chtěl nejdříve zmapovat pomocí diagramu tříd. To se ukázalo, pro potřeby této práce, jako velice nepřehledné řešení, a to hlavně z důvodu velkého množství tříd(100+). Proto jsem vypracoval zjednodušený diagram tříd, popisující pouze ty nejvýznamnější viz obr. 3.2.

- **Třída main.php** Vstupní brána pro uživatele a skládá se z main_menu.php a main_dashboard.php
- **Třída main_dashboard.php** Zobrazuje přehled probíhajících testů a kolik je z každého testu hotovo. Umožňuje dostat se kliknutím na vybraný test. Testy lze také řadit podle jména, zbývajících času nebo zbývajících procent.
- **Třída main_menu.php** Rozcestník pro všechny funkce TestRunneru.
- **Třída login.php** Zobrazení login dialogu a případných chybových hlášek.
- **Třída logout.php** Odhlášení uživatele.
- **Třída authorization.php** Ověření práv uživatele.
- **Třída task_search.php** Rychlé vyhledání testu.
- **Třída test_view_selection_form.php** Hlavní obrazovka pro 'Test Results' a 'Create Tests, Perform Tests'.
- **Třída test_view_detail_form.php** Detailní výběr pro 'Test Results' a 'Create Tests, Perform Tests'.
- **Třída test_bug_selection_form.php** Hlavní obrazovka pro 'Bug Statistics'.
- **Třída test_bug_detail_form.php** Detailní výběr pro 'Bug Statistics'.

- **Třída `test_description_selection_form.php`** Hlavní obrazovka pro 'Test Case Description'.
- **Třída `test_description_detail_form.php`** Detailní výběr pro 'Test Case Description'.
- **Třída `test_autotest_selection_form.php`** Hlavní obrazovka pro 'Autotest Description'.
- **Třída `test_autotest_detail_form.php`** Detailní výběr pro 'Autotest Description'.
- **Třída `test_manage_selection_form.php`** Hlavní obrazovka pro 'Create or Edit Test Cases'.
- **Třída `test_manage_detail_form.php`** Detailní výběr pro 'Create or Edit Test Cases'.
- **Třída `manage_comments.php`** Hlavní obrazovka pro 'Manage Comments'.
- **Třída `manage_comments_task.php`**
- **Třída `task_comment_window_form.php`**
- **Třída `manage_tr_form.php`** Hlavní obrazovka pro 'Manage TestRunner'.
- **Třída `detail_users_form.php`** Hlavní obrazovka pro 'Users'.
- **Třída `edit_users_form.php`** Editace položek 'Users'.
- **Třída `detail_platforms_form.php`** Hlavní obrazovka pro 'Platforms'.
- **Třída `edit_platforms_form.php`** Editace položek 'Platforms'.
- **Třída `detail_dependencies_form.php`** Hlavní obrazovka pro 'Dependencies'.

- **Třída `edit_dependencies_form.php`** Editace položek 'Dependencies'.
- **Třída `detail_components_form.php`** Hlavní obrazovka pro 'Components'.
- **Třída `edit_components_form.php`** Editace položek 'Components'.
- **Třída `detail_versions_form.php`** Hlavní obrazovka pro 'Versions'.
- **Třída `edit_versions_form.php`** Editace položek 'Versions'.
- **Třída `detail_products_form.php`** Hlavní obrazovka pro 'Products'.
- **Třída `edit_products_form.php`** Editace položek 'Products'.
- **Třída `detail_logs_form.php`** Hlavní obrazovka pro 'Logs'.



Obrázek 3.2: Zjednodušený popis tříd

3.7 Oprava chyb

V této části se jednalo především o chyby, které vznikly v nové funkcionalitě poslední verze. Všechny byly postupně opraveny. Jejich opravou jsem navíc získal ještě hlubší znalost kódu. Nyní uvedu stručný přehled:

3.7.1 Tlačítka Won't test a Reset

Tato tlačítka jsou součástí detailu testovacího případu. Pro bližší popis doporučuji prostudování kapitoly 6 v uživatelské příručce.

1. Ukládají, kdo je nastavil.
2. Ukládají testerův komentář.
3. Nefungují, když jsou vyšedlá.
4. Jejich funkci správně indikují i ikony pro stav celé kapitoly.

3.7.2 Manažer komentářů

Popis manažera komentářů je uveden v uživatelské příručce kapitola 12.

1. Nenačítá se znovu při kliknutí na edit.
2. Při změně se obnoví, ale drží zvolená nastavení.
3. Stisk tlačítka tlačítko nikam neposouvá.
4. Nefunkční potvrzující dialog jsem odstranil.

3.7.3 Ostatní

1. Položce Title v test casu jsem zvětšil velikost.
2. Ikony jsem upravil na transparentní.
3. Vyhledávání už nemá zbytečně agresivní červenou barvu při neúspěchu.
4. Upravil jsem některé dotazy do Firebirdu kvůli přechodu na novou verzi.

3.8 Odebrání nadbytečných funkcí

V rámci zajištění budoucí vitality TestRunneru bylo třeba odebrat nadbytečné funkce.

3.8.1 Zrušení dvojjazyčnosti

Postupem času se pro zaměstnance společnosti Kerio stala angličtina standardem. V tu chvíli se ze zajišťování dvojjazyčnosti stala zbytečná režie, která musí být odstraněna. Bohužel byla ze své podstaty v programu poměrně pevně zakořeněna a její odstranění vyžadovalo rozsáhlý průzkum kódu i důkladné testování.

Skoro na každém z příložených obrázků můžeme nalézt některou ze součástí této funkcionality. Konkrétně se jedná o položku Translate Test Case Description viz obr. 3.3, která zajišťovala funkcionality pro překladatele. Dále se jedná o vlaječky pro přepínání popisů viz obr. 3.11 a 3.5. Posledním ilustračním příkladem je položka translated a dvojí popis test casu viz obr. 3.9. Za zmínku ještě například stojí zobrazování ikon u nepřeložených testovacích případů přímo ve stromu v levé části, které na obrázcích není vidět nebo také jazykové preference pro každého uživatele.

3.8.2 Redukce uživatelských rolí

Vzhledem k tomu, že je většinou ke každému týmu přiřazen právě jeden tester, který zajišťuje kompletní servis okolo testů, staly se specializovanější role nepotřebné. Naopak vznikla potřeba jediné role pro přihlášeného uživatele, která byla dočasně řešena přidělením administrátorských práv. Bylo tedy rozhodnuto zredukovat původní role:

- Guest
- Tester
- Autotester
- Translator
- Project Manager
- Administrátor

a to konkrétně na:

- Guest
- Tester
- Administrátor

Guest je nepřihlášený uživatel, Tester sloučí veškeré potřeby pro přihlášeného uživatele až na samotnou administraci, kterou zajistí administrátor.

3.9 Nové funkce

3.9.1 DashBoard

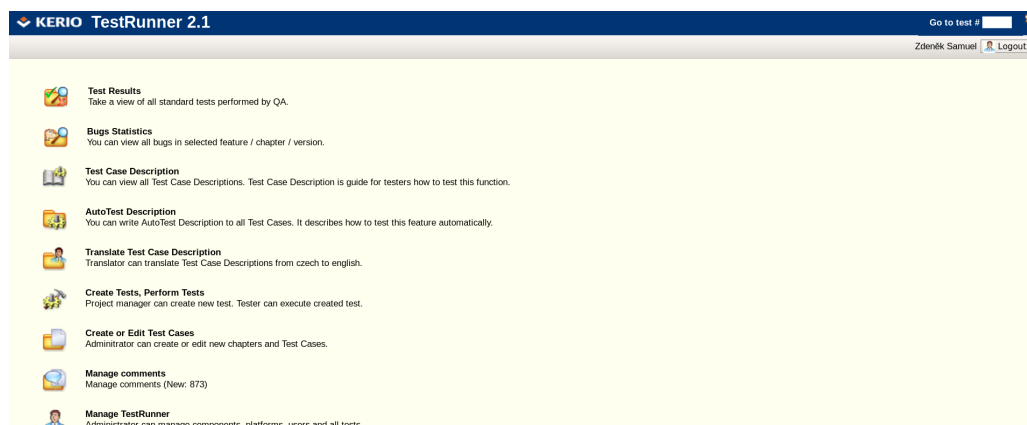
Dashboard zobrazuje rychlý přehled aktuálně běžících testů. Měl zrychlit orientaci v TestRunneru, protože ve většině případů nás zajímají právě aktuálně běžící testy produktu. Uživatel tedy bude ušetřen zdlouhavého proházení stromem.

Analytická část

Stránku jsem rozdělil na dvě části, aby ji bylo možné lépe organizovat. Dále jsem experimentálně zjistil nejlepší rozvržení stránky.

Implementační část

K rozdělení obrazovky posloužil tzv. Split pane, který byl součástí JQuery UI Layout. Díky tomu je možné obě části libovolně zvětšovat či zmenšovat. DashBoard obsahuje odkazy na jednotlivé testy, a nabízí tak přihlášenému uživateli možnost rychlého přesunu na kterýkoliv test. Pro lepší orientaci jsem implementaci doplnil o řazení podle sloupců (kliknutím na hlavičku) a barevné odlišení sudých a lichých řádek. Obě tyto funkce zajišťuje JQuery. Výsledek implementace viz obr. 3.4.



Obrázek 3.3: Hlavní stránka - původní.



Obrázek 3.4: Hlavní stránka - nová.

3.9.2 Vyhledávání

Analytická část

Vyhledávání bylo implementováno z předchozí verze v pohledu 'Test Case Description'. Zadavatel požadoval doplnění do částí 'AutoTest Description' a 'Create or Edit Test Cases'.

Implementační část

Vyhledávání bylo připraveno poměrně obecně a jeho doplnění nebylo příliš náročné. Ukázkou včetně příkladu lze nalézt v uživatelské příručce.

3.9.3 Součty průměrných časů testů pro kapitoly

Hlavním důvodem pro implementaci byla možnost lepší organizace času testerů. V praxi by nyní měl tester lépe odhadnout, který test má zvolit, když chce za hodinu odejít domů. Funkce viz obr.

Analytická část

Nejdříve bylo nutné rozhodnout, jakým způsobem budeme čas měřit. Testrunner využívá klasický průměr, modus a medián. Po dohodě se zadavatelem byl vybrán klasický průměr. Dále bylo nutné vybrat správné místo zobrazení. Hlavně kvůli nedostatku místa bylo zvoleno zobrazení v tooltipu.

Implementační část

Vlastní implementace spočívala v přidání ikonky hodin ke každému testu i celým kapitolám. Po najetí kurzorem myši na ikonku je proveden dotaz do databáze. Průměr testu je z posledních 10 časů. Pro kapitoly jsou tyto časy sečteny viz obr. 3.6

3.9.4 Proměnlivá velikost levého panelu

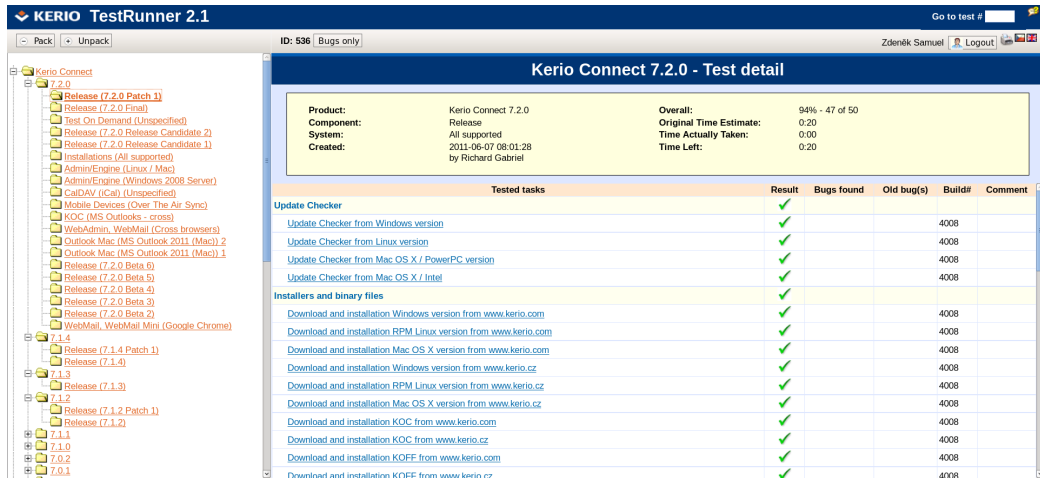
Levý panel sloužící k vyhledávání je po většinu času nepotřebný. Proto je požadována možnost zmenšení nebo skrytí tohoto panelu.

Implementační část

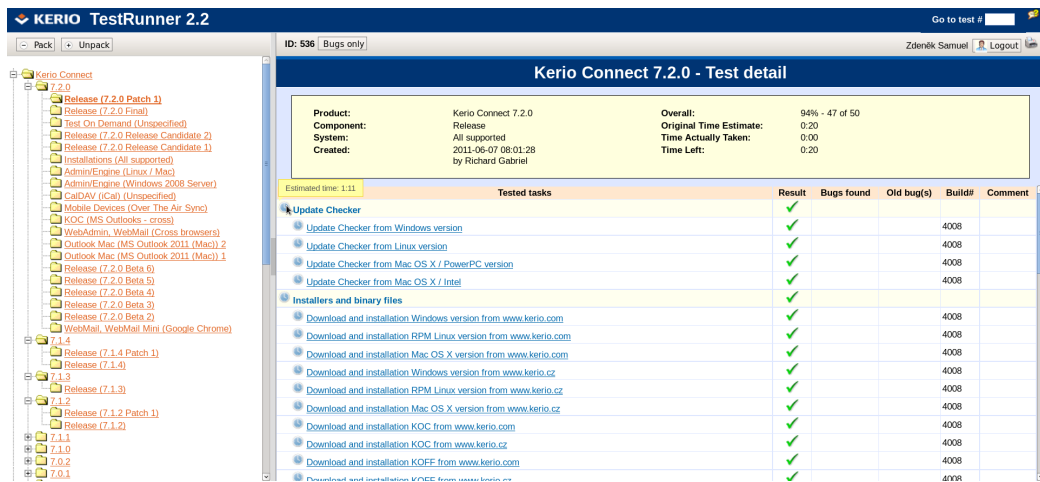
Implementace proběhla pomocí frameworku jQuery [5], který byl v projektu již dříve použit, doplněného o JQuery UI Layout. JQuery UI Layout jsem zvolil, protože plně vyhovuje požadavkům a doplňuje celkový koncept modulů JQuery. Umožňuje levý panel zmenšit či úplně schovat dle aktuální potřeby viz obr. 3.6 a 3.10.

3.9.5 Ověřování přes LDAP

V závislosti s přechodem na novější operační systém, bylo rozhodnuto přejít na firmou více využívanou autentizaci přes LDAP. Ta je pro implementaci v PHP poměrně dobře zdokumentována. K realizaci je potřeba doinstalovat modul PHP-LDAP.



Obrázek 3.5: Detail test suite - původní.



Obrázek 3.6: Detail test suite - nový.

3.9.6 Vazba testovacích případů s platformou

Analytická část

Asi nejsložitější částí bylo provázání jednotlivých testovacích případů s platformou. Prvním krokem byla volba vhodné implementace. Dospěl jsem k reprezentaci stromem, viz 3.7. Strom byl oproti grafu jednoznačný a lépe implementovatelný.

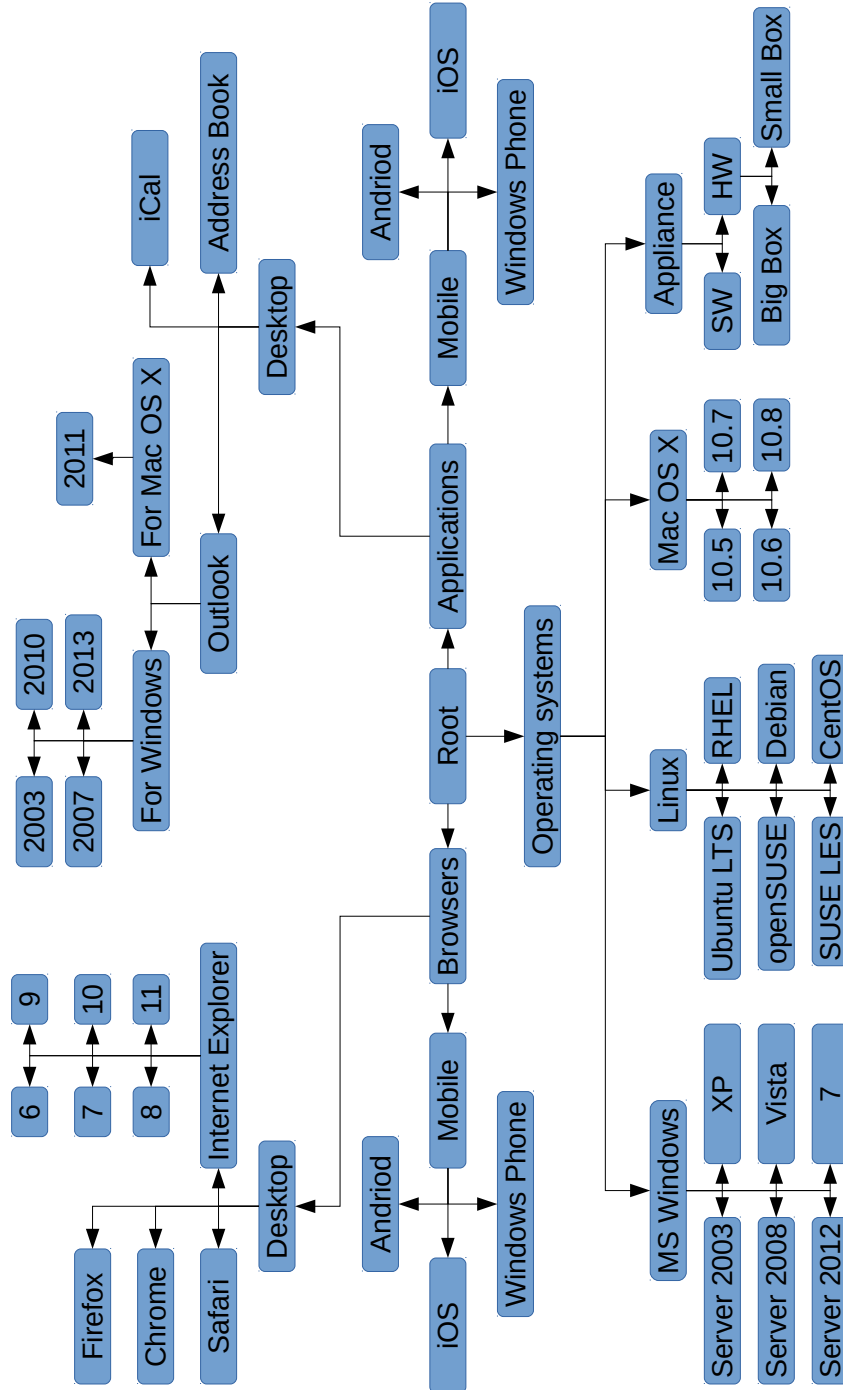
Výsledný strom nejevil sebemenší známky symetričnosti nebo pravidelnosti. Navíc do budoucna hrozilo, že by se mohl ještě výrazně měnit. Proto jsem dospěl k názoru, že nejlepší bude implementace pomocí rekurze a to hlavně kvůli nepravidelnosti výsledného stromu. Protože se nepředpokládá výrazný růst stromu, neměla by být rekurze překážkou.

Implementační část

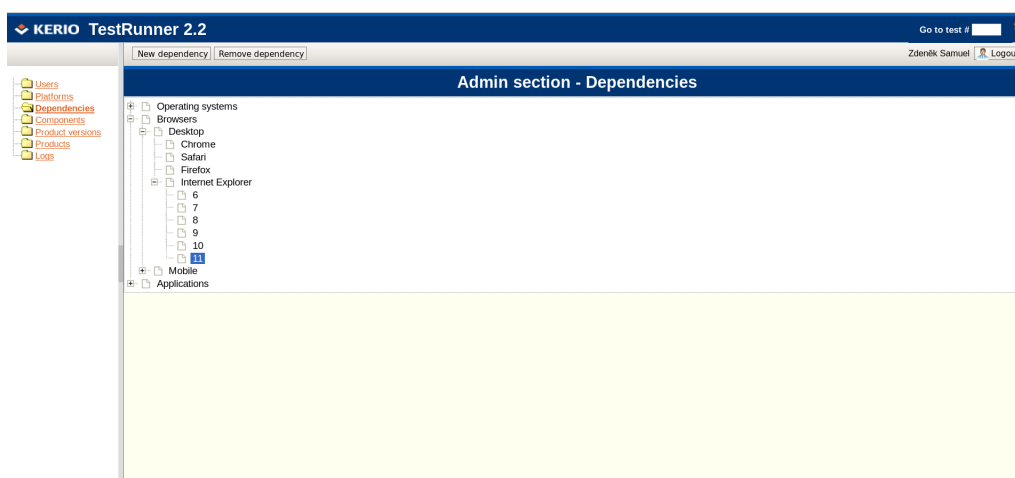
V praxi bude každá platforma jedním řádkem v tabulce databáze a bude si uchovávat pouze jméno, identifikační číslo své a svého rodiče. Každý testovací případ může mít více závislostí na platformě a zároveň každá závislost může patřit k více testovacím případům. Jedná se tedy o vazbu M:N, kde bude dále nutná rozkladová tabulka.

Aby bylo možné strom v budoucnu nadále upravovat, musel jsem do managementu TestRuneru přidat položku pro správu stromu viz obr. 3.8. První závislosti budou vloženy do databáze již při tvorbě tabulky, další pak podle potřeby. Přiřazení závislostí k jednotlivým testovacím případům bude nutné provést ručně. To je možné jak pro nové, tak i aktivní testovací případy. Ukázka přidávání a zobrazení jednotlivých závislostí viz obr. 3.10.

Důvodem implementace této funkce je především lepší zaměření testů.



Obrázek 3.7: Strom platformem.



Obrázek 3.8: Tvorba stromu závislostí.

3.9.7 Tooltips

Jak už bylo zmíněno v předchozí části, okno pro tvorbu test suite je neúměrně pomalé (u mě na počítači pravidelně více než 10s). Jedním z důvodů bylo načítání všech tooltipů při otevření okna. Dalším problémem původní implementace byla nefunkčnost při větším množství textu.

Analytická část

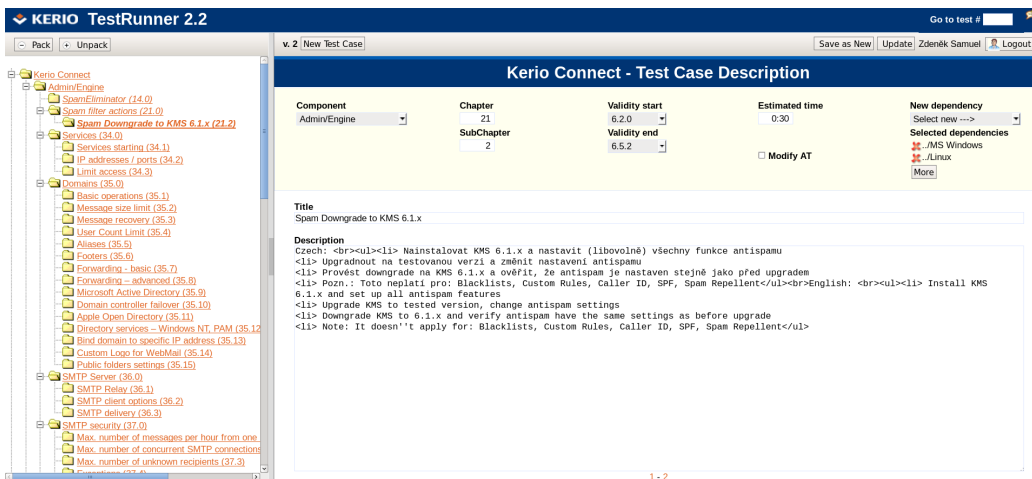
Zběžným průzkumem mezi testery jsem zjistil, že je velká část z nich tooltipy vůbec nepoužívá. Rozhodl jsem se je tedy načítat až ve chvíli, kdy o to bude někdo stát, konkrétně až při najetí myši. Původní implementace zahrnovala využití špatně rozšiřitelné a neudržované knihovny. Z těchto důvodů jsem se rozhodl ji nahradit knihovnou, která bude těmto potřebám více vyhovovat.

Implementační část

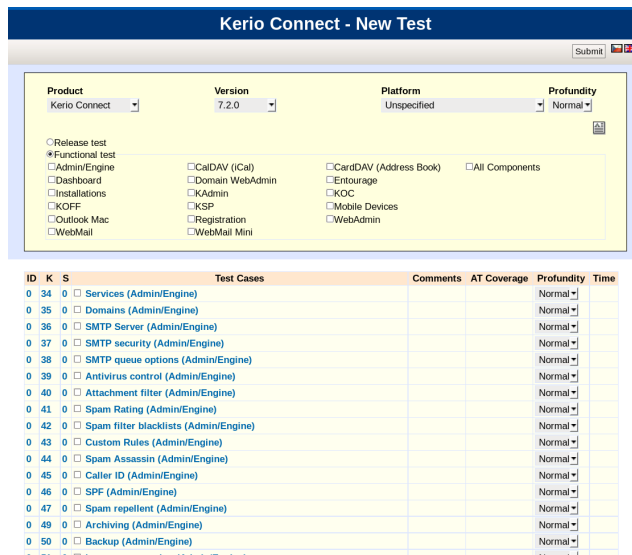
Pro samotnou implementaci jsem zvolil knihovnu qTip2 k frameworku jQuery, který byl v projektu již dříve použit. Tato knihovna zvládla pokrýt původní funkčnost, načítat tooltips na požádání. Nemalou výhodou je také výborně zpracovaný manuál. Původní implementace tooltipů byla z projektu TestRunneru odstraněna.



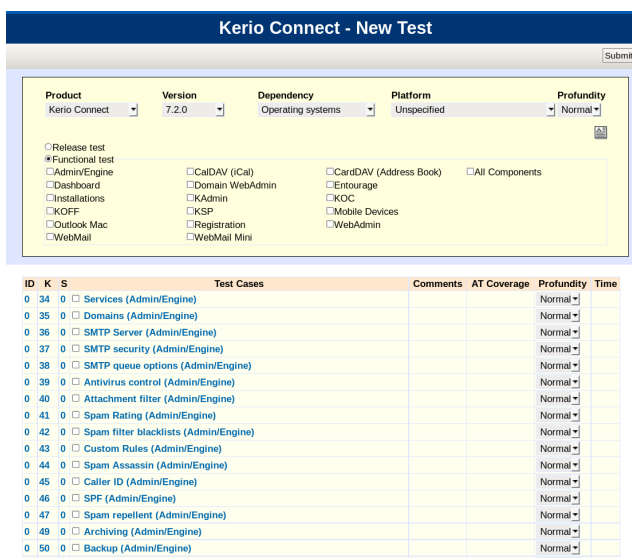
Obrázek 3.9: Editace test case - původní.



Obrázek 3.10: Editace test case - nová.



Obrázek 3.11: Tvorba test suite - původní.



Obrázek 3.12: Tvorba test suite - nová.

4 Testy TestRunneru

4.1 Automatické testování

Automatické testy jsem využil pouze v menším rozsahu. Mou volbou byl framework Selenium. Rozhodující byly mé dřívější pozitivní zkušenosti a také vývoj v mnou preferovaném jazyce Java. Jako neocenitelný pomocník se zde ukázala kniha Selenium Testing Tools Cookbook [14].

Testy pokrývají malou oblast nejčastěji využívaných a tím pádem kritických funkcí aplikace. Spadají do kategorie Smoke testů, které mají za úkol odhalit zásadní chyby.

Kritické funkce, vhodné pro pokrytí automatickými testy, jsou následující:

1. Login/Logout .
2. Kontrola práv pro nepřihlášeného uživatele, testera a administrátora.
3. Vytvoření testovacího případu.

4.2 Manuální testování

První série testů probíhala již v rané fázi vývoje, kdy jsem testoval, zdali funkcionality odpovídá specifikaci. Pro testování jsem zvolil funkční testy.

Následně jsem funkcionality konzultoval se zadavatelem a členy QA. Pokud nebyl zadavatel spokojen, změnil jsem ji, upravil testy, znovu otestoval a předvedl. Ve chvíli, kdy byla funkcionality schválena, přešel jsem k další. Ukázka testovacího protokolu viz obr. 4.1.

Přibližně v polovině práce jsem přikročil k akceptačním testům aplikace. Testovací scénáře jsem sestavoval na základě konzultací s jednotlivými členy QA. Důraz byl kladen na otestování TestRunneru pro časté úkony. Následně byl TestRunner uveden do ostrého provozu.

Druhá polovina práce měla totožný průběh.

| Číslo testu | Popis testované situace | Kroky k provedení provedení | Úspěšné |
|-------------|------------------------------------|--|---------|
| 1. | Přihlášení uživatele | <ol style="list-style-type: none"> 1. Kliknutí na tlačítko login. 2. Zadání přihlašovacích údajů. 3. Ověření změny tlačítka na logout. 4. Ověření zobrazení aktuálně přihlášeného uživatele. | Ano |
| 2. | Odhlášení uživatele | <ol style="list-style-type: none"> 1. Kliknutí na tlačítko logout. 2. Ověření změny tlačítka na login. 3. Není zobrazeno jméno žádného uživatele. | Ano |
| 3. | Ověření práv - Nepřihlášený | <ol style="list-style-type: none"> 1. Povolen přístup do Tests Results, Bug Statistics a Test Case Description. 2. Zakázán přístup do Autotest Description, Create tests, Perform, Tests, Create or Edit Test Cases, Manage Comments a Manage TestRunner. | Ano |
| 4. | Ověření práv - Tester | <ol style="list-style-type: none"> 1. Povolen přístup do Tests Results, Bug Statistics, Test Case Description, Autotest Description, Create tests, Perform, Tests, Create or Edit Test Cases a Manage Comments. 2. Zakázán přístup do Manage TestRunner. 3. V Tests Results lze editovat, mazat odemykat/zamykat testy. | Ano |
| 5. | Ověření práv - Administrátor | <ol style="list-style-type: none"> 1. Povolen přístup všude. 2. V Tests Results lze editovat, mazat odemykat/zamykat testy. | Ano |
| 6. | DashBoard | <ol style="list-style-type: none"> 1. Korektní zobrazení na hlavní stránce. 2. Kliknutím na test, se lze dostat na jeho detail. 3. V sloupci progress, po najetí na procentuální hodnotu, zobrazení zobrazení tooltipu s konkrétním počtem testů. 4. Řazení testů po kliknutí na hlavičku sloupce. | Ano |
| 7. | Administrace - Uživatelé | <ol style="list-style-type: none"> 1. Vytvoření nového uživatele v Manage TestRunner – Users. 2. Ověření správného jména, emailu, domény a přidělených práv. | Ano |
| 8. | Administrace - Platformy | <ol style="list-style-type: none"> 1. Vytvoření nové platformy v Manage TestRunner – Platforms. 2. Ověření správného jména a možnosti smazat platformu. 3. V Create Tests, Perform Tests vytvořit test s touto platformou. 4. Ověřit, že již nelze smazat platformu. | Ano |

Obrázek 4.1: Ukázka z testovacího protokolu.

5 Závěr

V rámci zadání jsem se detailně seznámil s principy testování, fungováním QA oddělení společnosti Kerio Technologies s.r.o. a doplnil si vzdělání v oblastech virtualizace, operačního systému Linux, SQL databází, PHP, JavaScriptu, HTML, CSS, typografického systému L^AT_EX a mnohých dalších. Dále jsem se seznámil s nástrojem TestRunner a požadavky na jeho změnu.

Připravil jsem nové prostředí pro běh TestRunneru. Na novém OS jsem zprovoznil novou verzi databáze FireBird, PHP a pluginu InterBase. Seznámil jsem se s databázovým modelem a zdrojovými kódy. Odstranil jsem všechny chyby, které byly zadány. Odebral jsem řadu nepotřebných funkcí a následně úspěšně implementoval nové funkce. Celou práci jsem důkladně otestoval a zdrojové kódy včetně konverzních skriptů k databázi předal zadavateli.

V práci jsem si zkusil od všeho něco. Jednalo se zatím o největší projekt, na kterém jsem pracoval. Přesto si myslím, že se mi všechny body zadání povedlo úspěšně splnit.

Přehled zkratk

| | |
|------|---------------------------------------|
| AJAX | Asynchronous JavaScript and XML |
| CSS | Cascading Style Sheets |
| QA | Quality Assurance |
| HTML | HyperText Markup Language |
| LDAP | Lightweight Directory Access Protocol |
| PHP | Hypertext Preprocessor |
| RC | Release Candidate |
| SQL | Structured Query Language |
| UI | User Interface |

Seznam obrázků

| | | |
|-----|---|----|
| 2.1 | Optimální hladina v softwarovém testování.[15] | 2 |
| 2.2 | Náklady na opravu chyby v průběhu výrobního cyklu [16]. | 3 |
| 2.3 | Rozdíl v plánování mezi klasickým a agilním přístupem [13]. | 4 |
| 2.4 | Jak funguje Scrum [10]. | 6 |
| 2.5 | Příklad scrumboardu v Kerio Technologies [16]. | 8 |
| 2.6 | Black-box testování [2]. | 12 |
| 2.7 | Stabilizace produktu [8]. | 16 |
| 2.8 | Priorita vs závažnost chyby. | 18 |
| 2.9 | Životní cyklus chyby podle Bugzilly [3]. | 19 |
| 3.1 | DB model. | 26 |
| 3.2 | Zjednodušený popis tříd | 30 |
| 3.3 | Hlavní stránka - původní. | 35 |
| 3.4 | Hlavní stránka - nová. | 35 |
| 3.5 | Detail test suite - původní. | 38 |

| | | |
|------|---|----|
| 3.6 | Detail test suite - nový. | 38 |
| 3.7 | Strom platforem. | 40 |
| 3.8 | Tvorba stromu závislostí. | 41 |
| 3.9 | Editace test case - původní. | 42 |
| 3.10 | Editace test case - nová. | 42 |
| 3.11 | Tvorba test suite - původní. | 43 |
| 3.12 | Tvorba test suite - nová. | 43 |
| 4.1 | Ukázka z testovacího protokolu. | 46 |

Literatura

- [1] *Apache Tutorials* [online]. 2013. [cit. 2.8.2013]. Dostupné z: <http://httpd.apache.org/docs/2.0/misc/tutorials.html>.
- [2] *Black-box testing* [online]. 2012. [cit. 2.8.2013]. Dostupné z: http://en.wikipedia.org/wiki/Black-box_testing.
- [3] *The Bugzilla Guide - 4.2.6 Release* [online]. 2013. [cit. 2.8.2013]. Dostupné z: <http://www.bugzilla.org/docs/4.2/en/pdf/Bugzilla-Guide.pdf>.
- [4] *Firebird 2.5 Quick Start Guide* [online]. 2011. [cit. 2.8.2013]. Dostupné z: http://www.firebirdsql.org/file/documentation/reference_manuals/user_manuals/html/qsg25.html.
- [5] *jQuery API* [online]. 2013. [cit. 2.8.2013]. Dostupné z: <http://api.jquery.com/>.
- [6] *Kerberos Protocol Tutorial* [online]. 2007. [cit. 2.8.2013]. Dostupné z: <http://www.kerberos.org/software/tutorial.html>.
- [7] *Lightweight Directory Access Protocol* [online]. 2013. [cit. 2.8.2013]. Dostupné z: <http://php.net/manual/en/book.ldap.php>.
- [8] *Software release life cycle* [online]. 2012. [cit. 2.8.2013]. Dostupné z: http://en.wikipedia.org/wiki/Software_release_life_cycle.
- [9] *PHP Manual* [online]. 2013. [cit. 2.8.2013]. Dostupné z: <http://php.net/manual/en/index.php>.

-
- [10] *Introduction to Scrum* [online]. 2012. [cit. 2.8.2013]. Dostupné z: <http://www.mountangoatsoftware.com/presentations/an-introduction-to-scrum>.
- [11] *SQL Tutorial* [online]. 2013. [cit. 2.8.2013]. Dostupné z: <http://www.w3schools.com/sql/default.asp>.
- [12] *Software Testing Methods* [online]. 2014. [cit. 2.5.2014]. Dostupné z: http://www.tutorialspoint.com/software_testing/testing_methods.htm.
- [13] BUCHALCEVOVÁ, A. Agilní metodiky. In *Objekty 2002*. Praha: Česká zemědělská univerzita (ČZU), 2002. ISBN 80-213-0947-4.
- [14] GUNDECHA, U. *Selenium Testing Tools Cookbook*. Community experience distilled. Packt Publishing, Limited, 2012. Dostupné z: <http://books.google.cz/books?id=tTyeCLSH2Z8C>. ISBN 9781849515757.
- [15] PATTON, R. *Testování softwaru*. Programování: Pro každého uživatele. Computer Press, 2002. Dostupné z: <http://books.google.cz/books?id=m1nHAAAACAAJ>. ISBN 9788072266364.
- [16] SAMUEL, Z. Agile Testing. Prezentováno jako externí přednáška k předmětu ZSWI, 2010.

Přílohy

Uživatelský manuál

| | |
|--|----|
| 1. Titulní stránka..... | 3 |
| 1.1. Uživatelská práva | 3 |
| 1.2. Popis funkcí..... | 4 |
| 2. Seznam testů..... | 5 |
| 2.1. Strom testů..... | 5 |
| 2.2. Seznam testů..... | 5 |
| 2.3. Ostatní funkce | 6 |
| 3. Vytváření nebo editace testů..... | 7 |
| 3.1. Globální hodnoty | 7 |
| 3.2. Komponenty | 8 |
| 3.3. Výběr testovacích případů a jejich vlastností | 8 |
| 3.4. Test na žádost | 9 |
| 3.5. Ostatní..... | 9 |
| 4. Detail testu | 10 |
| 4.1. Hlavička..... | 11 |
| 4.2. Tasks Tested | 11 |
| 4.3. Výsledky testů | 11 |
| 4.4. Chyby | 12 |
| 4.5. Ostatní..... | 12 |
| 5. Seznam chyb..... | 13 |
| 6. Testovací formulář..... | 14 |
| 6.1. Hlavička..... | 14 |
| 6.2. Odeslání výsledku | 15 |
| 6.3. Vracení do stavu „netestováno“ | 15 |
| 6.4. Won't test..... | 15 |
| 6.5. Chyby | 15 |
| 6.6. Descriptions..... | 16 |
| 6.7. Ostatní..... | 16 |
| 7. Test Case Update | 17 |
| 8. Seznam testovacích případů..... | 18 |
| 8.1. Strom testovacích případů..... | 18 |
| 8.2. Test Case detail..... | 18 |

| | |
|--|----|
| 9. Vyhledávání v testovacích případech | 19 |
| 10. Manipulace s testovacími případy | 19 |
| 10.1. Strom testovacích postupů | 20 |
| 10.2. Řídící informace | 20 |
| 10.3. Editační pole | 21 |
| 10.4. Ukládání nově vytvořených testovacích případů..... | 21 |
| 10.5. Ukládání updatů testovacích případů (klony)..... | 21 |
| 11. Popis autotestu | 22 |
| 11.1. Strom testovacích případů | 22 |
| 11.2. Editační pole | 22 |
| 12. Manažer komentářů | 23 |
| 12.1. Schválení návrhu na update..... | 24 |
| 13. Admin sekce..... | 25 |
| 13.1. Uživatelé..... | 25 |
| 13.2. Platformy..... | 26 |
| 13.3. Závislosti na platformě..... | 26 |
| 13.4. Komponenty..... | 27 |
| 13.5. Produktové verze..... | 28 |
| 13.6. Produkty | 28 |
| 13.7. Logy | 29 |

1. Titulní stránka

| Test name | Remaining | Progress |
|--|-----------|----------|
| Kerio Connect - Release (7.2.0 Patch 1) | 00:20 h | 94 % |
| Kerio Connect - Release (7.2.0 Release Candidate 2) | 00:05 h | 88 % |
| Kerio Connect - Admin/Engine (Linux / Mac) 2 | 00:45 h | 98 % |
| Kerio Connect - CalDAV (iCal) (Unspecified) 3 | 00:00 h | 93 % |
| Kerio Connect - WebAdmin, WebMail (Cross browsers) | 00:10 h | 99 % |
| Kerio Connect - Outlook Mac (MS Outlook 2011 (Mac)) 1 | 00:00 h | 78 % |
| Kerio Operator - Release 7 | 00:00 h | 93 % |
| Kerio Connect - Release (7.2.0 Beta 6) | 00:20 h | 95 % |
| Kerio Connect - Release (7.2.0 Beta 5) | 00:05 h | 97 % |
| Kerio Operator - Release 6 | 00:00 h | 94 % |
| Kerio Operator - Kerio Operator (Unspecified) | 00:00 h | 99 % |
| Kerio Connect - Release (7.1.1) | 00:00 h | 85 % |
| Kerio Connect - Admin/Engine (Windows 2003 Server) 2 | 01:00 h | 99 % |
| Kerio Connect - Release (7.1.0 RC 2) | 00:20 h | 91 % |
| Kerio Connect - CalDAV (iCal) (Unspecified) 2 | 00:00 h | 62 % |
| Kerio Connect - KOC (MS Outlooks - cross) 3 | 03:00 h | 94 % |
| Kerio Connect - Admin/Engine (Linux / Mac) 1 | 03:30 h | 91 % |
| Kerio Connect - Dashboard, WebMail (Internet Explorer 7) | 01:30 h | 94 % |
| Kerio Connect - Domain WebAdmin, WebMail (Safari 5.x) | 01:15 h | 98 % |
| Kerio Connect - Release (7.1.0 Beta 3) | 00:20 h | 70 % |
| Kerio Connect - KOC (MS Outlook 2010) 1 | 29:00 h | 14 % |
| Kerio Sales System - KISS (Unspecified) | 00:00 h | 11 % |

Tento dialog bude zobrazovat domovskou stránku aplikace TestRunner 2.2. Standardně se zobrazují možnosti nepřihlášeného uživatele.

Na tuto stránku lze přejít z ostatních dialogů kliknutím na logo aplikace. To neplatí pro následující dialogy:

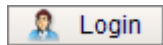
- Test Case Detail - formulář, kde tester vyplňuje výsledek testu
- Send Update - formulář pro odeslání návrhu na změnu testovacího postupu

1.1. Uživatelská práva

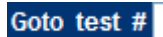
Seznam zobrazených ikon a jejich popisů závisí na roli uživatele. Vazby možných akcí na jednotlivé role ukazuje následující tabulka:

| akce \ role | nepřihlášený | tester | administrátor |
|------------------------------------|--------------|--------|---------------|
| Test Results | ANO | ANO | ANO |
| Bugs Statistics | ANO | ANO | ANO |
| Test Case Description | ANO | ANO | ANO |
| AutoTest Description | NE | ANO | ANO |
| Create Tests, Perform Tests | NE | ANO | ANO |
| Create or Edit Test Cases | NE | ANO | ANO |
| Manage Comments | NE | ANO | ANO |
| Manage TestRunner | NE | NE | ANO |

1.2. Popis funkcí



Po stisknutí tohoto tlačítka se otevře standardní přihlašovací dialog pro zadání jména a hesla. Po úspěšném přihlášení z jakéhokoliv dialogu se provede přesměrování na titulní stránku, uživateli se zobrazí položky, ke kterým má práva (viz tabulka v kapitole 1.1), a tlačítko v každém dialogu se změní na „Logout“ (popis i funkce). Při stisknutí tlačítka "Logout" se provede odhlášení uživatele ze systému a přesměrování na titulní stránku aplikace.



Každý test má svoje unikátní ID. Po zadání tohoto ID se otevře detail zadaného testu. To platí z každého dialogu, ve kterém se tato funkce vyskytuje.



Zobrazí se aktuální stránka online html manuálu. Podoba ani obsah manuálu není součástí této specifikace.



Přesměruje na stránku s výsledky testů.



Přesměruje na statistiku chybovosti jednotlivých funkcí.



Přesměruje na stránku s popisem testovacích případů.



Přesměruje na stránku, kde lze k popisu testovacího případu přidat popis autotestu.



Přesměruje na stránku, odkud lze test vytvořit nebo spustit.



Přesměruje na stránku, odkud lze vytvářet nebo editovat testovací případy.



Přesměruje na stránku, odkud lze schvalovat návrhy na úpravu testovacích případů.



Přesměruje na stránku, odkud lze TestRunner spravovat - správa produktů, komponent, platforem, závislostí, uživatelů a logů.

2. Seznam testů

The screenshot shows the KERIO TestRunner 2.2 application. The main window displays a list of tests for Kerio Connect 7.2.0. The table below represents the data shown in the screenshot.

| Test ID | Test name | Start | End | Progress | Bugs | Lock | Edit | Erase |
|---------|---------------------------------------|---------------------|-----|----------|------|------|------|-------|
| 536 | Release (7.2.0 Patch 1) | | | 94 % | | | | |
| 534 | Release (7.2.0 Final) | | | 100 % | | | | |
| 533 | Test On Demand (Unspecified) | 2011-05-16 10:07:23 | | 100 % | | | | |
| 532 | Release (7.2.0 Release Candidate 2) | | | 88 % | | | | |
| 530 | Release (7.2.0 Release Candidate 1) | | | 0 % | | | | |
| 528 | Installations (All supported) | | | 0 % | | | | |
| 527 | Admin/Engine (Linux / Mac) | 2011-04-26 15:37:46 | | 98 % | | | | |
| 526 | Admin/Engine (Windows 2008 Server) | 2011-05-11 13:43:33 | | 100 % | | | | |
| 525 | CalDAV (iCal) (Unspecified) | | | 93 % | | | | |
| 524 | Mobile Devices (Over The Air Sync) | | | 100 % | | | | |
| 523 | KOC (MS Outlooks - cross) | 2011-04-26 09:56:58 | | 100 % | | | | |
| 522 | WebAdmin, WebMail (Cross browsers) | 2011-04-26 09:30:14 | | 99 % | | | | |
| 521 | Outlook Mac (MS Outlook 2011 (Mac)) 2 | 2011-04-26 10:15:35 | | 100 % | | | | |
| 520 | Outlook Mac (MS Outlook 2011 (Mac)) 1 | 2011-04-13 10:48:10 | | 78 % | | | | |
| 516 | Release (7.2.0 Beta 6) | 2011-03-25 12:23:34 | | 95 % | | | | |
| 512 | Release (7.2.0 Beta 5) | | | 97 % | | | | |
| 505 | Release (7.2.0 Beta 4) | | | 100 % | | | | |
| 501 | Release (7.2.0 Beta 3) | | | 0 % | | | | |
| 492 | Release (7.2.0 Beta 2) | | | 0 % | | | | |
| 485 | WebMail, WebMail Mini (Google Chrome) | 2010-10-14 09:16:31 | | 100 % | | | | |

2.1. Strom testů

Základem jsou produkty, které se budou zobrazovat v abecedním pořadí. Release testy budou nově součástí standardních testů. V těch přibude komponenta „Release“. Strom testů lze rozbalit až na jednotlivé testy. Názvy jednotlivých testů bude tvořit název testované komponenty (max. dvě) a platforma v závorce (viz detail testů). Pokud budou v jednom testu vybrány více než dvě komponenty (typicky u WinRoute), bude název testu složen z názvu produktu a testované platformy v závorce (viz detail testů). Pokud by se stalo, že by v jedné verzi byly dva testy stejného jména, budou od sebe navzájem odlišeny pořadovým číslem (viz detail testů) na základě časové značky "Created".

Seznam testů se zobrazí při kliknutí na verzi produktu (zobrazí se všechny testy v této verzi). Při kliknutí na jednotlivý test (buď ve stromu testů, nebo v seznamu testů) se v pravém okně zobrazí detail vybraného testu (viz detail testů).




Tlačítko zabalí celý strom testů (platí pro všechny výskyty).

Tlačítko rozbalí celý strom testů (platí pro všechny výskyty).

2.2. Seznam testů

Seznam testů má následující vlastnosti:

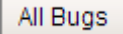
- „Test ID“: jednoznačný identifikátor testu (pomocí něj lze test vyhledávat a odkazovat se na něj z venku)
- „System“: vybraná testovaná platforma
- „Start“, „End“: časové značky začátku a konce testu. Začátek testu je v okamžiku, kdy tester poprvé spustí test. Konec testu je ve chvíli, kdy se test zamkne. Do té doby je položka "End" prázdná. Konec testu se aktualizuje po každém opětovném zamknutí.

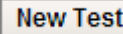
- „Progress“: aktuální vývoj testu. Počítá se jako poměr otestovaných a neotestovaných testovacích případů.
- „Bugs“: Tato funkce je dostupná pro všechny uživatele a po kliknutí na značku  u příslušného testu se v tom samém okně zobrazí stránka s detaily chyb nalezených v konkrétním testu.
- „Lock“: Tato funkce je použita proto, aby byl vidět finální stav testu při vydání produktu. Funkce ukazuje stav zámku testu (pro všechny uživatele):
 -  test je odemknutý a lze v něm provádět změny (testovat, editovat test). Při kliknutí na tuto značku se zobrazí dotaz, zda si uživatel opravdu přeje test zamknout.
 -  test je zamknutý - zmražený - nelze v něm provádět žádné akce (nelze testovat, editovat, aktualizovat). Při kliknutí na tuto značku se zobrazí dotaz, zda si uživatel opravdu přeje test odemknout.

Změnit stav zámku může pouze uživatel s právy tester nebo administrátor.

- „Edit“: Tento sloupec se zobrazí pouze uživatelům s právy tester nebo administrátor. Značka bude aktivní pouze u odemknutých testů. Při kliknutí na tuto značku se v novém okně otevře stránka pro editaci vytvořeného testu. Při editaci testu budou předvyplněny testovací případy vybrané při vytváření testu.
- „Erase“: Tento sloupec se zobrazí pouze uživatelům s právy tester nebo administrátor. Značka bude aktivní pouze u odemknutých testů a po kliknutí na tuto značku se zobrazí dotaz, zda si uživatel opravdu přeje test odebrat.

2.3. Ostatní funkce

 Po stisku tohoto tlačítka se otevře stránka se seznamem nalezených chyb při testu v celé verzi. Pokud nebude vybrána konkrétní verze, nebude tlačítko aktivní.

 Toto tlačítko je aktivní pouze pro uživatele s právy tester nebo administrátor. Po stisku tohoto tlačítka se v novém okně zobrazí formulář pro vytváření nového testu. Do formuláře pro vytváření nového testu se odešle jméno produktu a číslo verze dle aktuálního výběru:

- pokud kliknu na jméno produktu a následně na tlačítko "New Test", odešle se pouze jméno produktu (pravděpodobně se vytváří test pro novou verzi)
- pokud kliknu na verzi nebo na detail jiného testu a následně na tlačítko "New Test", odešle se jméno produktu a vybraná verze (pravděpodobně se vytváří nový test pro stávající verzi)

Pozn.: Obě hodnoty lze v cílovém formuláři změnit

3. Vytváření nebo editace testů

Kerio Connect - New Test

| | | | | |
|---------------------------------|-------------------------|--|--------------------------------|-----------------------------|
| Product Kerio Connect | Version 7.3.0 | Dependency Operating systems | Platform Unspecified | Profundity Normal |
|---------------------------------|-------------------------|--|--------------------------------|-----------------------------|

Release test
 Functional test

| | | | |
|--|--|---|---|
| <input type="checkbox"/> Admin/Engine | <input type="checkbox"/> CalDAV (iCal) | <input type="checkbox"/> CardDAV (Address Book) | <input type="checkbox"/> All Components |
| <input type="checkbox"/> Dashboard | <input type="checkbox"/> Domain WebAdmin | <input type="checkbox"/> Entourage | |
| <input type="checkbox"/> Installations | <input type="checkbox"/> KAdmin | <input type="checkbox"/> KOC | |
| <input type="checkbox"/> KOFF | <input type="checkbox"/> KSP | <input type="checkbox"/> Mobile Devices | |
| <input type="checkbox"/> Outlook Mac | <input type="checkbox"/> Registration | <input type="checkbox"/> WebAdmin | |
| <input type="checkbox"/> WebMail | <input type="checkbox"/> WebMail Mini | | |

| ID | K | S | Test Cases | Comments | AT Coverage | Profundity | Time |
|----|----|---|---|----------|-------------|------------|------|
| 0 | 34 | 0 | <input checked="" type="checkbox"/> Services (Admin/Engine) | | | Normal | 0:25 |
| 0 | 35 | 0 | <input checked="" type="checkbox"/> Domains (Admin/Engine) | | | Normal | 4:10 |
| 0 | 36 | 0 | <input type="checkbox"/> SMTP Server (Admin/Engine) | | | Normal | |
| 0 | 37 | 0 | <input type="checkbox"/> SMTP security (Admin/Engine) | | | Normal | |
| 0 | 38 | 0 | <input checked="" type="checkbox"/> SMTP queue options (Admin/Engine) | | | Normal | 0:50 |
| 0 | 39 | 0 | <input type="checkbox"/> Antivirus control (Admin/Engine) | | | Normal | |
| 0 | 40 | 0 | <input checked="" type="checkbox"/> Attachment filter (Admin/Engine) | | | Normal | 0:30 |
| 0 | 41 | 0 | <input type="checkbox"/> Spam Rating (Admin/Engine) | | | Normal | |
| 0 | 42 | 0 | <input type="checkbox"/> Spam filter blacklists (Admin/Engine) | | | Normal | |
| 0 | 43 | 0 | <input type="checkbox"/> Custom Rules (Admin/Engine) | | | Normal | |
| 0 | 44 | 0 | <input type="checkbox"/> Spam Assassin (Admin/Engine) | | | Normal | |
| 0 | 45 | 0 | <input type="checkbox"/> Caller ID (Admin/Engine) | | | Normal | |

3.1. Globální hodnoty

„Product“: Přednastavená hodnota bude dle výběru na stránce seznamu testů. V případě potřeby lze hodnotu změnit.

„Version“: Přednastavená hodnota bude dle výběru na stránce seznamu testů. V případě potřeby lze hodnotu změnit. Pokud se bude hodnota zadávat ručně nebo se bude měnit, bude toto pole odkazovat do Bugzilly do políčka "version" příslušného produktu a tyto hodnoty bude nabízet přes drop-down menu.

„Dependency“: Odkazuje do stromu závislostí. Přednastavená hodnota bude "none". Před zobrazením testů je třeba hodnotu změnit. Změna způsobí načtení příslušných testů.

„Platform“: Odkazuje do seznamu platform. Platformy budou v drop-down menu abecedně řazeny.

„Profundity“: Vybraná hloubka testu se přednastaví pro celý test. Hodnotu lze ale u jednotlivých kapitol či testovacích případů změnit. Drop-down menu bude obsahovat hodnoty „Deep“, „Normal“, „Simple“. K těmto hodnotám budou přiřazeny koeficienty, kterými se bude násobit časový odhad pro konkrétní testovací případ. Koeficienty jsou následující: Deep = 3, Normal = 1, Simple = 0.33. Přednastavenou hodnotou bude hodnota „Normal“.

Při kliknutí na ikonku  se otevře malé okno (podobně jako třeba Sticky Notes ve

WebMailu) pro zadání komentáře k celému testu. V případě, že je komentář zadaný, změní ikonka barvu na červenou.

3.2. Komponenty

V TestRunneru je třeba odlišit dvě základní skupiny testů: Release testy a Funkční testy. Release test slouží k tomu, aby se ověřilo, zda byl produkt korektně vypublikován. Při výběru tohoto testu nelze vybrat žádnou komponentu ani testovací případ z kategorie "Funkční testy". Navíc se při výběru "Release" testu zobrazí textové pole pro zadání verze nebo čísla buildu. Při výběru "Release" testu se předvyberou všechny testovací případy, které spadají do komponenty "Release". Výběr těchto testovacích případů lze měnit.

Komponenty pro funkční testy budou přehledně zobrazeny do sloupců a řádků (max. 3 sloupce). Komponenty obsahují kapitoly a ty jednotlivé testovací případy. V pravé horní části je umístěna speciální komponenta „All Components“.

Funkčnost komponent je následující:

- pokud se označí komponenta, označí se všechny kapitoly a testovací případy, které jsou v komponentě obsaženy
- pokud se odznačí vybraná komponenta, odznačí se všechny označené kapitoly a testovací případy patřící do vybrané komponenty
- pokud se označí komponenta „All Components“, označí se všechny komponenty a všechny testovací případy a naopak

3.3. Výběr testovacích případů a jejich vlastností

Při vytváření testu budou zobrazeny pouze poslední klony testovacích případů platných pro zadanou verzi! Při editaci testu se budou zobrazovat poslední platné klony testovacích případů, pouze pokud testovací případ nebyl ještě testován. Otestovaný testovací případ nepůjde při editaci testu odebrat.

Testovací případy lze označovat i odznačovat buď jednotlivě, nebo po kapitolách (označí / odznačí se všechny testovací případy v kapitole). Všechny kapitoly se budou standardně zobrazovat zabalené. Kliknutím na název kapitoly se zobrazí jednotlivé testovací případy (opětovné kliknutí testovací případy schová). Pokud se vybere konkrétní testovací případ, bude se ve vytvořeném testu zobrazovat i název jeho kapitoly, i když kapitola nebude označena. V tomto případě zůstane komponenta, do které testovací případ spadá, neoznačená, nicméně do názvu testu se s ní bude počítat.

„Comments“: kliknutím na ikonku u kapitoly nebo jednotlivého testovacího případu se otevře malé okno (podobně jako výše) pro zadání komentáře k jednotlivým kapitolám nebo testovacím případům. To, že je někde již komentář vyplněn, je signalizováno ikonkou. Při najetí myši na tuto ikonku se vypíše obsah komentáře (hint, ...).

Popis testovacího případu (ve zvoleném jazyce - standardně dle jazyka OS + autotest) bude k dispozici přes on-mouse-over akci (0.5s) (stejně jako v Bugzille a položky „Status Whiteboard“).

„Profundity“: Hloubka testu bude přednastavena výběrem globální hloubky. Hloubku testu lze měnit pro celou kapitolu (změní se všechny testovací případy v kapitole) nebo pro jednotlivé testovací případy.

„AT Coverage“: jedná se pouze o informativní hodnotu. Zobrazuje, jaká část testovacího případu je pokryta autotestem. Tuto hodnotu zadává tvůrce popisu autotestu v okně pro popis autotestu testovacího případu.

„Time“: opět se jedná pouze o informativní hodnotu. Hodnota udává čas potřebný k manuálnímu testu daného testovacího případu. Čas se bude měnit dle vybrané hloubky testu. Zároveň se mění i výsledný čas potřebný pro celý test, který je uveden v dolní části dialogu. Čas uvedený u názvu kapitoly bude součtovým časem všech časů jednotlivých testovacích případů obsažených v kapitole.

Názvy kapitol budou od jednotlivých testovacích případů barevně odlišeny (viz. obrázek).

3.4. Test na žádost

Formulář pro vytvoření nového testu bude standardně bez položky pro test na žádost. Pole pro zadání testu na žádost lze zadat stisknutím tlačítka. Opakovaným stisknutím lze vytvořit více polí. Posílat se bude pouze pole, ve kterém není prázdný řetězec. U každého testu na žádost se zobrazí i checkbox "Permanent". Testy na žádost jsou přidány do seznamu testovacích případů vytvořeného testu. Pokud bude vytvořený test obsahovat pouze testovací případy zadané přes formulář testu na žádost, bude název testu "Test on demand" (případně i pořadové číslo).

„Permanent“: Při zaškrtnutí checkboxu u vybraného testu na žádost se navíc v Bugzille vytvoří nový záznam s následujícími parametry:

- product = dle vytvářeného testu
- Component = QA
- Reporter = tvůrce testu
- Assignee = uživatel s administrátorskými právy. Pokud je těchto uživatelů více, přiřadí se do assignee první v pořadí (uživatelské jméno dle abecedy); ostatní jsou přidány do položky "Cc:"
- Severity = Task
- Summary = stručný popis testu, který je zapsán v poli u testu na žádost
- Description = stejné jako u Summary

3.5. Ostatní

Tlačítkem se vytvoří nový nebo aktualizuje stávající test. Při aktualizaci již vytvořeného testu nepůjdou odebrat již otestované testovací případy (checkboxy budou neaktivní). Po stisknutí tohoto tlačítka se zavře formulář a provede se přesměrování na stránku seznamu testů tak, aby byl vidět právě vytvořený nebo aktualizovaný test.

Pokud nebude zvolen alespoň jeden testovací případ (ze seznamu nebo v testu na žádost), nebude tlačítko aktivní.

Po vytvoření nebo aktualizaci testu se uživateli s administrátorskými právy odešle o této skutečnosti následující zpráva.

TestRunner info message

| Test details | |
|--------------|---------------------------------|
| Test ID: | 219 was created / updated |
| Product: | MailServer |
| Version: | 6.2.2 |
| System: | Mac OS |
| Link: | HERE |
| Fixed bugs: | HERE |
| Created by: | Lubos Rybnicek (rybnicek) |
| Date: | Tue, 29 Aug 2006 12:10:34 +0200 |

Test description:

Informative message, notes, etc. could be placed here.

This message is automatically generated by Kerio TestRunner, please do not reply to this e-mail.


Adresa odesilatele bude: testrunner@kerio.com

4. Detail testu

The screenshot displays the Kerio TestRunner 2.2 interface. On the left is a tree view of test categories, including Kerio Connect 7.2.0 (with sub-items like Release, Test On Demand, and Admin/Engine) and various other versions (7.1.4 to 6.6.1). The main window shows the 'Kerio Connect 7.2.0 - Test detail' for ID 527. A summary box at the top right indicates: Product: Kerio Connect 7.2.0, Component: Admin/Engine, System: Linux / Mac, Created: 2011-04-21 10:58:23 by Zdeněk Samuel. Overall status is 98% - 183 of 186. Time estimates: Original 44:52, Actual 36:40, Left 0:45. Below this is a table of test results:

| Test Item | Status | Count | Icon |
|---|--------|-------|------|
| Spam repellent | ✓ | | |
| Delay SMTP greeting | ✓ | 3662 | |
| Enforced correct flow of SMTP communication | ✓ | 3662 | |
| Spam settings - Upgrade / Downgrade from / to KMS 6.1.x | ✓ | | |
| Estimated time: 0:45 from KMS 6.1.x | ✓ | 3753 | |
| Archiving | ✗ | | |
| Actions | ✗ | 59478 | |
| Archive | ✓ | 3753 | |
| Archive before antivirus check | ✓ | 3753 | |
| Access and rights | ✓ | 3753 | |
| Backup | ✓ | | |
| Target backup directory | ✓ | 3753 | |
| Backup scheduling | ✓ | 3753 | |
| Functionality | ✓ | 3753 | |
| Advanced options | ✓ | 3753 | |
| Internet connection | ✓ | | |
| Online | ✓ | 3595 | |
| Offline | ✓ | 3595 | |

4.1. Hlavička

- „Product“: název produktu + číslo verze
- „Component“: název testované komponenty. Pokud je vybráno více komponent v jednom testu, bude zde uvedeno jméno produktu.
- „Systém“: vybraná platforma
- „Created“: časová značka vytvoření testu a tvůrce testu
- „Overall“: procentuální a číselné vyjádření průběhu testu
- „Original Estimate“: čas v hodinách potřebný k otestování celého testu
- „Actual Taken“: doba trvání testu. Tester u každého testovacího případu bude vyplňovat dobu, kterou věnoval testu.
- „Time Left“: součet časů u doposud netestovaných testovacích případů
- Při on-mouse-over akci na ikonu  se zobrazí komentář pro celý test uložený při vytváření testu.

4.2. Tasks Tested

Pokud na stránku detailu testu nahlíží uživatel, který nemá práva *tester*, nebudou názvy kapitol a testovacích postupů klikatelné. Na on-mouse-over se pouze zobrazí popis testovacího případu dle jazyka, který lze vybrat kliknutím na vlaječku. Standardní jazyk je dle jazyka systému.

Pokud na tuto stránku nahlíží uživatel s právy *tester* nebo *administrátor*, budou navíc názvy testovacích případů klikatelné a po kliknutí na testovací případ se objeví formulář pro provedení testu.







Názvy kapitol budou od jednotlivých testovacích případů barevně odlišeny (viz. obrázek).

4.3. Výsledky testů

Výsledky testů jednotlivých testovacích případů a následně i kapitol jsou v závislosti s nově nalezenými chybami. Staré chyby nebudou mít na výsledek testu vliv. Pro potřeby TestRunneru budeme rozlišovat dvě úrovně chyb:

- vážné – zásadní chyba ve funkčnosti nebo regrese, produkt by se s nimi neměl vydat
- „lehké“ – jsou to většinou light, triviality, suggestions – produkt lze s těmito chybami vydat

Výsledné stavy testu testovacích případů mohou být následující (dle závažnosti chyb):

-  - v testu nebyla nalezena žádná chyba
-  - v testu byla nalezena minimálně jedna lehká chyba. Všechny nalezené lehké chyby byly v Bugzille do uzamknutí testu převedeny do stavu RESOLVED.
-  - v testu byla nalezena minimálně jedna vážná chyba. Všechny nalezené vážné chyby byly v Bugzille do uzamknutí testu převedeny do stavu RESOLVED
-  - v testu byla nalezena minimálně jedna lehká chyba. Tento stav platí, dokud bude zbývat alespoň jedna neopravená lehká chyba před zamknutím testu.
-  - v testu byla nalezena minimálně jedna vážná chyba. Tento stav platí, dokud bude zbývat alespoň jedna neopravená vážná chyba před zamknutím testu.
-  - případ se netestuje

Do pole "Result" pro kapitolu se propaguje nejhorší dosažený stav z testovacích případů obsažených v kapitole. Aktualizace se provádí vždy při načtení stránky. Po zamknutí testu

se zmrazí aktuální stav a výsledky se již nebudou aktualizovat. Další aktualizaci lze provést až po odemknutí testu.

4.4. Chyby


Tester vkládá do testovacího formuláře čísla všech nalezených chyb. Ty se pak zobrazují ve sloupci *Bugs Found* a to následujícím způsobem:

- **červeně** – chyby se severitou *critical*
- **tučně** – chyby se severitou *serious*
- normálně – chyby se severitou *light* nebo *triviality*
- *kurzívou* – chyby se severitou *suggestion*
- ~~přeškrtnuté~~ – vyřešené chyby (jsou ve stavu **RESOLVED** nebo **VERIFIED**) a mají shodný TM jako testovaná verze
- ~~[přeškrtnuté v hranatých závorkách]~~ – vyřešené chyby (jsou ve stavu **RESOLVED** NEBO **VERIFIED**), jejich TM nepatří do testované verze

"Bugs found":

Při on-mouse-over akci na číslo chyby se do *hintu* vypíše summary, status, resolution. Při kliknutí na číslo chyby se Bugzilla otevírá v novém okně.

"Old bugs":

Značka  znamená, že v minulosti byla v testovacím případě nalezena chyba, která byla zanesena do TestRunneru a doposud nebyla opravena. Při on-mouse-over akci na tuto značku se do hintu vypíše číslo chyby, summary, status, resolution. V případě, že chyb je více, zobrazí se chyby v hintu pod sebe. Při kliknutí na tuto značku se otevře stránka v Bugzille se seznamem chyb uvedených v testovacím případě. (Pozn.: link do Bugzilly je: https://bugzilla.kerio.com/buglist.cgi?bug_id=14091,13750,...)

4.5. Ostatní

„Build #“: toto pole má pouze informativní hodnotu. Tester při vyplňování formuláře vyplňuje i číslo buildu, na kterém se testování provádí.

„Comment“: Tester má rovněž možnost připojit k testu krátký komentář. Připojený komentář značí ikonka. Při on-mouse-over akci na tuto ikonku se do *hintu* vypíše testerův komentář (bude obsahovat jméno testera a jeho komentář).

„Bugs only“: přesměruje na stránku s výpisem chyb v tomto testu.

„Print“: vytiskne seznam úkolů s časy na jednotlivé úkoly i souhrnně pro celé kapitoly. Tisknutelná verze je zjednodušená graficky i obsahově a nezobrazuje aktuální stav testu (je určena pro vytisknutí před/během testu, například pro případ, kdy tester potřebuje/chce "off-line" návod k testu).

5. Seznam chyb

The screenshot shows the 'Kerio Connect - Bugs statistics' page. The left sidebar contains a tree view of test categories. The main area displays a table of bugs under 'Opened bugs' and 'Resolved or Verified bugs'.



| Opened bugs | |
|----------------------------|---|
| Admin/Engine (Linux / Mac) | Archiving // Actions |
| 59478 | |
| Admin/Engine (Mac OS X) | Archiving // Actions |
| 17523 | |
| 17915 | |
| Admin/Engine (Linux) | Archiving // Archive before antivirus check |
| 9763 | |
| Admin/Engine (Mac OS X) | Archiving // Archive before antivirus check |
| 9763 | |
| Admin/Engine (Unspecified) | Archiving // Archive before antivirus check |
| 9763 | |
| Admin/Engine (Unspecified) | Archiving // Access and rights |
| 12240 | |
| 12296 | |

| Resolved or Verified bugs | |
|---------------------------|--|
|---------------------------|--|

- Seznam chyb půjde zobrazit buď pro konkrétní test, nebo pro celou verzi (několik testů).
- Prohlížeč si bude pamatovat poslední stav rozbalovátek pro každého uživatele (pomocí cookies).

"Opened bugs" bude zobrazovat všechny chyby objevené v testu, které dosud nejsou vyřešeny.

"Resolved or Verified bugs" bude zobrazovat chyby vyřešené či čekající na potvrzení, že jsou vyřešeny.

Tlačítka  a  rozbalují či zabalují výše popsání části. Prohlížeč si bude v cookies pamatovat stav rozbalovátek pro každého uživatele.

Dále platí, že každá chyba bude obsahovat: jméno testu (komponenta + platforma v závorce), jméno kapitoly a testovacího případu, číslo chyby bude zároveň linkem do Bugzilly, summary, severity, status.

6. Testovací formulář

Kerio Connect - Test Case Detail

| | | | |
|-------------------|---|------------------------------------|-----------|
| Component: | Release | Possible AutoTest Coverage: | 0% |
| Chapter: | Installers and binary files | Actual AutoTest Coverage: | 0% |
| Test Case: | Download and install Kerio Connector for BlackBerry from www.kerio.cz | Profundity: | Normal |
| | | Estimated Time (average): | 0:05 |
| | | Estimated Time (median): | 0:10 |
| | | Estimated Time (modus): | 0:10 (1x) |

Won't test Reset Build # Elapsed time 5 Send & Close

Found bugs Bug # Failed Add bug

Test Case Description

Czech:

- Stáhnout produkt ze stránky <http://www.kerio.cz/connect/download>
- Nainstalovat staženou verzi

English:

- Download product from <http://www.kerio.cz/connect/download>
- Install downloaded version

Update description

AutoTest Description +

Tester's short comment

History -

| Version | Build | Result | Tested by | System | Note | Time |
|---------|-------|--------|------------------|---------------|------|------|
| 7.2.0 | 3211 | OK | Jaroslav Buric | All supported | | 0:00 |
| 7.1.4 | 2985 | OK | Miroslav Osladil | All supported | | 0:00 |
| 7.1.3 | | OK | Zdeněk Samuel | All supported | | 0:00 |
| 7.1.0 | 1792 | Failed | Jaromir Obr | All supported | | 0:10 |

Old bugs -

6.1. Hlavička

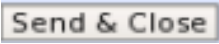

- „Component“: jméno komponenty, ve které se kapitola a testovací případ nachází
- „Chapter“: jméno kapitoly, ve které se testovací případ nachází
- „Test Case“: jméno testovacího případu
- „Possible AutoTest Coverage“: možné pokrytí testovacího případu autotestem. Zadává tvůrce popisu autotestu.
- „Actual AutoTest Coverage“: aktuální pokrytí testovacího případu autotestem. Zadává tvůrce popisu autotestu.
- „Profundity“: hloubka testu platná pro tento testovací případ
- „Estimate Time (average)“: průměrný čas určený pro test tohoto testovacího případu (již po přepočítání koeficientem hloubky testu).
- „Estimate Time (median)“: střední hodnota všech časů z předchozích testů. (Tato hodnota je pouze informativní a jako technologický test, NENÍ přepočítávána)

koeficientem hloubky testu. Slouží pro zhodnocení možnosti získávání přesnějších odhadů než je aritmetický průměr.)

- „Estimate Time (modus)“: nejčastější hodnota času z předchozích testů. (Tato hodnota je pouze informativní a jako technologický test, NENÍ přepočítávána koeficientem hloubky testu. Slouží pro zhodnocení možnosti získávání přesnějších odhadů než je aritmetický průměr.)



Pozn.: Tvůrcem testovacího postupu je nastaven časový odhad testu testovacího případu. Tester v každém testu zadává skutečný čas, který strávil testováním konkrétního testovacího případu. Zpět do databáze se však posílá čas v závislosti na "Profundity" (výsledný čas = skutečný čas / profundity). V hlavičce uvedený "Estimated Time" je průměrem těchto časů, který je následně vynásoben koeficientem hloubky testu. Vypočtený průměrný čas nesmí zahrnovat nulové hodnoty, které se do databáze dostanou importem z předchozí verze TestRunneru. V současné verzi již nulový čas nelze zadat!

6.2. Odeslání výsledku





Stisknutím tlačítka  se odesílá výsledek testu, okno se zavře a zobrazí se aktualizované okno s detailem testu. Pokud není nalezena žádná chyba, je test automaticky s výsledkem . Pokud se najdou nějaké chyby, je výsledek závislý na jejich závažnosti, kombinaci, stavu (viz kapitola 4.3). Výsledek testu je ovlivněn pouze nově zaevidovanými chybami.

Test nelze odeslat, pokud nebudou vyplněná pole **Build #** a **Elapsed Time**. **Build #** je pole textové a **Elapsed Time** číselné (pouze celá čísla > 0). Čas bude zadán v minutách.

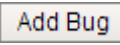

6.3. Vrácení do stavu „netestováno“

Pokud tester odeslal výsledek testu např. omylem, je možné vrátit test to původního stavu a to stiskem tlačítka . Po stisku se okno zavře a zobrazí se aktualizované okno s detailem testu. U testu je nastaven stav .

6.4. Won't test

Stisknutím tlačítka  se okno zavře a zobrazí se aktualizované okno s detailem testu. Testu je nastaven status . Po znovuotevření testovacího dialogu jsou volby pro přidání bugu a odeslání výsledku zašedlé. Pro návrat do původního stavu slouží tlačítko , po jehož stisknutí se dialog zavře a test se nastaví do stavu .

6.5. Chyby

Každá nalezená chyba je nejdříve zaevidovaná v Bugzille. Její číslo pak tester zadá do pole **Bug #** a stiskne se tlačítko . Chyba se přidá do seznamu chyb nalezených v tomto testu. Pokud se zaškrtně i checkbox  **Failed**, jedná se o vážnou chybu. Lze zadat více chyb. V případě potřeby půjde záznam odstranit (zobrazí se potvrzující dotaz).

Zobrazení chyb bude dle grafického návrhu, číslo chyby bude odkazovat do Bugzilly (otevře se nové okno s detailem chyby v Bugzille).

Zadá-li se neexistující číslo chyby (rozumí se neexistující v Bugzille), zobrazí se informativní okno s hláškou **Bug # does not exist in Bugzilla**. a chyba se do seznamu nepřidá.

„Old Bugs“: všechny chyby (nevyřešené i vyřešené), které byly k tomuto testovacímu případu do TestRunneru zadány. Prohlížeč si bude pamatovat stav rozbalovátka pro každého uživatele (pomocí cookies).

6.6. Descriptions



„Test Case Description“: obecný popis testovacího případu

"Chapter info": bude se zobrazovat pouze v případě, že kapitola obsahuje nějaký popis.


„AutoTest Description“: popis automatického testu testovacího případu. Prohlížeč si bude pamatovat stav rozbalovátka pro každého uživatele (pomocí cookies).

„History“: informace pro testera, jak se vyvíjely předchozí testy u tohoto testovacího případu. Detail viz obrázek.


Nejhorší a nejlepší čas je zvýrazněn **červeně**, respektive **zeleně**. Tyto časy nejsou zvýrazňovány pro účely "odměňování" nebo "trestání" testerů, ale pro snadné nalezení extrémních hodnot, zadaných například omylem. (Záměna čísla buildu a času, "uklepnutí se" apod.)

Tlačítka  a  rozbalují či zabalují výše popsané části. Prohlížeč si bude v cookies pamatovat stav rozbalovátek pro každého uživatele.

6.7. Ostatní

Pokud se v levém horním okraji vyskytuje ikonka , znamená to, že tvůrce testu vložil k tomuto testovacímu případu nějaký komentář. Komentář se zobrazí po akci on-mouse-over na této ikonce.

„Comment“: Krátká poznámka testera. V detailu testu se pak zobrazuje ve sloupci *Comment* (odešle se s formulářem po stisku tlačítka "Tested").

 Otevře nové okno pro odeslání aktualizace testovacího postupu administrátorovi.

7. Test Case Update

Update Checker from Windows version - Send update

Send Update

Actual description:

Czech:

- Nainstalovat poslední vydanou verzi na platformu Windows
- Nastavit IP adresu do rozsahu určeného pro testy vydání
- Ověřit, že vydávaná verze se zobrazí jako dostupná pro upgrade
- Udatovat tuto verzi verzí testovanou přes Update Checker
- Ověřit, že po updatu na tuto verzi nebude nabízena Update Checkerem žádná novější verze

English:

- Install the last released version on Windows
- Configure IP address to the release tests range
- Verify new version is available for the upgrade
- Upgrade this version to the tested one via Update Checker
- Verify there is no new version available after upgrading

Update:

Old updates:

- „Actual description“: popis testovacího případu
- „Update“: pole pro zapsání návrhu na změnu testovacího případu
- „Old updates“: Uchovává se historie toho, jak se testovací případ aktualizoval. Zobrazit či schovat tuto položku lze pomocí tlačítek a . Prohlížeč si bude v cookies pamatovat stav rozbalovátek pro každého uživatele.
- Po stisknutí tlačítka se návrh na změnu odešle ke schválení.

8. Seznam testovacích případů

The screenshot displays the Kerio TestRunner 2.2 application. On the left is a hierarchical tree view of test cases, with 'Manual account configuration' selected under 'Outlook Mac'. The main window shows the 'Test case detail' for this specific case. At the top, a 'Bugs Stat.' button is visible. The detail view includes a header with component information and a table of coverage statistics. Below this, there are sections for 'Test Case Description' (with Czech and English text), and 'Autotest Description'.

| Component: | Outlook Mac | Possible AutoTest coverage: | 0% |
|------------|------------------------------|-----------------------------|----|
| Chapter: | Identity/Account (Profile) | Actual AutoTest coverage: | 0% |
| Task: | Manual account configuration | | |

Test Case Description

Czech:
Cíl: ručně nakonfigurovat nový účet v Outlooku pro připojení k serveru

- Spustíte nabídku Tools->Accounts-> + -> Exchange...
- Nakonfigurujete účet pro připojení k serveru včetně veřejných složek a LDAP serveru (obeje v záložce Advanced).
- Vyzkoušíte i připojení přes SSL WebDAV a alternativní porty Tipy: + pro LDAP je nutné zaškrtnout "Override default LDAP port" a používat 389 (resp. 636 pro SSL) + veřejné složky: "hostname/public"

English:
GOAL: manually configure new account in Outlook for connection with server

- Start from menu "Tools->Accounts-> + -> Exchange..."
- Configure account for connection to server, including public folders and LDAP server (both are in tab Advanced)
- Test WebDAV over SSL and nondefault ports. Hints: + for LDAP must be enabled "Override default LDAP port" and used 389 (or 636 for SSL) + public folders has path: "hostname/public"

Autotest Description

8.1. Strom testovacích případů

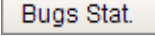
Hierarchie: Produkt -> Komponenta -> Kapitola -> Testovací případ

Produkty budou řazeny abecedně (i navzdory obrázku). Komponenty budou řazeny abecedně, kapitoly a testovací případy podle čísla kapitol a testovacích případů.

Označit lze komponentu, kapitolu, testovací případ (kvůli statistice chyb). Při kliknutí na testovací případ se do pravého okna zobrazí jeho detail. Při kliknutí na kapitolu nebo komponentu se do pravého okna zobrazí její obsah.

8.2. Test Case detail

Údaje v hlavičce viz obrázek, vysvětlení viz hlavička testovacího formuláře. Jednotlivé části jsou formátovány dle html (,
).

Stisknutím tlačítka  se přejde na stránku statistiky konkrétního testovacího případu nebo kapitoly.

10.1. Strom testovacích postupů

Základní funkčnost stromu je stejná jako v předchozí kapitole s tím rozdílem, že se vedle kapitol a testovacích postupů zobrazují čísla kapitol a testovacích postupů. Číslo kapitoly nebo testovacího postupu je sice unikátní, ale může se časem měnit.

Při kliknutí na existující testovací případ se otevře tento testovací případ pro editaci – obsahuje všechny již zadané údaje. Dialog pro vytvoření nového testovacího případu se otevře po stisknutí tlačítka **New Test Case**. Po stisknutí tohoto tlačítka a zobrazení dialogu pro vytvoření nového testu se toto tlačítko zablokuje.

Pokud je otevřen dialog pro vytvoření nového nebo editaci stávajícího postupu, lze ve stromu testovacích případů klikat na značky pro rozbalení či zabalení části nebo celého stromu bez toho, aniž by se měnil dialog v pravém okně. Pokud kliknu přímo na komponenty, kapitolu nebo testovací případ, změní se i obsah pravého (ztráta rozdělané práce) okna a aktivuje se tlačítko **New Test Case**.

10.2. Řídící informace

„Component“: Komponenty nelze vytvářet v tomto dialogu – jsou tedy dané. Komponenta se přednastaví dle toho, kde se nachází ukazatel ve stromu při stisku tlačítka. V případě potřeby lze komponentu měnit. Tato položka je povinná.

„Chapter“: Do tohoto pole půjde zadat pouze celé kladné číslo (kapitoly). Chování bude pak následující:

- Pokud se ukazatel ve stromu testovacích případů nachází na konkrétní kapitole, bude hodnota přednastavena na číslo oné kapitoly.
- Pokud se ukazatel ve stromu testovacích případů nachází na konkrétní komponentě, bude hodnota přednastavena na první nepoužité číslo v řadě v této komponentě. V případě, že toto číslo používá již jiná kapitola, tedy provádí se vkládání, ostatní kapitoly se posunou (musí platit jedinečnost).
- Pokud se ukazatel ve stromu složek nachází na nově vytvořené komponentě, bude číslo kapitoly prvním volným číslem v seznamu všech kapitol.
- Tato položka je povinná a v případě potřeby lze přednastavené číslo změnit.

„SubChapter“: Do tohoto pole půjde zapsat pouze celé kladné číslo včetně nuly. Přednastavenou hodnotou bude první volné číslo v rámci vybrané kapitoly. V ostatních případech se hodnota nepřednastavuje. Pokud se zadá nula, značí to, že se vytváří nová kapitola. Tato položka je povinná.

Pokud budu chtít vložit novou kapitolu nebo testovací případ mezi existující, zadá se pouze pozice, na které by měly kapitola nebo testovací případ být, a ostatní kapitoly nebo testovací případy, které jsou za touto pozicí, se posunou (musí platit jedinečnost).

„Validity Start“: Zadává se sem číslo verze, OD které tento testovací postup platí. Pole bude provázané s Bugzillou s políčkem "version" příslušného produktu. Tato položka je povinná.

„Validity End“: Zadává se sem číslo verze, DO které tento testovací postup platí. Pole bude provázané s Bugzillou s políčkem "version" příslušného produktu. Tato položka je povinná a v případě, že se nezná toto omezení, bude se zadávat *unspecified*.

„Estimated Time“: Odhad tvůrce testovacího případu, jaký čas je potřeba na otestování

tohoto testovacího případu. Tato položka je povinná. Jedná se ale pouze o inicializační hodnotu. Tato hodnota se bude přepočítávat v závislosti na skutečném trvání testu.

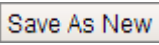
„New dependency“: Položka pro výběr nové závislosti na platformě. Na základě těchto závislostí jsou potom Test casey filtrovány. Pokud nejsou uvedeny závislosti, předpokládá se dostatečná obecnost pro jakýkoliv filtr.

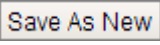
„Selected dependency“: Seznam všech závislostí na platformě. Test case je platný pouze pro tyto platformy (včetně jejich potomků, tzn. závislost operačních systémů se vztahuje i na závislost Linux, MS Windows, Mac OS atd.)

10.3. Editační pole

- Klasické textové pole s možností zápisu libovolných znaků – podpora češtiny, znaků jako: „“ / ; ' \
- Bude interpretovat následující html tagy: ,
, , html odkazy.

10.4. Ukládání nově vytvořených testovacích případů

Uložení testovacího případu se provede tlačítkem .

Základní a nejjednodušší variantou je, když se vytváří nová kapitola nebo testovací případ. Při stisku tlačítka  se provede uložení následujícího:

- všech řídicích informací (produkt, komponenta, kapitola, ...)
- název testovacího případu – Title
- popis testovacího případu - Description
- Příznak *autotest* se nastaví na *false*.
- Číslo klonu se nastaví na 0.

Ve spodní části lze přepínat mezi jednotlivými klony.

10.5. Ukládání updatů testovacích případů (klony)

V případě, že se testovaná funkce změnila, je třeba změnit i testovací postup. Zároveň se ale nesmí měnit zpětně pro již vytvořené testy. Dále je třeba dát vědět tvůrcům autotestu, že se změnil popis.

Z předchozích důvodů bude každé nové uložení testovacího postupu vytvářet novou verzi téhož – klon. V tomto případě bude ukládání probíhat následovně:

- uloží se všechny řídicí informace
- název testovacího případu – Title
- popis testovacího případu – Description
- překopíruje se položka „Autotest Description“ z předchozího klonu
- Příznak *autotest* se nastaví na *false*.
- Číslo klonu se zvýší o 1.

11. Popis autotestu

The screenshot shows the Kerio TestRunner 2.2 application. On the left is a tree view of test cases, with 'WebMail - New e-mail actions' selected. The main area displays the 'AutoTest Description' for this test case. The description includes the component (WebMail), chapter (WM - New e-mail actions), and task (WM - Drafts). It also shows the test case description in Czech and English, the responsible person (Miroslav Kroc), and the AT coverage (95% possible, 95% actual). The 'Test Case Update' button is visible.

11.1. Strom testovacích případů

Základní funkčnost je stejná jako v předchozí části s tím rozdílem, že se nezobrazují čísla kapitol.

U všech testovacích případů, u kterých se nerovnájí hodnoty „Possible AT Coverage“ a „Actual AT Coverage“ nebo je příznak *autotest* nastaven na *false* (byl vytvořen nebo updatován testovací případ), je zobrazena ikonka **U**. Tato informace se propaguje dále do názvu kapitoly a komponenty.

11.2. Editační pole

- Pro editaci se otevírá vždy poslední klon testovacího případu.
- „AutoTest Description“: Platí stejná pravidla jako u editace testovacích případů.
- „Test Case Description“: pouze informativní pole – nelze do něj zapsat.
- „Possible AT Coverage“: může nabývat hodnot 0 – 100. Toto pole je povinné. Vyjadřuje možné pokrytí testovacího případu autotestem
- „Actual AT Coverage“: může nabývat hodnot 0 – „Possible AT Coverage“. Toto pole je povinné. Vyjadřuje skutečné pokrytí testovacího případu autotestem.
- „Test Case Update“: Otevře dialog pro odeslání aktualizace testovacího postupu.

12. Manažer komentářů

| Status | Created | id test case | Test Case | Comment creator |
|--------|-----------------|--------------|--|---|
| New | 11.5.2011 15:04 | 612 | Kerio Connect - Admin/Engine -- ETRN download --- Send ETRN | Radek Šíp (rsip@kerio.com) Edit test case |
| New | 11.5.2011 15:00 | 612 | Kerio Connect - Admin/Engine -- ETRN download --- Send ETRN | Radek Šíp (rsip@kerio.com) Edit test case |
| New | 11.5.2011 14:39 | 607 | Kerio Connect - Admin/Engine -- POP3 download --- Sorting and Delivery | Radek Šíp (rsip@kerio.com) Edit test case |
| New | 11.5.2011 14:39 | 602 | Kerio Connect - Admin/Engine -- Scheduling --- Time condition | Radek Šíp (rsip@kerio.com) Edit test case |
| New | 11.5.2011 14:38 | 612 | Kerio Connect - Admin/Engine -- ETRN download --- Send ETRN | Radek Šíp (rsip@kerio.com) Edit test case |
| New | 11.5.2011 14:38 | 606 | Kerio Connect - Admin/Engine -- POP3 download --- POP3 Account | Radek Šíp (rsip@kerio.com) Edit test case |
| New | 11.5.2011 13:16 | 1505 | Kerio Connect - Admin/Engine -- Other tests --- Report Store Statistics | Radek Šíp (rsip@kerio.com) Edit test case |
| New | 11.5.2011 13:13 | 1505 | Kerio Connect - Admin/Engine -- Other tests --- Report Store Statistics | Radek Šíp (rsip@kerio.com) Edit test case |
| New | 10.5.2011 17:17 | 560 | Kerio Connect - Admin/Engine -- Antivirus control --- Sophos antivirus update | Radek Šíp (rsip@kerio.com) Edit test case |
| | | | Kerio Connect - Admin/Engine | |

- V manažeru komentářů jsou přijaté návrhy na úpravu testovacích případů.
- V rolovací nabídce je možno vybrat, pro který produkt chceme návrhy na update zobrazit. Po vybrání produktu a stisku **Show** se návrhy zobrazí.
- Zaškrťovací pole slouží k vybrání, jaké návrhy na update se mají zobrazit, zdali nové, schválené, zamítnuté anebo staré. Je možno vybrat více možností najednou.
- Stiskem volby „Edit test case“ se otevře nové okno s formulářem pro schválení navrhovaného updatu.

12.1. Schválení návrhu na update

| v. 1 New Test Case | | | | | Save as New | Update | Zdeněk Samuel | Logout |
|---|------------|----------------|----------------|---------------------|------------------------------------|--------|---------------|--------|
| Kerio Connect - Test Case Description | | | | | Comment detail | | | |
| Component | Chapter | Validity start | Estimated time | New dependency | | | | |
| Admin/Engine | 60 | 6.1.0-RC2 | 0:05 | Select new ---> | | | | |
| | SubChapter | Validity end | | Selected dependency | | | | |
| | 1 | unspecified | | | | | | |
| | | | | | <input type="checkbox"/> Modify AT | | | |
| Title | | | | | | | | |
| General | | | | | | | | |
| Description | | | | | | | | |
| Czech: Vytvořit několik uživatelů (login name, full name, description, internal user database, password) Ověřit (webmail, telnetem na pop3, ...) že vytvoření uživatelé mají možnost přijímat a odesílat e-maily Zkusit se přihlásit se špatným heslem Ověřit funkčnost změny hesla v administrační konzoli English: Create some users (login name, full name, description, internal user database, password) Verify (WebMail, telnet to pop3, ...) created users can send and receive emails Try to login with incorrect password - must be unsuccessful Verify functionality of password changing in KMS Admin | | | | | | | | |
| 1 | | | | | | | | |

- Po obdržení komentáře je potřeba návrh nejdříve schválit tlačítkem **Accept comment** a následně upravit popis testovacího případu nebo návrh zamítnout tlačítkem **Reject & Close**. Při zamítnutí návrhu dojde k zavření formuláře.
- Po úpravě popisu testovacího případu a stisku tlačítka **Update** se aktualizuje stávající popis testovacího případu. Při rozsáhlejší změně je možno testovací případ uložit jako nový a to stiskem tlačítka **Save as New**.

13. Admin sekce

13.1. Uživatelé

The screenshot displays the 'Admin section - Users' interface in KERIO TestRunner 2.2. On the left, there is a sidebar with navigation links: Users, Platforms, Dependencies, Components, Product versions, Products, and Logs. The main area is split into two parts. On the left, a 'New user' form is open, featuring input fields for 'User name', 'Full name', and 'E-Mail address'. It also includes dropdown menus for 'Domain' and 'Role' (currently set to 'tester'), and a checkbox for 'Be assigned to bugs from permanent On Demand (Administrators only)'. A 'Create' button is located at the top right of the form. On the right, a table lists existing users. The table has three columns: 'User Name', 'Role', and 'Edit'. The 'User Name' column contains the text 'tester' repeated 15 times. The 'Role' column contains 'guest' for the first 10 rows and 'tester' for the remaining 5 rows. Each row has an 'Edit' icon in the third column.

- Uživatelé budou řazeni abecedně dle uživatelského jména (zobrazuje se také).
- Username je shodné s usernamem do domény (kvůli automatickému ověřování).
- Vytvářený uživatel musí mít vyplněné všechny položky.
- Mazání uživatele není povoleno. Řeší se odebráním práv.
- Uživatelům, kteří budou mít v TestRunneru účet, lze přiřadit preferovaný jazyk (čeština, angličtina). Dle tohoto nastavení se budou zobrazovat testovací postupy bez ohledu na jazyk prohlížeče.
- Každému nově přidanému uživateli přijde informativní zpráva o jeho vytvoření v TestRunneru. To samé platí i pro změnu údajů.

13.2. Platformy

The screenshot displays the 'Admin section - Platforms' interface. On the left, a sidebar contains navigation links: Users, Platforms, Dependencies, Components, Product versions, Products, and Logs. The main area shows a table of platforms with columns for 'Platform', 'Edit', and 'Delete'. A modal window titled 'New platform' is open, showing a form with a 'Name' field and a 'Create' button. The list of platforms includes: All supported, Apple Mail, BlackBerry, Cross browsers, Cross platforms, Debian, Debian / Ubuntu, Firefox, Firefox 1.x, Firefox 2 / 3, Firefox 2.x, Firefox 3.x, Firefox 4.x, Google Chrome, HWA1000 (small box), HWA3000 (big box), Internet Explorer, Internet Explorer 6, Internet Explorer 7, Internet Explorer 8, Linux, Linux / Mac, MS Entourage 2004, MS Entourage 2008, and MS Outlook 2000.

- Platformy budou řazeny abecedně.
- Nelze vytvořit platformu, která by jako jméno měla prázdný řetězec. Nelze vytvořit dvě stejné platformy.
- Platformu lze vymazat, pouze pokud není použita v žádném testu. V takovém případě se zobrazí symbol pro mazání.

13.3. Závislosti na platformě

The screenshot displays the 'Admin section - Dependencies' interface. On the left, a sidebar contains navigation links: Users, Platforms, Dependencies, Components, Product versions, Products, and Logs. The main area shows a tree view of dependencies. The tree structure is: Operating systems, Browsers, Desktop, Chrome, Safari, Firefox, Internet Explorer (6, 7, 8, 9, 10, 11), Mobile, and Applications. The '10' version of Internet Explorer is selected.

- Závislost bude vytvořena s ohledem na právě vybraný prvek stromu. To znamená že se stane potomkem zvolené závislosti.

- Závislost lze vymazat, pouze pokud není použita v žádném testu nebo není rodičem jiné závislosti.

13.4. Komponenty

The screenshot displays the 'Admin section - Components' interface in KERIO TestRunner 2.2. A modal window for creating a new component is active, showing a 'Name' input field and an 'in product' dropdown menu currently set to 'Kerio Connect'. The background table lists various components and their associated products.

| Component | Product | Edit | Delete |
|------------------------|---------------|------|--------|
| Admin/Engine | Kerio Connect | | |
| CalDAV (iCal) | Kerio Connect | | |
| CardDAV (Address Book) | Kerio Connect | | |
| Dashboard | | | |
| Domain WebAdmin | | | |
| Entourage | | | |
| Installations | | | |
| KAdmin | | | |
| KOC | | | |
| KOFF | | | |
| KSP | | | |
| Mobile Devices | | | |
| On demand | | | |
| Outlook Mac | | | |
| Registration | | | |
| Release | | | |
| WebAdmin | | | |
| WebMail | | | |
| WebMail Mini | | | |
| Admin/Engine | | | |
| Installations | Kerio Control | | |
| KAdmin | Kerio Control | | |
| On demand | Kerio Control | | |
| Registration | Kerio Control | | |
| Release | Kerio Control | | |

- Jednotlivé jména komponent budou řazeny abecedně.
- Nelze vytvořit prázdnou komponentu nebo komponentu, aniž by bylo uvedeno, ke kterému je produktu.
- Nelze vytvořit dvě stejné komponenty v jednom produktu.
- Komponentu lze smazat, pouze pokud není v žádném testu a pokud neobsahuje žádný testovací případ. V takovém případě se zobrazí symbol pro mazání.

13.5. Produktové verze

| Product | Product version | Edit |
|---------------|-----------------|------|
| Kerio Connect | 7.3.0 | |
| Kerio Connect | 7.2.1 | |
| Kerio Connect | 7.2.0 | |
| Kerio Connect | 7.1.4 | |
| Kerio Connect | 7.1.3 | |
| Kerio Connect | 7.1.2 | |
| Kerio Connect | 7.1.1 | |
| Kerio Connect | 7.1.0 | |
| Kerio Connect | 7.0.2 | |
| Kerio Connect | 7.0.1 | |
| Kerio Connect | 7.0.0 | |
| Kerio Connect | 6.7.3 | |
| Kerio Connect | 6.7.2 | |
| Kerio Connect | 6.7.1 | |
| Kerio Connect | 6.7.0 | |
| Kerio Connect | 6.6.2 | |
| Kerio Connect | 6.6.1 | |
| Kerio Connect | 6.6.0 | |
| Kerio Connect | 6.5.2 | |
| Kerio Connect | 6.5.1 | |
| Kerio Connect | 6.5.0 | |
| Kerio Connect | 6.4.3 | |
| Kerio Connect | 6.4.2 | |
| Kerio Connect | 6.4.1 | |
| Kerio Connect | 6.4.0 | |

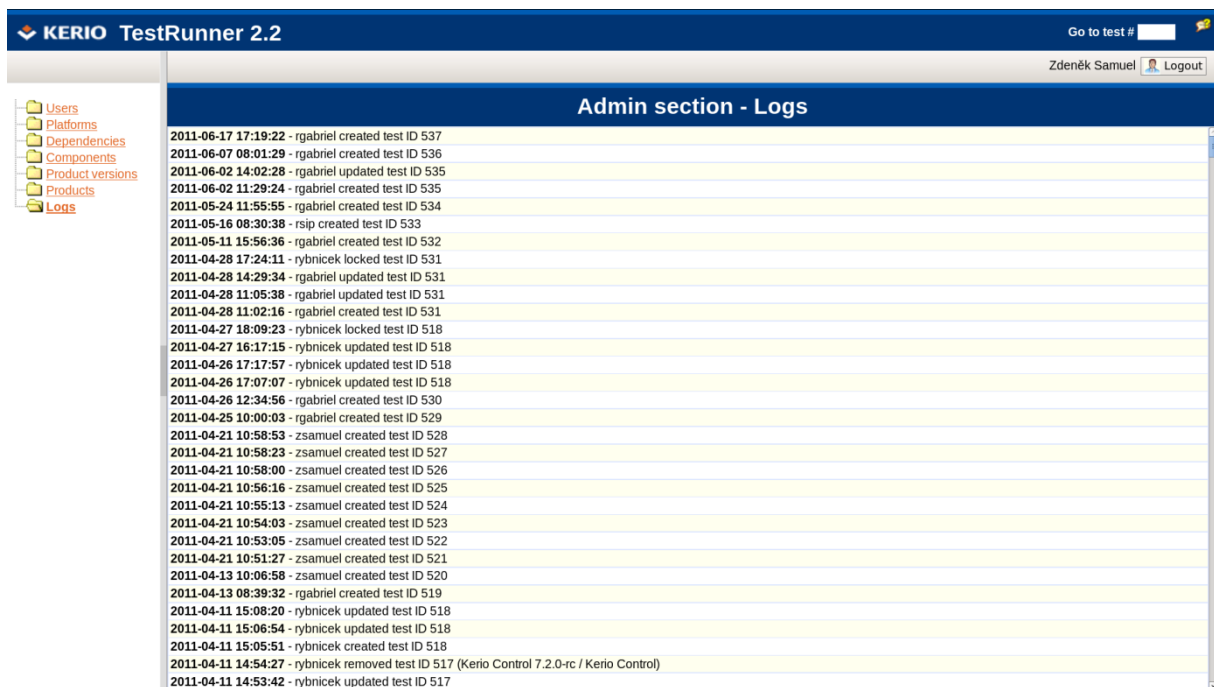
- Produktové verze budou načítány z Bugzilly.

13.6. Produkty

| Name | Short name | Edit |
|--------------------|----------------|------|
| Kerio Connect | Kerio Connect | |
| Kerio Control | Kerio Control | |
| Kerio Directory | KDir | |
| Kerio Operator | Kerio Operator | |
| Kerio Sales System | KISS | |
| Kerio Workspace | | |

- Produkty budou řazeny abecedně.
- Produktu bude přiřazeno jméno, zkrácené jméno a ID v Bugzille.

13.7. Logy



KERIO TestRunner 2.2 Go to test #

Zdeněk Samuel [Logout](#)

Admin section - Logs

| | |
|---------------------|---|
| 2011-06-17 17:19:22 | - rgabriel created test ID 537 |
| 2011-06-07 08:01:29 | - rgabriel created test ID 536 |
| 2011-06-02 14:02:28 | - rgabriel updated test ID 535 |
| 2011-06-02 11:29:24 | - rgabriel created test ID 535 |
| 2011-05-24 11:55:55 | - rgabriel created test ID 534 |
| 2011-05-16 08:30:38 | - rsp created test ID 533 |
| 2011-05-11 15:56:36 | - rgabriel created test ID 532 |
| 2011-04-28 17:24:11 | - rybnicek locked test ID 531 |
| 2011-04-28 14:29:34 | - rgabriel updated test ID 531 |
| 2011-04-28 11:05:38 | - rgabriel updated test ID 531 |
| 2011-04-28 11:02:16 | - rgabriel created test ID 531 |
| 2011-04-27 18:09:23 | - rybnicek locked test ID 518 |
| 2011-04-27 16:17:15 | - rybnicek updated test ID 518 |
| 2011-04-26 17:17:57 | - rybnicek updated test ID 518 |
| 2011-04-26 17:07:07 | - rybnicek updated test ID 518 |
| 2011-04-26 12:34:56 | - rgabriel created test ID 530 |
| 2011-04-25 10:00:03 | - rgabriel created test ID 529 |
| 2011-04-21 10:58:53 | - zsamuel created test ID 528 |
| 2011-04-21 10:58:23 | - zsamuel created test ID 527 |
| 2011-04-21 10:58:00 | - zsamuel created test ID 526 |
| 2011-04-21 10:56:16 | - zsamuel created test ID 525 |
| 2011-04-21 10:55:13 | - zsamuel created test ID 524 |
| 2011-04-21 10:54:03 | - zsamuel created test ID 523 |
| 2011-04-21 10:53:05 | - zsamuel created test ID 522 |
| 2011-04-21 10:51:27 | - zsamuel created test ID 521 |
| 2011-04-13 10:06:58 | - zsamuel created test ID 520 |
| 2011-04-13 08:39:32 | - rgabriel created test ID 519 |
| 2011-04-11 15:08:20 | - rybnicek updated test ID 518 |
| 2011-04-11 15:06:54 | - rybnicek updated test ID 518 |
| 2011-04-11 15:05:51 | - rybnicek created test ID 518 |
| 2011-04-11 14:54:27 | - rybnicek removed test ID 517 (Kerio Control 7.2.0-rc / Kerio Control) |
| 2011-04-11 14:53:42 | - rybnicek updated test ID 517 |

Do logu se budou vypisovat následující akce:

- vytvoření, update a smazání testu
- zamknutí a odemknutí testu

Každý log bude mít následující schéma:
datum (rok-měsíc-den) čas (hh:mm) - uživatelské_jméno akce ID_testu (viz obrázek).