Vysoká škola ekonomická v Praze Fakulta informatiky a statistiky Katedra informačních technologií

> Studijní program: Aplikovaná informatika Obor: Informatika

Řízení manuálních testů v nástroji HP Quality Center verze 12

BAKALÁŘSKÁ PRÁCE

Student	:	Patrik Luksch
Vedoucí	:	doc. Ing. Alena Buchalcevová, Ph.D.
Oponent	:	Ing. et Ing. Michal Doležel

2015

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a že jsem uvedl všechny použité prameny a literaturu, ze které jsem čerpal.

V Praze dne 5. 5. 2015

Patrik Luksch

Poděkování

Na tomto místě bych rád poděkoval paní doc. Ing. Aleně Buchalcevové, Ph.D. za vedení mé práce, ochotu a vstřícnost. Děkuji také své rodině a blízkým za poskytovanou podporu.

Abstrakt

Cílem práce je představení testovacího nástroje Quality Center verze 12 od firmy Hewlett Packard, pomocí něhož lze testovat funkčnost softwaru.

V teoretické části práce jsou vysvětleny základní pojmy z testování a jeho metodiky. Představeny jsou také druhy testovacích nástrojů a samotný Quality Center je zařazen do příslušné kategorie. V praktické části je nejdříve představen HP Quality Center, návod na jeho instalaci a konfiguraci, dále názorně předvedena manipulace s tímto nástrojem, která spočívá v tvorbě požadavků, plánování a spouštění manuálních testů a zaznamenávání defektů. Po přečtení práce je potenciální uživatel nástroje schopný vytvořit požadavky na testování, testovací sady a napsat testy, které si následně spustí a vyhodnotí. Vlastním přínosem práce je vznik jakési příručky v češtině, která je využitelná při výuce testování softwaru na Vysoké škole ekonomické či v Kompetenčním centru VŠE.

Klíčová slova

Testování, software, testovací nástroj, ALM, HP Quality Center 12, manuální testy.

Abstract

The aim of this thesis is to describe Quality Center version 12, which is a tool developed by Hewlett Packard for testing software functionality.

The first part of this thesis is devoted to basic terms of testing and its methods. The types of testing tools are introduced and Quality Center itself is categorized into appropriate category. The next part contains description of installation and settings of Quality Center and also the manipulation with this tool which consists of planning, start-ups of manual tests and failure reporting. With this thesis the potential future user of this tool should be able to create requirements for testing, the testing sets and also to write the tests, run them and evaluate them. The contribution of this thesis is the emergence of a guide in Czech, which is useful for teaching software testing at the University of Economics or Competence Center.

Keywords

Testing, software, testing tool, ALM, HP Quality Center 12, manual tests.

Obsah

1	Úvo	J
	1.1	Vymezení tématu práce a důvod výběru tématu1
	1.2	Cíl práce1
	1.3	Způsob dosažení cíle
	1.4	Předpoklady a omezení práce
	1.5	Struktura práce
	1.6	Výstupy práce a očekávaný přínos práce
	1.7	Rešerše zdrojů
2	Test	ování softwaru4
	2.1	Základní pojmy4
		2.1.1 Definice testování softwaru4
		2.1.2 Verifikace vs. validace4
		2.1.3 Řízení testování
		2.1.4 Stěžejní dokumenty pro proces testování softwaru
		2.1.5 Řízení životního cyklu aplikace7
		2.1.6 Základní typy testů7
	2.2	Metodiky testování
		2.2.1 Rational Unified Process
		2.2.2 OpenUP10
		2.2.3 MMSP11
		2.2.4 Extrémní programování (XP)12
	2.3	Chyba, defekt, bug13
		2.3.1 Závažnost a priorita bugu14
		2.3.2 Reportování bugů14
	2.4	Druhy testovacích nástrojů15
		2.4.1 Nástroje pro řízení testování15
		2.4.2 Nástroje pro funkční testování16
		2.4.3 Nástroje pro zátěžové testování16
3	Nást	roj HP Quality Center 12 17
	3.1	Krátká historie17
	3.2	HP ALM edice18
	3.3	Instalace a konfigurace HPQC 1219
		3.3.1 Příprava prostředí19
		3.3.2 Konfigurace HPQC20
		3.3.3 Vytvoření domény a projektu22
		3.3.4 Přihlášení do HPQC 1224

4	Říze	ní manuálních testů	25
	4.1	Představení testované aplikace	25
		4.1.1 Cíle a druhy testů	26
	4.2	Plánování testování	26
	4.3	Tvorba požadavků	28
		4.3.1 Import požadavků z MS Word a Excel	29
	4.4	Příprava testování	30
		4.4.1 Tvorba a návrh testů	30
		4.4.2 Přiřazování testů k požadavkům	32
	4.5	Spouštění testů	33
	4.6	Zaznamenávání bugů	36
		4.6.1 Zaznamenané bugy	37
	4.7	Vyhodnocování testování	38
		4.7.1 Analysis View	38
		4.7.2 Dashboard View	41
5	Závě	ěr	42
Ter	mino	logický slovník	43
Sez	znam	literatury	44
Sez	znam	obrázků a tabulek	47
		Obrázky	47
		Tabulky	47
Příl	loha ′	1	48
Příl	loha 2	2	56

1 Úvod

Testování softwaru je v současné době velice důležitou součástí procesu vývoje softwaru. Pokud se podíváme do historie lidstva, co dokázaly chyby v softwaru způsobit za události, není se čemu divit, že testování již dnes není opomíjeno. Firmy vynakládají velké finanční prostředky na testovací týmy, které mají být zárukou toho, že budou splněny požadavky zákazníků do maximální možné míry a dodaný software bude kvalitní.

Z ekonomického hlediska je pro firmu výhodnější testovat software již od začátku vývoje, než software jako hotový produkt. Dle Pattona rostou náklady na odstranění chyb logaritmicky, to znamená, že pokud jsme chyby objevili například již ve specifikaci, budou náklady téměř nulové. Pokud ale objeví chyby až koncový uživatel, mohou být náklady až stonásobně vyšší [1].

V současnosti je obecný trend vše automatizovat, mít zajištěn chod bez pomoci člověka, a to nejen v IT odvětví. U testování softwaru tomu není jinak, avšak ne vše se dá zautomatizovat dostatečně efektivně, a tak na řadu přichází pomoc člověka – testera. Tester provádí manuální testování, které nelze ničím nahradit. Pro řízení manuálních testů jsou používány různé nástroje, jedním z nich je Quality Center od firmy HP. V době tvorby zadání bakalářské práce byla nejaktuálnější verzí tohoto nástroje verze 12. O tom, jak se v tomto nástroji řídí manuální testy, vypovídá tato práce.

1.1 Vymezení tématu práce a důvod výběru tématu

Tématem práce je představení testovacího nástroje Quality Center verze 12 od firmy Hewlett Packard, pomocí něhož lze testovat funkčnost softwaru. V práci jsou vysvětleny základní pojmy z testování a jeho metodiky. Představeny jsou také druhy testovacích nástrojů a samotný Quality Center je zařazen do příslušné kategorie. Následuje instalace a nastavení QC a popis postupů pro řízení manuálních testů ve výše uvedeném nástroji.

Toto téma jsem si vybral, protože s nástrojem od HP přicházím do styku v praxi při testování webové aplikace. Když jsem začínal s prací v tomto nástroji, netušil jsem, co se jak dělá a k čemu je to dobré. To byl další důvod pro zvolení tohoto tématu, záměrem práce je totiž usnadnit uživateli první kontakt s HP Quality Center verze 12.

1.2 Cíl práce

Hlavním cílem této bakalářské práce je představit nástroj HP Quality Center verze 12 a tím vytvořit příručku pro první použití, která zatím v českém jazyce není dostupná. Ukázkové

řízení manuálních testů je předvedeno na aplikaci TTTournament [2]. Vedlejším cílem je vysvětlení základních pojmů týkajících se testování softwaru.

1.3 Způsob dosažení cíle

K úspěšnému dosažení cíle jsem si nainstaloval na svůj notebook volně dostupnou trial verzi Quality Center 12 z webových stránek [3] firmy Hewlett Packard, kterou jsem použil při tvorbě této bakalářské práce. Dále jsem čerpal z odborné literatury věnující se oblasti testování softwaru a ze svých dosavadních zkušeností.

1.4 Předpoklady a omezení práce

U čtenáře této práce se nepředpokládají žádné předchozí znalosti v oblasti testování softwaru ani zkušenosti s jakoukoli verzí testovacího nástroje Quality Center. Práce je tedy vhodná jak pro začínající testery, tak i pro naprosté laiky. Teoretická část (viz kapitola 1.5 Struktura práce) je z hlediska přínosu méně vhodná pro již aktivní testery, pro které může být přínosnější až část praktická. Cílem práce rozhodně není kompletně vysvětlit problematiku testování softwaru.

1.5 Struktura práce

Práce je členěna do několika kapitol. Práci lze rozdělit na dvě hlavní části a to na teoretickou a praktickou. Teoretická část je zastoupena druhou kapitolou, která pojednává obecně o testování softwaru, základních pojmech, vybraných metodikách testování, softwarových chybách a druzích testovacích nástrojů. Část praktická je pak reprezentována následujícími dvěma kapitolami. Postupně je představen nástroj Quality Center verze 12, následuje návod na jeho instalaci a konfiguraci a poté samotné řízení manuálních testů v tomto nástroji. Pátou kapitolou, poslední této práce, je Závěr, kde se nachází vyhodnocení stanovených cílů této bakalářské práce.

1.6 Výstupy práce a očekávaný přínos práce

Výstupem práce je vznik základní příručky pro práci s testovacím nástrojem HP Quality Center verze 12, podle níž bude čtenář schopen zprovoznit a používat nástroj. Vlastním přínosem práce je vznik jakési příručky v češtině, která je využitelná při výuce testování softwaru na VŠE či v Kompetenčním centru VŠE. Po přečtení práce je potenciální uživatel nástroje schopný vytvořit požadavky na testování, testovací sady a napsat testy, které si následně spustí a vyhodnotí. Mimo to jsou v práci vysvětleny základní pojmy z oblasti testování softwaru a jeho vybrané metodiky.

1.7 Rešerše zdrojů

V této podkapitole jsou uvedeny akademické práce a literatura, které se zabývají testováním softwaru a jsou tak relevantními zdroji pro tuto bakalářskou práci.

Jelikož je dnes testování softwaru rozšířenou disciplínou, existuje řada prací, které se tímto tématem zabývají. Velmi podrobně se testováním zabývá Jan Tryzna ve své diplomové práci, která nese název *Testování softwaru* [4]. Autor zde v druhé kapitole píše o základních pojmech testování softwaru a v třetí pak o metodikách testování softwaru.

Dalšími pracemi zabývajícími se testováním softwaru jsou bakalářské práce a to od Martina Kučery, *Testování softwaru a metodika RUP* [5], dále Dalibora Pavlíčka nesoucí název *Popis procesů business testování a jejich optimalizace* [6] a taktéž práce Tomáše Jandáka na téma *Možnosti automatizovaného testování aplikací* [7]. Automatizovaným testováním se zabývá diplomová práce Michala Pietrika – *Automatizované testování webových aplikací* [8]. Ve všech zmíněných pracích jsou velice podrobně rozebrány základní pojmy testování softwaru.

Co se týče zdrojů k manuálnímu testování, těch už není zdaleka tolik jako k testování automatizovanému. Spousta prací včetně těch zmíněných výše se zmiňuje o manuálním testování pouze okrajově a tak nejvhodnějším zdrojem pro tuto práci je bakalářská práce Jiřího Jandla na téma *Application Lifecycle Management – porovnání produktů* [9]. Autor v práci primárně představuje nástroj Polarion ALM 2012 a s ním porovnává další nástroje pro správu životního cyklu aplikace – mimo jiné i HP ALM 11, jehož součástí je Quality Center verze 11. Další prací, která se okrajově zmiňuje o HP Quality Center je diplomová práce Jana Doubravy s názvem *Vyhodnocování SLA nad HP Quality Center* [10].

K dispozici jsou samozřejmě i knižní zdroje. Základním zdrojem je kniha od Rona Pattona s názvem *Testování softwaru* [1]. Tato kniha je skutečně podrobným průvodcem světem testování softwaru. V současnosti je pravděpodobně nejlepším českým zdrojem kniha od Anny Havlíčkové a Petra Roudenského nesoucí název *Řízení kvality softwaru: průvodce testováním 1. vydání* [11], která je dalším hlavním zdrojem pro tuto práci. Spoluautorka knihy Anna Havlíčková, dříve Borovcová, v knize čerpá i ze své diplomové práce – *Testování webových aplikací* [12], která je i doporučenou literaturou ke kurzu Základy testování SW aplikací na VŠE.

Dalšími vhodnými knižními zdroji jsou knihy *Metodiky vývoje a údržby informačních systémů* [13] a *Metodiky budování informačních systémů* [14] od Aleny Buchalcevové, ze kterých pro svou práci čerpám informace zejména pro podkapitolu Metodiky testování.

2 Testování softwaru

V této kapitole jsou vysvětleny základní pojmy z oblasti testování. Poté následuje podkapitola, ve které jsou stručně popsány vybrané metodiky testování, dále podkapitola pojednávající o softwarových chybách a závěrečná podkapitola pojednává o druzích testovacích nástrojů.

2.1 Základní pojmy

V této podkapitole jsou uvedeny definice, které vysvětlují co je testování softwaru, definovány jsou pojmy verifikace a validace, co je řízení testování, popsány jsou stěžejní dokumenty pro proces testování softwaru, poté co je řízení životního cyklu aplikace a základní typy testování.

2.1.1 Definice testování softwaru

Je známo několik definic testování softwaru, některé si dokonce odporují, čili neexistuje jedna jediná definice, o které se dá říci, že tato definice je skutečně přesná a správná. Podle Kanera je definice následující: Softwarové testování je empirický technický výzkum kvality testovaného produktu nebo služby, prováděný za účelem poskytnutí těchto informací stakeholderům [15]. Podle [16] je testování pouze dynamická kontrola toho, že chování programu odpovídá specifikaci.

2.1.2 Verifikace vs. validace

Pojmy verifikace a validace jsou často zaměňovány a tak je užitečné vědět, který co znamená. Při verifikaci si pokládáme otázku: Vytváříme produkt správně? Podle [11] lze verifikaci chápat jako proces, jehož cílem je ověření, že dílčí produkt softwaru náležitě odráží specifikované požadavky. Verifikace tak zajišťuje, že výstupy jednotlivých fází vývoje odpovídají vstupům – specifikacím odvozeným z požadavku.

Při validaci si pokládáme tuto otázku: Vytváříme správný produkt? Roudenský s Havlíčkovou ve své knize definují validaci jako potvrzení toho, že dílčí produkt vývoje softwaru je správný s ohledem na požadavky na jeho zamýšlené použití, takže funguje podle představ a potřeb zákazníka a také je software podle toho testován [11].

Na příkladu požadavků si shrneme tyto dva pojmy [11]:

- Verifikace požadavků ověřuje správnost, úplnost a proveditelnost.
- Validace požadavků zjišťuje, zda skutečně zachycují potřeby uživatele.

2.1.3 Řízení testování

Proces testování softwaru není pouze o samotném testování daného produktu, je v něm několik činností, které mají na starost zodpovědné osoby. Každý má tedy na projektu svou roli a vykonává disciplínu jemu svěřenou. Podle [11] jsou ve struktuře týmu středních a větších projektů zastoupeny tyto role:

- Tester je zodpovědný za provádění testu podle testovacích scénářů (tento a další pojmy v této podkapitole jsou vysvětleny v podkapitole 2.1.4), zaznamenává výsledky testů a reportuje nalezené defekty. Jakmile je defekt opraven, provádí retestování. Často se tato role dělí na juniorskou a seniorskou.
- Test analytik tato role v sobě často obsahuje i roli testera. Test analytik vytváří testovací scénáře a testovací případy, tomu předchází analýza požadavků a stanovení rizikovosti a priority pro jejich provádění. Další disciplínou je pak tvorba sad regresních testů a jejich následná údržba, hlášení o pokrytí testů vzhledem k požadavkům či podpora zákazníka [11]. Role test analytika se taktéž často dělí na juniorskou a seniorskou.
- Vedoucí testování má na starosti tvorbu testovacího plánu, kontrolu jeho dodržování a koordinaci jeho změn. Kromě toho nastavuje a spravuje systém pro hlášení defektů, jednotlivým členům zadává úkoly a dohlíží na jejich plnění. Současně podává informace o progresu testování manažerovi testování a může být zapojen i do procesu navrhování a provádění testů [11].
- Manažer testování nejvyšší role v testovacím týmu, je zodpovědný za veškeré aktivity týmu a předkládá zprávy vedení projektu nebo vyššímu managementu. Dále spolupracuje s vedoucím testování při tvorbě testovacího plánu, nastavuje cíle testování a hodnotí jejich naplnění. Je prostředníkem mezi testovacím týmem a managementem, má na starosti komunikaci se zákazníky a vedoucím vývoje. Samozřejmostí je pravomoc pro sestavení testovacího týmu. U menších projektů je pak tato pozice sloučena s rolí vedoucího testování [11].

2.1.4 Stěžejní dokumenty pro proces testování softwaru

Pro proces testování softwaru je potřeba vytvořit minimálně tyto dokumenty:

Testovací plán – je dokument, který podrobně popisuje systematický přístup k testování systému, jako jsou stroje nebo software [17]. Pomocí testového plánu testovací tým sděluje vývojovému týmu, kdo, co a jak hodlají testovat a kolik času na to budou potřebovat. V případě webové aplikace to může být například test zobrazení v různých prohlížečích. Dále jaké prostředky budou potřeba, tím může být myšleno to, že bude potřeba pět počítačů s určitým hardwarovým vybavením a s jakým časovým rozvrhem či plánem počítají [1].

- Testovací případ nebo také test case, popisuje konkrétní akce prováděné s určitou softwarovou komponentou a jejich očekávané výsledky [18]. V tomto dokumentu jsou definovány podmínky, které musí být splněny, aby bylo možné daný případ testovat, a určuje podobu výsledku, který očekáváme. Testovacím případem může být například otestování funkčnosti nějaké aplikace pro správu zaměstnanců. Jednotlivé testy pak mohou testovat přihlášení uživatele do aplikace, přidání zaměstnance, odebrání zaměstnance či změnu údajů zaměstnance. Tyto testy pak obsahují konkrétní testovací skripty (viz dále).
- Testovací nápady (test ideas) jsou podle [11] krátkým popisem toho, co by mělo být testery otestováno. Testovací nápady mohou vznikat při brainstormingu s kolegy, na základě zkušeností testera či stížností zákazníka. Jak z názvu vyplývá, testovací nápad je pouhou myšlenkou, nikoliv detailním popisem případného provedení [11]. Testovací nápady pro webovou aplikaci vyhledávající dopravní spoje mohou vypadat takto:
 - o zadání stejného místa odjezdu a příjezdu
 - zadání nekorektního místa odjezdu či příjezdu (kombinace čísel a speciálních znaků apod.)
 - o nezadání místa odjezdu a příjezdu
 - o zadání pouze místa odjezdu nebo pouze místa příjezdu
 - o zadání nekorektního času odjezdu (např. 25:16, písmem)
 - o zadání času s tečkou, středníkem, čárkou (18.10 18;10 18,10)
 - o zadání nekorektního data (32.10.)
- Testovací skript udává, jak budeme postupovat při testování softwaru. Obsahuje soupis kroků (stepů), které má tester vykonat a pro každý krok je uveden očekávaný výsledek. Tester pak po provedení kroku vyhodnocuje, zda se jev, který nastal, shoduje s tím očekávaným. Pokud se všechny výsledky kroků shodují s těmi očekávanými, je testovací skript považován za úspěšně otestovaný. Na příkladu přihlašování do aplikace pro správu zaměstnanců může vypadat jednoduchý testovací tak jako na obrázku 1:

	Přihlašování do	webové aplikace		
číslo kroku	definice kroku	očekávaný výsledek	skutečný výsledek	status
1	Zadej nekorektní login a stiskni PŘIHLÁSIT	přihlášení neúspešné	přihlášení neúspešné	ok
2	Zadej nekorektní heslo a stiskni PŘIHLÁSIT	přihlášení neúspešné	přihlášení neúspešné	ok
3	Zadej nekorektní login a heslo a stiskni PŘIHLÁSIT	přihlášení neúspešné	přihlášení neúspešné	ok
4	Zadej korektní login a heslo a stiskni PŘIHLÁSIT	přihlášení úspěšné	uživatel přihlášen v aplikaci	ok
5	Stiskni tlačítko ODHLÁSIT	odhlášení úspěšné	uživatel odhlášen z aplikace	ok

Obrázek 1 – Testovací skript pro přihlášení do aplikace, (zdroj: autor)

 Testovací scénář – v sobě obsahuje několik testovacích skriptů, které na sebe logicky navazují a jsou vykonány v určitém pořadí tak, aby byla otestována určitá část softwaru. Příkladem může být již zmíněné přidání nového zaměstnance v aplikaci pro správu zaměstnanců. Budeme mít několik testovacích skriptů, které ověří korektnost zadání jména, příjmení, adresy apod. a společně budou tvořit testovací scénář, který se bude souhrnně nazývat Přidání nového zaměstnance.

2.1.5 Řízení životního cyklu aplikace

Životní cyklus aplikace je časový úsek, který začíná v momentě, kdy vznikne myšlenka vytvořit software a končí, když se software přestane používat [14]. Řízení životního cyklu aplikace (application lifecycle management) se dělí podle [19] na tři základní aspekty, kterými jsou správa, vývoj a údržba. V těchto aspektech je zahrnuta správa požadavků, softwarová architektura, programování, testování softwaru, údržba softwaru, řízení změn, nepřetržitá integrace, projektový management a správa releasů.

Slovo release v překladu z anglického jazyka znamená uvolnit, zveřejnit, pustit. V oblasti informačních technologií označuje vydanou verzi softwaru, která byla dána k dispozici pro používání a do češtiny se běžně nepřekládá. V dalším textu této práce tak budu používat právě tento pojem.

2.1.6 Základní typy testů

Testy můžeme dělit podle mnoha kritérií. V rámci této práce uvedu jen ty nejzákladnější testy:

- Manuální jsou prováděny testerem, který postupuje buď podle testovacích skriptů anebo podle svých zkušeností, fantazie či intuice. Hlavní výhodou manuálního testování je určitě jednoduchost, dále možnost lépe reagovat na vzniklé situace a nižší nákladovost. Nevýhodou je pak možná chybovost testera a časová náročnost. Pokud je například při testování webové aplikace potřeba vyplňovat velké množství různých textových polí, manuální testování je pak zdlouhavé a je nasnadě uvažovat o testování automatizovaném.
- Automatizované podle [6] jsou automatizované testy používány tam, kde testování softwaru nevyžaduje významný podíl lidského úsudku a funkčnost se nemění na rozdíl od zadávaných vstupních parametrů. K tomu se využije libovolný nástroj, který podporuje automatizované testování softwaru, kde se naprogramuje, co se má s jakými vstupními parametry otestovat a test spustí.

Manuální a automatizované testy se velice často navzájem doplňují.

Statické – toto testování probíhá bez jakéhokoliv spuštění softwaru, proto je možné začít s tímto testováním ještě před začátkem vývoje. Statické testování může být například i zkoumání dokumentace, může se tak přijít na to, že je chybný časový plán či nedostatečný počet hardwaru.

- Dynamické dynamické testování je opakem statického. Běh spustitelného softwaru je zde nutný a zkoumají se zde výstupy softwaru oproti zadaným vstupům.
- Černé skříňky black box testy lze chápat tak, že na software pohlížíme jako na skříňku, do které nevidíme. Softwaru poskytneme vstup, ten nám zas výstup. A protože nevidíme dovnitř, do zdrojového kódu, netušíme, jak se z našeho vstupu stal obdržený výstup [11].
- Bílé skříňky white box testy fungují přesně naopak než black box testy. Do zdrojového kódu máme možnost nahlédnout a tak přesně víme, jak se z našeho vstupu stal obdržený výstup.
- Šedé skříňky grey box testy jsou kombinací dvou výše zmíněných.
- Jednotkové unit testing provádí nejčastěji programátor. Jednotkami mohou být funkce, procedury či metody nebo třídy. Cílem je otestovat každou jednotku nezávisle na ostatních, že fungují tak, jak mají. Kód testu ověří obdržené výstupy se zadanými vstupy, pokud se neshodují, test ohlásí chybu [11].
- Integrační integration testing přichází na řadu po jednotkovém testování. Máme-li nějaký systém, do kterého se přidává například nová funkcionalita, je nutnost otestovat, zda právě tato nová funkcionalita nezpůsobila nefunkčnost softwaru.
- Systémové system testing následuje po integračním testování. V této fázi se podle specifikace od zadavatele ladí software. Výsledkem tohoto testu by měla být téměř stoprocentní jistota, že příjemce softwaru bude spokojen a produkt převezme. Záměrně píši téměř stoprocentní, jelikož nikdy nejde všechno otestovat stoprocentně.
- Akceptační acceptance testing je prováděno zadavatelem podle zadavatelského dokumentu. Zadavatel testuje, zda jsou splněna tzv. akceptační kritéria, která si sám zadavatel stanovil pro přijetí softwaru a byla odsouhlasena i ze strany dodavatele. Jejich nesplnění znamená nepřevzetí softwaru ze strany zadavatele [11].

2.2 Metodiky testování

Jak už jsem zmínil v úvodu, testování je velmi důležitou součástí při vývoji softwaru. Proces vývoje softwaru popisují metodiky vývoje softwaru, které definují principy, procesy, praktiky, role, techniky, nástroje a produkty používané při vývoji, a to jak z hlediska softwarově inženýrského, tak z hlediska řízení [13]. Metodik existuje velké množství a liší se zaměřením, rozsahem, oblastí, pro kterou jsou určeny, přístupem k řešení a v neposlední řadě také úrovní podrobnosti, na které jsou popsané jednotlivé postupy, procesy, doporučení, techniky a další složky metodiky. Rozdíly mezi metodikami nalezneme i v přístupu k testování softwaru, který při jeho vývoji uplatňují [20]. V této podkapitole uvedu několik vybraných metodik. Cílem této podkapitoly rozhodně není dopodrobna představit vybrané metodiky.

2.2.1 Rational Unified Process

Rational Unified Process (RUP) je založena na tzv. nejlepších praktikách softwarového vývo-je [21]:

- Iterativní vývoj
- Řízení požadavků
- Použití komponentové architektury
- Vizuální modelování
- Kontrola kvality softwaru
- Řízení změn

Základními prvky RUPu jsou [22]:

- Aktivita představuje jednotlivé úkoly a činnosti, které musí osoba s určitou rolí vykonat.
- Role je přesně definováno, kdo má jakou roli na projektu a kdo má jaký vztah k aktivitám a artefaktům.
- Artefakt dokument či jiný vstup nebo výstup aktivity.
- Workflow popisuje návaznost jednotlivých procesů (pomocí UML diagramu aktivit).

Podle [23] probíhá testování v této metodice postupně takto:

- Test manažer určuje, co a proč se bude testovat a jaká jsou výstupní kritéria z testů.
- Test analytik stanovuje způsob provádění testů, jejich konkrétní podobu, jaká je pro ně požadovaná konfigurace testované aplikace a jaké nástroje budou použity.
- Test designer vytváří testy.
- Test designer či tester provádí smoke testy. Ověřují, že aplikace je nainstalovaná, spuštěná a stabilní, tím pádem vhodná pro testy. Tento bod obsahuje jak přípravu tak provedení těchto testů.
- Tester provádí konfiguraci aplikace dle vstupních požadavků daného testu a spouští testy a případné rozdíly oproti očekávanému stavu hlásí v požadovaném formátu.
- Test manažer provádí analýzu výsledků testů. Nalezené rozdíly jsou v podobě žádostí o změnu předány k další analýze. Výstupem testů je Test report, který předává informace o rozsahu provedených testů a jejich výsledku.

Nedostatkem je podle [13] to, že je metodika zaměřena pouze na úroveň projektu a má malý rozsah, protože nezahrnuje provoz a údržbu a zaměřuje se pouze na vývoj řešení a na softwa-rově inženýrské role a dimenze.

2.2.2 OpenUP

Tato open-source metodika, která je vhodná spíše pro projekty s menšími týmy, využívá iterační nebo inkrementační model životního cyklu softwaru a je postavena na případech užití, řízení rizik a architektuře [14]. Metodika má čtyři základní principy, které vycházejí z pravidel Manifestu agilního vývoje softwaru. Tyto pravidla a základní principy OpenUP jsou pro srovnání uvedeny v tabulce 1.

Pravidla Manifestu	Principy OpenUp
Individuality a interakce jsou upřednostňovány před procesy a nástroji.	Spolupráce na společných zájmech a sdílení vědomos- tí.
Spolupráce se zákazníkem před vyjednáváním o smlouvě.	Vyvážení konkurenčních priorit s cílem maximalizo- vání užitku stakeholdera.
Fungující software před komplexní dokumentací.	Organizování vývoje se zaměřením se na architekturu s cílem minimalizovat rizika.
Náklonnost ke změnám oproti předem stanovenému plánu.	Získávání průběžné zpětné vazby a zlepšování.

Tabulka 1 – Pravidl	a Manifestu a	a principy	OpenUP	(zdroj:	[24])
---------------------	---------------	------------	--------	---------	-------

OpenUP je uspořádána do dvou vzájemně závislých dimenzí, kterými jsou metodický a procesní obsah. V metodickém obsahu jsou definovány role, úlohy, produkty a návody bez ohledu na způsob využití v životním cyklu projektu. Procesní obsah určuje životní cyklus daného projektu a sleduje jej z časového hlediska [24].

Metodický obsah má tři úrovně – mikro-přírůstek (měřitelný krok přispívající k dosažení cíle iterace), iterace životního cyklu (časově omezené intervaly, jejichž výsledkem je plně otestovaná spustitelná verze produktu) a životní cyklus projektu, který má čtyři fáze (Zahájení, Rozpracování, Konstrukce, Zavedení) [24].

V malém týmu jsou reprezentovány tyto role: zainteresovaná osoba, analytik, architekt, vývojář, tester, vedoucí projektu a kdokoli. Metodický obsah metodiky je pak rozdělen do následujících disciplín: Požadavky, Architektura, Vývoj, Testování, Řízení projektu, Řízení konfigurací a změn [24]. Úlohy jsou jednotky práce, kterých je v této metodice 18, jsou vykonávány výše zmíněnými rolemi.

Podle [14] se v rámci úloh vytvářejí, modifikují a používají pracovní produkty. Za tyto činnosti jsou zodpovědné určité role. Prvky metodického obsahu se seskupují do tzv. vzorů schopností, ty mohou být dílčí (zaměřené například na řízení iterace či zahájení projektu), složené (šablonové vzory pro iterace) či výsledné procesy, které představují kompletní a integrovanou procesní šablonu pro specifický typ projektu [14].

V metodice OpenUP je podporováno testování softwaru v rámci softwarově specifických procesů. Díky podrobnému popisu jednotlivých činností, množství návodů a šablon dokumentů může projekt vést i méně zkušený projektový manažer [14].

2.2.3 MMSP

Metodika MMSP je metodikou pro menší softwarové projekty a primárně byla vytvořena uzpůsobením metodiky OpenUP (verze 1.5.1.1) pro projekty v rámci výuky softwarového inženýrství na VŠE v Praze [25]. Struktura metodiky je dělena do čtyř základních oblastí. Vzhledem k tématu této práce v jednotlivých oblastech uvedu pojmy související s testováním. Oblasti jsou následující:

- Role projektový tým je tvořen architektem, projektovým manažerem, analytikem, vývojářem, nedefinovanou rolí, zainteresovanou stranou a testerem. Tester má za úkol identifikovat, definovat a provést potřebné testy, zaznamenávat a analyzovat jejich výsledky. V případě shledaného nesouladu je pak potřeba komunikace s ostatními členy týmu [25].
- Disciplíny jsou jimi Řízení projektu, Požadavky, Architektura, Vývoj, Řízení změn a konfigurací a Testování. Požadavky definují, jakou funkcionalitu má software poskytovat, běžnou praxí je, že se v průběhu životního cyklu projektu zpřesňují a mění. Testování je disciplínou, která poskytne vývojovému týmu včasnou zpětnou vazbu, zda vyvíjený software uspokojuje definované požadavky, dále identifikuje chyby v řešení a ověřuje, zda nově implementované funkčnosti nezpůsobily, že již původně funkční otestovaný software nefunguje [25]. V rámci této disciplíny je prováděno:
 - Plánování testů výstupem je testovací plán, viz podkapitola 2.1.4.
 - Příprava testů je podle [25] jednou z nejnáročnějších celého testovacího cyklu a je zahajována v případě, kdy je nutné otestovat nově implementovanou funkcionalitu.
 - Provedení testů jsou provedeny naplánované testy, po zaznamenání výsledků následuje jejich analýza.
- Pracovní produkty v MMSP jsou pracovní produkty děleny do oblastí projektového řízení, řízení změn a konfigurací, požadavků, architektury, vývoje a testování. Pracovními produkty testování jsou:
 - o Plány testů viz podkapitola 2.1.4.
 - Seznamy testovacích nápadů viz podkapitola 2.1.4.
 - Testovací data množina dat využívaná při testování softwaru. V případě manuálního testování je vhodné uvést v testovacích případech odkaz na testovací data [25].
 - o Testovací případy viz podkapitola 2.1.4.
 - Testovací sada je pracovní produkt, který sdružuje uspořádané testovací případy do logické posloupnosti tak, aby kompletně otestoval určitou část softwaru [25].

- Záznamy výsledků testů v tomto dokumentu jsou informace o provedených testech a jejich výsledcích. Obsahuje i report chyb, o něm viz podkapitola 2.3.2.
- Životní cyklus životní cykly jsou totožné jako v metodice OpenUP, má tedy 4 fáze, kterými jsou Zahájení, Rozpracování, Konstrukce a Zavedení.

2.2.4 Extrémní programování (XP)

Metodika vychází z principů a postupů běžných při vývoji softwaru, které však dovádí do extrémů [26]. Podle [14] je tato metodika disciplinovaná a zavádí specifické praktiky jako neustálé revize zdrojového kódu (párové programování), neustálé testování vývojáři (testování jednotek), každodenní návrh (refaktorizace) a další. Na obrázku 2 jsou zachyceny praktiky XP, z nichž vybrané uvedu v této práci. Detailněji jsou všechny praktiky popsány v knize od Aleny Buchalcevové [14].



Obrázek 2 – Praktiky XP, (zdroj: [27])

- Whole Team (tým jako celek) základem týmu je zákazník denně spolupracující na projektu společně s programátory. Součástí týmu jsou často testeři, kteří se zákazníkem tvoří akceptační testy, dále analytici napomáhající zákazníkovi definovat požadavky a vedoucí projektu [14].
- Planning Game (plánovací hra) spočívá ve dvou činnostech:
 - plánování dodávky zákazník prezentuje požadované vlastnosti programátorům a ti odhadují jejich náročnost
 - plánování iterace tým si určuje, co v rámci dvoutýdenní iterace bude vyvíjet, zákazník prezentuje vlastnosti, které by měly být realizovány v těch dvou týdnech a programátoři je rozdělí na úkoly a odhadují náročnost.

Požadavky jsou definovány formou uživatelských příběhů (user stories), ve kterých zákazník vyjadřuje potřeby, které má systém podpořit. Text je psaný běžnou termino-

logií uživatele a je stručný [14].

- Small Releases (malé verze) podle [14] je XP založeno na iterativním, přírůstkovém vývoji, výsledkem iterace je implementované řešení dodané zákazníkovi, který si zvolil, co bude v dané iteraci realizováno. Verze jsou dodávány v krátkých časových horizontech, jakými jsou den, týden či měsíc. Díky nepřetržitému testování je zde malá chybovost a tak po uvolnění nové verze trvá testování krátkou dobu.
- Customer tests (zákaznické testy) zákazníkem jsou specifikovány akceptační testy (viz podkapitola 2.1.6) pro každý požadavek, jsou prováděny často a jejich výsledky jsou zveřejňovány [14].

2.3 Chyba, defekt, bug

Existuje mnoho výrazů, které vyjadřují to, že software nepracuje tak jak má. Používají se slova, jakými jsou vada, omyl, problém, selhání, chyba, defekt, bug a mnohé další. Často však záleží na samotné instituci, jaký výraz si zvolí pro svou interní komunikaci.

Definice softwarové chyby od [1] je následující:

"O softwarovou chybu se jedná, je-li splněna jedna nebo více z následujících podmínek:

- Software nedělá něco, co by podle specifikace dělat měl.
- Software dělá něco, co by podle specifikace produktu dělat neměl.
- Software dělá něco, o čem se specifikace nezmiňuje.
- Software nedělá něco, o čem se produktová specifikace nezmiňuje, ale měla by se zmiňovat.
- Software je obtížně srozumitelný, těžko se s ním pracuje, je pomalý nebo podle názoru testera softwaru jej koncový uživatel nebude považovat za správný."

Roudenský s Havlíčkovou popisují softwarový defekt jako vadu v kódu či datech, jejíž příčinou je nejčastěji pochybení programátora. Takovým pochybením může být například chyba v kódu, špatný návrh nebo nepochopení specifikace [11].

Podle [11] je defekt následkem pochybení člověka a je příčinou chyby. Chyba je stav systému, který může vést k selhání. Selhání je pak nesoulad mezi aktuálním chováním systému oproti tomu, které je ve specifikaci.

Jako bug se jednoduše označuje jakýkoli defekt softwaru, toto označení budu používat v dalších částech práce.

2.3.1 Závažnost a priorita bugu

Pokud se mluví o bugu, tak je dobré zmínit i jejich dělení podle závažnosti a priority. V závislosti na tom, jaký dopad bude mít bug na systém, jsou rozlišovány tyto stupně závažnosti [11]:

- Kritický tento bug ovlivňuje kritické funkce pro daný systém a znemožňuje jeho správné užívání. Je to například riziko ztráty dat.
- Vysoký bug ovlivňuje významné funkce systému, systém je ale nadále použitelný, byť s určitými obtížemi.
- Střední bug neomezuje významnou funkčnost systému, uživatelům může při používání způsobovat větší potíže.
- Nízký bug, který absolutně neovlivní funkčnost systému. Je to například překlep v popiscích apod.

Z pohledu testování jsou pak známy 4 úrovně priorit [11]:

- 1 Kritická v tomto případě nelze pokračovat v testování, dokud není bug opraven.
 Oprava je v tomto případě akutní.
- 2 Vysoká bug z části blokuje proces testování, opravu je nutné provést co nejrychleji.
- 3 Střední bug neovlivňuje proces testování. Oprava může přijít na řadu v rámci běžného pořadí řešení úkolů.
- 4 Nízká tento bug nemusí být často ani opraven. Jeho oprava přijde na řadu, až když jsou vyřešeny všechny důležitější úkoly.

2.3.2 Reportování bugů

Na reportování bugů existují různé nástroje k tomu vhodné, ale postačí i obyčejný textový editor.

Podle [22] by měly být dodrženy tyto tři body:

- Reportováno je cokoliv, co snižuje hodnotu softwaru v očích některého stakeholdera.
- Manažer projektu či povolaná osoba rozhodne, zda se skutečně jedná o bug.
- Nutnost existence záznamu o tom, že daný nesoulad byl nalezen a kdo rozhodl o tom, že se o bug nejedná.

Report by měl být především stručný, přesný a srozumitelný. Důležité je bug vhodně pojmenovat, popsat, proč je to bug a popis kroků k jeho zreprodukování. Dále kdo, kdy a v kolik bug zaznamenal a jeho stav. Žádoucí je i připojit přílohu, snímek obrazovky či video je kolikrát to nejvýstižnější, co lze předat vývojovému týmu.

Příklad bug reportu je na obrázku 3:

ldentifikace defektu	build0705_13_04_2015
Chyba	Spadnutí serveru při ukládání uživatelských dat
Lokace chyby	Sekce DATA UŽIVATELŮ
Cesta k chybě	MENU -> sekce DATA UŽIVATELŮ -> proveď změny -> stiskni ULOŽIT
Identifikace verze a release	Build 7.05
Datum nálezu	13.04.2015
Typ bugu	Chyba databáze, chyba kódu
Závažnost	blocker
Priorita	high
Popis defektu	po stisku buttonu Uložit se nezobrazí uživateli potvrzující hláška o uložení, která by se měla dle specifikace objevit, aplikace spadne a je vypsaná chybová hláška serveru.
Autor reportu	Patrik Luksch
Aktuálně přiděleno na:	Vývojář – Adam Novák, retester Patrik Luksch
Status defektu	nový

Obrázek 3 – Příklad reportování bugu, (zdroj: autor)

2.4 Druhy testovacích nástrojů

V této podkapitole jsou uvedeny nástroje pro testování softwaru. Protože jich v současnosti existuje velké množství, uvádím pouze několik vybraných, které jsou rozděleny do třech kategorií, přičemž v každé z nich je zástupce jak volně dostupného nástroje, tak komerčního:

2.4.1 Nástroje pro řízení testování

- RTH (Requirements and Testing Hub) je volně dostupný nástroj pro tvorbu a správu požadavků, spouštění testů, zaznamenávání a následné sledování stavu chyb v softwaru [28]. Nevýhodou však je, jak to u volně dostupných aplikací často bývá, velký výskyt chyb v samotné aplikaci.
- TestLink je další volně dostupný test management nástroj pro správu testů. Výhodou je snadná úprava pro svou potřebu, jelikož je psaný v PHP jazyku [29].
- HP Application Lifecycle Management (HP ALM) je komerční software pro řízení životního cyklu aplikace od firmy Hewlett Packard, jehož součástí je HP Quality Center. Více o nástroji HP Quality Center v kapitole 3.

 IBM Rational Quality Manager – je komerční nástroj od firmy IBM, který nabízí obdobnou funkčnost jako předchozí nástroj od firmy HP, tedy komplexní správu a plánování testů od požadavků až po bugy. Cena licence na tento a předchozí nástroj se pohybuje od desetitisíců až po statisíce korun. Využívají je tak většinou velké firmy či banky.

2.4.2 Nástroje pro funkční testování

- Selenium je hojně používaný, volně dostupný nástroj pro tvorbu automatizovaných testů webových aplikací. Nástroj se skládá z dalších komponent, které se navzájem doplňují [30]:
 - Selenium IDE doplněk do prohlížeče Mozilla Firefox, nahrávání testů je zde podobné jako maker v Microsoft Office.
 - Selenium RC tvorba testů pomocí programovacích jazyků (Java, C#, Python).
 - Selenium WebDriver oproti verzi RC poskytuje jednodušší přístup k vytváření testů (není nutné používat Selenium server ke spouštění testů) [30].
 - Selenium Grid umožňuje spouštět testy na více počítačích a prohlížečích současně.
- vTest je komerční nástroj od firmy Verisium, který umožňuje komplexně testovat webové aplikace v mnoha různých prostředích [31].

2.4.3 Nástroje pro zátěžové testování

- JMeter je volně dostupný nástroj napsaný v jazyce Java sloužící pro vykonávání zátěžových testů a měření výkonu [32]. Je zde možnost nasimulovat aktivitu několika klientů současně a tím otestovat stabilitu aplikace.
- vPerformer je komerční nástroj od firmy Verisium pro vykonávání zátěžových testů a měření výkonu podobně jako výše uvedený JMeter, který dokáže nasimulovat aktivitu stovky virtuálních uživatelů [31].

3 Nástroj HP Quality Center 12

HP Quality Center (dále HPQC), který je součástí komplexního softwaru nesoucí název Application Lifecycle Management (dále ALM), je komerční nástroj pro tvorbu požadavků, správu, plánování a spouštění testů a zaznamenávání defektů. HPQC je nabízen i jako SaaS. V této kapitole je uvedena stručná historie nástroje, jeho edice a jak ho co nejsnáze nainstalovat.

3.1 Krátká historie

Quality Center se v různých verzích a názvech objevuje již od roku 2001, kdy byl vyvíjen firmou Mercury Interactive jako Test Director. Od roku 2006, kdy byla firmou Hewlett Packard (dále HP) koupena, spadá pod HP Software divizi a Quality Center přešel do portfolia softwaru firmy HP **[33]**. V tabulce 2 je uveden výčet verzí:

verze	vvdáno	vlastník
5.5	2001	Mercury Interactive
6.0	2002	Mercury Interactive
6.5	2003	Mercury Interactive
7.0	nikdy nevydána	
8.0	2004	Mercury Interactive
8.2	2005	Mercury Interactive
9.0	2007	Mercury Interactive
9.1	2007	Mercury Interactive
9.2	2007	Hewlett Packard
9.5	2008	Hewlett Packard
10.0	2009	Hewlett Packard
11.0	2010	Hewlett Packard
11.5	2012	Hewlett Packard
11.52	červen 2013	Hewlett Packard
11.53	listopad 2013	Hewlett Packard
12.0	březen 2014	Hewlett Packard
12.01	červenec 2014	Hewlett Packard
12.02	březen 2015	Hewlett Packard

Tabulka	2 –	Výčet	verzí	(zdroj:	[34]	[35])
		•		· •	-	/

3.2 HP ALM edice

HP nabízí následující edice ALM, čísla v závorce jsou použita pro zjednodušení orientace v tabulce 3:

- HP ALM (1) tato edice obsahuje moduly pro požadavky, testy, bugy, řízení vývoje, releasy a projektové plánování. Z této edice vycházejí zbylé tři edice.
- HP ALM Essentials (2) je navržena tak, aby byly menší týmy schopny rychle a efektivně pracovat na projektu. Poskytuje základní funkcionality pro správu požadavků, řízení testů a správu defektů.
- HP QC Enterprise (3) poskytuje základní funkcionality pro řízení kvality. Řízení kvality je podporováno integrací dalších nástrojů, jako jsou HP Unified Functional Testing, HP Business Process Testing a HP Sprinter.
- HP ALM Performance Center (4) nástroj pro kompletní správu, plánování a běh zátěžových testů, součástí této edice je nástroj HP LoadRunner, sloužící právě pro testování výkonu aplikací při zatížení.

V tabulce 3 je přehled funkcí jednotlivých edic. Z důvodu složitějšího překladu funkcí do českého jazyka jsem ponechal v tabulce jejich anglické názvy.

Funkce	(1)	(2)	(3)	(4)
Release management	1	1	1	1
Project Planning and Tracking	1	x	x	x
Risk Based Quality Management	1	x	1	x
Test Authoring and Execution	1	1	1	1
Test Resources	1	1	1	~
Test Configurations	1	x	1	~
HP Sprinter	1	х	1	x
Lab Management	1	х	х	~
Versioning	1	х	1	1
Baselining	1	х	1	1
Cross Project Customization	1	1	х	1
Sharing Defects	1	x	х	х

Tabulka 3 - Přehled funkcí jednotlivých edic (zdroj: [34])

Z tabulky jsou patrné jisté rozdíly v jednotlivých edicích, zákazník má tedy možnost se rozhodnout, která edice je pro jeho projekt tou nejvhodnější.

Tato práce popisuje prostředí edice číslo 3, tedy HP Quality Center Enterprise verze 12 (dále HPQC 12).

3.3 Instalace a konfigurace HPQC 12

V této podkapitole je popsáno, jak nejjednodušeji nainstalovat HPQC 12 na osobním notebooku či počítači a jak nakonfigurovat nástroj před prvním spuštěním.

Pro snazší orientaci se budu dále v textu odkazovat na obrázky, které obsahují číselné popisky v kulatých závorkách.

3.3.1 Příprava prostředí

Instalace HPQC se skládá z klientské a serverové části, podrobnější informace o systémových požadavcích pro jednotlivé verze HPQC jsou uvedeny na stránkách HP [36]. Po stažení podporovaného operačního systému (dále OS) lze instalaci provést virtualizací serveru, například pomocí VMware Playeru. Práce v tomto nástroji je velice jednoduchá, stručný postup lze najít na [37]. Aby virtualizace fungovala, je jí potřeba povolit v BIOSu. Nastavení se může lišit v závislosti na značce stroje, u HP je cesta následující:

System Configuration → Device Configurations → zaškrtnout Virtualization Technology (VTx)

Pro HPQC 12 jsem shledal jako nejvhodnější OS Windows Server 2008, který je i doporučen samotným HP. Novější verzi nedoporučuji, jelikož poté nastane problém s příliš novou verzí Internet Exploreru (dále IE), protože HPQC 12 podporuje pouze IE verze 8,9 a 10, zatímco ve Windows Serveru 2012 je defaultně daná verze 11. Naopak pro databázi doporučuji nainstalovat Microsoft SQL Server 2012, která oproti MS SQL Server 2008 funguje bezchybně. Nutná je pak ještě instalace MS SQL Server Management Studia 2012.

I přesto, že si lze při virtualizaci navolit parametry virtuálního stroje dle libosti, doporučuji provést menší úpravu xml souboru s názvem validations v instalační složce ALM_Win64. Konkrétně v řádce <memory enabled="true" threshold="8"/> přepsat číslo 8 v uvozovkách napří-klad na 4 či 3 a poté uložit. Tato řádka kontroluje velikost RAM paměti stroje při spuštění instalace ALM.

3.3.2 Konfigurace HPQC

Po prvotní instalaci spuštěním souboru setup.exe přichází na řadu konfigurace QC. Předpokladem pro spuštění instalace HP ALM je funkční lokální server. Pro snazší orientaci jsou kroky očíslovány, viz obrázek 4.

% HP ALM Server Configuration Wizard		<u>_ ×</u>
Welcome (1)	Database Server	
License Key (2)		
Database Server (3)	Database Type	
Site Administration Database Schema (4) Security (5)	MS-SQL (Win Auth.)	
Site Administrator User (6)		
File Repository Path (7)		
Application Server (8)	O Database Parameters	
HP ALM Service (9)	DB host name: localhost	¬
Mail Server (10)		-
Installation Summary (11)	DB port number: 1433	
Apply Configuration (12) (13)	Oracle SID:	
Finish	O Connection String	
	jdbc:mercury:sqlserver://localhost:1433	
	Database Administrator Login	
	DB admin user name:	
	DB admin password:	
	Back Next Cancel	Help

Obrázek 4 – Instalace QC, 3. krok, (zdroj: autor)

(1) Vítání – v prvním kroku je pouze vítání uživatele v instalaci.

(2) Licenční klíč – v druhém kroku je uživatel vyzván k zadání licenčního klíče, pro trial verzi je třeba zaškrtnout Use Evaluation Key a z nabídky vybrat požadovanou edici ALM.

(3) Databázový server – ve třetím kroku přichází na řadu nastavení databázového serveru. Database Type je potřeba zvolit v závislosti na tom, jaká autentifikace byla zvolena při instalaci MS SQL Serveru. Já jsem zvolil Windows autentifikaci. Pokud by uživatel zvolil SQL Server autentifikaci, vyplňoval by kolonky DB admin user name a DB admin password, viz obrázek 4. DB host name je localhost (stejně jako při přihlašování v MS SQL Server Management Studiu), DB port number lze ponechat předvyplněný 1433.

(4) Databázové schéma – čtvrtý krok vytváří databázové schéma, je vhodné ponechat předvybranou volbu Create a New Schema a předvyplněné Schema name.

(5) Zabezpečení – v pátém kroku je nastavováno zabezpečení. V Confidential Data Encryption jsem použil volbu Use default value, při přechodu na další krok se objeví varování, že to není bezpečná a doporučovaná volba, je na uživateli zda přijme toto riziko nebo zadá heslo své.

Zadání hesla pro Communication Security je vyžadováno a bez jeho vyplnění není možné přejít na další krok.

Security

ALM Server encrypts confidential data, such as passwords to external systems (DB, LDAP), and secures communication with other HP BTO applications.

Confidential Data Encryption
Enter a passphrase with at least 12 characters for secure storage of confidential data. Important: If you are installing a cluster of servers, make sure you enter the same passphrase on all nodes.
Use default value (unsecure)
Confidential data passphrase:
Retype Confidential data passphrase:
Communication Security
Enter a passphrase with at least 12 characters for secure communication.
Retype Communication security passphrase:

Obrázek 5 – Nastavování zabezpečení, 5. krok, (zdroj: autor)

(6) Správce webu – v šestém kroku jsou nastavovány administrátorské přihlašovací údaje do QC. Uživatel může vyplnit dle libosti.

Site Administrator User

Type user name and schema name and p	d password to be used when logging in to Site Administration. This is not the same as the Site Administration database bassword.
User name:	Patrik-admin
Password:	••••••
Retype password:	•••••

Obrázek 6 – Nastavování přihlašovacích údajů, 6. krok, (zdroj: autor)

(7) Úložiště souborů – v tomto kroku uživatel nastavuje, kam se budou ukládat soubory v rámci QC. Volby lze ponechat tak, jak jsou přednastaveny.

(8) Aplikační server – zde je nastavován port, na kterém poběží server. I v tomto kroku lze ponechat přednastavenou volbu.

(9) HP ALM Service – v 9. kroku jsou nastavovány přihlašovací údaje do HP ALM Service. Jméno a heslo je potřeba zadat stejné jako má uživatel pro přihlášení do systému. Pokud uživatel heslo do systému nemá, zvolí libovolné. Do Domain je potřeba vyplnit název domény,

ten lze zjistit přes Tento počítač, ve Vlastnostech.

HP ALM Service

Type your Windows login user name, password, and domain to enable HP ALM Server to run as a service using your local network.

If you do not specify a user account, the local system account will be used to run the HP ALM service. In this case, HP ALM Server will not have access to files over the network.

User Name:	Patrik
Password:	
Domain:	WIN-H GTK THU39

Obrázek 7 – Nastavení údajů do HP ALM Service, 9. krok, (zdroj: autor)

(10) Mailový server – v kroku číslo 10 je nastavován mailový server, to pokud by uživatel chtěl v QC posílat zprávy jednotlivým uživatelům, u kterých administrátor zadá jejich mailovou adresu. Lze ponechat volbu None.

(11) Shrnutí instalace – krok 11 je shrnutím konfigurace, stiskem tlačítka Next je konfigurace zahájena.

(12) Potvrzení konfigurace – zde probíhá kompletní instalace, tento krok informuje uživatele o úspěšně či neúspěšně provedené konfiguraci.

(13) Konec – pokud proběhl předchozí krok úspěšně, je nabídnuto uživateli spustit ALM server.

3.3.3 Vytvoření domény a projektu

Po úspěšné konfiguraci se uživateli spustí IE s přihlašovací stránkou. Po zadání přihlašovacích údajů se zjeví chybová hláška. To proto, že není vytvořená doména, projekt a uživatel není přidělen k projektu.

Do prohlížeče je potřeba zadat adresu <u>http://localhost:8080/qcbin/SiteAdmin.jsp</u>. Port 8080 platí za předpokladu, že ho uživatel v 8. kroku konfigurace nezměnil na jiný. Po zadání přihlašovacích údajů už nic nebrání vytvoření domény a nového projektu. Postup je následující:

(1) Vytvořit doménu.

(2) Vytvořit projekt.

(3) Vytvořit uživatele a přiřadit je k projektu.

Podrobněji viz dále.

CHP ALM - Quality Center 12.00 - Site Administration - Windows Internet Explorer		_ 8 ×
🚱 🗢 🗟 http://ocalhost:8080/qcbin/SiteAdmin.jsp 📃 🖻	Bing 🗲 🗙 📴 Bing	₽ •
👷 Favorites 🛛 🙀 🕨 Suggested Sites 🔹 🖉 Web Silce Gallery 👻		
Q HP ALM - Quality Center 12.00 - Site Administration	🏠 🔹 🔂 👻 🚍 🖶 👻 Page 🔹 Safety 🔹	Tools 🔹 🕜 🕶
Quality Center - Site Administration	Site Administrator: Patrik	LOGOUT
Tools + Help + Site Projects Site Users Site Connections Licenses Servers DB Servers Site Configuration Site Analysis		
Create Domain Create Project Projects Create Project Projects Directoryic Directoryic Create Domain Physical Directoryic Create Domain Viscolianeous Create Domain Contact E-mail: User Quots: User Quots: Create Domain	able Versioning	
	Server Time: 4/20/2015	1:12 PM
HP ALM - Quality Center 12:00 - Site Administration	Local intranet Protected Mode: Off	€ 100% -
Arstart 🚡 🛛 📜 💆 🧭	* 🖻 🖗 🕻	3 1:12 PM 📃

Obrázek 8 – Vytvoření domény, projektu, přiřazení uživatelů k projektu, (zdroj: autor)

(1) Vytvořit doménu – doména se vytvoří stiskem tlačítka (1) viz obrázek 8, zadáním libovolného názvu a stiskem OK.

(2) Vytvořit projekt – nový projekt se vytvoří stiskem tlačítka (2). V zobrazeném okně uživatel zvolí Create an empty project, poté ho pojmenuje a přiřadí do již vytvořené domény.

(2a) V následujícím kroku je potřeba zvolit MS-SQL jako Database Type a localhost jako Server Name v závislosti na našem nastavení v instalaci.

(2b) V dalším kroku přiřadíme uživatele k projektu kliknutím na jméno uživatele a šipku viz obrázek 9.

atrik	Available Users	d hi
	User Name	Full Name

Obrázek 9 – Přiřazení uživatele k projektu, (zdroj: autor)

(2c) V dalším kroku je možnost přidat dostupná rozšíření pro danou verzi QC.

23

(3) Uživatelé – kliknutím do záložky Site Users (3) viz Obrázek 8 lze spravovat, přidávat či odstraňovat uživatele.

3.3.4 Přihlášení do HPQC 12

Doména i projekt jsou vytvořeny, přes adresu <u>http://localhost:8080/qcbin/start_a.jsp</u> je nyní možno se přihlásit. Přihlašovací stránku lze vidět na obrázku 10.

C HP ALM - Quality Center 12.00 - Windows Interne	et Explorer							_ 8 ×
C C + ttp://localhost:8080/qcbin/start_a.jsp					💌 🗟 🐓 🗙	Bing		P -
🔆 Favorites 🛛 🍰 Suggested Sites 👻 💋 Web Slice	e Gallery 🔻							
HP ALM - Quality Center 12.00					合	• 🔊 - 🗈 🖶 • P	age • Safety •	Tools + 🔞 +
								-
		Applicati	on Lifecycle N	lanagement				
		Quality Cen	ter					
		Name:	Patrik					
			- define					
		Password:	********					
			Automatically log in to and project on this ma	o my last domain achine				
			Forgot password	Authenticate				
		Domain:	TTTOURNAMENT	~				
			,					
		Project:	BP_testing	~				
				Login				
				Login				
								-
HP ALM - Quality Center 12.00					🚯 📢 Local intrane	t Protected Mode: Off	e	100% • //
🎦 Start 🐁 🛛 🍃 💆 🌿 🏉							* 🖻 🗑 🕼	1:32 PM 📃

Obrázek 10 – Přihlášení do HPQC12, (zdroj: autor)

Po zadání jména a hesla je třeba nejprve stisknout tlačítko Authenticate a poté až Login.

4 Řízení manuálních testů

V této kapitole je předvedeno, jak řídit manuální testy v HPQC. Zacházení s nástrojem demonstruji při příležitosti testování pracovní verze aplikace TTTournament, kterou v rámci své bakalářské práce vyvinula a mně poskytla Lucie Růžičková [2]. Cílem této kapitoly je vznik příručky pro řízení manuálních testů v nástroji HPQC 12.

4.1 Představení testované aplikace

Aplikace TTTournament slouží pro organizování turnajů ve stolním tenisu. Skládá se z hlavního okna, kde si uživatel volí požadované výstupy a z pomocných oken, která se zobrazí po stisku některého z tlačítek. Hlavní okno po spuštění aplikace a vytvoření turnaje je vyobrazeno na obrázku 11. Uživatelská příručka k aplikaci je v příloze 1 této práce.

			TTTou	mament		-	×
Nový Tisk							
					Poh	ar rektora, VŠE-Žižko	ov 20.4. 2015
		Uložit	Přidat hráče Vybrat vše	Tisk	Vylosovat		^
			Vyberte pi	řítomné hráče.			
	Nasazení*	Jméno*	Příjmení*	Rok narození	Tým*		
	1	Prokopcov	Dmitrij	1993	Hennebont GV	Odebrat	
	2	Konečný	Tomáš	1992	Carbo-Koks TTS Polonia Byto	Odebrat	•
	3	Širůček	Pavel	1996	STEN marketing HB Ostrov	Odebrat	•
	4	Vráblík	Jiří	1993	KS Dekorglass Działdowo	Odebrat	
	5	Jančařík	Lubomír	1992	TTC Zugbrücke Grenzau	Odebrat	
	6	Vožický	Bohumil	1993	Post SV Mühlhausen	Odebrat	
	7	Košťál	Radek	1992	TJ Sokol Králův Dvůr	Odebrat	
	8	Tregler	Tomáš	1996	SF SKK EL Niňo Praha	Odebrat	
	9	David	Petr	1993	Post SV Mühlhausen	Odebrat	
	10	Olejník	Martin	1992	DTJ Hradec Králové	Odebrat	
	11	Výborný	Richard	1993	TSV Bad Königshofen	Odebrat	
	12	Reitčnies	David	1992	SE SKK EL Niňo Praha		•
Testovací	seznam hráčů						

Obrázek 11 - Hlavní okno aplikace [2], (zdroj: autor)

4.1.1 Cíle a druhy testů

Cílem testování bylo otestovat funkcionalitu aplikace a tím vyhodnotit, zda je aplikace připravena k užívání. Testovány byly stěžejní funkce aplikace, jakými jsou založení turnaje, vytvoření seznamu hráčů, přidání a editace hráče, losování turnaje, tisk rozlosování a zadávání výsledků.

Testování proběhlo manuálně, k dispozici jsem měl pouze hotovou aplikaci a uživatelskou příručku, bylo to tedy black box testování. Abych mohl aplikaci skutečně otestovat, bylo nutné ji spustit, současně se tedy jednalo i o dynamické testování.

4.2 Plánování testování

Před započetím tvorby požadavků a testovacích sad je nejprve nutné založit Release. Ten se zakládá v závislosti na tom, jak je naplánovaný projektovým manažerem. Release může být po týdnech, měsících nebo klidně po půl roce.

V rámci releasu se plánují cykly. V jednom releasu může být jeden či více cyklů. Pokud by například měl vývojový tým releasy každý měsíc, může mít cykly plánované po týdnech, čili 4 cykly.

Základní kroky pro vytvoření releasu a cyklu, které jsou níže detailněji popsány, jsou:

- (1) Kliknout do modulu Management.
- (2) Vytvořit složku pro release
- (3) Kliknout na vytvořenou složku.
- (4) Založit release.
- (5) Zadat jméno releasu.
- (6) Zadat datum, kdy release začíná.
- (7) Zadat datum, kdy release končí.
- (8) Zadat popis releasu.
- (9) Založit cyklus.

V následujícím textu pomocí čísel kroků odkazuji na obrázek 12. Tímto způsobem je odkazováno na obrázky v dalším textu práce.

HP ALM - Quality Center 12.00 - Windows Internet Explorer		_ 8 ×
	💌 🗟 🖘 🔀 Bing	P -
😭 Favorites 🛛 🍰 Suggested Sites 🔹 🙋 Web Slice Gallery 🔹		
C HP ALM - Quality Center 12.00	🏹 🕶 🔜 🖛 🕶 Page 🕶 Safety	/ • Tools • 🔞 •
		×I ^
Application Lifecycle Management		Logout
A Beleases Edit View		
	Name: TTTournament_testing_04_15] — <u> </u>
(▲) Dashboard × (→) (↓) (↓) (↓) (↓) (↓) (↓) (↓) (↓) (↓) (↓	E Details Details	ă
Management (1) \hat{z} (2) (4) (9) \hat{z} Releases	0 Attachments	- <u>B</u>
(3) TTTournament	◆Start Date: 4/19/2015 ▼ ●End Date: 5/3/2015 ▼	
Libraries	(0) (7)	
Requirements *		
Requirements		
Teating 0	4	
Transmission A	Description	-
Co Test Resources		5
🚺 Test Lab	B I ⊻ A 🎂 ≣ ∰ ⊡ ⊡ ▷୩ ୩০ ୭ (♥ ▦ № ଵ, ଵ, ⊠	
🖄 Test Runs	Testovani zakladni funkcionality aplikace TTTournament od Lucie Ruzickove.	
Defects		
		_
HP ALM - Ouality Center 12.00	ОК Сюзе нер	100% -

Obrázek 12 – Tvorba releasu, (zdroj: autor)

Pro založení releasu je nutné poklepat na modul Management (1), v něm na Releases. Poté kliknout na ikonku New Release Folder (2) nebo lze použít klávesovou zkratku Ctrl+Alt+N. Po zadání jména složky a uložení je potřeba na ní poklepat (3) a stisknout tlačítko pro vytvoření releasu (4) – klávesová zkratka Ctrl+R.

V zobrazeném okně je potřeba vyplnit název releasu (5), odkdy dokdy trvá (6+7) a popis (8). Pro testování aplikace jsem zvolil dvoutýdenní release. Vlastnosti releasu jako datové rozmezí či název lze jednoduše změnit kliknutím pravým tlačítkem myši na daný release nebo pomocí ikonek, jež se nachází ve stejné řadě jako ikony (2), (4) a (9).

Pro založení nového cyklu je potřeba stisknout tlačítko (9) nebo klávesovou zkratku Ctrl+Y. V cyklu je pak možnost sledovat progres testování, počet testů přiřazených k danému cyklu nebo počet zbývajících testu ke spuštění.

Záložka Libraries, která se také nachází v modulu Management, jak podle názvu napovídá, je knihovnou všech entit vytvořených v projektu, tedy požadavků, zdrojů a testů.

4.3 Tvorba požadavků

Jak už jsem zmínil v předchozím textu, funkční požadavky definují, jakou funkcionalitu má software poskytovat. Zakládají se v modulu Requirements. Požadavky jsou vlastně testovacími případy, které vytváří zadavatel či vývojář a které je v rámci projektu potřeba otestovat. Požadavky jsou často dodávány ve Wordu či Excelu. Daný dokument je pak vhodné přiložit do přílohy k požadavku, či nakopírovat do popisu viz obrázek 13 tak, jako jsem to provedl při tvorbě požadavků pro aplikaci TTTournament, kdy jsem od autorky obdržel popis základních funkcí, který posloužil pro specifikování požadavků.

Požadavky se tvoří následovně:

- (1) Založit složku pro požadavek.
- (2) Poklepat na vytvořenou složku.
- (3) Vytvořit požadavek.
- (4) Zadat údaje o požadavku.

Requirem	Requirements Edit View Favorites Analysis (1) (3) (3) (3) (3) (1) (3) (2) (3) (3) (3) (4) (4) (4) (4) (4) (4) (4) (4) (4) (5) <					
No Filter Defined						
68 🗭 🏲	Name	4	Req ID	Direct Cover Status	Author	
	- Requirements (2)	<u>0</u>				
	CTE 01_Zakladani turnaje	>3			patrik	
	<u> 01_</u> 01-Zakladani t	urnaje <u>9</u>		Vot Covered	patrik	
	🛴 01_02-Vytvoreni s	ezna <u>11</u>		🐨 Not Covered	patrik	
	👻 🔚 02_Sprava hracu	<u>5</u>			patrik	
	<u> 02_</u> 01-Pridani hrac	æ <u>12</u>		🐨 Not Covered	patrik	
	02_02-Editace hrad	ce <u>13</u>		🐨 Not Covered	patrik	
	🗕 🔚 03_Losovani	<u>6</u>			patrik	
	<u> 03</u> _01-Losovani	<u>14</u>		🐨 Not Covered	patrik	
	🚽 🥅 04 Tisk	7			patrik	
Descripti	on Comments Rich Text	Attachme	ents History			
BI	⊻႔ၿ ≣≣∎वव	⊳¶ ¶⊲	[©] פ ⊞ ′	🗞 ବ୍ 🔍 🖾		
Cilem je ote vytvoreni s	estovat zakladni funkcnost aplikac seznamu hracu.	e, tou je za	lozeni turnaje a			

Utevre se po spusteni aplikace okno, kde si uzivatel muze zvolit, zda vytvořit nový turnaj neho načíst noslední noužitý. Pokud uživatel zvolí nový turnaj

Obrázek 13 – Tvorba složek požadavků, (zdroj: autor)

Pro založení požadavků je třeba poklepat na Requirements, poté na tlačítko (1) New Folder (Ctrl+Alt+N), pomocí něhož vytvoříme složku pro požadavky. Poté poklepat na nově vytvořenou složku (2) a stisknout ikonu (3) New Requirement (Alt+N). V každé složce požadavků může být několik jednotlivých požadavků, složky je dobré vhodně logicky dělit v závislosti na funkcích aplikace. Za zmínku stojí i sloupec Direct Cover Status, který je ve statusu Not Covered (nepokryt). To proto, že k danému požadavku prozatím není přiřazen žádný test. Pro

testovanou aplikaci jsem vytvořil pět složek požadavků, které jsou částečně vidět na obrázku 13, do kterých jsem založil jednotlivé požadavky, kterým jsou později přiřazeny samotné testy (viz kapitola 4.4.2).

Při tvorbě požadavku (4) je potřeba zadat jméno (4a) viz obrázek 14, typ požadavku (4b) a přiřadit k cyklu releasu (4c), čímž se automaticky přiřadí i k samotnému releasu, do něhož patří přiřazený cyklus. Je zde možnost přidat popis či přílohu (4d), což už často není třeba, v mém případě souhlasí popis složky požadavků s popisem samotného požadavku, odkazuji se tedy na popis složky. Zadávané údaje se liší podle projektu či firmy, pokud jsou požadavky vedeny podle priority, je možnost zadat i prioritu. Okno pro založení požadavku je na obrázku 14.

📴 Requirement Details					
Image: Second		(4a) ani turnaje	Requir	rement Type: 🛴 Test	(4b)
 Petalis Rich Text (4d) Attachments Linked Defects Requirement Trace Test Coverage Business Models Li Risk Assessment History 	Author: Creation Time: Modified: Product: (4C) Target Cycle	patrik ⋈ ✓ 8:42:53 PM ✓ 4/22/2015 8:42:53 PM ✓ 01_cyklus ✓ 01_cyklus ✓	Creation Date: Direct Cover Stat Priority: Reviewed: Target Release:	4/22/2015 Not Covered Not Reviewed TTTournament_testi	
	B I U A ab Popis viz. popis pozada	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ↔ ✓	esting_04_15 5 Clear	२ छ । ।K Cancel	Негр

Obrázek 14 – Tvorba požadavků, (zdroj: autor)

4.3.1 Import požadavků z MS Word a Excel

Pro import požadavků z Wordu a Excelu je potřeba doinstalovat přídavné nástroje do HPQC.

Přes adresu http://localhost:8080/qcbin/addins.html (pokud uživatel při instalaci nezvolil jiný port), je potřeba kliknout na More HP ALM Add-ins, v horní části na ikonku Add-Ins a pod Add-ins for Microsoft Applications najít odkazy, ze kterých je možnost stažení těchto nástrojů.

Po nainstalování doplňků je potřeba zadat data do Excelu či Wordu a poklepat na záložku Doplňky, poté kliknout na Export To HP ALM. Pomocí průvodce, kde je mimo jiné potřeba zadat URL serveru, přihlašovací jméno a heslo do QC, zvolit doménu a projekt, je možno nahrát požadavky, ale i testy či defekty v osmi krocích.

4.4 Příprava testování

V této podkapitole je předvedeno jak se v nástroji HPQC tvoří a navrhují testy. Příprava testů se provádí v modulu Testing, v záložce Test Plan.

4.4.1 Tvorba a návrh testů

Tvorbu a návrh testů probíhá následovně:

- (1) Vytvořit složku pro testy.
- (2) Založit test do složky.
- (3) Přepnutí se do záložky pro návrh testu.
- (4) Samotný návrh testu.



Obrázek 15 – Tvorba testovacích složek, (zdroj: autor)

Proces tvorby a návrhu testů začíná vytvořením složky Ctrl+Alt+N nebo kliknutím na (1) viz obrázek 15. Složky mohou být tvořeny a plněny testy libovolně avšak je opět vhodné složky logicky členit podle testovacího plánu, jak bude probíhat testování komponent aplikace.

Pro testovanou aplikaci jsem zvolil stejné členění složek tak jako v případě požadavků. Stejné názvy nevadí, naopak je to podle mého názoru přehlednější při přiřazování testů k požadavkům viz 4.4.2.

Do vytvořených složek se vkládají a navrhují testy. Testy se tvoří pomocí klávesové zkratky Ctrl+N nebo kliknutím na (2). Mně se názvy testů shodují s názvy požadavků, na projektu kde byly názvy složitější a požadavků mnohem více se shodnost názvů osvědčila. Testy se navrhují kliknutím na daný test a poté klikem na (3) Design Steps v obrázku 16. Zde se zaznamenávají testovací skripty klikem na (4) nebo Alt+N.

Tests Edit View Favo	orites Analysis	🗒 Design Step Details	<u>_ X</u>
📓 🖏 🗙 😷 🔽 - i	💷 0 🌬 🛪 🖂 -	⊠ < > © Ø ≫ % % < ≌ * <u>®</u> ≝	
No Filter Defined	Details Design Ste	Step Name: Step 14	
Name	(4)	Details Details	
- Subject		Attachments (4a)	
▶ ¿ Unattached	Step Name	Description	
🚽 🔚 01_Zakladani	Step 1	BIUA 🏙 🧮 🖽 🖬 🖛 🗤 🦃 🥮 🖏 🔍 👦	
🍃 01_01-Zał			
01_02-Vyt	Step 2		
👻 🔚 02_Sprava hra	Step 3		
02_01-Pric			
02_02-Edi	Step 4	4	
🗕 🖂 03_Losovani		(4b)	
03_01-Los	Step 5		
- 🛁 04_Tisk			
04_01-Tis	Step 6		
■ 04_02-Tis	Step 7		
05_2adavani v 05_01-Zac	Step 8		
	Step 9		
<	Step 10	OK Cancel	Help

Obrázek 16 – Tvorba testu, (zdroj: autor)

Poté se zobrazí okno, kde se vyplňuje popis kroku, co má tester provést za úkon (4a) a předpokládaný jev (4b), který má nastat po vykonaném úkonu testera. Uloží se stiskem OK. Na obrázku 17 lze vidět testovací skript k testu Zakládání turnaje.

Details Design Ste	Parameters Test Configurations Atta	chments Req Coverage Linked Defects D)ері 🔇 🔪
2 6 423 🗙 🖎	🖺 0 & 🖗 - % 0 🛄 🖽	≙ ♥	
Step Name	Description	Expected Result	-
Step 1	Spust aplikaci.	Aplikace se spustí, nabídne založení nového turnaje.	
Step 2	Stiskni tlacítko NE.	Aplikace nacte predchozí turnaj z databáze.	
Step 3	Ukonci a znovu spust aplikaci.	Aplikace se spustí, nabídne založení nového turnaje.	
Step 4	Stiskni tlacitko ANO.	Zobrazí se upozornení že dojde k odstranení predchozích dat.	
Step 5	Stiskni NE.	Okno se zavre, aplikace je stále spuštená.	
Step 6	Zopakuj stepy 3+4 a stiskni ANO.	Otevre se okno pro zadání názvu a místa turnaje.	
Step 7	Zadej nekorektní údaje (císla) a ULOŽIT.	Upozornení na zadání nekorektních údaju.	
Step 8	Nevypln ani jedno povinné pole.	Výzva k vyplnení všech polí.	
Step 9	Vypln pouze jedno pole.	Výzva k vyplnení všech polí.	
Step 10	Zadej korektní údaje a ULOŽIT.	Turnaj vytvoren.	•

Obrázek 17 – Testovací skript a jeho kroky, (zdroj: autor)

Při tvorbě testovacích skriptů je dobré řídit se pravidlem méně je více. Tím je myšleno, že je lepší ve skriptech psát krátké, jasné kroky ve větším počtu. Pokud by tester měl číst dlouhý krok pro vykonání, hrozí pak jeho nekompletní provedení a tím by celý test ztrácel svou podstatu. Pokud je například požadavek na otestování zobrazení webové aplikace, je vhodné přiložit do přílohy snímek obrazovky požadovaného výstupu a na něj se odkazovat v daném kroku. Přehled všech testovacích skriptů je v příloze 2.

4.4.2 Přiřazování testů k požadavkům

Po vytvoření testů je potřeba je přiřadit k jednotlivým požadavkům následovně:

- (1) Kliknout do záložky pro přiřazení testu k požadavkům.
- (2) Kliknout na strom požadavků.
- (3) Najít určitý požadavek a přiřadit k danému testu.

Testy se k požadavkům přiřazují v záložce Req Coverage (1) – obrázek 18. V pravé části se nachází Requirement Tree (2), v něm je potřeba najít a kliknout na příslušný požadavek a šipkou Add to Coverage (3) přiřadit.

📑 🕺 🗙 💋 V - 💷 🛛 Þ 🕴 - A	. 🗐		(1)	
No Filter Defined	Details Design Steps	Parameters Test Configurations	Attachments Req Coverage	Linked Defects Dependencies Business M
Name	🛐 Select Req 🔒 😭 🕅			(2)
▶ 2 Unattached	Entity Name	Req: Description	· · · · · · · · · · · · · · · · · · ·	Requirement Tree
🚽 🔚 01_Zakladani turnaje	🔥 01 01-Zakladani turnaje		(3	
🕞 01_01-Zakladani turnaje			•	
D1_02-Vytvoreni seznamu				✓
🚽 🔚 02_Sprava hracu				✓
02_01-Pridani hrace				风 01 01-Zakladani turnaje
02_02-Editace hrace				
🚽 🔚 03_Losovani				D 2 Sprava bracu
🕞 03_01-Losovani				
- ▼ 04_Tisk				b Co_Costvan
📑 04_01-Tisk pdf				
04_02-Tisk printer				F 05_2adavani vysledku
🕶 🔚 05_Zadavani vysledku				
05_01-Zadavani vysledku				

Obrázek 18 – Přiřazování testu k požadavku, (zdroj: autor)

Tím je docíleno toho, že požadavky již nejsou ve statusu Not Covered, ale ve statusu No Run, jak lze vidět na obrázku 19. To znamená, že testy jsou úspěšně přiřazeny ke svým požadavkům, nebyly však zatím spuštěny. Jak se spouští testy je popsáno v podkapitole 4.5.

📸 🐮 🗶 🔁 T - 眠 🛛 Q 주 🔤 - 🏲 Q 슈 - U - 💷											
No Filter Defined											
6 🛚 单 🏲	Name	•	Req ID	Direct Cover Status	Author						
- E	Requirements	<u>0</u>									
-	🔚 01_Zakladani turnaje	1			patrik						
	<u> 01_</u> 01-Zakladani turnaje	<u>6</u>		🖸 No Run	patrik						
	01_02-Vytvoreni seznamu	7		🖸 No Run	patrik						
-	🔚 02_Sprava hracu	2			patrik						
	02_01-Pridani hrace	<u>8</u>		🖸 No Run	patrik						
	02_02-Editace hrace	<u>9</u>		🖸 No Run	patrik						
-	🔚 03_Losovani	<u>3</u>			patrik						

Obrázek 19 – Přiřazené testy k požadavkům, (zdroj: autor)

Jak už jsem popisoval v podkapitole 4.4.1, v tuto chvíli přijde vhod mít požadavky pojmenovány stejně jako testy, vyhledání a přiřazení testu k požadavku je pak jednodušší.

4.5 Spouštění testů

V této podkapitole je předvedeno spouštění navrhnutých testů. Testy jsou spouštěny ze záložky Test Lab, která se nachází v modulu Testing.

Pro spuštění testů uživatel postupuje takto:

- (1) Vytvoří složku pro testovací sady.
- (2) Vytvoří testovací sadu.
- (3) Přiřadí vytvořené testy do vytvořených testovacích sad.
- (4) Spustí test.

Po vytvoření složky poklepáním na ikonu (1) viz obrázek 20, je potřeba vytvořit testovací sadu kliknutím na ikonu (2). Složky testů lze dělit například podle sprintu či releasu. Vzhledem k časovému horizontu testování a rozsahu funkčnosti aplikace TTTournament jsem testovací sadu pojmenoval Zakladni_funkcionalita, která zahrnuje všechny testy. V případě, že by byly přidány další funkce do aplikace, bylo by záhodno testovací sady pojmenovat a rozdě-lit vhodněji.

Test Sets Edit View Tests F	🔊 New Test Set					<u>_ 0 ×</u>
(1) (2) (2) ★ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓						
No Filter Defined	Name: Zakladni_funkcionalit	а		Тур	e: <u>(</u> Default	~
👻 🔚 Root						
Unattached	Details	Details				
TTTournament	Attachments	Baseline:		 Close Date: 		~
		Modified:		Open Date:		~
		Status:		 Target Cycle: 		
		Test Set Folder:	TTTournament			
		Description				
		BIUA	≣≣∎⊡ ⊡ № ¶	₄ 🖻 🥙 🖽 🍫 🤤	(@. 🛛	
		Testovaci sada, ve kte	ere se nachazeji testy na zakla	adni funkcionalitu aplikace T	TTournament.	
					OK Close	Help

Obrázek 20 – Tvorba testovací sady, (zdroj: autor)

Testy se přidávají kliknutím na vytvořenou testovací sadu a poté kliknutím na Execution Grid (3), viz obrázek 21. Testy lze přidat jak přes Test Plan Tree, tak přes Requirements Tree a to klikem na daný test a na šipku (3a).

🛄 Sele		Run Test Set	X 🔁 🛛 -	R 0 Þ 🕅	» *	Test Plan Tree Requirements Tree 🗙
Details	Execution Grid	Execution Flow	Automation Atta	chments Linked	Def 🔇 🔪	
n 🔉 👝 Þ	Name	Test: Test Name	Туре	Status	Â	👻 🚍 Subject
						Unattached
	[1]01_01-Zaklad	눩 01_01-Zaklad	MANUAL	🕒 No Run		✓ = 01_Zakladani turnaje
	[1]01 02-Vytvore	🕞 01_02-Vytvor	MANUAL	🕒 No Run		🗈 01_01-Zakladani turnaje
	[1]02 01-Pridani	🕞 02_01-Pridani	MANUAL	🔁 No Run		🐎 01_02-Vytvoreni seznamu
	[1]02 02-Editace	02_02-Editac	MANUAL	🔁 No Run		👻 🔚 02_Sprava hracu
	[1]03 01-Losovani	🕞 03_01-Losov	MANUAL	😒 No Run		02_01-Pridani hrace
	[1]04 01-Tisk pdf	🕞 04_01-Tisk pdf	MANUAL	😒 No Run		02_02-Editace hrace
	[1]04_02-Tisk pri	🕞 04_02-Tisk pr	MANUAL	🔁 No Run		
	[1]05_01-Zadava	🕞 05_01-Zadav	MANUAL	🔁 No Run	~	03_01-Losovani
<					>	🔻 🥅 04 Tisk
Last Rur	n Report				¥	04_01-Tisk pdf
No resu	lts found.	3	Steps Details			▶ 04_02-Tisk printer
						👻 🔚 05_Zadavani vysledku
						💦 05_01-Zadavani vysledku

Obrázek 21 – Přiřazení testů do testovací sady, (zdroj: autor)

Testy jsou tímto připraveny ke spuštění, v této fázi se už jedná o samotné testování aplikace, dosud to byly pouze přípravné procesy. Spouštějí se klikem na malou šipku u Run (4) viz obrázek 21 a vybráním volby Run with Manual Runner (Ctrl+Alt+F9). Poté se zobrazí okno, které je vidět na obrázku 22, kde je potřeba kliknout na Begin Run.

D Manual Runner: T	est Set Zakladni_funkcion	alita, Test [1]01_01-Zal	kladani turnaje	
Begin Run E	End Run 🗙 Cancel Run	🖞 🍟 🔹 🛈 OS Info		?
Run Details (1)				
Run Name:	Run_4-25_22-53-10	•Status:	Not Completed	-
Test Instance:	[1]01_01-Zakladani tur 📃	Test Set:Name:	Zakladni_funkcionalita	
•Tester:	Patrik 🖂 🗸	Baseline:		
Change Detectio		Change Status:		
Configuration ID:	1001	Configuration: Na	01_01-Zakladani turnaje	
Draft Run:	N 🗸	Duration:	0	-
Comments				
BIUAab	≣ <u>≣</u> ⊡⊡ ≥¶ ¶⊲) (° III 4% Q	Add Co ≪ ⊠	omment
Test Details				
Name: 01_01-Zaklad	ani turnaje		Test Detail	<u>s</u>

Obrázek 22 – Spouštění testu, (zdroj: autor)

V nově zobrazeném okně se objeví testovací skript, který byl vytvořen v kapitole 4.4.1. Současně s tím jsem měl spuštěnou aplikaci TTTournament a podle jednotlivých kroků testu jsem ji postupně testoval. Na obrázku 23 lze vidět test na zakládání turnaje. V Description je pokyn pro testera, co má vykonat, v Expected je očekávaný výstup aplikace a v kolonce Actual píše tester, jaký výstup skutečně dostal. Pokud se očekávaný a skutečný výstup shoduje, používám stručný a jasný komentář a to OK.

Krok se označuje jako úspěšný klikem na tlačítko (a), pokud by byl krok shledán testerem jako neúspěšný, použije ikonu (b). I v této fázi je možnost spravovat kroky testu, pokud tester vyhodnotí, že by bylo záhodno nějaký krok přidat (c) nebo naopak nějaký odebrat (d). Test se ukončuje klikem na ikonu (e) nebo klávesovou zkratkou Ctrl+Q. Výsledky testů jsou zobrazeny v kapitole 4.7.

D Manual Runner: T	est Set Zakladni_	funkcionalita, Te	st [1]01_01-Zaklad	ani turnaje 📃	l ×						
	[] - 10 - 10 - 10 [] - 10 - 10 [] - 10	¥† 🗗 All		(e)	?						
Step Name	Status	Exec Date	Exec Time		~						
Step 7	Passed	4/25/2015	11:26:32 PM								
Step 8	Passed	4/25/2015	11:26:50 PM								
Step 9	Passed	4/25/2015	11:26:58 PM								
Step 10	Passed	4/25/2015	11:27:04 PM		11						
Step 11	Passed	4/25/2015	11:27:19 PM								
Step 12	Passed	4/25/2015	11:27:41 PM								
Step 13	Passed	4/25/2015	11:28:06 PM								
Description B I U A ab Stiskni ANO.	Description B I U A ⊕ ⋮Ξ ⋮Ξ □ □ ▷୩ ୩୦ ▷ ♥ ♥ ⊞ № ♀ ♀ ⊡ Stiskni ANO.										
Expected:		(Ac	tual:								
B I U A ab	zadání údaju o novér	▷¶ ¶⊲ ♀ :		≣ <u>≣</u> ⊡ ₪	*						

Obrázek 23 – Spuštěný test, (zdroj: autor)

Přehled všech spuštěných testů a jejich status lze najít v záložce Test Runs, viz obrázek 24, jsou zde současně informace o tom, kdy, v kolik, kým a z jakého počítače byl test spuštěn. Statusy testů je možno zjistit i přes modul Requirements poklepáním na vybraný požadavek, ve sloupci Direct Cover Status.

y: Exec Date[Descending];Exec Tin	ne[Descending]								-
Run ID	Run Name	Test: Test Name	Configuration:	Status	State	Duration	Exec Date	Exec Time	Host	Teste
10	B Run_4-26_13	05_01-Zadavani	05_01-Zadavani	Passed		156	4/26/2015	1:59:39 PM	WIN-1L1Q475B	patrik
9	🖹 Run_4-26_13	04_02-Tisk printer	04_02-Tisk printer	😮 Failed		142	4/26/2015	1:56:39 PM	WIN-1L1Q475B	patrik
8	B Run_4-26_13	04_01-Tisk pdf	04_01-Tisk pdf	😧 Failed		258	4/26/2015	1:54:28 PM	WIN-1L1Q475B	patrik
Z	B Run_4-26_13	03_01-Losovani	03_01-Losovani	Passed		106	4/26/2015	1:50:47 PM	WIN-1L1Q475B	patrik
<u>6</u>	B Run_4-26_13	02_02-Editace hr	02_02-Editace hr	😧 Failed		210	4/26/2015	1:48:54 PM	WIN-1L1Q475B	patrik
5	B Run_4-26_13	02_01-Pridani hr	02_01-Pridani hr	😮 Failed		1719	4/26/2015	1:46:20 PM	WIN-1L1Q475B	patrik
4	Ph Run 4-25 23	01 02-Vytvoreni	01 02-Vvtvoreni	Passed		14	4/25/2015	11:58:01 PM	WIN-1I 1Q475B	natrik

Obrázek 24 – Přehled spuštěných testů, (zdroj: autor)

4.6 Zaznamenávání bugů

V této kapitole je předvedeno zaznamenávání bugů v HPQC. V podkapitole 4.6.1 jsou okomentovány nalezené bugy v aplikaci TTTournament.

Bugy lze zaznamenávat dvěma způsoby:

(1) Zaznamenat bug během testování kliknutím na ikonu (obrázek 25).

(2) Zaznamenat bug zpětně v modulu Defects.

První způsob, podle mého názoru elegantnější, je bug zaznamenat přímo během testování. V momentě, kdy tester shledá testovací krok jako neúspěšný (Failed), může ihned přes ikonu New Defect (1) tento bug zanést. Automaticky se vyplní jeho popis, tím pádem i cesta k zreprodukování a komentář testera k danému kroku.

Tester doplňuje povinně pouze kolonky, které jsou červeně, bez kterých QC nedovolí bug zaznamenat. Povinnými údaji jsou kolonky Summary (shrnutí) a Severity (závažnost), volitelnými pak datum, do kdy má být bug vyřešen, v jakém cyklu a releasu byl bug zaznamenán, zda je možno bug zreprodukovat či jeho priorita. Jak jsem již zmínil v kapitole 2.3.2, je při zaznamenávání bugu vhodné připojit přílohu, většinou snímek obrazovky, k tomu je vytvořena záložka Attachments.

Manual Runner: Test Set Zakladni_funkcionalita, Test [1]02_01-Pridani hrace											
🤣 🤹 🖉 - 60 - 10 - 10 - 10 - 10 - 10 - 10 - 10	New Defect	_O×									
Step Name Status (1)	💈 🗙 🖓 🖌 🤻 👧 🖆 🔍 Use Default Values 💷 Set Default Values										
Step 1 🜍 Passed	Summary: Nekorektni zadavani roku narozeni										
Step 2 💋 Passed											
Step 3 🔗 Passed	Details Details										
Step 4 🚫 Passed	Attachments										
Step 5 😯 Failed	Detected By: patrik Detected on Date: 4/26/2015	_ → _									
Step 6 Step 6	Severity: 2-Medium Actual Fix Time:										
	Assigned To: patrik 🗸 Closed in Version:	▼									
	Closing Date: V Detected in Cycle: D1_cyklus	▼									
Description	Detected in Delan TTTsuranment testi 🗐 🗤 Detected in Varai										
B I U A 💩 🗏 🗮 💷 💷	Description:										
Zadei rok parození v nekorektním tvaru (víc	B I 및 A 🏙 🗮 🔚 💷 💷 🕬 🕬 🖉 🌐 🤣 옥 옥 🖾										
	Test Set: Zakladni_funkcionalita Test: 02_01-Pridani brace	_									
Expected:	Run: Run_4-26_13-40-20										
	U A ab E F G										
Chybová hláška, max 4-místné.	Step: Step 5										
	OK Close	Неір									

Obrázek 25 – Zaznamenávání bugu během testování, (zdroj: autor)

Druhou možností jak zaznamenat bugy je přes modul Defects, kliknutím na ikonu New Defect (Alt+N). To se objeví stejné okno jako je na Obrázku 25, avšak popis bugu musí tester kompletně napsat celý sám.

Velkou nevýhodou při tomto způsobu je nutnost pamatování si v jaké testovací sadě, testu

a kroku bug nastal a jak jej zreprodukovat, což v případě testování velkého projektu není pohodlné a tester pak může zaznamenat bug, který vlastně ani neexistuje. V přehledu bugů pak není rozdíl v tom, zda byl bug zaznamenán způsobem prvním či druhým. Přehled bugů v modulu Defects je na Obrázku 26. Sloupce s jednotlivými údaji lze libovolně podle potřeby přemisťovat tahem myši.

Defects	Edit V	iew Favo	rites Analysis											
New [Defect	× 🔉	▼ - 📖 🏊 4	₽ 🖂 - 6	ի -	a 📴 🛛								
No Filter D	efined													
Actu	Assi	Defect	Description	Detecte	CI	Detected	Detected in	Detected on	Severity	Subject	Summary	Reprod	Status	Мо
	natrik	1	Test Set: 7aklad	natrik		01. cyklus	TTTournament t	4/26/2015	2-Medium	02 Sprava bracu	Nekorektni zada	Y	New	4/26/
	patrik	2	Test Set: Zaklad	patrik		01_cyklus	TTTournament_t	4/26/2015	2-Medium	02_Sprava hracu	Chybejici chybov	Y	New	4/26/2
	patrik	3	Test Set: Zaklad	patrik		01_cyklus	TTTournament_t	4/26/2015	2-Medium	02_Sprava hracu	Chybova hlaska	Y	New	4/26/2
	patrik	4	Test Set: Zaklad	patrik		01_cyklus	TTTournament_t	4/26/2015	3-High	04_Tisk	Chybova hlaska	Y	New	4/26/2
	patrik	<u>5</u>	Test Set: Zaklad	patrik		01_cyklus	TTTournament_t	4/26/2015	2-Medium	04_Tisk	Chybova hlaska	Y	New	4/26/2
Descriptio	on Atta	Chybova h	Linked Entities	History										*
Descriptio	n:							Comments:					Add Co	omment
BI	UA a	<u>▶</u> ≣]≡	• • •	90	▦	A 🖗 🔍 🔍	8	BIUA	한 🗄 🗄 🖬	I 🖸 🛯 🗤 🦻	🖱 🖽 🦓 🔍	•		
Test Set: Test: 04_(Run: Run_	Zakladni_ I2-Tisk pri 4-26_13-5	funkcionaliti inter 54-44	3				-							

Obrázek 26 – Přehled zaznamenaných bugů, (zdroj: autor)

4.6.1 Zaznamenané bugy

V této podkapitole jsou okomentovány bugy aplikace, které jsem zaznamenal. Komentář byl předán vývojářce aplikace.

- Bug 1 se nachází v testovací sadě Správa hráčů, test Přidání hráče. Testuje se přidání nového hráče do seznamu hráčů, kdy se zadávají povinné údaje o hráči. Ačkoliv rok narození není povinný, aplikace umožní uživateli zadat více než čtyřmístné číslo, byla by zde vhodná chybová hláška vyzývající k zadání korektního formátu roku. Závažnost jsem ohodnotil stupněm Medium, tedy střední.
- Bug 2 se nachází v testovací sadě Správa hráčů, test Editace hráče. Testuje se úprava údajů již existujícího hráče. Po změně roku narození na textový řetězec, aplikace umožní uživateli pokračovat v editaci dalších údajů, kolonka pro rok narození je pouze ohraničena červeným obdélníkem. Byla by zde vhodná chybová hláška, aby uživatel nemohl pokračovat v editaci. I s tímto nekorektně zadaným údajem je možné vylosovat turnaj. Závažnost je hodnocena stupněm Medium.
- Bug 3 se nachází v testovací sadě Správa hráčů, test Editace hráče. Je testována editace existujícího hráče. Aplikace nechá uživatele zadat stejné číslo nasazení, které už má jiný uživatel. V seznamu hráčů jsou pak dva hráči se stejným číslem nasazení. Závažnost ohodnocena stupněm Medium.
- Bug 4 se nachází v testovací sadě Tisk, test Tisk PDF. Je zde testován tisk seznamů do souboru PDF. Pro tisk je potřeba zaškrtnout checkbox, který seznam chce uživatel

vytisknout. Pokud checkbox uživatel nezaškrtne a stiskne TISK, dialogové okno i tak nabídne uložení seznamu. Ve skutečnosti žádný soubor uložen nebyl. Zde by bylo vhodné zabránit jevu, kdy má možnost uživatel tisknout "prázdný PDF soubor." Tento bug je ohodnocen stupněm High, tedy vysoký.

 Bug 5 – se nachází v testovací sadě Tisk, test Tisk printer. Je zde testován tisk seznamů klasickou tiskárnou. Pro tisk je potřeba zaškrtnout checkbox, který seznam chce uživatel vytisknout. Pokud checkbox uživatel nezaškrtne a stiskne TISK, nic se neděje a uživatel neví, zda aplikace přestala pracovat nebo naopak stále ještě pracuje. Zde by byla vhodná chybová hláška, že je pro tisk potřeba vybrat seznam hráčů. Závažnost byla stanovena na Medium.

4.7 Vyhodnocování testování

V této podkapitole je předvedeno, jak vygenerovat v HPQC grafy a reporty, ve kterých se nachází nejrůznější informace, například kolik testů bylo spuštěno, kolikrát byly dané testy spuštěny, jejich výsledky, progres testování v rámci releasu či cyklu, jak se měnilo množství bugů v čase a další. Tyto informace slouží test manažerům či test analytikům, ti je poté zpracovávají a interpretují jejich nadřízeným či zákazníkům. Vyhodnocování testů se provádí v modulu Dashboard, v záložkách Analysis View a Dashboard View.

4.7.1 Analysis View

Pro tvorbu grafů či reportu uživatel postupuje takto:

- (1) Vytvoří složku pro graf/report.
- (2) Stiskne ikonu a vybere volbu dle libosti.
- (3) Nadefinuje parametry pro graf/report.
- (4) Přidá graf/report do Analysis Tree.

Nejprve je nutné založit novou složku kliknutím na ikonu (1), která je stejná jako v předchozích případech. Novou složku může uživatel vytvořit do složky Private nebo Public. Jak z názvu vyplývá, ve složce Private budou reporty přístupné pouze pro určité uživatele, které definuje administrátor.

Po stisku ikony (2) je možné vybrat z několika možností, které jsou vidět na obrázku 27.



Obrázek 27 – Tvorba grafu, (zdroj: autor)

Pro ukázku v této práci jsem vybral první možnost Graph Wizard, tedy průvodce grafy, kdy je ve čtyřech krocích možné vygenerovat různé druhy grafů, které mohou být zaměřeny výhradně na pokrytí požadavků, spuštěné testy, dosud nespuštěné testy, defekty a jiné jak ukazuje obrázek 28. Pokud by si uživatel nebyl jistý, co která volba znamená, tak po kliknutí na volbu je níže popis.

Vzhledem k tomu, že testování aplikace probíhalo v krátkém časovém sledu, nemá smysl generovat nějaké trendové grafy apod. Je pak na uživateli, aby zvolil graf podle své potřeby. Pro rychlé vyhodnocení testování aplikace TTTournament úplně postačí dva základní grafy. Prvním je graf týkající se bugů, druhý pak výsledků spuštěných testů.



Obrázek 28 – Průvodce tvorby grafu, (zdroj: autor)

Zajímalo mě, jak bude vypadat graf v závislosti na prioritě bugů. Graf je vidět na obrázku 29.



Obrázek 29 – Graf výskytu bugů, (zdroj: autor)

Celkem jsem zaznamenal při testování pět bugů, z toho tři byly v testové složce Správa hráčů a dva ve složce Tisk.

Druhý graf, který je na obrázku 30 zobrazuje výsledky všech testů. Osa y znázorňuje, kolikrát byl daný test spuštěn, osa x ukazuje, který test to byl. První test se testoval zakládání turnaje. Modrá část označená jako <unknown> je tam proto, že jsem spustil test a řádně neukončil. Poté jsem ho spustil ještě dvakrát, oba s výsledkem Passed, tedy že prošly úspěšně.



Obrázek 30 – Výsledky testů, (zdroj: autor)

Z grafu je vidět, že čtyři testy mají status Failed. Neznamená to, že by testované části aplikace nefungovaly, ale v momentě, kdy je jeden krok v testu označen jako Failed, je pak celý test označen také jako Failed. Po vytvoření grafu přichází na řadu volba Add to Analysis Tree, kde je poté potřeba vybrat složku, do které se uloží.

4.7.2 Dashboard View

Tato záložka slouží pro tvorbu souhrnných reportů. Je potřeba vytvořit novou stránku a to buď do složky Private, Public nebo do nově vytvořené. Poté kliknout do Configuration (1) viz obrázek 31, vpravo v Graphs Tree a zde najít vytvořené grafy ze záložky Analysis View a přidat šipkou (2). Poté, co jsou grafy přidány, je možné vidět výsledek kliknutím na View. Souhrnný report je vidět na obrázku 32, nechybí zde ikona pro export do PDF (1).



Obrázek 31 - Tvorba reportu, (zdroj: autor)



Obrázek 32 – Souhrnný report, (zdroj: autor)

5 Závěr

Tato bakalářská práce se zabývala oblastí testování softwaru, především pak nástrojem HP Quality Center verze 12. Teoretická část práce pojednávala o základních pojmech týkajících se testování, metodikách vývoje softwaru z pohledu testování, defektech a druzích testovacích nástrojů. V praktické části byl stručně představen nástroj HP Quality Center, konkrétně jeho krátká historie a přehled nabízených edic. Poté je popsána instalace a konfigurace před prvním spuštěním, řízení manuálních testů, zaznamenávání bugů a vyhodnocování testování v tomto nástroji.

Hlavním cílem bylo stanoveno představení nástroje a vytvoření příručky pro řízení manuálních testů v tomto nástroji, která bude využitelná při výuce testování softwaru na Vysoké škole ekonomické v Praze či v Kompetenčním centru VŠE. Cílů bylo dosaženo v kapitolách 3 a 4.

Kromě hlavního cíle byl vymezen i cíl vedlejší. Tím bylo představit základní pojmy z oblasti testování. Postupně jsem uvedl definice testování softwaru, vysvětlil pojmy verifikace a validace, dále co je řízení testování, jaké jsou stěžejní dokumenty pro proces testování softwaru, co je řízení životního cyklu aplikace a základní typy testů. Dále jsem uvedl čtyři vybrané metodiky testování, vysvětlil pojem defekt a popsal základní rozdělení testovacích nástrojů. Tento cíl byl splněn kapitolou 2.

K dosáhnutí hlavního cíle jsem demonstroval testování v HPQC na aplikaci TTTournament. Tím jsem vytvořil příručku k řízení manuálních testů v HPQC 12, což je hlavním přínosem této práce a současně otestoval zmíněnou aplikaci. Po vyhodnocení testů jsem shledal všechny základní funkcionality aplikace jako výborně fungující, zaznamenané bugy, které budou předány vývojářce, nijak výrazně nesnižují kvalitu softwaru a po jejich doladění nic nebrání releasu.

Terminologický slovník

Termín	Zkratka	Význam [zdroj]		
Aplikace		Programové vybavení počítače [vlastní definice].		
Basic Input/Output System	BIOS	Základní vstupně-výstupní systém. Základ pro fun gování počítače, resp. základní desky [38].		
Brainstorming		Metoda volné spontánní diskuze na dané téma [39].		
Funkce		V programování druh podprogramu s návratovou hodnotou [40].		
Hardware		Hmotné vybavení počítače [vlastní definice].		
Iterace		Opakované volání funkce v aplikaci [vlastní defini- ce].		
Metodika		Obecně pracovní postup [vlastní definice].		
Metoda		Je blok kódu, který obsahuje sled příkazů [41].		
Microsoft	MS	Softwarová firma [vlastní definice].		
Open-source		Software s otevřeným zdrojovým kódem, uživatel může při dodržení jistých podmínek upravovat a prohlížet kód [vlastní definice].		
Software as a Service	Saas	Model nasazení softwaru, kdy dochází k hostování aplikace provozovatelem služby. Služba je nabízena zákazníkům přes Internet [42].		
Software		Nehmotné vybavení počítače (programy) [vlastní definice].		
Specifikace		Dokument upřesňující zadání [vlastní definice].		
Stakeholder		Člověk, který má zájem na určité záležitosti [vlastní definice].		
Trial verze		Časově omezená licence, tzv. "na vyzkoušení" [vlastní definice].		
Unified Modeling Language	UML	Grafický jazyk pro vizualizaci, specifikaci, navrho- vání a dokumentaci programových systémů [43].		
Zdrojový kód		Zápis textu programu v programovacím jazyce [vlastní definice].		

Seznam literatury

1. Patton, Ron. Testování softwaru. 1. Praha : Computer Press, 2002. 80-722-6636-5.

2. **Růžičková, Lucie.** Aplikace pro organizování turnajů ve stolním tenisu. *Bakalářská práce*. Praha : VŠE, 2015.

3. QUALITY CENTER ENTERPRISE TRIALS. *HP*. [Online] [Citace: 12. únor 2015.] http://www8.hp.com/cz/cs/software-solutions/quality-center-quality-management/try-now.html.

4. Tryzna, Jan. Diplomová práce. Testování softwaru. Praha : VŠE, 2007.

5. Kučera, Martin. Bakalářská práce. *Testování softwaru a metodika RUP*. Praha : VŠE, 2006.

6. **Pavlíček, Dalibor.** Popis procesů business testování a jejich optimalizace. *Bakalářská práce*. Praha : Unicorn College, 2012.

7. **Jandák, Tomáš.** Bakalářská práce. *Možnosti automatizovaného testování aplikací*. Plzeň : Západočeská univerzita, 2012.

8. **Pietrik, Michal.** Diplomová práce. *Automatizované testování webových aplikací*. Brno : Masarykova univerzita, 2012.

9. **Jandl, Jiří.** Bakalářská práce. *Application Lifecycle Management - porovnání produktů*. Praha : VŠE, 2012.

10. **Doubrava, Jan.** Diplomová práce. *Vyhodnocování SLA nad HP Quality Center*. Praha : VŠE, 2013.

11. **Roudenský, Petr a Havlíčková, Anna.** *Řízení kvality softwaru: průvodce testováním.* Brno : Computer Press, 2013. 978-80-251-3816-8.

12. **Borovcová, Anna.** Diplomová práce. *Testování webových aplikací*. Praha : Univerzita Karlova, 2008.

13. **Buchalcevová, Alena.** *Metodiky vývoje a údržby informačních systémů*. 1. Praha : Grada, 2005. 80-247-1075-7.

14. —. *Metodiky budování informačních systémů*. Praha : Oeconomica, 2009. 978-80-245-1540-3.

15. **DevelopSense.** DevelopSense. *Blog: Testing vs. Checking.* [Online] [Citace: 12. duben 2015.] http://www.developsense.com/blog/2009/08/testing-vs-checking/.

16. Abran, Alain a Moore, James W. SWEBOK, Guide to the Software Engineering Body of *Knowledge*. [Online] 2004. [Citace: 20. březen 2015.]

http://www.inf.ed.ac.uk/teaching/courses/seoc/2006_2007/resources/SWEBOK_Guide_2004. pdf. 0-7695-2330-7.

17. **Wikipedia.** Wikipedia. *Testovací plán*. [Online] [Citace: 15. duben 2015.] http://cs.wikipedia.org/wiki/Testovac%C3%AD_pl%C3%A1n.

18. **Page, Alan, Johnston, Ken a Rollison, Bj.** *Jak testuje software Microsoft*. 1. Brno : Computer Press, 2009. 978-80-251-2869-5.

19. **Chappell, David.** *What is Application lifecycle managemen?* [Online] [Citace: 30. duben 2015.] http://www.davidchappell.com/writing/white_papers/What_is_ALM_v2.0--Chappell.pdf.

20. *Hodnocení metodik vývoje informačních systémů z pohledu testování*. **Buchalcevová**, **Alena a Kučera, Jan.** Praha : Česká společnost pro systémovou integraci, 2008, měsíčník ČSSI. 1210-9479.

21. **IBM.** Rational Unified Process - Best Practices for Software Development Teams. [Online] listopad 2001. [Citace: 11. duben 2015.] https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpr actices_TP026B.pdf.

22. **Vomáčko, Vít.** kurz Základy testování SW aplikací 4IT354. *powerpoint prezentace na 7. cvičení*. Praha : VŠE, 2014.

23. **SW Testování.** Metodika RUP a testování. *SW Testování*. [Online] [Citace: 10. duben 2015.]

http://www.swtestovani.cz/index.php?option=com_content&view=article&id=43:metodika-rup-a-testovani&catid=3:zaklady&Itemid=11.

24. **Eclipse.** *Introduction to OpenUP (Open Unified Process).* [Online] [Citace: 30. duben 2015.] http://www.eclipse.org/epf/general/OpenUP.pdf.

25. Rejnková, Petra. MMSP. [Online] [Citace: 30. duben 2015.] http://mmsp.czweb.org/.

26. Beck, K. Extrémní programování. Praha : Grada, 2002. 80-247-0300-9.

27. **Jeffries, R.** XProgramming. *What is Extreme Programming?* [Online] [Citace: 29. duben 2015.] http://ronjeffries.com/xprog/what-is-extreme-programming/.

28. RTH Blog. *About*. [Online] [Citace: 13. duben 2015.] https://requirementsandtestinghub.wordpress.com/about/.

29. **Testování, softwaru.** Testování softwaru. *TestLink – správa a organizace testování softwaru*. [Online] [Citace: 13. duben 2015.] http://testovanisoftwaru.cz/nastroje/testlink/.

30. **Wikipedia**. Wikipedia. *Selenium (software)*. [Online] [Citace: 13. duben 2015.] http://cs.wikipedia.org/wiki/Selenium_%28software%29.

31. Verisium. Verisium. vPerformer - Performance Testing Tool. [Online] [Citace: 13. duben

2015.] http://www.verisium.com/products/vPerformer/index.html.

32. **Foundation, Apache Software.** The Apache Software Foundation. [Online] [Citace: 13. duben 2015.] http://jmeter.apache.org/.

33. **Wikipedia**. Wikipedia. *Mercury Interactive*. [Online] [Citace: 14. duben 2015.] http://en.wikipedia.org/wiki/Mercury_Interactive.

34. **Tutorialspoint.** Tutorialspoint. *Quality Center - Introduction*. [Online] [Citace: 14. duben 2015.] http://www.tutorialspoint.com/qc/qc_overview.htm.

35. **Wikipedia**. Wikipedia. *HP QuickTest Professional*. [Online] [Citace: 14. duben 2015.] http://en.wikipedia.org/wiki/HP_QuickTest_Professional.

36. **HP.** ALM & QC Enterprise Technical Specifications. *HP Live Network*. [Online] [Citace: 20. duben 2015.] https://hpln.hp.com/page/alm-qc-enterprise-technical-specifications#ALM1200.

37. **stahuj.cz.** VMware Player - virtualizace v pěti krocích. *stahuj.cz magazín*. [Online] [Citace: 20. duben 2015.] http://magazin.stahuj.centrum.cz/vmware-player-virtualizace-v-peti-krocich/.

38. **Svět Hardware.** svethardware.cz. *Slovník*. [Online] http://www.svethardware.cz/slovnik/b.

39. **ABZ.cz.** Brainstorming. *Slovník cizích slov*. [Online] [Citace: 22. duben 2015.] http://slovnik-cizich-slov.abz.cz/web.php/slovo/brainstorming-1.

40. **Wikipedia**. Wikipedia. *Funkce*. [Online] [Citace: 23. duben 2015.] http://cs.wikipedia.org/wiki/Funkce.

41. **Microsoft.** Microsoft developer network. *Metody (Průvodce programováním v C#)*. [Online] [Citace: 26. duben 2015.] https://msdn.microsoft.com/cs-cz/library/ms173114.aspx.

42. **Wikipedia**. Wikipedia. *Software as a Service*. [Online] [Citace: 22. duben 2015.] http://cs.wikipedia.org/wiki/Software_as_a_service.

43. —. Wikipedia. *UML*. [Online] [Citace: 22. duben 2015.] http://cs.wikipedia.org/wiki/Unified_Modeling_Language.

Seznam obrázků a tabulek

Obrázky

Obrázek 1 – Testovací skript pro přihlášení do aplikace, (zdroj: autor)	6
Obrázek 2 – Praktiky XP, (zdroj: [27])	12
Obrázek 3 – Příklad reportování bugu, (zdroj: autor)	15
Obrázek 4 – Instalace QC, 3. krok, (zdroj: autor)	20
Obrázek 5 – Nastavování zabezpečení, 5. krok, (zdroj: autor)	21
Obrázek 6 – Nastavování přihlašovacích údajů, 6. krok, (zdroj: autor)	21
Obrázek 7 – Nastavení údajů do HP ALM Service, 9. krok, (zdroj: autor)	22
Obrázek 8 – Vytvoření domény, projektu, přiřazení uživatelů k projektu, (zdroj: autor)	23
Obrázek 9 – Přiřazení uživatele k projektu, (zdroj: autor)	23
Obrázek 10 – Přihlášení do HPQC12, (zdroj: autor)	24
Obrázek 11 - Hlavní okno aplikace [2], (zdroj: autor)	25
Obrázek 12 – Tvorba releasu, (zdroj: autor)	27
Obrázek 13 – Tvorba složek požadavků, (zdroj: autor)	28
Obrázek 14 – Tvorba požadavků, (zdroj: autor)	29
Obrázek 15 – Tvorba testovacích složek, (zdroj: autor)	30
Obrázek 16 – Tvorba testu, (zdroj: autor)	31
Obrázek 17 – Testovací skript a jeho kroky, (zdroj: autor)	31
Obrázek 18 – Přiřazování testu k požadavku, (zdroj: autor)	32
Obrázek 19 – Přiřazené testy k požadavkům, (zdroj: autor)	32
Obrázek 20 – Tvorba testovací sady, (zdroj: autor)	33
Obrázek 21 – Přiřazení testů do testovací sady, (zdroj: autor)	34
Obrázek 22 – Spouštění testu, (zdroj: autor)	34
Obrázek 23 – Spuštěný test, (zdroj: autor)	35
Obrázek 24 – Přehled spuštěných testů, (zdroj: autor)	35
Obrázek 25 – Zaznamenávání bugu během testování, (zdroj: autor)	36
Obrázek 26 – Přehled zaznamenaných bugů, (zdroj: autor)	37
Obrázek 27 – Tvorba grafu, (zdroj: autor)	39
Obrázek 28 – Průvodce tvorby grafu, (zdroj: autor)	39
Obrázek 29 – Graf výskytu bugů, (zdroj: autor)	40
Obrázek 30 – Výsledky testů, (zdroj: autor)	40
Obrázek 31 - Tvorba reportu, (zdroj: autor)	41
Obrázek 32 – Souhrnný report, (zdroj: autor)	41

Tabulky

Tabulka 1 – Pravidla Manifestu a principy OpenUP (zdroj: [24])	10
Tabulka 2 – Výčet verzí (zdroj: [34] [35])	17
Tabulka 3 - Přehled funkcí jednotlivých edic (zdroj: [34])	18

Příloha 1

Uživatelská příručka k aplikaci TTTournament [2]

Uživatelská příručka je určená pro koncové uživatele, tedy samotné rozhodčí, kteří budou turnaj řídit. Na ukázkovém turnaji předvedu všechny možnosti, kterých může uživatel v průběhu řízení turnaje využít. Součástí jsou rovněž ukázky oken výsledné aplikace.

Obecné ovládací prvky

Aplikace se skládá z jednoho hlavního okna, ve kterém se podle přání uživatele zobrazují data, a několika pomocných oken, například pro přidání nového turnaje nebo hráče. V ukázce na obrázku č.1 je v horní části umístěno hlavní menu, které umožňuje vytvořit nový turnaj nebo nový seznam. Tlačítko Tisk v hlavním menu umožní hromadný tisk všech sestav v aktuálním turnaji. V pravém horním rohu jsou zobrazeny informace o turnaji ve formátu název turnaje, místo konání turnaje a datum. V prostřední části obrazovky se nachází veškerý uživatelský obsah. Na zmíněném obrázku je ukázán seznam účastníků. Všechna tlačítka poskytující uživateli nějakou funkci, kterou je možno s obsahem provádět, se nacházejí v horní části obrazovky a liší se podle zobrazeného obsahu. Dalším společným prvkem je zobrazení záložek v dolní části obrazovky. Kliknutím na záložku si uživatel zvolí, jaký obsah chce zobrazit.

			TTTou	urnament		- 🗆 🗙
Nový	Tisk					
N	ový turnaj	[Pr	aque GrandPrix, Praha 19.4. 2015
N	ový seznam 🕨	Ze souboru	Přidat hráče Vybrat vš	e Tisk	Vylosovat	-
		Prázdný list	Vyberte	přítomné hráče.		
	Nasa	zení* Jméno*	Příjmení*	Rok narození	Tým*	
	1	Vacenovská	lveta	1993	Linz-Froschberg AG	Odebrat
	2	Matelová	Hana	1992	TTG Bingen/Münster-Sarmshe	Odebrat
	3	Štrbíková	Renáta	1996	KST Tarnobrzeg	Odebrat
	4	Pěnkavová	Kateřina	1993	SKST Mart Hodonín	Odebrat
	5	Čechová	Dana	1992	TTG Bingen/Münster-Sarmshe	Odebrat
	6	Harabaszová	Lenka	1993	SKST Mart Hodonín	Odebrat
	7	Bartošová	Hana	1992	SK Frýdlant n.O.	Odebrat
	8	Adámková	Karin	1996	SK Frýdlant n.O.	Odebrat
	9	Tomanová	Tamara	1993	TTC Rödinghausen	Odebrat
	10	Ságlová	Miroslava	1992	Wolframs-Eschenbach	Odebrat
	11	Kučerová	Aneta	1993	MSK ČP Břeclav Gumotex	Odebrat
	12	Rlašková	Dagmar	1992	SK En/dlant n O	C Oddated Y
Seznai	m dorostenky					

Obrázek 1 – Hlavní okno

Spuštění aplikace

Pro spuštění aplikace není potřeba žádná instalace. Uživatel obdrží složku se všemi potřebnými soubory. Samotná aplikace se spouští pomocí .exe souboru. Po spuštění aplikace se uživateli otevře okno, ve kterém je dotázán, zda chce vytvořit nový turnaj. Pokud ano, veškerá předchozí data se z databáze vymaží. Pokud ne, pokusí se aplikace načíst předchozí turnaj, jestliže v databázi již nějaký je.

V případě vytváření nového turnaje se uživateli zobrazí okno, kde vyplní základní informace o turnaji. Po potvrzení se uživatel ocitne v hlavním okně aplikace a v hlavním menu je mu zpřístupněna možnost vytvořit nový seznam účastníků. Ukázka okna pro nový turnaj je na obrázku č.2.

	Přidat turnaj 🗧 –		×
	Nazev turnaje*]	
	Místo konání*	í I	
	* Takto označená pole isou povinná	J	
-			

Obrázek 2 - Nový turnaj

Vytvoření nového seznamu

Po vytvoření nebo načtení turnaje se uživateli uvolní možnost vytvořit nový seznam. Je mu dána možnost načíst seznam z .xlsx souboru nebo vytvořit nový prázdný seznam. Okno pro vyplnění požadovaných informací je na obrázku č. 3. Pro prázdný seznam je zobrazeno pouze pole Název seznamu.



Obrázek 3 - Nový seznam

Při možnosti načtení ze souboru je potřeba vložit .xlsx soubor ve správném formátu, který je zobrazen na obrázku č. 4. Na názvech sloupců nezáleží, důležité je jejich pořadí. Po potvrzení dojde k načtení hráčů do aplikace, jejich uložení do databáze a přidání záložky na dolní lištu.

	A	В	С	D
1	Pořadí	Jméno	Oddíl	Rok narození
2	1	Vacenovská Iveta	Linz-Froschberg AG	1993
3	2	Matelová Hana	TTG Bingen/Münster-S	1992
4	3	Štrbíková Renáta	KST Tarnobrzeg	1996
5	4	Pěnkavová Kateřin	SKST Mart Hodonín	1993
6	5	Čechová Dana	TTG Bingen/Münster-S	1992
7	6	Harabaszová Lenka	SKST Mart Hodonín	1993
8	7	Bartošová Hana	SK Frýdlant n.O.	1992
9	8	Adámková Karin	SK Frýdlant n.O.	1996
10	9	Tomanová Tamara	TTC Rödinghausen	1993
11	10	Ságlová Miroslava	Wolframs-Eschenbach	1992
12	11	Kučerová Aneta	MSK ČP Břeclav Gumot	1993
13	12	Blašková Dagmar	SK Frýdlant n.O.	1992
14	13	Tomanovská Kateř	SK Frýdlant n.O.	1996
15	14	Mikulcová Kristýna	SKST Mart Hodonín	1993
16	15	Vachovcová Alena	SKST Mart Hodonín	1992

Obrázek 4 - Vzorový soubor hráčů

Editace seznamu hráčů

			TTTouri	nament		
lový Tisk						
					Pr	ague GrandPrix, Praha 19.4.
		Uložit Při	dat hráče Vybrat vše	Tisk	Vylosovat	
			Vyberte při	tomné hráče.		
	Nasazení*	Jméno*	Příjmení*	Rok narození	Tým*	
	1	Vacenovská	lveta	1993	Linz-Froschberg AG	Odebrat
	2	Matelová	Hana	1992	TTG Bingen/Münster-Sarmshe	Odebrat
	3	Štrbíková	Renáta	1996	KST Tarnobrzeg	Odebrat
	4	Pěnkavová	Kateřina	1993	SKST Mart Hodonín	Odebrat
	5	Čechová	Dana	1992	TTG Bingen/Münster-Sarmshe	Odebrat
	6	Harabaszová	Lenka	1993	SKST Mart Hodonín	Odebrat
	7	Bartošová	Hana	1992	SK Frýdlant n.O.	Odebrat
	8	Adámková	Karin	1996	SK Frýdlant n.O.	Odebrat
	9	Tomanová	Tamara	1993	TTC Rödinghausen	Odebrat
	10	Ságlová	Miroslava	1992	Wolframs-Eschenbach	Odebrat
	11	Kučerová	Aneta	1993	MSK ČP Břeclav Gumotex	Odebrat
	12	Rlačková	Dagmar	1992	SK Frédlant n O	D datast

Obrazovka se seznamem hráčů je zobrazena na obrázku č. 5.

Obrázek 5 - Editace seznamu hráčů

Uživatel může libovolně upravovat hodnoty v textboxech. Pro uložení změn do databáze slouží tlačítko Uložit. Všechna pole označená hvězdičkou jsou povinná a kontrolovaná před uložením. Rovněž se kontroluje, zda sloupce Nasazení a Rok jsou číselné hodnoty.

Tlačítko Přidat hráče otevře okno zobrazené na obrázku č. 6. V něm uživatel vyplní všechny požadované hodnoty, které jsou kontrolovány stejným způsobem, jako v hlavním okně. Pokud jsou v seznamu hráči s číslem nasazení 1,2 a 3 a uživatel vloží hráče s číslem nasazení 2, tak se všem hráčům s rovným nebo vyšším číslem nasazení toto číslo o 1 zvětší. Nový hráč tedy bude v seznamu vložen na pozici 2.

 Přidat hráče	-	×
Nasazeni*		
2		
Jméno*		
Klára		
Příjmení*		
Vymyšlená		
Oddíl*		
Slavoj Houslice		
Rok narození		
1996		
* Takto označená pole jsou povinná		
Uložit		
	_	

Obrázek 6 - Přidat hráče

Odebrání hráče ze seznamu je možné stiskem tlačítka Odebrat, které se nachází vedle každého záznamu.

Tlačítko Tisk umožní uživateli vytisknout zobrazený seznam.

Tlačítko Vylosovat vezme všechny hráče, kteří mají zaškrtnutý checkbox, a nabídne uživateli okno s nastavením losování, které podrobněji popíši dále.

Pro zjednodušení se v horní části nachází tlačítko Vybrat vše, které zaškrtne všechny hráče v seznamu.

Po vylosování již není možné hráče přidávat ani odebírat a rovněž není možné provést nové losování, příslušná tlačítka jsou deaktivována. Hráči, kteří nebyli zařazeni do losování, jsou ze seznamu a z databáze vymazáni.

Možnosti losování

Po stisknutí tlačítka Vylosovat je uživateli zobrazeno okno s nastavením losování, které je na obrázku č. 7. V horní části okna je povinné pole pro vyplnění názvu, který se po vylosování zobrazí na záložce. Následuje volba mezi vylosováním skupin a ko- systémem, tedy pavoukem.

	Nastavení losování 🛛 🗖 🗙
Název*	
	O Skupiny
	O Pavouk
	* Takto označená pole jsou povinná
	Pokračovat

Obrázek 7 - Nastavení losování

Při vybrání ko-systému a stisknutí tlačítka Pokračovat se provede losování, uloží se do databáze a do spodní části se přidá nová záložka. Podmínkou vylosování je, že počet zvolených hráčů musí být více než jedna a méně než 129.

Druhou možností je vylosování tříčlenných nebo čtyřčlenných skupin. Při zvolení této varianty je uživateli zobrazeno okno se všemi kombinacemi, které je možné ze zadaného počtu účastníků vylosovat. Toto losování je omezeno podmínkou, že zvolený počet účastníků musí být větší než pět a menší než 257. Uživatel si může zvolit automatické přiřazení rozhodčích k jednotlivým zápasům zaškrtnutím pole Určit rozhodčí. Následující obrázek č. 8 ukazuje okno s kombinacemi skupin.

	Nastavení skupin	- 🗆 🗙
Počet čtyi	íčlenných skupin x počet tříčlen 2 x 9 5 x 5 8 x 1	ných skupin
	Určit rozhodčí	Pokračovat

Obrázek 8 - Nastavení skupin

Obě možnosti losování implementují pravidla popsaná v Soutěžním řádu stolního tenisu.

Obrazovka skupin

Obrazovka s načtenými skupinami se nachází na obrázku č. 9.

louní Tisk									
NOVY TSK								Praque Gra	ndPrix, Praha 19.
Hlavní soutěž Vylosovat 1 Vacenovská lveta	Skupina 1	1	2	3	Body	Pořadi	11 Kučerová Aneta	8 Adámková Karin	8;9;5
11 Kučerová Aneta	1 Vacenovská lveta Linz-Froschberg AG	×	3:0	3:0	2	1	Koznodci: 1 Vacenovská lveta	1 Vacenovska iveta 11 Kučerová Aneta	8;6;4
	11 Kučerová Aneta MSK ČP Břeclav Gumotex	0:3	x	3:0	1	2	Rozhodčí: 1 Vacenovská lveta	8 Adámková Karin 8 Adámková Karin	3;9;5
	8 Adámková Karin SK Frjálant n.O.	0:3	0:3	×	0	3	Rozhodčí:	11 Kučerová Aneta	
	Skupina 2	1	2	3	Body	Pořadi	15 Vachovcová Alena	7 Bartošová Hana	
	2 Matelová Hana TTG Bingen/Münster-Sarmsheim	x			0	0	Rozhodčí: 2 Matelová Hana	2 Matelová Hana 15 Vachovcová Alena	
	15 Vachowcowić Alena		x		0	0	Rozhodčí: 2 Matelová Hana	7 Bartošová Hana 7 Bartošová Hana	
	SKST Mart Hodonin								

Obrázek 9 - Zobrazení skupin

V pravém horním rohu má uživatel tlačítko Tisk. Po jeho stisknutí se zobrazí okno s možnostmi tisku, které je na obrázku č. 10. Uživatel si může vybrat, zda chce vytisknout skupiny nebo zápisy k jednotlivým zápasům ve skupinách.

🔳 Vytisknout 🗕 🗖 🗙
Skupiny
✓ Zápisy
✓ Skupina 1
✓ Skupina 2
Skupina 3
Skupina 4
Vytvorit PDF Vytisknout

Obrázek 10 - Volba tisku

Hlavní částí obrazovky jsou skupiny, do kterých se průběžně ukládají hodnoty podle výsledků zápasů. Hráčům se automaticky dopočítávají body a konečné umístění.

Po pravé straně vedle každé skupiny se nachází seznam zápasů v takovém pořadí, v jakém mají být odehrány. Pod každým zápasem je napsán rozhodčí, pokud uživatel zvolil jejich automatické přiřazení.

Ke každému zápasu náleží textbox, do kterého se vyplňují výsledky ve formátu, který uvedu na příkladu z obrázku č. 11.

1 Vacenovská lveta	15 Vachovcová Alena	8;-11;7;6

Obrázek 11 - Příklad zadání výsledků

V obrázku je na první pozici Vacenovská a Vachovcová na druhé pozici. Výsledky se vždy zadávají z pohledu hráče na první pozici. Pokud tento hráč set vyhraje, zapíše se kladná hodnota, pokud prohraje, tak záporná. Na obrázku je zadaný výsledek zápasu 8;-11;7;6. To znamená, že Vacenovská zvítězila v poměru setů 3:1. První set vyhrála 11:8, druhý prohrála 11:13, třetí a čtvrtý vyhrála 11:7 a 11:6. Oddělení jednotlivých setů středníkem je povinné.

V levé části obrazovky se po zadání všech výsledků ve skupině automaticky doplňují hráči postupující do hlavní soutěže a soutěže útěchy. Z těchto seznamů je pak možné vylosovat další fázi turnaje. Možnost vylosování je podmíněna vyplněním všech výsledků ve všech skupinách. Podobně jako u seznamu hráčů není již možné po vylosování provést los znovu. Rovněž jsou deaktivovány všechny textboxy s výsledky, aby je již nebylo možné upravovat. Možnosti losování jsem podrobně popsala v kapitole 7.5.

Hráči jsou do dalších fází rozdělováni na základě umístění ve skupině. První dva postupují do hlavní soutěže, zbylí hrají soutěž útěchy. Umístění ve skupině se počítá na základě získaných bodů. Za každé vítězství obdrží hráč bod. V případě rovnosti bodů u dvou hráčů rozhoduje o konečném pořadí vzájemný zápas, jeho vítěz skončí s lepším umístěním. V případě rovnosti bodů u tří hráčů rozhoduje poměr součtu vyhraných a prohraných setů ve vzájemných zápasech. Postup předvedu na následujícím příkladu.

Hráč A porazil hráče B 3:0, hráč B porazil hráče C 3:1 a hráč C porazil hráče A 3:2. Poměry setů jsou následující:

- Hráč A 5:3
- Hráč B 3:4
- Hráč C 4:5

Nejlepší poměr setů má v tomto případě hráč A (5/3=1,67), dále hráč C (4/5=0,8) a jako poslední hráč B (3/4=0,75). Pokud dojde k rovnosti v poměru setů, postupuje se obdobným způsobem s míčky získanými v jednotlivých setech.

Obrazovka pavouka

Poslední možnou obrazovkou je zobrazení pavouka, která je na obrázku č. 12.



Obrázek 12 - Vyřazovací fáze turnaje

V pravém horním rohu se opět nachází možnost tisku. Uživatel si po otevření dialogového okna může zvolit, zda vytisknout pavouka nebo zápisy k zápasům, obdobně jako u skupin.

V hlavní části obrazovky je vykreslen samotný pavouk. Vítěze zápasu uživatel určuje dvojklikem na jméno hráče, který je automaticky posunut do dalšího kola. Není možné do dalšího kola posunout hráče, který ještě nemá soupeře. V prvním kole, pokud jsou v pavouku volná místa, označuje se volné kolo jako "bye".

Příloha 2

Testovací skripty k aplikaci TTTournament

TC01 Založení turnaje

Otevře se po spuštění aplikace okno, kde si uživatel může zvolit, zda vytvořit nový turnaj nebo načíst poslední použitý. Pokud uživatel zvolí nový turnaj, otevře se mu dialogové okno pro vyplnění základních informací o turnaji, například název a místo konání turnaje. V opačném případě se načte předchozí turnaj [2].

Step	Popis stepu	Očekávaný výsledek	Skutečný výsledek
1	Spusť aplikaci	Aplikace se spustí, nabídne	
		založení nového turnaje.	
2	Stiskni tlačítko NE.	Aplikace načte předchozí tur-	
		naj z databáze.	
3	Ukonči a znovu spusť	Aplikace se spustí, nabídne	
	aplikaci.	založení nového turnaje.	
4	Stiskni tlačítko ANO.	Zobrazí se upozornění, že	
		dojde k odstranění předcho-	
		zích dat.	
5	Stiskni NE.	Okno se zavře, aplikace je	
		stále spuštěná.	
6	Zopakuj stepy 3+4 a	Otevře se okno pro zadání	
	stiskni ANO.	názvu a místa turnaje.	
7	Zadej nekorektní údaje	Upozornění na zadání neko-	
	(čísla) a ULOŽIT.	rektních údajů.	
8	Nevyplň ani jedno povin-	Výzva k vyplnění všech polí.	
	né pole.		
9	Vyplň pouze jedno pole.	Výzva k vyplnění všech polí.	
10	Zadej korektní údaje a	Turnaj vytvořen.	
	ULOŽIT.		
11	V horním menu stiskni	Aplikace nabídne založení	
	Nový a Nový turnaj.	nového turnaje.	
12	Stiskni ANO.	Zobrazí se upozornění, že	
		dojde k odstranění předcho-	
		zích dat.	
13	Stiskni ANO.	Prázdný formulář pro zadání	
		údajů o novém turnaji.	

Tabulka 1 - Testovací skripty k požadavku Založení turnaje, (zdroj: autor)

TC02 Vytvoření seznamu hráčů

Uživatel má v hlavním menu možnost vytvořit nový seznam načtením ze souboru nebo vytvořením

prázdného seznamu. Načtená obrazovka se bude skládat z:

- Textového pole pro název seznamu
- Textového pole pro cestu k souboru při možnosti načíst ze souboru
- Tlačítka Vybrat pro nalezení souboru v počítači při možnosti načíst ze souboru
- Tlačítka Potvrdit pro vytvoření seznamu

Vstupní soubor bude ve formátu .xls nebo .xlsx. Po potvrzení se uživateli v hlavním okně zobrazí záložka s názvem seznamu, kterým bude možné jej načíst [2].

Step	Popis stepu	Očekávaný výsle-	Skutečný výsledek
		dek	
1	Spusť aplikaci a založ nový turnaj.	Aplikace spuštěna,	
		turnaj založen.	
2	V horním menu stiskni Nový a	Zobrazení okna pro	
	Nový seznam a zvol Ze souboru.	zadání názvu sezna-	
		mu a vyhledání sou-	
		boru.	
3	Zadej název a stiskni NAČÍST.	Chybová hláška,	
		nelze najít soubor.	
4	Zopakuj step 2, stiskni VYBRAT a	Chybová hláška,	
	vyber excel soubor (dodaný vývojá-	vyplnit název.	
	řem) a stiskni NAČÍST.		
5	Zopakuj step 2, stiskni VYBRAT a	Chybová hláška,	
	vyber libovolný excel a stiskni	chybné hodnoty.	
	NAČÍST.		
6	Zopakuj step 2, zadej název a a	Vytvořen seznam	
	vyber excel soubor (dodaný vývojá-	v levém dolním rohu.	
	řem) a stiskni NAČÍST.		
7	Klikni na seznam.	Seznam vyobrazen	
		v hlavním oknu.	
8	V horním menu stiskni Nový a	Zobrazení okna pro	
	Nový seznam a zvol Prázdný list.	zadání názvu sezna-	
		mu.	
9	Stiskni načíst.	Chybová hláška,	
		zadej název.	
10	Zadej název, stiskni Načíst.	Vytvořen seznam	
		v levém dolním rohu.	

Tabulka 2 - Testovací skripty k požadavku Vytvoření seznamu, (zdroj: autor)

TC03 Přidání hráče

Hráče je možné do seznamu přidávat také ručně. Na obrazovce se seznamem uživatel stiskne tlačítko Přidat hráče. Zobrazené okno bude obsahovat následující textová pole:

• Umístění v žebříčku

58

- Jméno
- Příjmení
- Oddíl
- Datum narození [2]

Tabulka 3 - Testovací skripty k požadavku Přidání hráče, (zdroj: autor)

Step	Popis stepu	Očekávaný výsle-	Skutečný výsledek
		dek	
1	Spusť aplikaci a otevři/založ turnaj a	Spuštěná aplikace,	
	otevři/založ seznam.	turnaj i seznam zob-	
		razen v hlavním	
		okně.	
2	Stiskni tlačítko Přidat hráče.	Zobrazeno okno pro	
		zadání údajů.	
3	Zadej do všech kolonek textový řetě-	Chybová hláška,	
	zec a stiskni ULOŽIT.	v polích pro číselný	
		řetězec se nachází	
		textový.	
4	Nech některá povinná pole prázdná a	Chybová hláška,	
	stiskni ULOŽIT.	nutnost vyplnit	
		všechna povinná	
		pole.	
5	Zadej rok narození v nekorektním	Chybová hláška,	
	tvaru (více než čtyřmístné).	max. čtyřmístné.	
6	Přidej hráče s číslem nasazení, které	Hráč přidán se zada-	
	už má v seznamu jiný hráč.	ným nasazením;	
		hráč, který měl pů-	
		vodně číslo nasazení	
		má původní číslo+1 .	

TC04 Editace hráče

Hráče již přidané do seznamu je možné také editovat přímo v seznamu. Jednoduše přepíše zobrazené hodnoty [2].

Step	Popis stepu	Očekávaný výsle-	Skutečný výsledek
		dek	
1	Spusť aplikaci a otevři/založ turnaj a	Spuštěná aplikace,	
	otevři/založ seznam.	turnaj i seznam zob-	
		razen v hlavním	
		okně.	
2	Změň jméno u libovolného hráče.	Jméno úspěšně změ-	
		něno.	
3	Změň rok na textový řetězec.	Chybová hláška,	

Tabulka 4 - Testovací skripty k požadavku Editace hráče, (zdroj: autor)

		nelze použít textový
		řetězec.
4	Změň číslo nasazení u libovolného	Chybová hláška, že
	hráče na číslo, které má jiný hráč.	číslo už je obsaze-
		né/číslo změněno,
		seznam zaktualizo-
		vaný.
5	Označ libovolného hráče a stiskni	Hráč odebrán, se-
	ODEBRAT.	znam zaktualizován.

TC05 Losování

Při stisknutí tlačítka Vylosovat se uživateli nově otevřeném okně zobrazí následující nastavení:

- TextBox pro název, který se následně zobrazí na záložce
- Volbu mezi vylosováním skupin nebo ko systémem

V případě zvolení možnosti skupiny se uživateli zobrazí okno, ve kterém bude:

- Checkbox pro automatické přiřazení rozhodčích
- Radiobuttony s možnostmi pro počet tříčlenných a čtyřčlenných skupin, ze kterých si uživatel zvolí

Proces implementuje všechna pravidla, která jsou popsána v soutěžním řádu stolního tenisu (citace). Po stisknutí tlačítka Pokračovat proběhne vylosování a přidání tlačítka na hlavní panel, kterým bude možné skupiny zobrazit [2].

Step	Popis stepu	Očekávaný výsle-	Skutečný výsledek
		dek	
1	Spusť aplikaci a otevři/založ turnaj	Spuštěná aplikace,	
	a otevři/založ seznam.	turnaj i seznam zob-	
		razen v hlavním	
		okně.	
2	Stiskni tlačítko Vylosovat.	Chybová hláška,	
		nutnost zvolit hráče	
		pro losování.	
3	Vyber libovolný počet hráčů, stisk-	Objeví se okno pro	
	ni Vylosovat.	zadání Názvu a vý-	
		běru z možností	
		Skupiny či Pavouku.	
4	Stiskni Pokračovat.	Chybová hláška,	
		nutnost vyplnit ná-	
		zev.	
5	Vyplň název a stiskni Pokračovat.	Chybová hláška,	
		nutnost vyplnit mož-	

Tabulka 5 -	Testovací skripty	k požadavku	Losování,	(zdroj: autor)
-------------	-------------------	-------------	-----------	----------------

		nost.	
6	Vyplň název a libovolnou možnost	Objeví se okno na-	
	a stiskni Pokračovat.	stavení.	
7	Stiskni Pokračovat.	Chybová hláška,	
		nutnost vyplnit mož-	
		nost.	
8	Zvol možnost a stiskni Pokračovat.	Seznam úspěšně	
		vylosován.	

TC06 Tisk sestav

Uživateli bude poskytnuta možnost vytisknout skupiny, pavouky nebo zápisy k zápasům [2].

Step	Popis stepu	Očekávaný výsle-	Skutečný výsledek
		dek	
1	Spusť aplikaci a otevři/založ turnaj	Spuštěná aplikace,	
	a otevři/založ seznam.	turnaj i seznam zob-	
		razen v hlavním	
		okně.	
2	Vyber libovolný počet hráčů, stisk-	Objeví se okno pro	
	ni Vylosovat.	zadání Názvu a vý-	
		běru z možností	
		Skupiny či Pavouku.	
3	Vyplň název a libovolnou možnost	Objeví se okno na-	
	a stiskni Pokračovat.	stavení.	
4	Zvol možnost a stiskni Pokračovat.	Seznam úspěšně	
		vylosován.	
5	Stiskni Tisk.	Zobrazí se okno	
		s možností vytisk-	
		nout nebo vytvořit	
		PDF.	
6	Odzaškrtni checkbox a stiskni Vy-	Chybová hláška,	
	tvořit PDF.	nebyl vybrán seznam	

Tabulka 6 - Testovací skripty k požadavku Tisk sestav, (zdroj: autor)

	Zaškrtni checkbox a stiskni Vytvo- řit PDF.	PDF vytvořeno a uloženo.	
7			
	Zaškrtni checkbox a stiskni Vy-	Objeví se okno pro	
	tisknout.	tisk.	
8			

TC07 Zadávání výsledků

V průběhu turnaje má uživatel možnost zadávat výsledky do připravených kolonek u skupin. Aplikace předpokládá zápasy pouze na tři vítězné sety. Tedy výsledek 3:0, 3:1 nebo 3:2. Na základě těchto výsledků pak probíhá automatické doplňování vítězů. V případě K.O. systému se vítěz určuje dvojklikem na jméno hráče [2].

Step	Popis stepu	Očekávaný výsle-	Skutečný výsledek
		dek	
1	Spusť aplikaci a otevři/založ turnaj	Spuštěná aplikace,	
	a otevři/založ seznam.	turnaj i seznam zob-	
		razen v hlavním	
		okně.	
2	Vyber libovolný počet hráčů, stisk-	Objeví se okno pro	
	ni Vylosovat.	zadání Názvu a vý-	
		běru z možností	
		1	1

Tabulka 7 - Testovací skripty k pož	adavku Zadávání výsledků	, (zdroj: autor)
-------------------------------------	--------------------------	------------------

		razen v hlavním	
		okně.	
2	Vyber libovolný počet hráčů, stisk-	Objeví se okno pro	
	ni Vylosovat.	zadání Názvu a vý-	
		běru z možností	
		Skupiny či Pavouku.	
3	Vyplň název a libovolnou možnost	Objeví se okno na-	
	a stiskni Pokračovat.	stavení.	
4	Zvol možnost a stiskni Pokračovat.	Seznam úspěšně	
		vylosován	
5	Zadej do libovolného zápasu texto-	Chybová hláška,	
	vý řetězec.	nekorektní skóre.	
6	Zadej do libovolného zápasu skóre	Chybová hláška,	
	4:0.	nekorektní skóre.	
7	Zadej do libovolného zápasu skóre	Chybová hláška,	
	3:3.	nekorektní skóre.	
8	Zadej do libovolného zápasu skóre	Chybová hláška,	
	ve formátu 3-1.	nekorektní skóre.	
9	Zadej do libovolného zápasu skóre	Skóre zadáno úspěš-	
	ve formátu 3;8;9.	ně, sety zaznamená-	
		ny v tabulce.	