

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 8. srpna 2013

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	Tým	4
2.3	Vývoj	4
2.3.1	Agilní metody	4
2.3.2	Scrum obecně	6
2.3.3	Testování ve scrumu	8
2.3.4	Unit testování	10
2.3.5	Zátěžové testování	10
2.4	Stabilizace produktu	11
2.4.1	Obecně	11
2.4.2	Pre-alpha	11
2.4.3	Alpha	12
2.4.4	Beta	12
2.4.5	Release candidate	12
2.5	Automatické testování	13
2.5.1	AutoBuild	13
2.5.2	Autotest	13
3	TestRunner	14
3.1	Obecně	14
3.2	Popis funkčnosti	15
3.3	Specifikace požadovaných změn	15

3.3.1	Dashboard	15
3.3.2	Redukce práv	15
3.3.3	Závislost test casů na platformě	16
3.3.4	Vyhledávání	16
3.3.5	Dvojjazyčnost	16
3.3.6	Proměnlivá velikost levého panelu	16
3.3.7	Bugfix	16
3.4	Aktuální programové prostředí	18
3.5	Přechod na nový operační systém	19
3.6	Analýza	19
3.7	Oprava chyb	19
3.7.1	Tlačítka Won't test a Reset	20
3.7.2	Manažer komentářů	20
3.7.3	Ostatní	20
3.8	Odebrání nadbytečných funkcí	21
3.8.1	Redukce uživatelských rolí	21
3.8.2	Zrušení dvojjazyčnosti	22
3.9	Nové funkce	22
3.9.1	DashBoard	22
3.9.2	Vyhledávání	24
3.9.3	Součty průměrných časů testů pro kapitoly	24
3.9.4	Proměnlivá velikost levého panelu	24
3.9.5	Vazba testovacích případů s platformou	26
3.9.6	Tooltipy	28
4	Testy TestRunneru	31
5	Závěr	32

1 Úvod

V této práci se budu zabývat testováním softwarových produktů. Tato oblast spadá do odvětví, které se nazývá proces vývoje softwaru. 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. K tomu využiji reverzního inženýrství. Drobné změny v kódu mi pomohou v lepším pochopení celého programu. Program následně vyčistím od přebytečných funkcí a některé části kódu inovuji. Nakonec implementuji nový autentizační systém a další požadované funkce.

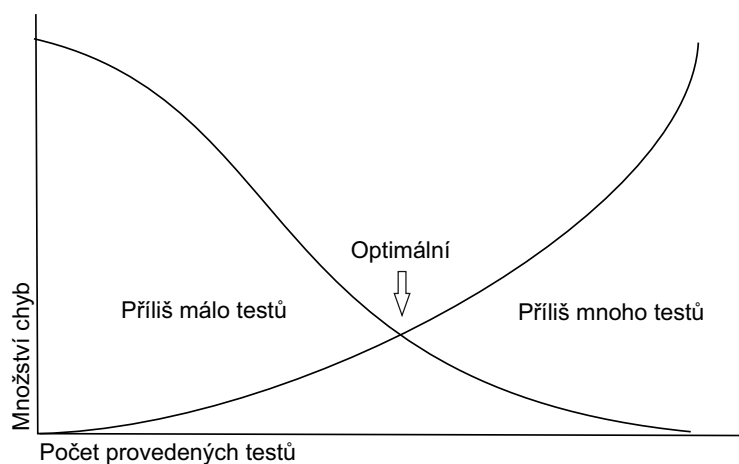
Před předáním program důkladně otestuji. V rámci předání, budu změny prezentovat celému QA oddělení Kerio Technologies s.r.o.. Vzhledem k tomu, že rozsah změn je poměrně velký, program předám a nasadím ve dvou iteracích.

2 Problematika testování

2.1 Testování obecně

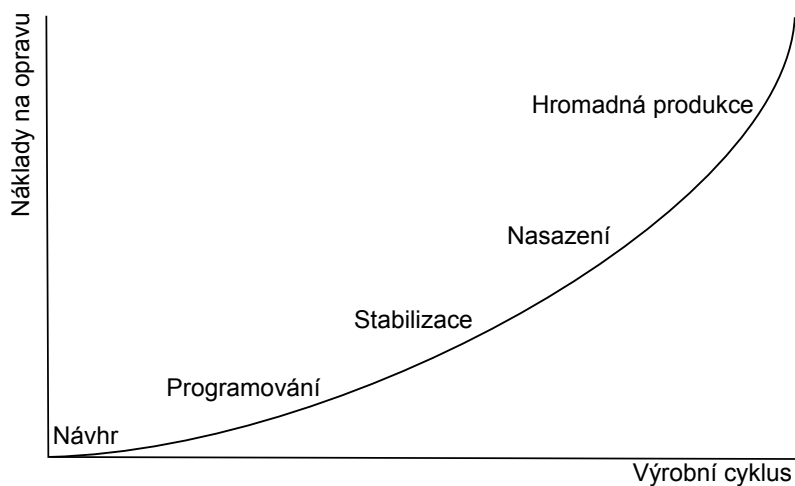
Hlavní myšlenkou testování je zjišťování kvality testovaného produktu. V našem případě je oním produktem software. Důvod, proč se zajímáme o testování, je jednoduchý. U softwaru nemůžeme nikdy dosáhnout perfektního vývojového cyklu. Abychom se mu mohli alespoň přiblížit, je bezpodmínečně nutné, aby byl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

Bohužel i sebelepší testovací strategie nám 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.[17]

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 částech vývoje.



Obrázek 2.2: Náklady na opravu chyby v průběhu výrobního cyklu [18].

Základním stavebním kamenem testování jsou iterace. Produkt se z pravidla předkládá k testování s každou novou verzí. Proto je velké množství chyb odhaleno velice brzy. Bohužel je velké množství chyb, které se projeví až v pozdější části vývoje. Je to způsobeno hlavně tím, že v raných etapách ještě není možné jednoznačně zkontrolovat splnění některých požadavků.

2.2 Tým

Oddělení Quality assurance (zkráceně QA) je nezávislý tým testerů, který je součástí vývojového oddělení. Jejich hlavní zásadou je: Testovat lépe než více. Protože náplní testera je nacházet chyby v práci programátora, mohou vznikat konflikty. Proto tester musí mít nejen talent k hledání chyb ale i určité diplomatické kvality.

2.3 Vývoj

2.3.1 Agilní metody

Kerio Technologies s.r.o. k vývoji svých využívá agilních metod. Program nemá 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.



Obrázek 2.3: Rozdíl mezi klasickým a agilním přístupem [15].

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

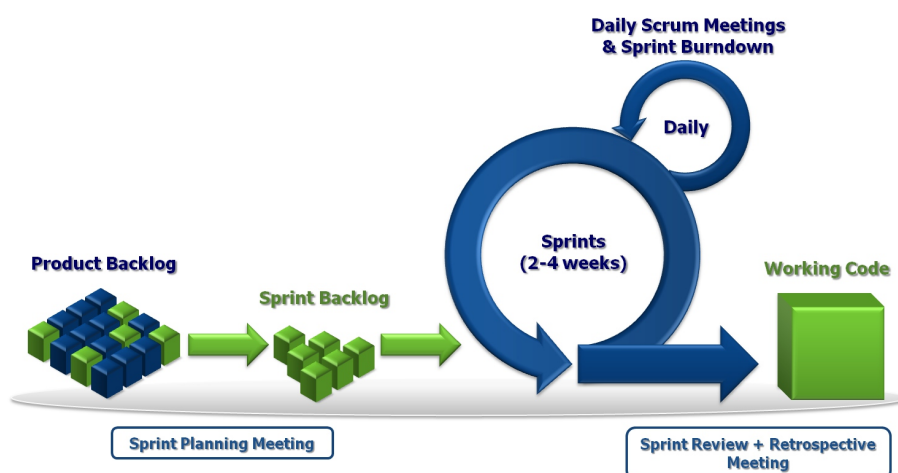
- Nultá iterace: Základ aplikace
 1. Analýza problému
 2. Naprogramování základní kostry programu.
 3. Hlavním cílem je základ, který lze demonstrovat.
- První až N-tá iterace: Vývoj aplikace.
 1. Výběr implementované funkcionality a analýza postupu.
 2. Implementace.
 3. Build, Testování, Předvedení.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

A nyní již k hlavním výhodám agilních technik:

1. Je dobře známo kolik práce tým zvládne v jedné iteraci udělat a díky tomu řídit rychlost týmu.
2. Možnost změny zadání za běhu.
3. Rychlé doručení produktu k zákazníkovi.

2.3.2 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. Scrum dostal své jméno podle každodenních schůzek týmu, scrumů. Na těchto schůzkách se aktualizuje postup na projektu, dokončené úkoly, následující úkoly a případné problémy. Součástí je aktuální a plně funkční verze produktu. Nespornou výhodou scrumu je také schopnost samostatné organizace týmu. Manažerovi tedy výrazně ubude práce. A protože k scrumu není potřeba žádný dodatečný trénink, je velice jednoduché s ním začít. Přibližná funkčnost scrumu viz obr. 2.4.



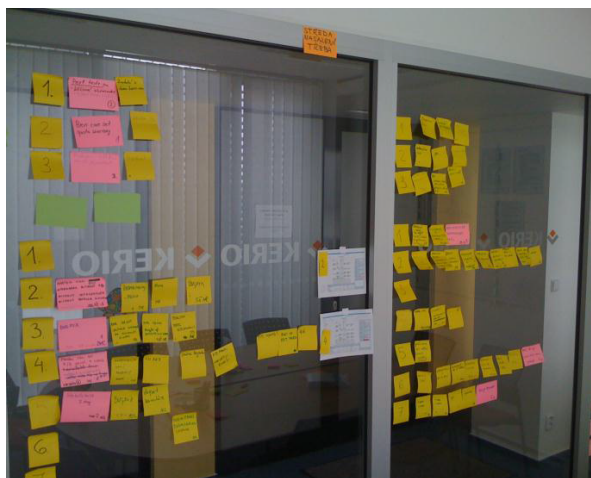
Obrázek 2.4: Jak funguje Scrum [13].

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 příslušenství ve formě takzvaných artefaktů:

- 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[13]. Sprint backlog má na starosti vývojový tým a bývá často realizován jako scrum board viz obr. 2.5.



Obrázek 2.5: Jak vypadá scrum board [18].

2.3.3 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. Na první pohled by se mohlo zdát výhodné vymezení konkrétního výstupu programu. Bohužel často existuje mnoho cest jak dosáhnout cíle a uživatel si často vybere jinou než námi testovanou. Proto zavedeme testování které toto zohlední a bude vytvářet různé cesty průchodem programu. Tzv. test case. 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.

Každý test je navržen bez znalosti jakýchkoliv vnitřních struktur programu. Tomuto postupu se říká Black-box testování viz obr. 2.6.



Obrázek 2.6: Black-box testování [2].

Další velice důležitou vlastností testu je jednoznačnost. U každého testu musíme být schopni určit, jestli test dopadl či nedopadl úspěšně. Takovéto testy je pak velice jednoduché zautomatizovat. 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, vytvoří testy a provedou je.

2.3.4 Unit testování

Zatím jsme testování brali pouze ze stránky testera. Ovšem i programátor může eliminovat velké množství chyb použitím unit testů. Ty jsou založené na testování části kódu ještě před samotnou implementací. Díky tomuto členění vzniká velice dobře testovatelný, upravitelný a čitelný kód.

2.3.5 Zátěžové testování

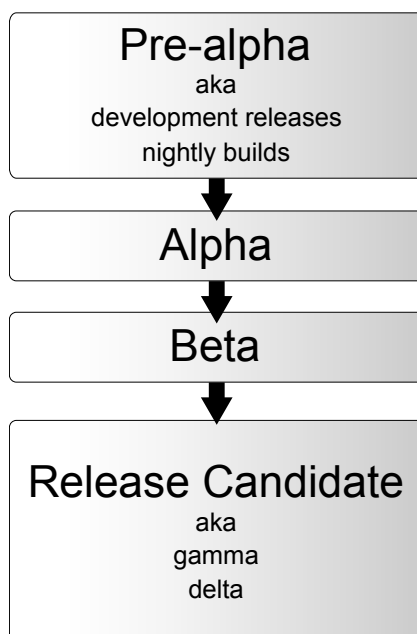
Testování nezahrnuje pouze funkčnost, ale také jak se program bude chovat v extrémních situacích. K tomu nám pomůže zátěžové testování, kde budeme testovat stabilitu v následujících situacích:

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

2.4 Stabilizace produktu

2.4.1 Obecně

V této fázi byla již naprogramována veškerá funkcionální a zbývá program doladit. Testeři zjišťují poslední chyby. Vývojáři kontrolují kód. Jakákoliv další vylepšení se už nedělají. Fáze obvykle trvá jeden až dva měsíce, proto se již neuplatňují agilní techniky. Je ovšem snaha tuto dobu pokud možno co nejvíce zkrátit. Stabilizační fázi lze dále rozdělit viz obr. 2.7.



Obrázek 2.7: Stabilizace produktu [11].

2.4.2 Pre-alpha

Předtím, než může produkt postoupit do alpha verze, je třeba ověřit, zdali jsou opraveny všechny nahlášené chyby. Všechny testovací případy (test cases) jsou sdruženy do skupin (test suites). 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.

2.4.3 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.

2.4.4 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.

2.4.5 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.

2.5 Automatické testování

2.5.1 AutoBuild

Autobuild překládá každou novou verzi programu ze zdrojových kódů. Díky tomu vývojář ihned ví, zdali neobsahuje kritickou chybu. Je neocenitelným pomocníkem při hledání právě takovýchto chyb. Pokud je chyba detekována, oprava získá nejvyšší prioritu. Tím je zajištěna vitalita celého systému.

2.5.2 Autotest

Autotest je nástroj, který v mnoha ohledech usnadňuje testerovi práci. To se projeví hlavně pokud máme mnoho různých prostředí, pro která je program vyvíjen. Autotest je zvládne pokrýt všechna zároveň. Zároveň funguje vždy stejně, a je tedy eliminován lidský faktor (v případě, že je test napsán správně). Jako nevýhody autotestu můžeme považovat časovou náročnost na vývoj a údržbu. Bohužel ani autotesty nemohou testovat všechno, a tak je nelze používat jako jedinou metodu testování.

Autotesty najdou nejčastěji využití pozdějších etapách testování, kdy se funkcionality programu příliš nemění. Důvodem je nižší režie na vývoj a údržbu.

3 TestRunner

3.1 Obecně

TestRunner je program pokrývající základní potřeby testování v QA Kerio Technologies s.r.o.. Byl vytvořen pro tyto účely:

1. vytvářet variabilní testy
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 ZČÚ. Od jeho vytvoření uběhlo již několik let a za tu dobu kód postupně rostl a stal se špatně udržovatelným [16]. 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

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 požadovaných změn

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 test casů na platformě

- některé test case nedávají na některých platformách smysl - měly by být svázány s konkrétní platformou
- vytvořit strom závislostí test casů 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 dvojjazyčnosti
- popisy testů musí být zachovány a sloučeny dohromady

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 testcase testován. Tlačítko ale funguje.
- Pokud je testcase 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 test case ve stavu wont"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 reloadne). 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 vyresetuje 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 [5] [14]
- PHP ve verzi 5.1.6 [12] [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 [9]), 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 tomu 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 [10]). 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 na projektu vystřídalo velké množství různých přístupů a řešení.

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

Tyto tlačítka jsou součástí detailu test case. 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 manažeru komentářů je uveden v uživatelské příručce kapitola 12.

1. Nereloaduje se 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 vitality TestRunneru i v budoucnu bylo třeba odebrat nadbytečné funkce.

3.8.1 Redukce uživatelských rolí

Vzhledem k tomu, že většinou ke každému týmu je 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

na polovinu a to konkrétně:

- Guest
- Tester
- Administrátor

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

3.8.2 Zrušení dvojjazyčnosti

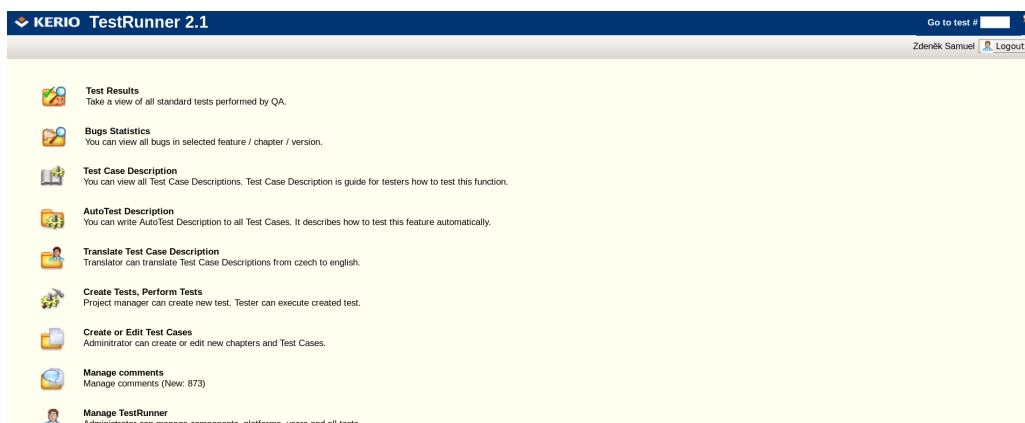
Postupem času se pro zaměstnance společnosti Kerio stala angličtina standardem. V tu chvíli se ze zajišťování dvojjazyčnosti stala zbytečná režie která musí být odstraněna. Bohužel byla ze své podstaty v programu poměrně pevně zakořeněna a její odstranění vyžadovalo rozsáhlý průzkum kódu i důkladné testování.

Skoro na každém z příložených obrázků můžeme nalézt některou ze součástí této funkcionality. Konkrétně se jedná o položku Translate Test Case Description viz obr. 3.1, která zajišťovala funkcionality pro překladatele. Dále se jedná o vlaječky pro přepínání popisů viz obr. 3.9 a 3.3. Posledním ilustračním příkladem je položka translated a dvojí popis test casu viz obr. 3.7. Za zmínku ještě například stojí zobrazování ikon u nepřeložených test case přímo ve stromu v levé části, které na obrázcích není vidět. Případně také jazykové preference pro každého uživatele.

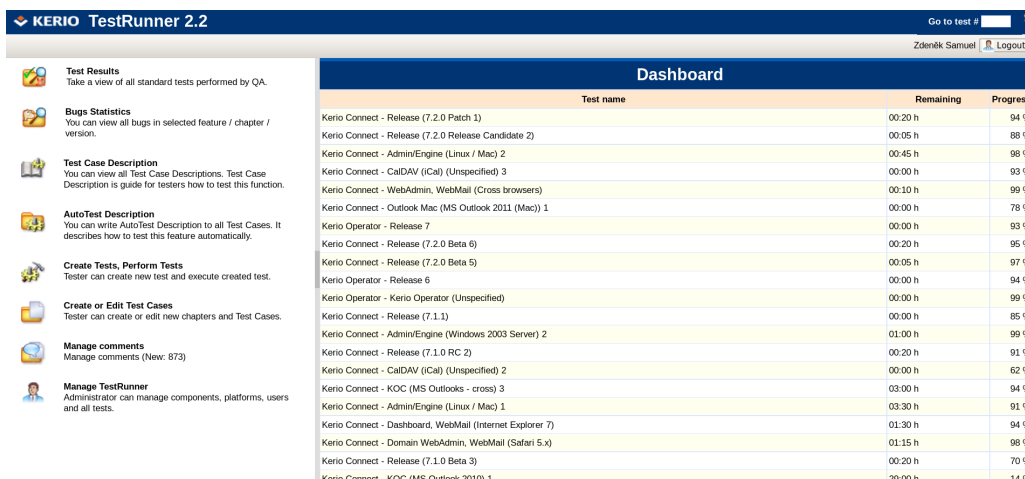
3.9 Nové funkce

3.9.1 DashBoard

Už na první pohled vypadá úvodní stránka nevyužitá. Bylo tedy rozhodnuto, že volný prostor bude sloužit k zobrazování aktuálně běžících testů. Stránka tedy byla rozdělena na dvě části. Aby bylo možné stránku lépe organizovat, rozhodl jsem se pro implementaci tzv. Split pane. Díky tomu je možné obě části libovolně zvětšovat či zmenšovat. DashBoard dále nabízí přihlášenému uživateli možnost rychlého přesunu na kterýkoliv test. Pro lepší orientaci v testech je implementováno řazení podle jakéhokoliv sloupce (po kliknutí na hlavičku) a barevné odlišení sudých a lichých řádek viz obr. 3.2.



Obrázek 3.1: Hlavní stránka - původní.



Obrázek 3.2: Hlavní stránka - nová.

3.9.2 Vyhledávání

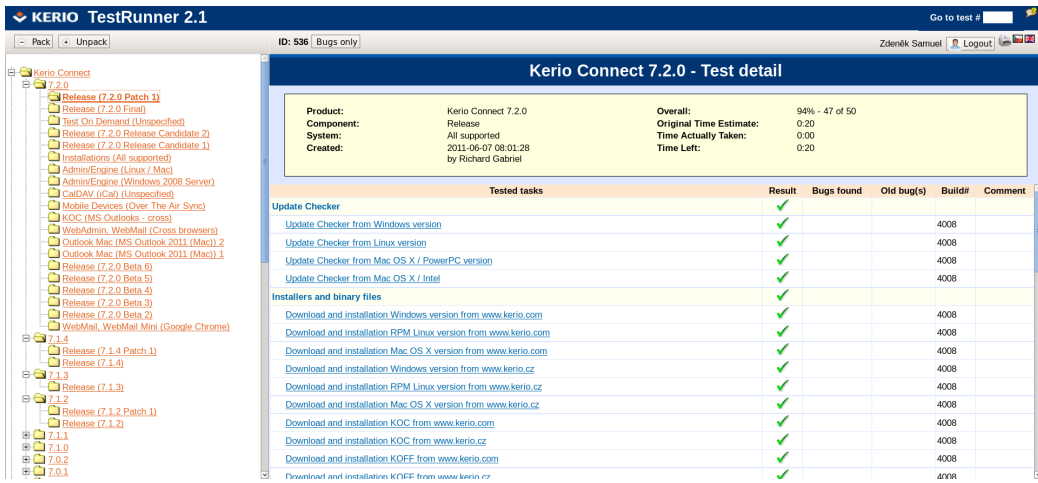
Vyhledávání bylo připraveno již z předchozí verze v pohledu "Test Case Description" a na mě zbývalo doplnění do částí "AutoTest Description" a "Create or Edit Test Cases". Naštěstí bylo vyhledávání připraveno poměrně obecně a jeho doplnění nebylo příliš náročné. Ukázkou včetně příkladu lze nalézt v uživatelské příručce.

3.9.3 Součty průměrných časů testů pro kapitoly

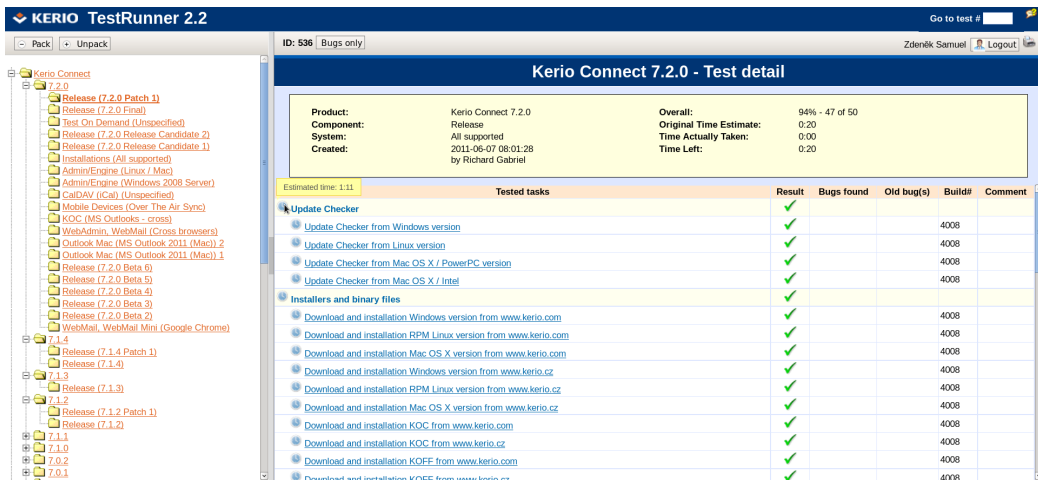
Hlavním důvodem pro implementaci byla možnost lepší organizace času testerů. V praxi by nyní měl tester lépe odhadnout, který test zvolit, když chce za hodinu odejít domů. Funkce viz obr. 3.4

3.9.4 Proměnlivá velikost levého panelu

S příchodem tabletů vznikla potřeba lepší optimalizace místa. Levý panel, který slouží k vyhledávání, je tedy po většinu času nepotřebný. Nyní může být v případě zmenšen či úplně schován dle aktuální potřeby viz obr. 3.4 a 3.8. Velikost levého panelu viz obr. 3.6 je předem známa, proto je aktivní pouze funkce schování. Implementace proběhla pomocí frameworku jQuery [8] a pluginu jQuery UI Layout.



Obrázek 3.3: Detail test suite - původní.



Obrázek 3.4: Detail test suite - nový.

3.9.5 Vazba testovacích případů s platformou

Analytická část

Asi nejsložitější částí bylo provázání jednotlivých test case s platformou. Prvním krokem byla volba vhodné implementace. Důkladným rozborem závislostí na platformě jsem dospěl reprezentaci stromem. viz 3.5. Strom byl oproti grafu jednoznačný a lépe implementovatelný.

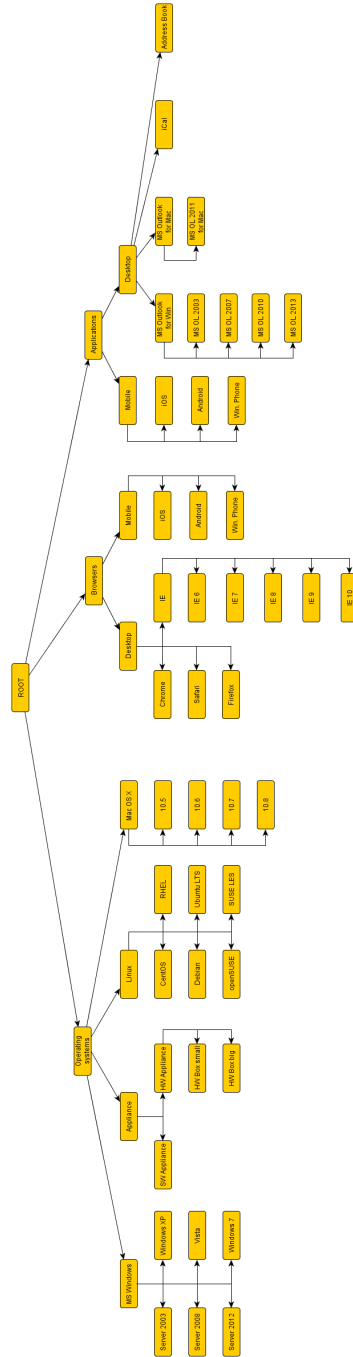
Výsledný strom nejevil sebemenší známky symetričnosti nebo pravidelnosti. Navíc do budoucna hrozilo, že by se mohl ještě výrazně měnit. Proto jsem dospěl k názoru, že nejlepší bude implementace pomocí rekurze a hlavně kvůli nepravidelnosti výsledného stromu. Protože se nepředpokládá výrazný růst stromu,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

Implementační část

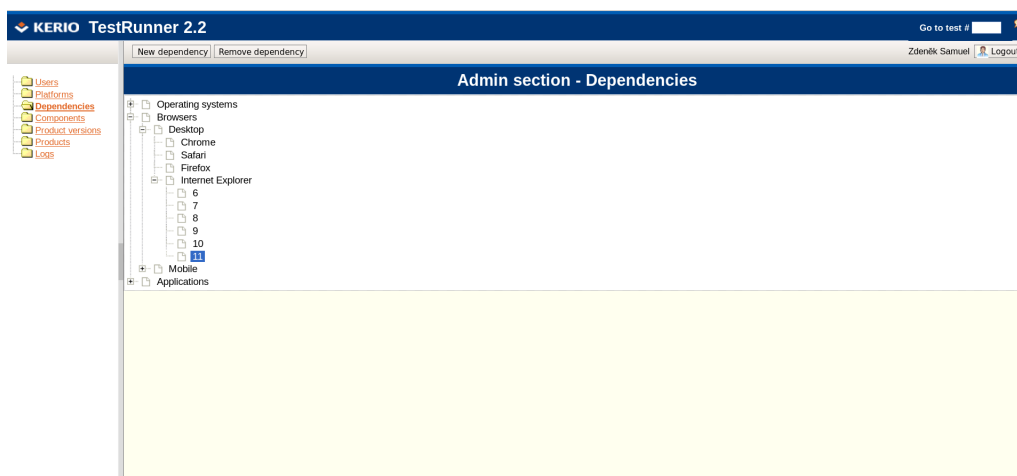
V praxi bude každá platforma jedním řádkem v tabulce databáze a bude si uchovávat pouze jméno, identifikační číslo své a svého rodiče. Každý test case může mít více závislostí na platformě a zároveň každá závislost může patřit k více test casům. Jedná se tedy o vazbu M:N, kde bude dále nutná rozkladová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

Aby bylo možné strom v budoucnu nadále upravovat, musel jsem do managementu TestRuneru přidat položku pro správu stromu viz obr. 3.6. První závislosti budou vloženy do databáze již při tvorbě tabulky, další pak podle potřeby. Bohužel přiřazení závislostí k jednotlivým test casům bude nutné provést ručně. To je možné jak pro nové, tak i aktivní test casey. Přidávání a zobrazení jednotlivých závislostí viz obr. 3.8.

Důvodem implementace této funkce je především okno pro vytváření test suite. Aktuální implementace načítá kompletně všechny test case, což nejen že trvá neúměrně dlouho, ale také znesnadňuje celkovou tvorbu.



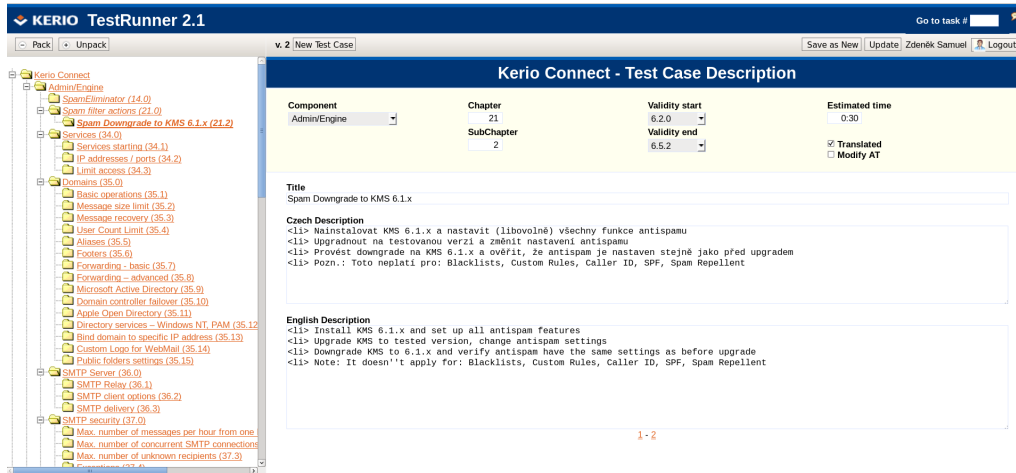
Obrázek 3.5: Strom platformem.



Obrázek 3.6: Tvorba stromu závislostí.

3.9.6 Tooltips

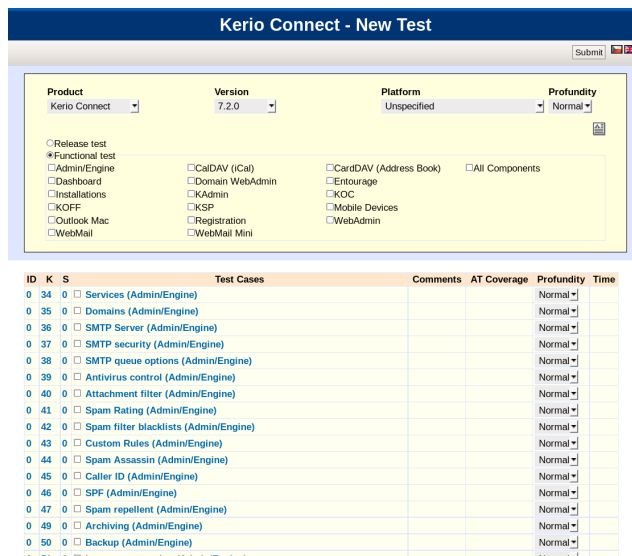
Jak už bylo zmíněno v předchozí části, okno pro tvorbu test suite je neúměrně pomalé. U mě na počítači pravidelně více než 10s. Jedním z důvodů bylo načítání všech tooltipů při otevření okna. Zběžným průzkumem mezi testery jsem zjistil, že je velká část z nich ani nepoužívá. Rozhodl jsem se je tedy načítat až ve chvíli, kdy o to bude někdo stát, konkrétně až při najetí myši, a to technologií Ajax pomocí frameworku jQuery a dodatečné knihovny qTip2. Původní implementace tooltipů byla z projektu TestRunneru odstraněna z důvodu nefunkčnosti při větším množství textu v tooltipu.



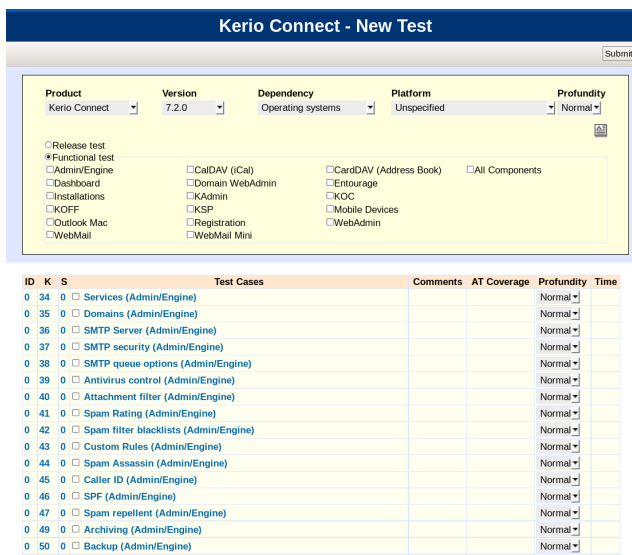
Obrázek 3.7: Editace test case - původní.



Obrázek 3.8: Editace test case - nová.



Obrázek 3.9: Tvorba test suity - původní.



Obrázek 3.10: Tvorba test suite - nová.

4 Testy TestRunneru

Kvůli vysoké časové režii, kterou vyžadují automatické testy, jsem se rozhodl využít pouze testy manuální.

První série testů probíhala již v rané fázi vývoje, kdy jsem testoval, zdali funkcionality odpovídá specifikaci. Následně jsem funkcionality konzultoval se zadavatelem a členy QA. Pokud nebyl zadavatel spokojen, změnil jsem ji, znovu otestoval a předvedl. Ve chvíli kdy byla funkcionality schválena, přešel jsem k další.

Přibližně v polovině práce jsem přikročil k testu celé aplikace. Průběhy jednotlivých testů probíhali na základě konzultací s jednotlivými členy QA. Důraz byl kladen na otestování TestRunneru pro časté úkony. Poté byla první verze nasazena k betatestingu a následně do ostrého provozu. Druhá polovina práce měla totožný průběh.

5 Závěr

V rámci zadání jsem se detailně seznámil s fungováním QA oddělení společnosti Kerio Technologies s.r.o. a doplnil vzdělání v oblastech virtualizace, operačního systému Linux, SQL databází, PHP, JavaScriptu[7], HTML[6], CSS[4], 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 se s databázovým modelem a zdrojovými kódy. Odstranil jsem všechny buggy 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.[17]	2
2.2	Náklady na opravu chyby v průběhu výrobního cyklu [18].	3
2.3	Rozdíl mezi klasickým a agilním přístupem [15].	4
2.4	Jak funguje Scrum [13].	6
2.5	Jak vypadá scrum board [18].	8
2.6	Black-box testování [2].	9
2.7	Stabilizace produktu [11].	11
3.1	Hlavní stránka - původní.	23
3.2	Hlavní stránka - nová.	23
3.3	Detail test suite - původní.	25
3.4	Detail test suite - nový.	25
3.5	Strom platforem.	27
3.6	Tvorba stromu závislostí.	28
3.7	Editace test case - původní.	29

3.8	Editace test case - nová.	29
3.9	Tvorba test suitu - původní.	30
3.10	Tvorba test suite - nová.	30

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] *CSS Tutorial* [online]. 2013. [cit. 2.8.2013]. Dostupné z: <http://www.w3schools.com/css/default.asp>.
- [5] *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.
- [6] *HTML Tutorial* [online]. 2013. [cit. 2.8.2013]. Dostupné z: <http://www.w3schools.com/html/default.asp>.
- [7] *JavaScript Tutorial* [online]. 2013. [cit. 2.8.2013]. Dostupné z: <http://www.w3schools.com/js/default.asp>.
- [8] *jQuery API* [online]. 2013. [cit. 2.8.2013]. Dostupné z: <http://api.jquery.com/>.
- [9] *Kerberos Protocol Tutorial* [online]. 2007. [cit. 2.8.2013]. Dostupné z: <http://www.kerberos.org/software/tutorial.html>.

-
- [10] *Lightweight Directory Access Protocol* [online]. 2013. [cit. 2.8.2013]. Dostupné z: <http://php.net/manual/en/book.ldap.php>.
- [11] *Software release life cycle* [online]. 2012. [cit. 2.8.2013]. Dostupné z: http://en.wikipedia.org/wiki/Software_release_life_cycle.
- [12] *PHP Manual* [online]. 2013. [cit. 2.8.2013]. Dostupné z: <http://php.net/manual/en/index.php>.
- [13] *Introduction to Scrum* [online]. 2012. [cit. 2.8.2013]. Dostupné z: <http://www.mountaingoatsoftware.com/presentations/an-introduction-to-scrum>.
- [14] *SQL Tutorial* [online]. 2013. [cit. 2.8.2013]. Dostupné z: <http://www.w3schools.com/sql/default.asp>.
- [15] BUCHALCEVOVÁ, A. Agilní metodiky. In *Objekty 2002*. Praha: Česká zemědělská univerzita (ČZU), 2002. ISBN 80-213-0947-4.
- [16] FOWLER, M. et al. *Refactoring: Improving the Design of Existing Code*. Addison-Wesley Professional, 1999. ISBN 0-201-48567-2.
- [17] PATTON, R. *Testování softwaru*. Computer Press, 2002. ISBN 80-7226-636-5.
- [18] SAMUEL, Z. Agile Testing. Prezentováno jako externí přednáška k předmětu ZSWI, 2010.


Přílohy


Uživatelský manuál


- [1. Titulní stránka](#)
 - [1.1. Uživatelská práva](#)
 - [1.2. Popis funkcí](#)
- [2. Seznam testů](#)
 - [2.1. Strom testů](#)
 - [2.2. Seznam testů](#)
 - [2.3. Ostatní funkce](#)
- [3. Vytváření nebo editace testů](#)
 - [3.1. Globální hodnoty](#)
 - [3.2. Komponenty](#)
 - [3.3. Výběr testovacích případů a jejich vlastností](#)
 - [3.4. Test na žádost](#)
 - [3.5. Ostatní](#)
- [4. Detail testu](#)
 - [4.1. Hlavička](#)
 - [4.2. Tasks Tested](#)
 - [4.3. Výsledky testů](#)
 - [4.4. Chyby](#)
 - [4.5. Ostatní](#)
- [5. Seznam chyb](#)
- [6. Testovací formulář](#)
 - [6.1. Hlavička](#)
 - [6.2. Odeslání výsledku](#)
 - [6.3. Vrácení do stavu „netestováno“](#)
 - [6.4. Won't test](#)
 - [6.5. Chyby](#)
 - [6.6. Descriptions](#)
 - [6.7. Ostatní](#)
- [7. Test Case Update](#)
- [8. Seznam testovacích případů](#)
 - [8.1. Strom testovacích případů](#)
 - [8.2. Test Case detail](#)
- [9. Vyhledávání v testovacích případech](#)
- [10. Manipulace s testovacími případy](#)
 - [10.1. Strom testovacích postupů](#)
 - [10.2. Řídící informace](#)
 - [10.3. Editační pole](#)
 - [10.4. Ukládání nově vytvořených testovacích případů](#)
 - [10.5. Ukládání updatů testovacích případů \(klony\)](#)
- [11. Popis autotestu](#)
 - [11.1. Strom testovacích případů](#)
 - [11.2. Editační pole](#)
- [12. Manažer komentářů](#)
 - [12.1. Schválení návrhu na update](#)
- [13. Admin sekce](#)
 - [13.1. Uživatelé](#)
 - [13.2. Platformy](#)
 - [13.3. Závislosti na platformě](#)
 - [13.4. Komponenty](#)
 - [13.5. Produktové verze](#)
 - [13.6. Produkty](#)
 - [13.7. Logy](#)


1. Titulní stránka


KERIO TestRunner 2.2


 **Test Results**
Take a view of all standard tests performed by QA.


 **Bugs Statistics**
You can view all bugs in selected feature / chapter / version.


 **Test Case Description**
You can view all Test Case Descriptions. Test Case Description is guide for testers how to test this function.

 **AutoTest Description**
You can write AutoTest Description to all Test Cases. It describes how to test this feature automatically.

 **Create Tests, Perform Tests**
Tester can create new test and execute created test.

 **Create or Edit Test Cases**
Tester can create or edit new chapters and Test Cases.

 **Manage comments**
Manage comments (New: 873)

 **Manage TestRunner**
Administrator can manage components, platforms, users and all tests.

Dashboard

Test name
Kerio Connect - Release (7.2.0 Patch 1)
Kerio Connect - Release (7.2.0 Release Candidate 2)
Kerio Connect - Admin/Engine (Linux / Mac) 2
Kerio Connect - CalDAV (iCal) (Unspecified) 3
Kerio Connect - WebAdmin, WebMail (Cross browsers)
Kerio Connect - Outlook Mac (MS Outlook 2011 (Mac)) 1
Kerio Operator - Release 7
Kerio Connect - Release (7.2.0 Beta 6)
Kerio Connect - Release (7.2.0 Beta 5)
Kerio Operator - Release 6
Kerio Operator - Kerio Operator (Unspecified)
Kerio Connect - Release (7.1.1)
Kerio Connect - Admin/Engine (Windows 2003 Server) 2
Kerio Connect - Release (7.1.0 RC 2)
Kerio Connect - CalDAV (iCal) (Unspecified) 2
Kerio Connect - KOC (MS Outlooks - cross) 3
Kerio Connect - Admin/Engine (Linux / Mac) 1
Kerio Connect - Dashboard, WebMail (Internet Explorer 7)
Kerio Connect - Domain WebAdmin, WebMail (Safari 5.x)
Kerio Connect - Release (7.1.0 Beta 3)
Kerio Connect - KOC (MS Outlook 2010) 1
Kerio Sales System - KISS (Unspecified)

Tento dialog bude domovskou stránku aplikace TestRunner 2.2. Standardně se budou zobrazovat 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
Translate Test Case 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


 **Login** 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 tlačítka v každém dialogu se změni na „Logout“ (popis i funkce). Při stisknutí tlačítka "Logout" se provede odhlášení uživatele ze systému a provede se přesměrování na titulní stránku aplikace.


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

 Zobrazí se aktuální stránka online html manuálu. Podoba a 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 přidat 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, platform, závislostí, uživatelů a logů.

2. Seznam testů




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

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 jak 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  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

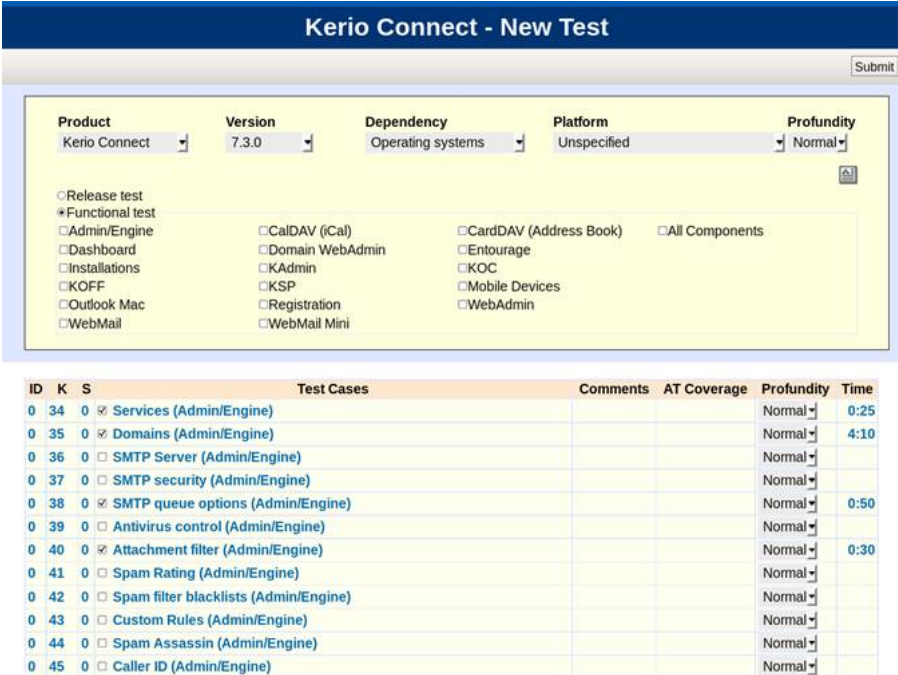
 Po stisku tohoto tlačítka se otevře stránka se seznamem nalezených chyb při testu v celé verzi. Pokud nebude vybrána konkrétní verze, nebude tlačítko aktivní.

 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



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 a testovací případ z kategorie "Funkční testy" a budou neaktivní (zašedlé). 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 testovacích případů (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 stránky 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 odesílatele bude: testrunner@kerio.com

4. Detail testu


The screenshot shows the KERIO TestRunner 2.2 interface. The left sidebar displays a tree view of test cases for Kerio Connect 7.2.0, including various release candidates, installations, and components like Admin/Engine, CalDAV, and Mobile Devices. The main content area is titled 'Kerio Connect 7.2.0 - Test detail' and shows a summary table with the following data:

Product:	Kerio Connect 7.2.0	Overall:	98% - 183 of
Component:	Admin/Engine	Original Time Estimate:	44:52
System:	Linux / Mac	Time Actually Taken:	36:40
Created:	2011-04-21 10:58:23 by Zdeněk Samuel	Time Left:	0:45

Below the summary table, a list of test tasks is shown with their status (green checkmark for success, yellow warning for partial success, and red X for failure):

- Spam repellent (Success)
- Delay SMTP greeting (Success)
- Enforced correct flow of SMTP communication (Success)
- Spam settings - Upgrade / Downgrade from / to KMS 6.1.x (Success)
- Archiving (Warning)
- Actions (Warning) - 59478
- Archive (Success)
- Archive before antivirus check (Success)
- Access and rights (Success)
- Backup (Success)
- Target backup directory (Success)
- Backup scheduling (Success)
- Functionality (Success)
- Advanced options (Success)
- Internet connection (Success)
- Online (Success)
- Offline (Success)

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*
- ~~průškrtnuté~~ vyřešené chyby (jsou ve stavu RESOLVED nebo VERIFIED) a mají shodný TM jako testovaná verze
- ~~průškrtnuté v hranatých závorkách~~ vyřešené chyby (jsou ve stavu RESOLVED NEBO VERIFIED) a 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 souhrně 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 TestRunner 2.2 interface. On the left is a tree view of test components, including 'Custom rules', 'Spam Assassin', 'Archiving', 'Backup', 'Internet connection', 'Scheduling', 'POP3 download', 'ETRN download', 'SSL certificates', 'Store Directory', 'Miscellaneous', 'Security policy', 'Update checker', 'Users', 'Groups', 'Aliases', 'Mailing lists', 'Other settings', 'Definitions', 'Message queue', 'Active connections', 'Traffic charts', 'Statistics', 'Logs', 'KAssist test', 'Other tests', 'CalDAV (iCal)', and 'Dashboard'. On the right, the 'Kerio Connect - Bugs statistics' panel is visible. It has two sections: 'Opened bugs' and 'Resolved or Verified bugs'. The 'Opened bugs' section contains a table of bugs with columns for the test name and the bug title. The 'Resolved or Verified bugs' section is currently empty.

Kerio Connect - Bugs statistics	
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 popsané 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 Build # Elapsed time

Found bugs Bug # Failed

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

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 a 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 naleznou 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 znovuootevření 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 (nevřešené 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).

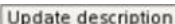
„History“: informace pro testera, jak se vyvíjely předchozí testy u tohoto testovacího případu. Detail viz obrázek. Nejhorší a nejlepší čas je zvýrazněn **červeně**, respektive **zeleně**. Tyto časy nejsou zvýrazňovány pro účely "odměňování" nebo "trestání" testerů, ale pro snadné nalezení extrémních 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

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
- Updateovat 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 shows the KERIO TestRunner 2.2 interface. On the left is a tree view of test cases under 'Kerio Connect'. The right pane shows the 'Test case detail' for 'Outlook Mac'. The detail pane includes a table with component, chapter, and task information, as well as coverage percentages. Below this is a 'Test Case Description' section with Czech and English text, and an 'Autotest Description' section.

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

Test Case Description

Czech:
Cíl: různé 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 (obojí v záložce Advanced).
• Vyzkoušejte i připojení přes SSL WebDAV a alternativní porty Tipy: + pro LDAP je nutné zaškrtnout "Override default LDAP port" a používat 3 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) "hostname/public"

Autotest Description

8.1. Strom testovacích případů

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

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

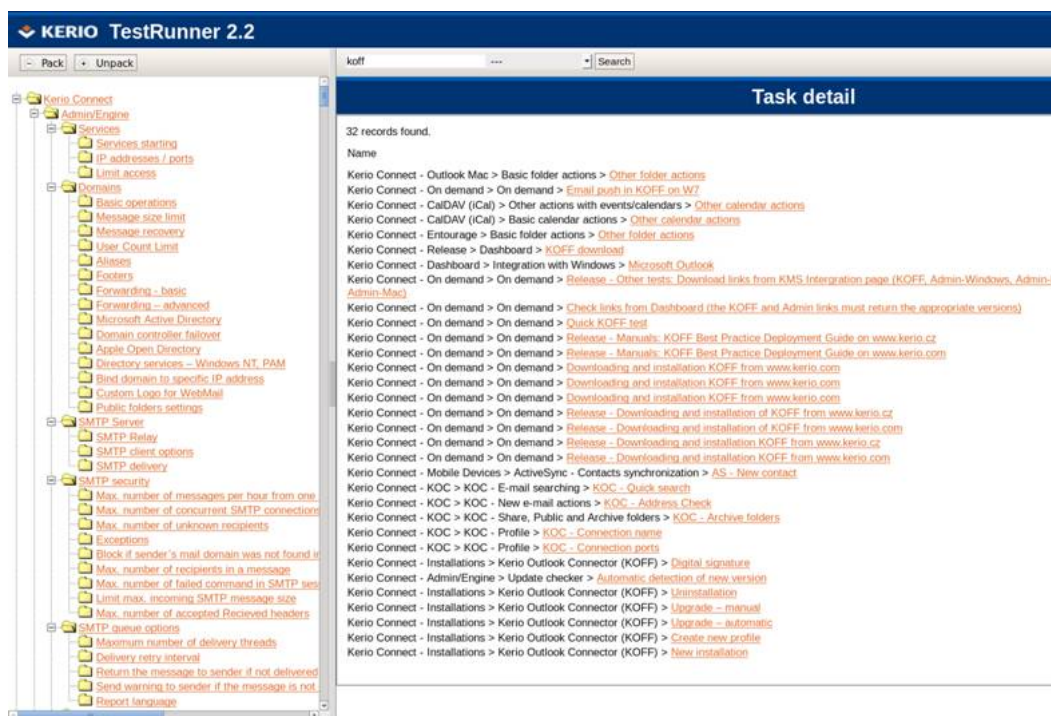
Označit lze komponentu, kapitolu, testovací případ (kvůli statistice chyb). Při kliknutí na testovací případ se do pravého okna zobrazí jeho detail. Při kliknutí na kapitolu nebo komponentu se do pravého okna zobrazí její obsah.

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.

9. Vyhledávání v testovacích případech



V horní části lišty TestRunneru se nachází vyhledávací dialog. Do textového pole je zadáván hledaný text, v rolovacím seznamu je vybrán produkt, kde se má vyhledávat a po stisku tlačítka bude text vyhledán.

10. Manipulace s testovacími případy

Kerio TestRunner 2.2

v. 2 | New Test Case

Kerio Connect - Test Case Description

Component	Chapter	Validity start	Estimated time
Installations	31	unspecified	0:00
	SubChapter	Validity end	
	7	unspecified	

Modify AT

Title
Kerio Connect size (Windows)

Description
Czech:

- Pozn.: tento test musí být proveden na finální veřejné verzi, ne na interní
- Pro udržování přehledu o velikosti Kerio Connect, zadejte do poznámky k tomuto testu následující info
- velikost instalačního balíku windowsové instalačky serveru
- velikost nainstalované verze (z Control panels / Add or remove program
- velikost binárky "mailserver.exe"

English:

- Porovnejte s předchozí finální vydanou verzí
- Note: this test must be done not internal!
- To keeping of Kerio Connect size, fill to the note of this test following information:
- size of server installation package for windows
- size of installed version (from Control panels / Add or remove program
- size of binary file "mailserver.exe" (Windows)
- Check it with previous final released version

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 operační systémy 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' expanded to show various actions like 'Drafts', 'Addresses from Contacts', etc. On the right, the 'Kerio Connect - AutoTest Description' window is open, displaying details for a test case.

Component:	WebMail	Validity:	6.4.0 - unspecified
Chapter:	WM - New e-mail actions		
Task:	WM - Drafts		

Test Case Description

Czech:

- Rozepsat zprávu a uložit ji do konceptů
- Ověřit uložení a možnost jejího otevření, doplnění a posílání
- Uložit do konceptu zprávu s přílohou (i více) a ověřit, že: o U příloh se zobrazují ikonky dle typu přílohy (ne z přílohy ale z Navigky) (viz http://...)
- Lze přidat novou přílohu – zobrazuje se u ní cesta k souboru a generická (standardní) ikonka o Libovolnou přílohu lze odebrat

English:

- Start writing a message and then save it to the drafts
- Verify the message was saved and it's possible to reopen, to completion and to send this message
- Save the message with attachments to Drafts (try more att. too) and verify that: o The icons by type are displayed beside the attachments (fro.../showid.php?id=2) o It is possible to add a new attachment – path to file and generic icon are displayed beside o It is possible to remove any...

Responsible: Miroslav Kroc Possible AT Coverage: Actual AT Coverage:

AutoTest Description

řodp: dherbolt (IMPLEMENTOVÁNO POUZE V KMS 6.5)

- rozepte zprávu a uložte ji do konceptů (s přílohou/bez přílohy)

- kontroluje počet a typ příloh

- NEPŘÍDAVÁ NOVE PŘÍLOHY (sets nemůže přistupovat na disk)

- testuje odebírání příloh

- testuje počet a typ příloh u doručené zprávy

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 . 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

KERIO TestRunner 2.2						
Manage comments						
<input checked="" type="checkbox"/> Show New		<input type="checkbox"/> Show Accepted		<input type="checkbox"/> Show Rejected		<input type="checkbox"/> Show Old
				Kerio Connect		<input type="button" value="Show"/>
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		
New	11.5.2011 15:00	612	Kerio Connect - Admin/Engine -- ETRN download --- Send ETRN	Radek Šíp (rsip@kerio.com) Edit		
New	11.5.2011 14:39	607	Kerio Connect - Admin/Engine -- POP3 download --- Sorting and Delivery	Radek Šíp (rsip@kerio.com) Edit		
New	11.5.2011 14:39	602	Kerio Connect - Admin/Engine -- Scheduling --- Time condition	Radek Šíp (rsip@kerio.com) Edit		
New	11.5.2011 14:38	612	Kerio Connect - Admin/Engine -- ETRN download --- Send ETRN	Radek Šíp (rsip@kerio.com) Edit		
New	11.5.2011 14:38	606	Kerio Connect - Admin/Engine -- POP3 download --- POP3 Account	Radek Šíp (rsip@kerio.com) Edit		
New	11.5.2011 13:16	1505	Kerio Connect - Admin/Engine -- Other tests --- Report Store Statistics	Radek Šíp (rsip@kerio.com) Edit		
New	11.5.2011 13:13	1505	Kerio Connect - Admin/Engine -- Other tests --- Report Store Statistics	Radek Šíp (rsip@kerio.com) Edit		
New	10.5.2011 17:17	560	Kerio Connect - Admin/Engine -- Antivirus control --- Sophos antivirus update	Radek Šíp (rsip@kerio.com) Edit		
			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 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

Component Admin/Engine	Chapter 60	Validity start 6.1.0-RC2	Estimated time 0:05	New dependency Select new --->
	SubChapter 1	Validity end unspecified	<input type="checkbox"/> Modify AT	Selected dependency

Title
General

Description

Czech:
 Vytvořit několik uživatelů (login name, full name, description, internal user database, password)
 Ověřit (webmailem, 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

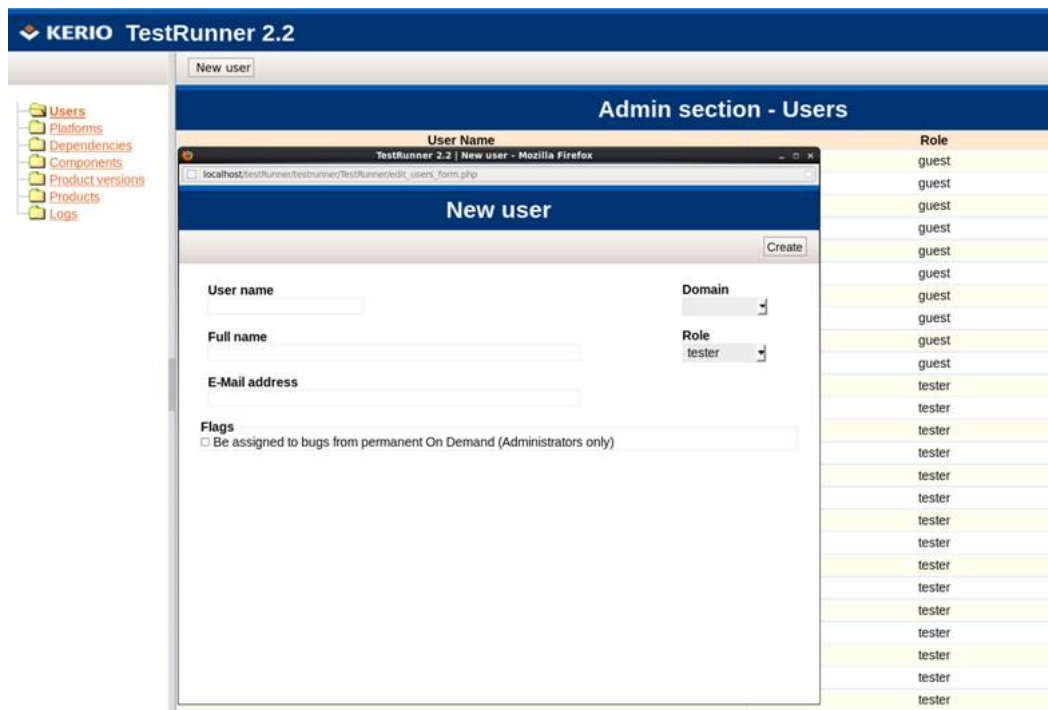
Comment status: Ne
Created: 10
Comment creator: Ra
Product: Ke
Component: Ad
Chapter: Us
Comment:
Chybí test zabudovaně

Accept comment
Reject & Close

- 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 jako nový a to stiskem tlačítka **Save as New**.

13. Admin sekce

13.1. Uživatelé



- Uživatelé budou řazeni abecedně dle uživatelského jména (zobrazuje se také).
- Username je shodné s usermame 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 shows the 'Admin section - Platforms' in the KERIO TestRunner 2.2 interface. On the left, there is a navigation menu with folders for Users, Platforms, Dependencies, Components, Product versions, Products, and Logs. The main area displays a table of supported platforms. A 'New platform' dialog box is open, showing a 'Name' input field and a 'Create' button. The dialog box is titled 'New platform' and is overlaid on the platform list.

- 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 shows the 'Admin section - Dependencies' in the KERIO TestRunner 2.2 interface. On the left, there is a navigation menu with folders for Users, Platforms, Dependencies, Components, Product versions, Products, and Logs. The main area displays a tree view of dependencies. The tree view includes categories like Operating systems, Browsers, Desktop, Chrome, Safari, Firefox, Internet Explorer (versions 6, 7, 8, 9, 10, 11), Mobile, and Applications. The '10' version of Internet Explorer is highlighted. Below the tree view is a large empty area.

- 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 shows the 'Admin section - Components' interface. On the left is a navigation menu with items like Users, Platforms, Dependencies, Components, Product versions, Products, and Logs. The main area contains a table with two columns: 'Component' and 'Product'. A 'New component' dialog box is open in the foreground, featuring a 'Name' input field and an 'in product' dropdown menu currently set to 'Kerio Connect'. A 'Create' button is visible in the top right of the dialog.

Component	Product
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

The screenshot shows the 'Admin section - Product versions' interface. On the left is the same navigation menu as in the previous screenshot. The main area contains a table with two columns: 'Product' and 'Product version'. The 'Product' column contains 'Kerio Connect' for all rows. The 'Product version' column lists various versions from 6.4.0 to 7.3.0. The row for version 7.2.1 is highlighted in pink. A 'Load from Bugzilla' button is located at the top of the table area.

Product	Product version
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

The screenshot shows the 'Admin section - Products' interface. On the left is a navigation menu with items: Users, Platforms, Dependencies, Components, Product versions, Products, and Logs. The main area contains a table with the following data:

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

Overlaid on the table is a 'New product' form with the following fields:

- Name
- Short name
- Bugzilla product id
- Create button

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

13.7. Logy

The screenshot shows the 'Admin section - Logs' interface. The left navigation menu is the same as in the previous screenshot. The main area displays a list of log entries:

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 - rsip 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) .