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

> Studijní program: Aplikovaná informatika Obor: Informační systémy a technologie

Automatizované testování s využitím nástroje HP QuickTest Professional

DIPLOMOVÁ PRÁCE

Diplomant	: Bc. Michal Malý
Vedoucí	: doc. Ing. Alena Buchalcevová, Ph.D.
Oponent	: Ing. et Ing. Michal Doležel

Prohlášení

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

V Praze 5.5.2014

.....

Michal Malý

Poděkování

Na tomto místě bych rád poděkoval paní doc. Ing. Aleně Buchalcevové, Ph.D. za trpělivé vedení a cenné rady, které mi v průběhu psaní této práce poskytla. Děkuji také přítelkyni a rodině za jejich podporu.

Abstrakt

Tato práce se zabývá testováním softwaru a jeho možnou automatizací a porovnává klady а zápory automatizovaného testování s manuálním testováním. V rámci automatizovaného přístupu ie představen nástroi pro automatizované testv HP QuickTest Professional (HP QTP). Hlavním cílem je analýza výhodnosti použití toho nástroje v podmínkách reálného projektu na základě předem definovaných kritérií.

V teoretické části představuji základní pojmy spadající do oblasti testování softwaru a zařazuji testování do procesu vývoje softwaru. Následuje porovnání manuálního a automatizovaného přístupu k testování softwaru a představení nástroje pro automatizované testy HP QTP.

Praktická část má za úkol analyzovat výhodnost automatizace prostřednictvím nástroje HP QTP na reálném projektu dle definovaných kritérií a doporučit za jakých podmínek se automatizace v tomto případě vyplatí. V rámci praktické části je také vytvořena příručka popisující základní funkcionalitu HP QTP, která je předvedena na příkladech. Příručka, která je zároveň výstupem této práce, má testerům pomoci rychleji pochopit práci s tímto nástrojem a tím i lépe naplánovat proces automatizace testů.

Klíčová slova

Testování softwaru, automatizované testy, HP QuickTest Professional, QTP, modely vývoje softwaru, úrovně testování.

Abstract

This thesis deals with software testing and its possible automation and compares the pros and cons of automated testing with manual testing. In the automated approach is introduced tool for automated testing HP QuickTest Professional (HP QTP). The main objective is to analyze the advantages of using this tool in conditions of a real project, based on predefined criteria.

In the theoretical section I introduce basic terms within the field of software testing and identify testing in the software development process. A comparison of manual and automated approach to software testing and introduction of tool HP QTP for automated tests follows.

Target of practical part is to analyze the advantages of automation by using HP QTP on a real project according to defined criteria and recommend for what conditions the automation in this case worthwhile. In the practical part is also created manual which describing the basic functionality of HP QTP, which is demonstrated on examples. Manual, which is also the output of this work, has to help testers to quickly understand the work with this tool and therefore better plan process of tests automation.

Keywords

Software testing, test automation, HP QuickTest Professional, QTP, software development models, levels of testing.

Obsah

1	Úvod.				
	1.1	Vymezení tématu práce			
	1.2	Důvod výběru tématu			
	1.3	Cíl prác	ce 2		
	1.4	Metod	a dosažení cíle 2		
	1.5	Struktu	ra práce 2		
	1.6	Výstup	y práce a očekávané přínosy 3		
	1.7	Rešerše	e		
		1.7.1	Odborné publikace 4		
		1.7.2	Akademické práce 4		
		1.7.3	Internetové zdroje 4		
2	Testo	vání soft	waru 6		
	2.1	Pozice	testování ve vybraných modelech životního cyklu vývoje softwaru6		
		2.1.1	Vodopádový model 6		
		2.1.2	Spirálový model		
		2.1.3	V-model 8		
	2.2	Definic	e testování softwaru		
	2.3	2.3 Softwarová chyba			
	2.4	Softwa	rová kvalita 10		
	2.5	Úrovně	testů 11		
		2.5.1	Testování jednotek 12		
		2.5.2	Integrační testování 12		
		2.5.3	Systémové testování 12		

		2.5.4	Akceptační testování (UAT – User Acceptance Testing)	. 12
		2.5.5	Regresní testování	. 13
	2.6	Manuá	ilní a automatizované testování	. 13
		2.6.1	Využití automatizovaného testování	. 14
		2.6.2	Využití manuálního testování	. 14
		2.6.3	Shrnutí	. 16
3	HP Q	uickTest	Professional	. 17
	3.1	Přístup	y k vytváření testů	. 18
		3.1.1	Expert view	. 18
		3.1.2	Keyword view	. 19
	3.2	Nahráv	/ání	. 19
		3.2.1	Kontextové nahrávání (Context sensitive recording)	. 20
		3.2.2	Analogové nahrávání (Analog recording)	. 20
		3.2.3	Low level recording	. 20
	3.3	Shrnut	í	. 20
4	Využi	tí HP Qu	ickTest Professional pro automatizaci testování na existujícím projektu	. 22
	4.1	Oracle	Siebel	. 22
	4.2	2 Situace na projektu		. 22
	4.3	3 Popis testovacího cyklu		. 23
		4.3.1	Zadávací dokumentace	. 23
		4.3.2	Testování programátorem	. 24
		4.3.3	Tvorba testovacích scénářů	. 24
		4.3.4	Systémové testy	. 24
		4.3.5	Integrační testy	. 25

		4.3.6 Regresní testy	26
	4.4	Předpoklady pro automatizaci testování	27
	4.5	Posouzení vhodnosti použití HP QTP na projektu	28
	4.6	Shrnutí	30
5	Příruč	ka k základní funkcionalitě HP QuickTest Professional	31
	5.1	Obsah příručky	31
	5.2	Úvod do výkladu funkcí	32
	5.3	Spuštění HP QTP	33
	5.4	Nahrání prvního testu	37
	5.5	Spuštění testu a analýza výsledků	40
	5.6	Parametrizace	44
	5.7	Vkládání checkpointů	46
	5.8	Ukládání výstupních hodnot	48
	5.9	Podmínky	52
	5.10	Reportování událostí pomocí metody Reporter.ReportEvent	53
	5.11	Akce	54
	5.12	Object repository (Objektový repositář)	66
		5.12.1 Local object repository (Lokální objektový repositář)	67
		5.12.2 Shared object repository (Sdílený objektový repositář)	71
	5.13	Typy nahrávání v QTP	74
		5.13.1 Kontextové nahrávání (Context sensitive recording)	74
		5.13.2 Analogové nahrávání (Analog recording)	75
		5.13.3 Low level recording	77
	5.14	Funkce	78

5.15 Deskriptivní programovaní				
5.15.1 Statická metoda deskriptivního programování				
5.15.2 Dynamická metoda deskriptivního programová	ní 83			
5.16 Doporučení				
6 Závěr				
Seznam použitých zkratek a termínů				
Seznam použité literatury				
Seznam obrázků a tabulek				
Obrázky				
Tabulky				

1 Úvod

Rozvoj podnikání, ale i běžného života je v současné době závislý na využívání softwaru a informačních technologií. Pro mnoho firem hraje jednu z klíčových rolí rozvoje vývoj a údržba softwaru v podobě různých informačních systémů, které zajišťují chod firmy. Aby software v průběhu času odpovídal aktuálním požadavkům uživatelů, je nutné ho neustále rozvíjet a udržovat. Nedílnou součástí tohoto procesu je mimo jiné i testování softwaru, které postupně nabývá na důležitosti, jak se zvyšují nároky na software a roste tím jeho složitost.

Stejně jako vývoj softwaru, tak i metody testování se mění a vyvíjí. Vzniká velké množství nástrojů, které mají proces testování zjednodušit a zefektivnit. Jednou skupinou takových nástrojů jsou programy pro automatizované testy. Právě zástupcem této skupiny je i nástroj HP QuickTest Professional, kterým se v této práci zabývám a hodnotím, zda se za daných podmínek vyplatí jeho využití v existujícím projektu, jehož úkolem je pravidelná dodávka CRM software Siebel pro nejmenovaného mobilního operátora.

1.1 Vymezení tématu práce

Diplomová práce se zabývá testováním softwaru, porovnáním manuálního a automatizovaného testování. V rámci automatizovaného testování je představen nástroj pro automatické testy HP QuickTest Professional a k němu je vytvořena příručka poskytující přehled základní funkcionality nástroje a předvedení nástroje na příkladech.

1.2 Důvod výběru tématu

Toto téma jsem si vybral, protože v testování softwaru již přes tři roky pracuji a náš testovací tým momentálně stojí před rozhodnutím, zda využít nástroj a HP QuickTest Professional v testovacím procesu a případně za jakých podmínek. Díky tomu, že máme k dispozici aktuálně nevyužívanou licenci, mám možnost učit se s nástrojem pracovat. Svoje poznatky zpracovávám jako příručku pro kolegy testery a právě tento úkol, spolu s určením za jakých podmínek a v jakých fázích testovacího procesu je využití automatizace na projektu vhodné, jsem se rozhodl zpracovat jako diplomovou práci.

1.3 Cíl práce

Hlavním cílem práce je posoudit vhodnost automatizace testování na reálném projektu pomocí nástroje HP QTP 11. Za tímto účelem provádím v práci analýzu podmínek a předpokladů pro užití toho nástroje a identifikuji vhodná místa pro jeho zavedení v rámci testovacího procesu na tomto projektu.

Jako dílčí cíle jsem definoval představení nástroje HP QTP a vytvoření příručky popisující jeho základní funkcionalitu. Tato příručka poskytuje testerům alokovaným na implementaci automatizace základní znalosti a přehled o funkcionalitě HP QTP a tím i informace umožňující lépe naplánovat automatizaci prostřednictvím toho nástroje a později rychleji zvládnout jeho obsluhu.

1.4 Metoda dosažení cíle

Aby byl splněn hlavní cíl práce a byla posouzena výhodnost zavedení HP QTP na reálném projektu, jsou v práci definována kritéria, za jakých se automatizace vyplatí. Poté je popsána situace na projektu a proces, jakým na projektu probíhá testování. Popis situace na projektu je následně porovnán s kritérii definujícími podmínky, za jakých se vyplatí automatizovat a na základě tohoto porovnání jsou určena místa na projektu, kde se zavedení automatizace pomocí HP QTP vyplatí.

Při tvorbě příručky je postupně popisována základní funkcionalita HP QTP a ta je následně předváděna na příkladech. Při psaní této příručky jsou využívány servery a blogy zaměřující se na testování, které poskytují informace, návody a rady jak pracovat nejen s HP QTP, ale i s dalšími nástroji.

1.5 Struktura práce

Práce je rozdělena na tři části. V úvodní teoretické části, která začíná druhou kapitolou, jsou definovány základní pojmy z oblasti testování softwaru a představeny vybrané modely vývoje softwaru, jejichž nedílnou součástí je i testování. Teoretický úvod potom pokračuje popisem jednotlivých úrovní testování porovnáním manuálního a automatizovaného přístupu k testování.

V další části, která je již zaměřena na praktický přínos, je nejprve ve třetí kapitole představen nástroj HP QuickTest Professional a ve čtvrté kapitole je popsán reálný projekt, kde se o zavedení automatizace pomocí tohoto nástroje uvažuje. Jsou zde definována kritéria, při jejichž splnění se automatizace vyplatí. Podmínky na projektu jsou konfrontovány s těmito kritérii a jejich pomocí jsou určeny místa v rámci testovacího procesu, kde se zavedení automatizace pomocí HP QTP vyplatí.

Poslední část, v práci kapitola pět, je věnována vytvoření příručky k nástroji HP QTP popisující základní funkcionalitu tohoto nástroje. Informace z této příručky mají pomoci testerům lépe naplánovat zavedení automatizace prostřednictvím tohoto nástroje a rychleji zvládnout výuku.

1.6 Výstupy práce a očekávané přínosy

Očekávaným přínosem práce je určení, zda je vhodné zavést automatizaci testování softwaru na reálném projektu pomocí nástroje HP QTP a případně za jakých podmínek. Výstupem práce je analýza předpokladů pro automatizaci na tomto projektu a určení v jakých fázích testovacího cyklu je využití automatizace nejvýhodnější a kde je to naopak neefektivní.

Součástí práce je i vytvoření příručky popisující základní funkcionalitu HP QTP, která je poté demonstrována na praktických příkladech. Příručka poskytuje nutné znalosti pro základní práci s tímto nástrojem testerům, kteří s ním začínají pracovat. Zároveň poskytuje i přehled o možnostech tohoto nástroje pro potřeby plánování testování z pohledu test managementu.

1.7 Rešerše

Podklady pro tuto diplomovou práci lze čerpat ze širokého spektra zdrojů. Téma testování softwaru je pokryto mnoha odbornými články a publikacemi a je i častým tématem akademických prací. Na internetu se nachází také velké množství serverů, které se na testování softwaru přímo zaměřují, a kde se publikují články, metodiky a návody z oblasti testování. Posledním, ale také významným zdrojem jsou blogy zakládané testery z různých oblastí testování, kde uživatelé sdílejí svoje zkušenosti, odpovídají na dotazy a publikují články, návody a tutoriály z oblasti testování a vývoje softwaru.

1.7.1 Odborné publikace

V současné době není problém najít v odborné literatuře knihy zabývající se testováním a řízením kvality softwaru obecně. Z odborných publikací jsem v práci čerpal především z knih *Řízení kvality softwaru: průvodce testováním* [21], *Testování softwaru* [16], které jsou v češtině a *Lessons learned in software testing: a context-driven approach* [12], která je v anglickém jazyce. Z těchto tří svazků jsem čerpal informace při psaní úvodní teoretické části a také při přípravě kritérií vhodnosti automatizace na projektu. Tyto publikace jsem využíval v průběhu psaní celé práce a mohl jsem si díky nim lépe utřídit informace převzaté z dalších internetových zdrojů, kterých je na toto téma poměrně velké množství.

1.7.2 Akademické práce

Akademických prací, které se zabývají řízením kvality softwaru, je na VŠE více, ale automatizované testování řeší jen několik z nich. Z diplomových prací je to například práce Terezy Faustové na téma *Nástroje na podporu testování* [2], kde je poměrně rozsáhlá kapitola zaměřená na automatizované testy a nástroje pro automatizované testy. Srovnáním testovacích nástrojů se zabývá také Robin Štolc v diplomové práci *Porovnání komerčních a open source nástrojů pro testování softwaru* [24], kde se do srovnání nástrojů pro automatizované testy dostává i HP QTP, kterým se zabývám v této práci.

Z bakalářských prací se přímo na automatizace testování zaměřuje práce Automatizované testování [17] od Lukáše Pouzara a Automatizované testování webových aplikací [8] od Ramana Holdberha.

1.7.3 Internetové zdroje

Významným zdrojem pro tvorbu příručky k HP QTP byly komunitní servery, blogy a internetové stránky poskytující tutoriály, fóra, návody, tipy a rady ohledně práce s různými testovacími nástroji. Odtud jsem čerpal znalosti potřebné pro naučení se s tímto nástrojem a následně je pak zapracovával do příručky. Nejvýznamnějšími prameny byly blog *Software testing help* [19], blog *QTP Tutorials & Interview Questions* [20] a server *tutorialspoint - Simply easy learning* [26]. Všechny tyto zdroje poskytují návody a tutoriály nejen na nástroj HP QTP, ale i mnoho dalších testovacích nástrojů a navíc i programovacích a skriptovacích jazyků a jsou tak cenným zdrojem informací pro kohokoliv zabývajícím se vývojem nebo testováním softwaru.

2 Testování softwaru

Tato kapitola obsahuje vymezení základních pojmů, vztahujících se k tématu testování softwaru. V následujících odstavcích je vymezen pojem testování jako součást cyklu vývoje softwaru i testování jako samostatný proces a pojmy, které s testováním softwaru úzce souvisí.

2.1 Pozice testování ve vybraných modelech životního cyklu vývoje softwaru

Cyklus vývoje softwaru je proces mapující všechny kroky vedoucí ke vzniku softwaru od prvotního záměru vytvořit takový software až k jeho uvedení na trh. Metod a modelů, které by definovaly průběh tohoto procesu, se v průběhu let objevila celá řada a každý z nich má své výhody a nevýhody a nedá se obecně říci, který z modelů je nejlepší. Často jde o variace či kombinace více zavedených modelů. V následujících odstavcích uvádím příklady vybraných často využívaných modelů. [16]

2.1.1 Vodopádový model

Ve vodopádovém modelu je proces vývoje rozdělen do jednotlivých etap, které tvoří sekvenci od prvotní myšlenky a formulace požadavků až po výsledný produkt. Pořadí etap je pevně dané a je nutné projít během vývoje všemi etapami postupně. Pro postup do další etapy je vždy nutné nejdříve dokončit tu aktuální. Přestože jde o jeden z nejstarších modelů, je i v současné době poměrně využíván nebo alespoň slouží jako výchozí pro nějakou inovovanou variantu modelu, protože jeho pevná pravidla a srozumitelnost přináší i současné době poměrně dobř výsledky. [16][22]



Obrázek 1 - vodopádový model, zdroj: [18]

2.1.2 Spirálový model

V tomto modelu prochází vývoj několika iteracemi, při kterých se opakovaně ve stejném sledu provádí definované kroky. Podle Barryho Boehma, který tento model definoval poprvé v roce 1986, se cyklus se dělí na čtyři hlavní části:

- Určení cílů, alternativ, omezení
- Vyhodnocení alternativ, identifikace a řešení rizik
- Vývoj a verifikace další úrovně produktu
- Plánování dalších fází [1]

Po každé další fázi tedy produkt projde opět stejným cyklem, kdy se naplánuje další fáze, vyhodnotí se cíle a omezení poté se zhodnotí rizika a alternativy a nakonec dojde vývoji a otestování další části produktu. Cyklus se opakuje, dokud produkt není hotový. [14]



Obrázek 2 - spirálový model, zdroj: [25]

2.1.3 V-model

V-model klade důraz na testování od samého počátku vývoje. Od startu projektu se tak testuje souběžně s vývojem. Model je rozdělen na dvě větve, kdy se na jedné nachází jednotlivé fáze vývoje softwaru a na druhé potom odpovídající úrovně testování. V každé fázi dochází k ověřování funkčnosti a splnění požadavků zadání vztahující se ke konkrétní fázi vývoje.

Výhodou tohoto modelu je, že se nejprve testují menší celky a komponenty a až poté větší sestavy. Před vstupem do každé další fáze dochází k otestování výstupu té aktuální, chyby se tak často odhalí dříve a snižují se tím náklady na opravu. [21][27]



Obrázek 3 - V-model vývoje softwaru, zdroj: [21]

Jak již bylo řečeno, modelů vývoje softwaru bylo již vytvořeno mnoho, ale ve většině z nich je testování jejich součástí. Přesto, že v praxi mu často není věnována dostatečná priorita, je testování velmi důležitou fází ve vývojovém cyklu softwaru a má velmi významný vliv na kvalitu finálního produktu.

2.2 Definice testování softwaru

Pojem testování softwaru vymezuje mnoho definicí, které se od sebe často liší buď kvůli časovému vývoji, nebo úhlu pohledu. Z toho důvodu není žádná definice ustálená, ale jako příklad jsou uvedeny alespoň dvě, se kterými jsem se setkal nejčastěji.

Proces zahrnující jakékoliv aktivity zaměřené na vyhodnocení vlastností nebo schopností programu nebo systému a určení, zda odpovídají požadovaným výsledkům. [4]

Šetření prováděné za účelem poskytnutí informací o testovaném produktu nebo službě stakeholderům. [11]

Ačkoliv většina definic pravdivě popisuje podstatu testování nebo alespoň její část, žádná z nich nedokáže zcela pokrýt tak komplexní proces, jakým je softwarové testování. Obecně se

pojem testování softwaru nejčastěji pojí s úkolem nalézt a odstranit chyby v softwaru a dodat software v požadované kvalitě.

2.3 Softwarová chyba

Softwarové chyba je jakákoliv odchylka od specifikace softwarového produktu, kvůli které produkt neplní svojí funkci nebo nepracuje tak, jak se od něj očekává. Dle Rona Pattona [16] se o chybu jedná, pokud splňuje jednu nebo více následujících podmínek:

- 1. Software nedělá něco, co by podle specifikace dělat měl.
- 2. Software dělá něco, co by podle specifikace produktu dělat neměl.
- 3. Software dělá něco, o čem se specifikace nezmiňuje
- 4. Software nedělá něco, o čem se produktová specifikace nezmiňuje, ale měla by se zmiňovat
- 5. Software je obtížně srozumitelný, těžko se s ním pracuje, je pomalý nebo podle názoru testera softwaru – jej koncový uživatel nebude považovat za správný. [16]

Úkolem testovacího týmu je tyto chyby vyhledávat, co nejpřesněji popsat a odreportovat a v co nejkratším možném čase zajistit jejich opravu.

2.4 Softwarová kvalita

Z pohledu softwaru kvalitou rozumíme míru splnění požadavků a schopnost plnit funkci, pro kterou byl software vytvořen. Přeneseně by se dalo říci, že kvalitní software je takový, který dokáže plnit svou funkci podle očekávání a bez problémů.

Dle modelu FURPS [3] vytvořeného společností Hewlett-Packard v roce 1987 se na kvalitu software nahlíží z následujících 5 dimenzí:

- Functionality (funkčnost) hodnotí hlavní funkčnost, schopnosti programu a míru splnění požadavků byznysu.
- Usability (vhodnost k použití) řeší použitelnost softwaru z pohledu koncového uživatele

- Reliability (spolehlivost) zkoumá četnost a závažnost chyb a odchylky při zpracování dat.
- Supportability (podporovatelnost) hlediskem hodnocení je oblast údržby, přizpůsobitelnosti, rozšiřitelnost, testovatelnosti a podpory aplikace, Důležitá je také schopnost zapojení aplikace do existujících procesů podpory a údržby SW.
- Performance (výkonnost) hodnotí celkovou výkonnost a schopnost plnění klíčových funkcí systému za různých podmínek. [3]

Model FURPS, jež je akronymem výše uvedených vlastností, je v současné době rozšiřován o znaménko "+", které představuje rozšíření o další dimenze, protože postupem času již původní nestačily ke komplexnímu zhodnocení kvality softwarového produktu.

Pod rozšířením "+" se nejčastěji skrývají následující kategorie [21]:

- Design constrains (omezení návrhu) omezení, která jsou kladena na návrh produktu.
- Implementation requirements (požadavky na implementaci) vymezení programovacích jazyků, standardů, zdrojů atd.
- Interface constrains (požadavky na rozhraní) popisuje omezení rozhraní a schopnost interakce s okolními systémy.
- Physical requirements (požadavky na fyzické vlastnosti) popis fyzických vlastností produktu. [21]

2.5 Úrovně testů

Dle výše uvedeného V-modelu vývoje softwaru, se testování dělí do čtyř úrovní, které odpovídají jednotlivým úrovním v procesu vývoje.

- Testování jednotek
- Integrační testování
- Systémové testování
- Akceptační testování [21]

2.5.1 Testování jednotek

Unit testing neboli testování jednotek má za úkol testovat separované části systému. Takové části mohou reprezentovat samostatně funkční moduly, části kódu (například nově přidávaná funkcionalita) nebo jednotlivé komponenty. Cílem je otestovat a odladit část před integračními tesy a ověřit že taková jednotka samostatně funguje. Tento typ testů zpravidla obstarávají sami vývojáři.

2.5.2 Integrační testování

Po otestování jednotek přichází na řadu testovací tým a integrační testování, které má za cíl otestovat spolupráci jednotek zapojených do funkčních celků a ověřit, že mezi sebou dokážou spolupracovat. Předpokladem je, že jsou k dispozici alespoň dvě jednotky, které mezi sebou interagují a tvoří systém. Integrační testy odhalují chyby mezi komponentami, platformami, subsystémy, ale i hardwarem a softwarem.

2.5.3 Systémové testování

Po úspěšném integračním otestování je k dispozici kompletní integrovaný systém, který se v této úrovni testuje jako celek. Tato fáze je poslední úrovní testování, které má za cíl ověření splnění funkčních i technických požadavků systému, než je výsledný produkt předán zákazníkovi. Součástí jsou tak jak funkční tak nefunkční testy.

2.5.4 Akceptační testování (UAT – User Acceptance Testing)

User Acceptance testing, v češtině uživatelské akceptační testy jsou, jak název napovídá, testy prováděné již na straně klienta. Testery jsou v tomto případě většinou koncoví uživatelé systému, kteří testují předem připravené scénáře zaměřené na klíčovou funkcionalitu systému.

Cílem této úrovně testování je ověření, zda byla splněna akceptační kritéria, což je soubor měřitelných a ověřitelných podmínek pro přijetí produktu vydefinovaných zákazníkem. [21]

2.5.5 Regresní testování

Regresní testy se dají vykonávat na všech výše uvedených úrovních. Regresní sada testů obsahuje základní scénáře ověřující klíčovou funkcionalitu aplikace, zároveň ale musí obsahovat testy, které pokrývají všechny součásti aplikace. Regresní testy se provádí vždy, když dojde ke změně v aplikaci nějakou změnou stávající funkcionality, případně přidáním funkcionality nové. Regresní testy se například provádí s každou novou verzí aplikace či systému, tak aby se odhalilo, jestli zaváděné změny negativně neovlivní ostatní součásti systému.

V závislosti na velikosti projektu může jít o velké množství scénářů, které je nutné do regresní sady zahrnout. S ohledem na fakt, že regresní testy je potřeba provést po každé změně aplikace, která může mít vliv její celkovou funkčnost, je vhodné regresní testy automatizovat, pokud je to možné.

2.6 Manuální a automatizované testování

Tato kapitola popisuje rozdíly mezi manuálním a automatizovaným přístupem k testování a porovnává vhodnost jejich použití.

Podle využití testovacích nástrojů se testovaní dělí na automatizované a manuální. Automatizované testování je tedy testování, které provádí software v podobě nějakého testovacího nástroje a manuální testování je vykonáváno lidským faktorem v podobě testerů.

V poslední době automatizované testování nabývá na důležitosti, protože nástroje pro automatizaci jsou stále sofistikovanější a dokážou zastoupit stále více činností v rámci testovacího procesu bez nutných zásahů lidského pracovníka. Dokážou mimo exekuce testů také instalovat či konfigurovat aplikace nebo umožňují vytváření testů z testovacích analýz. Přesto výhodnost použití automatizovaného testování vždy velmi záleží na podmínkách konkrétního projektu, kdy se automatizace nemusí vždy vyplatit nebo ani není možná a manuální testování je tak v některých případech stále nezastupitelné.

2.6.1 Využití automatizovaného testování

Automatizovaný přístup se nevyplatí vždy, někdy je jeho implementace příliš nákladná, například na menších projektech, nebo pro něj nejsou na projektu vhodné podmínky, například kvůli nekonzistenci prováděných testů. Jsou ale naopak případy, kdy se automatizace nabízí jako ideální řešení.

Automatizace je vhodná především pokud se provádí velké množství testů, které se nemění a pokrývají stejné části aplikace nebo pokud je potřeba otestovat velké množství vstupů a výstupů. V praxi se proto automatizace uplatňuje především při regresních testech. Využít je ale lze například i při testech jednotkových, integračních nebo testech nefunkčních, které nejsou zaměřeny přímo na ověření funkcionality aplikace, ale spíše na její technickou způsobilost. Sem spadají zátěžové testy nebo bezpečnostní testy.

Výhodou automatického testování je úspora času a nákladů na spouštění a údržbu testů. Vyšší pokrytí aplikace udržovanou a postupně rozšiřovanou sadou regresních testů. U některých testů, které vyžadují dlouhodobou interakci se systémem, dochází také k výraznému zvýšení efektivity a přesnosti výsledků. Dále se při automatických testech snadněji dohledávají chyby, vzhledem k tomu, že automatizace v každém testu generuje vždy stejný postup a je tak jednodušší případné chyby replikovat a zajistit tak jejich brzkou identifikaci a odstranění. [21]

I přes nesporné výhody, které s sebou automatizace přináší ji nelze využít vždy a v mnoha případech je tak manuální testování nenahraditelné nebo ho automatizace pouze v některých částech doplňuje.

2.6.2 Využití manuálního testování

Manuální testování hraje stále dominantní roli oproti automatizovanému testování a to hlavně kvůli nízkým vstupním nákladům a jednoduchosti provedení testů. Automatizaci navíc nelze v některých případech použít nebo lze, ale je to velice neefektivní. Například v případech, kdy vývoj softwaru prochází pravidelnými iteracemi, kde se mění, přidává nebo odebírá vždy odlišná funkcionalita. V takových případech se automatizace nevyplatí, protože nelze využívat stejné sady testů a pro každou iteraci je tak nutné generovat testy nové. Výhody manuálního testování spočívají v aspektech testování, ve kterých ho automatizace nedokáže nahradit. Automatizace sice naprosto přesně a bez chyby provede daný testovací skript mnohokrát rychleji, než ho tester provede manuálně, ale nedokáže ověřovat mimofunkční požadavky jako například nekorektní zobrazení barev, nečitelnost textu nebo přetékání textu mimo obrazovku. Tester navíc dokáže kreativně řešit nestandardní situace, během kterých už by automatizovaný test selhal a reportoval chybu i přesto, že jde o správné chování. Tester takovou situaci dokáže lépe posoudit a ihned analyzovat příčinu této nestandardní situace a v ideálním případě buď chybu rovnou z pozice uživatele eliminovat, nebo se jí změnou postupu testu vyhnout, popřípadě alespoň předat vývoji informace o problému i s analýzou co přesně je třeba změnit, aby byla chyba odstraněna. [21]

	Automatizované testování	Manuální testování
Výhody	 Nízké náklady a vysoká efektivita při dlouhodobé tvorbě a údržbě sady testů Rychlost a přesnost exekuce Přesná replikace chybových stavů Eliminace lidského faktoru Možnost zpracování velkého množství vstupů a výstupů 	 Nízké vstupní náklady Kreativní řešení nestandardních situací Přímá analýza chybových stavů Díky jednoduchosti mohou testy provádět i méně zkušení pracovníci
Nevýhody	 Náklady na testovací nástroje Časově náročná tvorba testovacích skriptů Potřeba zkušených pracovníků 	 Časově náročná exekuce Riziko chyb ze strany testerů S objemem testů úměrně narůstá pracnost a tím i náklady

Tabulka 1 - porovnání výhod a nevýhod manuálního a automatizovaného přístupu

2.6.3 Shrnutí

Přestože s postupem času a se zdokonalováním testovacích nástrojů narůstá význam automatizace v softwarovém testování, zůstávají manuální testy v některých případech stále nenahraditelné. Zatímco automatizace s sebou přináší rychlost, bezchybnost a v podstatě neomezený počet možných opakování, někdy je potřeba kvalita, kreativita, variabilita a analytické myšlení, které mohou nabídnout pouze testy manuální.

Je proto vždy potřeba zvážit výhodnost využití automatizace v podmínkách konkrétního projektu. Právě taková úvaha výhodnosti je provedena v následujících kapitolách, kde je nejprve představen nástroj pro automatické testy HP QTP a poté je analyzována vhodnost jeho využití na reálném projektu, kde jsem zaměstnán.

3 HP QuickTest Professional

HP QuickTest Professional je nástroj pro automatizované testování nejrůznějších aplikací a prostředí, který se dlouho době řadí ke špičce mezi komerčními nástroji pro automatické funkční testy. Původně byl od roku 2001 vyvíjený společností Mercury Interactive pod jménem QuickTest Professional. V roce 2006 koupila Mercury Interactive společnost Hewlett-Packard (HP) a produkt dále vydávala pod svou hlavičkou jako HP QuickTest Professional (HP QTP). [9][21]

Od verze 11.5 dochází ke spojení nástroje HP QTP, který obstarává především testování GUI, s nástrojem HP Service Test, který je zaměřený na testování API. Jako jeden balík firma HP nyní nástroj vydává pod jménem HP Unified Functional Testing. Aktuální verze je verze 12 vydaná v březnu 2014. [9]

Verze	Rok vydání	
5.5	2001	
6.0	2002	
6.5	2003	
7.0	Nikdy nevydána	
8.0	2004	
8.2	2005	
9.0	2007	
9.1	2007	
9.2	2007	
9.5	2008	
10.0	2009	
11.0	2010	
11.5	2012 (přejmenováno na Unified Funcional Testing)	
11.52	Červen 2013	
11.53	Listopad 2013	
12.0	Březen 2014	

Tabulka 2 - Přehled vydaných verzí HP QTP (později HP UFT), zdroj: [9]

Vzhledem k tomu, že je na projektu zakoupena pouze verze 11, je v práci užíváno pouze původní označení HP QTP.

Jak již bylo uvedeno HP QTP podporuje testování velkého množství aplikací a prostředí a tyto možnosti se dají ještě rozšířit volitelnými pluginy. Za všechny QTP podporuje například SAP, Siebel, Oracle, Web, .NET, Javu a další.

Samotné testy se v QTP vytváří ve skriptovacím jazyku VBSkript vyvinutým společností Microsoft a při psaní testů se tak využívá všech charakteristik VBSkriptu jako práce s proměnnými, funkcemi, objekty atd.

3.1 Přístupy k vytváření testů

HP QTP nabízí dva rozdílné přístupy při vytváření testovacích skriptů díky rozdělení na Keyword view (náhled na skript zobrazený pomocí klíčových slov) a Expert view¹ (skript zapsaný pouze pomocí VBSkriptu). Zatímco Keyword view nabízí možnost vytvářet testy i testerům nepoznamenaným programováním a skriptováním díky grafickému zobrazení objektů uspořádaných chronologicky podle provedených kroků do tabulky, Expert view na druhou stranu nabízí strohý pohled na zápis testu pouze pomocí VBSkriptu.

Obě view mapují stejné objekty a operace s nimi a informace mezi nimi se navzájem synchronizují. V podstatě obsahují stejná data pouze v jiném formátu.

3.1.1 Expert view

Expert view funguje v podstatě jako skript editor, kde je možné přímo upravovat nebo tvořit testovací skript. Každý řádek v Expert view reprezentuje krok testu zapsaný ve VB skriptu. V zápise jsou identifikovány objekty, se kterými se v kroku pracuje, a ty jsou odděleny tečkou. Operace, které se s objekty provádí, určuje metoda uvedena v závěru zápisu.

¹ Z důvodu lepší orientace v přiložených obrázcích z HP QTP budu dále užívat původní názvy v angličtině (tedy Keyword view a Expert view).

*	Action2	
3:	Dialog("Login").WinEdit("Password:").Set DataTable("Heslo", dtGlobalSheet)	
4:	Dialog("Login").WinButton("OK").Click	
5:	If Dialog("Login").Dialog("Flight Reservations").Exist Then	
6:	Dialog("Login").Dialog("Flight Reservations").Static("Please enter agent name").Output CheckPoint("Chybova zprava")	
7:	Dialog("Login").Dialog("Flight Reservations").WinButton("OK").Click	
8:	Dialog("Login").WinButton("Cancel").Click	
9:	Reporter.ReportEvent micFail, "Neúspěšné přihlášení", "špatný login nebo heslo"	
10:	Else	
11:	Window("Flight Reservation").Check CheckPoint("Flight Reservation")	
12:	Window("Flight Reservation"). Close	
13:	End If	-
H 4 🕨	N Keyword View Expert View	

Obrázek 4 - náhled na Expert view, zdroj: autor z nástroje HP QTP

3.1.2 Keyword view

Keyword view umožňuje grafické zobrazení kroků testu uspořádaných v tabulce, kde každý řádek reprezentuje krok testu a každý sloupec jednotlivé části tohoto kroku. V tomto grafickém rozhraní tester může jednoduše upravovat kroky testu, například změnou objektů, operací nebo vkládaných hodnot a poskytuje tak možnost tvořit a upravovat test i uživatelům, kteří neovládají skriptování.

Reack Cheve			
Item	Operation	Value	Documentation
🗢 💏 Action2			
🖳 📮 SystemUtil	Run	"C:\Program Files\HP\QuickTest Profe	Open the "C:\Program Files\HP\QuickTest Professional\samples\flight\a
👻 🛅 Login			
Agent Name:	Set	DataTable("Login", dtGlobalSheet)	Enter <the "login"="" column="" data="" of="" table="" the="" value=""> in the "Agent Name:"</the>
Password:	Set	DataTable("Heslo", dtGlobalSheet)	Enter <the 'heslo'="" column="" data="" of="" table="" the="" value=""> in the ''Password:'' ec</the>
🚾 OK	Click		Click the "OK" button.
👻 🍯 📰 Flight Reservations	Exist		Check whether the "Flight Reservations" dialog box exists. If so:
Please enter agent n	Output	CheckPoint("Chybova zprava")	Store the current values of selected properties of the "Please enter age
▲	lor i		
Keyword View / Expert View /			

Obrázek 5 - náhled na Keyword view, zdroj: autor z nástroje HP QTP

3.2 Nahrávání

Velkou výhodou QTP je možnost přímého nahrávání akcí uživatele. Nahrávání umožňuje ukládání nahrávaných objektů a rychlejší vytváření skriptu bez nutnosti dohledávat objekty a metody, které s nimi manipulují. QTP umožňuje následující 3 způsoby nahrávání:

3.2.1 Kontextové nahrávání (Context sensitive recording)

Kontextové nahrávání je nastavené jako výchozí způsob nahrávání v QTP. Umožňuje nahrávání, při kterém se rozpoznávají objekty bez ohledu na jejich umístění na obrazovce. Rozpoznávání objektů je také vázáno na aktivované pluginy pro konkrétní aplikace. Dle mého názoru jde o nejspolehlivější způsob nahrávání, ale jsou případy, kdy ho není možné využít, například kvůli nedostupnosti pluginu nebo jde o výjimečný případ, kdy je vhodnější užít analogové nahrávání, viz níže.

3.2.2 Analogové nahrávání (Analog recording)

Při analogovém nahrávání se zaznamenává přesný pohyb myši vzhledem k aktivnímu a oknu a akce klávesnice i myši. Tohoto způsobu se dá využít například při replikaci podpisu na podepisovacím appletu což je případ, kdy nelze využít kontextové nahrávání.

3.2.3 Low level recording

Při tomto nahrávání QTP pracuje s objekty i v případě, že je nerozpozná. Všem objektům přiřazuje obecnou skupinu buď "Window" nebo "WinObject". Tento typ nahrávání se dá využít hlavně v případech, kdy mají objekty své pevné místo na obrazovce, protože poloha objektu slouží zároveň jako jeho hlavní identifikátor.

3.3 Shrnutí

HP QTP se dlouhodobě řadí mezi špičku komerčních nástrojů určených k automatizaci testů. Kombinuje všechny běžně používané techniky automatizovaného testování, tedy:

- Přímé psaní skriptů vyžaduje znalost skriptovacího jazyka a testy se vytváří přímým psaním testovacího skriptu.
- Zachycení a přehrávání aktivity uživatele vytváření skriptu na základě nahrávání akcí uživatele.
- Modifikace generovaných skriptů ruční úprava skriptů pořízených při nahrávání uživatele

- Testování řízené daty tento přístup umožňuje provádět testy nad velkým množstvím dat. Využívá vstupy například z databázových tabulek nebo XLS souborů a stejným způsobem i ukládá výstupy.
- Testování řízené klíčovými slovy interpretace testu pomocí klíčových slov označující objekty, procedury, parametry a jejich hodnoty. Umožňuje skládání testu pouze za pomoci výběru a uspořádání těchto klíčových slov. [21]

Jde tedy o relativně univerzální nástroj, který dokáže automatizované testy bez problémů zastřešit. Pokud je projekt dost velký, disponuje finančními zdroji na zakoupení licence a vytváří podmínky pro využití automatizace, je určitě na místě využití automatizace za pomocí HP QTP případně nějakého alternativního nástroje zvážit.

4 Využití HP QuickTest Professional pro automatizaci testování na existujícím projektu

V této kapitole je popsán vývojový cyklus a podmínky na reálném projektu, kde pracuji a na základě tohoto popisu je posouzena vhodnost nasazení HP QTP a určeno, ve kterých fázích testovacího procesu se případné nasazení nejvíce vyplatí.

Před posouzením vhodnosti použití HP QTP je nejdříve představen projekt, pro který se má nástroj použít.

4.1 Oracle Siebel

CRM systém Siebel od společnosti Oracle je nejvýznamnější položkou vývoje softwaru na našem projektu, proto je zde v krátkosti představen.

Siebel je komplexní systém pro řízení vztahů se zákazníkem, který umožňuje spravovat kompletní přehled o zákaznících, jejich produktech, platbách, komunikační historii a mnoha dalších položkách z oblasti řízení vztahů se zákazníky. Slouží zároveň jako systém pro zřizování objednávek produktů a služeb na prodejnách.

4.2 Situace na projektu

Projekt, na kterém se nacházím, obstarává pravidelné dodávky softwaru v podobě úprav CRM systému Oracle Siebel pro nejmenovaného klienta poskytujícího telekomunikační služby. Klient musí neustále reagovat na vývoj na trhu tím, že vyvíjí, mění a udržuje produkty a služby, které nabízí. S tím přichází i potřeba udržovat softwarové řešení, které mu umožňuje provoz a prodej těchto produktů a služeb, a jehož součástí je právě i námi dodávaný CRM systém Siebel.

Klient si definuje změny, jaké chce v aplikaci provést, což většinou zahrnuje tvorbu nebo úpravu produktů a služeb. S tím pak souvisí i úpravy v integraci, protože CRM Siebel je pouze jedna součást složité softwarové architektury, na které participuje řádově několik desítek systémů, databází a integračních platforem. Část těchto systémů je ještě zastřešována naším projektem, většinu ale klient obstarává sám nebo outsourcuje² od dalších dodavatelů.

Z pohledu našeho projektu jsme zodpovědní především za dodávky CRM Siebel, a proto i testování se zaměřuje především na správnou funkcionalitu tohoto systému. Je ale nutné zajistit korektní zpracování vstupů a výstupů v rámci integrace a většinu testů je tak nutné provádět end-to-end³.

4.3 Popis testovacího cyklu

Změny v softwaru, které klient požaduje, jsou vyvíjeny v rámci dílčích projektů a tyto projekty jsou poté sdružovány do tzv. releasů⁴. Release tedy představuje nasazení změn v rámci jednoho období na produkční prostředí klienta. Počet releasů je zpravidla pět nebo šest za rok, v závislosti na jejich objemu. Testovací cyklus každého z dílčích projektů, ze kterých se release skládá, je popisován v následujících odstavcích.

Důležitá informace, kterou je třeba před tímto popisem uvést je, že oproti metodikám, kde jsou definovány pojmy jednotkových, systémových a integračních testů se chápání těchto pojmů na tomto projektu poněkud liší. Tak, jak jsou tyto pojmy vnímány na projektu, tak byly převzaty ze strany klienta a odlišnosti jsou podrobněji popsány dále v této kapitole.

4.3.1 Zadávací dokumentace

Změnové požadavky, které jsou zařazeny do aktuálního releasu, tvoří jednotlivé dílčí projekty s vlastním názvem a označením. Každý projekt řeší odlišnou funkcionalitu, ale zároveň na sebe v některých případech projekty mohou navazovat, pokud se vyvíjejí ve více fázích. V rámci každého takového projektu jsou zpracovány dokumenty popisující požadavky na změny funkcí a produktů případně obsahující specifikaci nových funkcí a produktů.

² Zajištění vybraných činností firmy externími zdroji.

³ Testování procesu od jeho počátku až do konce napříč celou integrační cestou.

⁴ Release je označení pro soubor dílčích projektů, jež zahrnují změny softwaru, které mají být implementovány v rámci jednoho období.

Tyto požadavky jsou následně konzultovány mezi zadavateli projektů a vývojovým týmem, což má za cíl vyřešit, jakým způsobem je možné změnové požadavky implementovat. Výsledkem je dokument ASD (Architecture Solution Dokument), což je dokument popisující technické řešení změnových požadavků, který je vytvořen pro každý dílčí projekt zvlášť.

Na základě ASD dokumentu se poté na straně vývoje dále vytváří DD (Detail Design), což je dokument obsahující detailní návrh řešení a na jeho základě poté probíhá vývoj.

4.3.2 Testování programátorem

Jakmile jsou požadované změny vyvinuty, dochází před předáním k systémovým testům, k ověření měněného kódu na straně programátora. Jde o testování na úrovni jednotkových testů, které provádějí sami programátoři. Zpravidla jde o jednoduché provolávání změnovaných funkcí v rámci jednotlivých projektů před předáním k systémovým testům, aby se ověřila správnost změn provedených v kódu.

4.3.3 Tvorba testovacích scénářů

Dokumenty ASD a DD vytvořené v předchozí části zadávací dokumentace slouží jako vstupy pro testovací tým. Na jejich základě testeři vytváří testovací analýzy pro projekty, které jsou jim přiřazeny. Výstupem jsou dokumenty obsahující testovací plán, testovací strategii, systémové a integrační testy. Součástí těchto výstupů je mimo jiné i definování priorit scénářů a požadavků na data. V případech větších projektu mohou požadavky na data tvořit až stovky položek.

Jakmile dorazí pokyn od vývoje, že je daný projekt vyvinutý a nasazený, může se začít se systémovými testy.

4.3.4 Systémové testy

Náš projekt je mimo CRM Siebel zodpovědný také za několik dalších systémů, které představují databáze nebo integrační platformy. Systémové testy u nás označují testy, které mají za cíl ověřit korektní chování každého samostatného systému po implementaci nových změn přidaných v rámci aktuálního releasu, předtím než je systém zapojen do integrace.

Chápání pojmu systémových testů se tedy na tomto projektu liší od toho, jak je chápáno v rámci výše uvedeného V-modelu, kde jsou systémové testy chápány jako testy ověřující finální produkt před předáním klientovi.

Systémové testy tedy v tomto případě ověřují implementaci změnované funkcionality každého systému zvlášť a z pohledu dílčího projektu. Provádějí se na vlastním testovacím prostředí nacházejícím se u nás na projektu. Vzhledem k izolaci každého testovaného systému, jde o jednoduché většinou jednokrokové testy, které ověřují provedené změny. Vzhledem k tomu, že na otestování systémových testů připadá v průměru 7-14 dní a testuje se několik projektů, které jsou s každým releasem jiné, tak se automatizace v tomto případě nevyplatí, protože tvorba nových automatizovaných testů pro každý projekt by zabrala více času, než testy manuální, které jsou v tomto případě jednoduché a jejich exekuce je rychlá.

Cílem těchto testů je odstranit maximum chyb, před zahájením integračních testů, které by mohly relativně snadno odhalitelné a triviální chyby zbytečně blokovat.

4.3.5 Integrační testy

Po ukončení systémových testů jsou zahájeny testy integrační. Ty jsou prováděny na kopii produkčního prostředí, označovaného jako "předprodukce". Na tomto prostředí jsou integrovány všechny systémy, včetně těch, za které náš projekt nenese zodpovědnost.

Integrační testy mají odhalit veškeré chyby v komunikaci mezi jednotlivými systémy a zajistit tak korektní chování produkčního prostředí po nasazení aktuálního releasu.

V rámci testovacího cyklu jde o největší objem testů, které by měli zaručit kvalitu dodávky vývoje do aktuálního releasu. V rámci každého dílčího projektu jsou naplánovány endto-end testy zahrnující všechny systémy, které jsou dotčené upravovanou funkcionalitou v rámci změnových požadavků. Na těchto testech tak spolupracují všichni dodavatelé. Dílčích projektů, ve kterých se řeší vždy jiné změnové požadavky, může být v závislosti na objemu releasu až několik desítek. V závislosti na počtu těchto dílčích projektu a na jejich náročnosti se tak objem testů vyšplhá obvykle na 1500 až 2000 testů za release. Od objemu releasu se poté odvíjí také časové období, které je na testy alokováno. V průměru se integrační testy testují cca měsíc a půl. Ve výjimečných případech se etapa integračních testů o týden prodlužuje nebo přichází na řadu přesčasy a pracovní víkendy pokud je nutné dotestovat v termínu.

Zde se automatizace opět nevyplatí vzhledem, k tomu že by bylo nutné připravovat nové testy ke každému releasu. S přihlédnutím k množství a náročnosti těchto testů by ani nebylo možné je v tak krátkém čase připravit a odexekuovat.

4.3.6 Regresní testy

Regresní testy přichází na řadu po ukončení integračních testů. Pokrývají klíčovou funkcionalitu aplikace nejčastěji používanými scénáři z pohledu uživatelů. Cílem je ověřit, že implementované změny neměli negativní dopad na původní funkčnost.

Regresní testy v současné době čítají zhruba 270 testovacích scénářů, ale toho číslo se může s každým releasem změnit, v závislosti na tom, zda je implementována nějaká významná funkčnost, kterou je nutné do regresních testů zařadit, nebo naopak jsou některé testy odebrány, pokud po implementaci změn již nejsou relevantní.

Na regresní testy bývá alokováno 3-6 testerů a testují se 3-5 dní. Na průběh testů má velký vliv počet a závažnost chyb zanesených implementací požadovaných změn. V ideálním případě pokud testy neblokuje žádná závažná chyba, jsou tři seniorní testeři schopni odexekuovat většinu testů z regresní sady za tři dny. Předpokladem pro hladký průběh testování je příprava dat ještě před začátkem testů nejlépe pro alespoň dvě iterace každého testu. Tato příprava zpravidla zabere více času než samotné testy a je potřeba na ní přiřadit testery ještě během dokončování integrační fáze.

V případě regresních testů je automatizace zcela na místě. Jde o ustálenou sadu end-to-end testů, které je možné provádět opakovaně po každém releasu. Pokud dojde ke změně v testech, týká se zpravidla jen několika málo testů.

Automatizace se vyplatí i v případě přípravy dat a to jak pro regresní testy, tak pro integrační a systémové testy, kde je často potřeba exekuovat testy nad konkrétními daty,
kterých není na produkci dostatek a je proto potřeba je připravit. Taková příprava může představovat například zřízení stovky objednávek jednoho konkrétního produktu, nad kterým pak bude probíhat další exekuce samotných testů.

4.4 Předpoklady pro automatizaci testování

V této kapitole jsou uvedeny obecně uznávané předpoklady pro zavedení automatizace, aby bylo možné lépe identifikovat, za jakých podmínek a v jaké fázi testování je vhodné nástroj HP QTP použít.

Na předpoklady pro úspěšné zavedení automatizace lze nahlížet ze dvou pohledů. Za prvé z pohledu test managementu, který řeší hlavně zázemí a připravenost projektu na automatizaci, plánování a strategii zavádění automatizace a dále potom finanční, lidské a časové zdroje.

Druhá kategorie předpokladů pro automatizaci je řešena z pohledu testovacích scénářů. Zde se řeší vlastnosti, časová náročnost a množství testovaných scénářů, při kterých se automatizace vyplatí a při kterých již ne.

Předpoklady z pohledu test managementu jsou dle [21] následující:

- Mít pro automatizaci tým pracovníků přiřazených pouze na tento úkol, tak aby se mu mohl věnovat na plný úvazek. Tento tým má tak v rámci automatizace za úkol identifikaci cílů a potřeb, navržení řešení, plánování implementace, volbu nástrojů, realizaci a podporu během testovacího procesu.
- Podpora managementu v podobě spolupráce na strategii, plánování a řešení problému, nikoliv pouze přiřazením zdrojů.
- Dostatečné časové a finanční zdroje na zajištění procesu automatizace.
- Strategie a plán automatizace s jasně definovanými přínosy.
- Zkušení pracovníci a vhodné nástroje.
- Dlouhodobá kontinuální údržba procesu automatizace.

Z pohledu testovacích scénářů jsou předpoklady pro zavedení automatizace následující [21]:

- Existující stabilní sada scénářů, která nepodléhá častým změnám.
- Testy, které se provádějí často a pravidelně.
- Testy, které se mají automatizovat, by měli testovat kritickou funkcionalitu systému.
- Automatizace přinese významnou úsporu času nebo lidských zdrojů.

4.5 Posouzení vhodnosti použití HP QTP na projektu

Pokud se bude vycházet z výše uvedených předpokladů pro zavedení automatizace, tak z pohledu managementu je projekt na automatizaci připraven. Automatizací byli pověření tři seniorní testeři, kteří úzce spolupracují s test managementem na plánování celého procesu. K dispozici je dosud nevyužívaná licence HP QTP, prostřednictvím které je možné automatizaci realizovat a vyhnout se tak počáteční investici do testovacího nástroje. Zbývá tak jen určit strategii jakým způsobem a ve kterých fázích procesu HP QTP využít.

Testovací proces na zmiňovaném projektu a vhodnost automatizace jednotlivých fází je zrekapitulován v následující tabulce, kde jsou popsány fáze testování z pohledu řešeného projektu (a z pohledu tamějších definic jednotlivých úrovní testování) spolu s možnostmi pro případné využití automatizace nebo naopak důvody proč automatizace vhodná není.

Fáze testovacího procesu	Možnost využití HP QTP	Vhodné pro automatizaci				
	V této fázi si programátoři provádí testy samostatně a zpravidla jde					
Testování	o jednoduché testy funkcionality ovlivněné změnami v aktuálním					
programátorem	rogramátorem releasu. Testy se provádí v malém objemu a prostor pro					
	automatizaci zde není.					
	V této fázi již dochází k vytváření většího balíku testů, ale vzhledem					
	k tomu, že se testuje každý systém a aplikace samostatně, tak jde					
Systémové testy	většinou o jednoduché a často jednokrokové testy. Navíc se	NE				
	systémové testy generují opakovaně pro každý release nové a					
	automatizace se zde tedy nevyplatí.					

Tabulka 3 - vhodnost použití HP QT	v jednotlivých fázích	testovacího procesu	na řešeném projektu
------------------------------------	-----------------------	---------------------	---------------------

Integrační testy	Z pohledu počtu a náročnosti testů se v tomto případě využití automatizace nabízí. Ovšem vzhledem k tomu, že se testy, stejně jako v případě testů systémových, mění s každým releasem, využití automatizace se zde opět nevyplatí a pravděpodobně ani není možné, vzhledem k náročnosti a množství integračních scénářů, které by se zřejmě v mezidobí mezi releasy ani nestihlo připravit.	NE
Regresní testy	Regresní testy tvoří ustálený balík testů ověřující klíčovou funkcionalitu systému. V některých případech jsou převzaté z integračních testů, ve kterých byla klíčová funkcionalita nasazována. Splňují tak vhodnost pro zavedení automatizace hned z několika kritérií. Testy se provádějí v pravidelných intervalech, zaměření testů se v čase mění jen nepatrně, a testy jsou dostatečně náročné i komplikované, aby se automatizace vyplatila. Navíc je pro ně nutná relativně časově náročná příprava dat, která se dá také snadno automatizovat. V tomto případě je tak automatizace zcela na místě.	ANO
Další případy	Přesto, že z výše uvedených testovacích fází se automatizace vyplatí pouze při regresních testech, je potřeba také zmínit možnost automatizace při přípravě dat, kdy je potřeba zakládat více stejných druhů objednávek, zákazníků nebo konfigurací produktů, které jsou předpokladem pro další testování. Ta je často nutná i mimo regresní testy a je možné přípravu automatizovat i v případě testů systémových a integračních. I když příprava dat často není složitá, tak je vzhledem k množství dat časově náročná a zbytečně vytěžuje testery, kteří by se mohli věnovat důležitějším činnostem a je proto dobré dle uvážení automatizovat i přípravu dat, tam kde je jasné, že se časová úspora vyplatí.	ANO

4.6 Shrnutí

Projekt, na kterém pracuji, je relativně velký a hlavně kvůli objemu dodávek bývá často problém dodržet termíny. Každá úspora času a zdrojů je proto velkým přínosem a pomáhá projektu efektivněji plnit dodávky alokací zdrojů na nejkritičtější místa. Automatizace, která může tyto zdroje ušetřit, byla proto spíše otázkou času.

Pro využití HP QTP na analyzovaném projektu mluví hned několik faktorů. Tím nejdůležitějším je fakt, že realizace automatizace má na tomto projektu smysl a v testovacím projektu jsou fáze, kde se vyplatí a dokáže ušetřit čas i testovací kapacity. Dalším důvodem, když už víme, že automatizace smysl má, je jednoznačně fakt, že licence na nástroj HP QTP byla zakoupena již dříve a aktuálně není využívána.

Dle výše uvedených poznatků by se tak plánovaná automatizace měla zaměřit především na dva cíle. Tím prvním jsou regresní testy, které se vyplatí automatizovat a takto vytvořený balík automatizovaných testů dlouhodobě udržovat, tak aby bylo kdykoliv v průběhu releasu možné regresně otestovat v krátkém čase původní funkčnost bez zbytečného vytěžování testerů.

Druhým cílem je zaměření automatizace na přípravu dat. Ta hraje velký význam při regresních testech, ale také při testech integračních a systémových. U větších projektů je často nutné splnit výchozí podmínky pro otestování nové funkčnosti a připravit velké množství dat. V takovém případě přináší automatizace velkou časovou úsporu a umožňuje testery alokovat na důležitější úkoly. Pokud připravená data dojdou kvůli velkému množství retestů, což je relativně častý jev, lze díky již automatizovanému skriptu v podstatě okamžitě poskytnout data nová.

5 Příručka k základní funkcionalitě HP QuickTest Professional

Cílem této kapitoly je vytvoření příručky, která by testerům, kteří se chtějí seznámit s nástrojem HP QuickTest Professional, posloužila jako přehled základních funkcí QTP a usnadnila jim výuku s tímto nástrojem.

Pro demonstraci funkcí je využita plná licence verze HP QTP 11.0 vydaná v roce 2010.

5.1 Obsah příručky

V příručce je popsána většina základních funkcí HP QTP, které jsou předvedeny na přiložené aplikaci pro rezervaci letenek, která je popsána níže. Obsah příručky je vytvořen tak, aby po jejím pročtení a vyzkoušení popisovaných funkcí mohl tester vytvářet testy. Popsat komplexně všechny funkce a možnosti HP QTP není v rozsahu diplomové práce možné, snažil jsem se proto vybrat funkce tak, aby tento výběr představoval efektivní paletu funkcí pro tvorbu testů a poskytl vstupní znalosti pro další rozvoj uživateli, který má zájem se s QTP naučit pracovat.

Obsah příručky je rozdělen do následujících kapitol odpovídajících konkrétní funkcionalitě:

- 1. Úvod do výkladu funkcí
- 2. Spuštění HP QTP
- 3. Nahrání prvního testu
- 4. Spuštění testu a analýza výsledků
- 5. Parametrizace
- 6. Vkládání checkpointů⁵ (kontrolních bodů)
- 7. Ukládání výstupních hodnot
- 8. Podmínky

⁵ Kontrolní body v průběhu testu – vzhledem k označení v aplikaci v HP QTP budu dále používat označení checkpoint.

- 9. Reportování událostí pomocí metody Reporter. ReportEvent
- 10. Akce
- 11. Object repository (Objektový repositář) (objektový repositář)
- 12. Typy nahrávání v QTP
- 13. Funkce
- 14. Deskriptivní programovaní
- 15. Doporučení

5.2 Úvod do výkladu funkcí

Pro potřeby výkladu funkcí QTP je využita aplikace *Flight reservation*, což je jednoduchá aplikace simulující rezervaci letenek, která je dodávána jakou součást softwarového balíku HP QTP. Nachází se ve *Start Menu\Programs\HP QuickTest Professional\Sample Applications*.

Jako login⁶ do aplikace slouží jakýkoliv řetězec znaků delší než 3 znaky a jako heslo je defaultně nastaveno "mercury" (nerozlišuje malá a velká písmena).

Aplikace vytváří rezervace letenek s parametry datum, start, cíl, jméno zákazníka atd. Po potvrzení vytvoří objednávku s novým číslem, kterou lze později upravit nebo smazat.

V menu Analysis lze vytvářet reporty a grafy na základě vytvořených objednávek.

⁶ Přejímám obecně používané označení pro přihlašovací jméno do aplikace.

😹 Flight Reservation		×
File Edit Analysis Help		
🖹 達 🗙 🂵 📰 ?		
Flight Schedule:		
Date of Flight: Fly From: Fly To:	Fights	
Corder Information:		
Flight No: Departure Time: Arrival Time:	e Airline:	
Name:		
	Tickets:	A
Class:	Price:	
O First O Business O Economy	Total:	
Update Order Delete Order	Insert Order	
	Order No:	

Obrázek 6 - základní obrazovka aplikace Flight reservation, zdroj: autor z aplikace Flight Reservation

5.3 Spuštění HP QTP

QTP se spouští z následujícího umístění, kam se instaluje defaultně: *Start Menu\Programs\HP QuickTest Professional*.

Po startu QTP se objeví menu s volbou pluginů pro QTP. Doporučuje se zapínat pouze pluginy, které jsou potřeba pro konkrétní test, protože hrozí, že spolu budou pluginy kolidovat a snižovat tak úroveň identifikace objektů v testu. (Pokud se výběr pluginů neobjeví, může se vyvolat v menu QTP *File/Settings/Properties/Associated add-ins*). Pro potřeby výuky je dostačující volba pluginu Visual basic, vzhledem k tomu, že aplikace Flight reservation, je napsaná právě v jazyce Visual basic.

🚾 Quick Test Professional	- Add-in Manager		×
QuickTest Professional	Add-in Manager Select add-ins to loa Add-in PowerBuil SAP Siebel Stingray Terminal Visual Basic Visual Age Web Web Web Web Web Tervides support for Tip: To maximize per identification reliabilit need.	d: Licensed Licensed Licensed Licensed Licensed Licensed Licensed Licensed Licensed Licensed Licensed	ic objects.
Show on startup		Cancel	Help
I▼ Show on startup			neip

Obrázek 7 - volba pluginů v QTP, zdroj: autor z nástroje HP QTP

Možnost zobrazení při spuštění se nechává zaškrtnutá, aby bylo možné pokaždé vybrat s jakými pluginy se bude pracovat. Volba se potvrzuje tlačítkem *OK* a nastavení pluginů se ukládá jako výchozí pro příští spuštění.

Po potvrzení následuje spuštění QTP s úvodní obrazovkou (Start page), zde se nachází:

- na pravé straně výpis změn aktuální verze
- levá strana úvodní obrazovky (viz Obrázek 8 níže) nabízí průvodce procesy zahrnující best practice⁷ popis procesů používaných v QTP. Poslední použité soubory (testy, knihovny a další soubory používané v QTP) a zkratkové tlačítka pro otevření nebo založení nového testu a dalších komponent QTP.
- zobrazování úvodní obrazovky lze zakázat zaškrtnutím checkboxu⁸ Don't show the start page on startup.

Start Page 🔬 Test
QuickTest Professional
Welcome !
Welcome to HP QuickTest Professional, the advanced solution for functional test and regression test automation. This next-generation automated testing solution deploys the concept of keyword-driven testing to enhance test creation and maintenance.
QuickTest Professional meets the needs of both technical and non-technical users, empowering the entire testing team to create sophisticated test suites.
Process Guidance List
Keyword-Driven Testing
<u>Application Areas</u>
Business Components
Descetty Used Files
- Cul Reserver Files HD QuickTest Resfersional Tests Test skaleni
<u>Cryprogram Priest PPO(Crypton Protestional) Tests (Fest Skoleni</u>
C:\Program Files\HP\QuickTest Professional\Tests\skoleni library DB_Excel.qti
<u>C:\Program Files\HP\QuickTest Professional\Tests\skoleni_library2.qtl</u>
Submit your feedback about QuickTest on the <u>Send Feedback and Win!</u> Web page to become eligible to win prizes in special prize drawings.

🗖 Don't show the Start Page window on startup

Obrázek 8 - startovní obrazovka QTP, zdroj: autor z nástroje HP QTP

⁷ Obecně uznávané nejlepší postupy.

⁸ Okénko pro zaškrtávací volbu v aplikaci.

Startovní obrazovka může posloužit jako rozcestník pro rychlé spuštění, pro potřeby výuky ale není nutné ji využívat, a tak se může odškrtnout volba zobrazovat *Start page* po spuštění QTP. V další části následuje popis hlavního okna QTP (viz Obrázek 9).

- Titulek okna zobrazuje jméno aktivního dokumentu. Pokud v dokumentu došlo ke změně oproti poslednímu uložení, pak je jméno dokumentu zobrazováno s hvězdičkou.
- Panel hlavního menu.
- Panel nástrojů.
- Keyword view graficky uspořádaná reprezentace testu pomocí klíčových slov. Jednotlivé kroky v testu jsou uspořádány chronologicky do tabulky a u každé akce, která se v daném kroku provádí lze měnit zadávané hodnoty. Ideální pro neprogramátory.
- Expert view průběh testu je zaznamenán pomocí VB Skriptu. Pro každý řádek skriptu v Expert view existuje korespondující řádek v Keyword view.
- Active Screen poskytuje náhled na aplikaci v konkrétním označeném kroku testu včetně zvýraznění komponent (tlačítek či polí), se kterými tento krok manipuluje.
- Data Table tabulka sloužící jako zdroj dat, jež lze vkládat do testu nebo naopak cíl zápisu výstupů z testu.
- Test Flow hierarchicky zobrazuje uspořádání akcí, tak jak jsou v průběhu testu volány.

🐨 QuickTest Professional - [I	C:\Program Files\HF	P\QuickTest Pr	ofessional\Tes	s\MMA\flight_l	ogin*]							
: Jie Edit View Inse	rt <u>A</u> utomation <u>R</u> e	sources <u>D</u> ebu	ıg <u>T</u> ools <u>W</u> ir	ndow <u>H</u> elp								- 8×
🔋 🛐 New 🕶 🛵 Open 🕶 🔚	🗊 😽 🍏 l 🔏 🗈	- C. F 🗊	Q	Q Q 44 A	: 🕞 🥹	r) III ()	83 🛱 📮					
🕴 🙆 Record 🕨 Run 👻 🔳 S	top 🗃 🔭 📰	e .			i 🚰 • 🐲	· 🚀 🛪	la ≙l ₌ i:	i 🥔 🐁	-			
Test Flow 🔻 🕈 🗙	Test Flow 🗸 🗛 🚮 flight Jogin*											
Flight_login	Action2		▼ ← B	ack 🔍 Show								
~	Item		Operation	Value			Documentati	on				
	🔫 🚒 Action2											
	🗐 🗐 💭 💭	mUtil	Run	"C:\Program	Files\HP\Quic	kTest Profe	Open the "C:	\Program File	es\HP\QuickT	lest Professio	nal\samples\fl	light\app
	🕨 🕞 Login	1										
	Flight	Reservation	Close				Close the "Fli	ght Reserval	tion" window.			
	Ken	word View	vnert View /									
	Data Table											▼ ‡ ×
	A1											
	A	В	C	D E	F	G	Н	I	J	K	L	<u>M</u>
	2	-8										
	3											
	4											
	5											
	6											
	7											-
	Global 🗸	Action2 /					•					
	🛱 To Do 📔 Info	rmation 🗐 🎒 Ad	tive Screen 🏢	Data Table								
											Re	ady //

Obrázek 9 - hlavní okno QTP, zdroj: autor z nástroje HP QTP

5.4 Nahrání prvního testu

Pro nahrání prvního testu je použit jednoduchý testovací scénář zalogování do aplikace pro rezervaci letenek Flight reservation.

Úkolem je se úspěšně zalogovat do aplikace za pomoci validního přihlašovacího jména a hesla a poté aplikaci vypnout.

Test tedy bude obsahovat následující kroky.

- 1. Otevřít aplikaci Flight reservation
- 2. Zadat validní přihlašovací jméno
- 3. Zadat validní heslo
- 4. Stisknout tlačítko OK
- 5. Zavřít aplikaci

Prvním krokem je stisknutí tlačítka *Record*. Poté se objeví nastavení nahrávání pro všechny povolené pluginy. Pro začátek jsou ponechána defaultní nastavení u všech pluginů, tedy volba *Record on any*. Později je doporučeno pro jednotlivé pluginy nastavit volbu *Record & Run on a specific window*. Nastavení lze v případě potřeby znovu vyvolat z menu *Automation**Record & run settings*.

Record and Run Settings	×					
Oracle Web Siebel Windows Applications						
Record and run test on any open Oracle application						
Open the following application when a record or run session begins						
Address:						
V						
🗖 Auto-login						
User name:						
Password:						
Log out of the application when the test closes						
Close the browser when the test closes						
Note: You can also use environment variables to set the Record and Run Settings. Click Help for more information.						
OK Cancel Apply Help						

Obrázek 10 - nastavení nahrávání, zdroj: autor z nástroje HP QTP

Po potvrzení začíná QTP nahrávat, což lze ověřit blikajícím textem *Recording* v pravém dolním rohu aplikace. Následně jsou prováděny požadované kroky testu, tzn.:

- 1. Spuštění aplikace na rezervaci letenek z cesty Start Menu\Programs\HP QuickTest Professional\Sample Applications
- 2. Zadání přihlašovacího jména (libovolné a delší než 4 znaky)
- 3. Zadání hesla ("mercury" defaultně nastavené v QTP)
- 4. Kliknutí na tlačítko OK
- 5. Po úspěšném zalogování do aplikace kliknutí na tlačítko Zavřít

Po zadání všech kroků se nahrávání ukončí kliknutím na tlačítko *Stop* opět v QTP. Výsledný skript vypadá následovně:

- 1. Window("Window").Window("Start Menu").WinButton("All
 Programs").Click
- 2. SystemUtil.Run "C:\Program Files\HP\QuickTest
 Professional\samples\flight\app\flight4a.exe","","C:\Program
 Files\HP\QuickTest Professional\samples\flight\app\","open"
- 3. Dialog("Login").WinEdit("Agent Name:").Set "Jmeno"
- 4. Dialog("Login").WinEdit("Agent Name:").Type micTab
- 5. Dialog("Login").WinEdit("Password:").SetSecure
 "532a16e3f375f0323f0967c1ab5085abe5bc552f"
- 6. Dialog("Login").WinButton("OK").Click
- 7. Window("Flight Reservation").Close

První krok označuje kliknutí ve *Start menu* systému *Windows*, kde se spouští aplikace na rezervaci letenek. Tento krok lze odstranit, protože QTP je schopné přímo identifikovat a spustit aplikaci v kroku číslo 2.

Stejně tak lze odstranit krok číslo 4, který zaznamená stisk tabulátoru a přesun do dalšího pole, vhledem k tomu, že QTP dokáže zadávat hodnoty do polí na základě jejich identifikace, aniž by bylo nutné jejich označení.

Pátý krok představuje zadání hesla, které QTP automaticky zakryptuje. Pro potřeby výuky je lepší, když je hodnota vidět, je tedy upravena metoda zadávání ze *SetSecure* na jednoduchý *Set*. Finální skript tedy vypadá následovně:

- 1. SystemUtil.Run "C:\Program Files\HP\QuickTest
 Professional\samples\flight\app\flight4a.exe","","C:\Program
 Files\HP\QuickTest Professional\samples\flight\app\","open"
- 2. Dialog("Login").WinEdit("Agent Name:").Set "Jmeno"

- 3. Dialog("Login").WinEdit("Password:").Set "mercury"
- 4. Dialog("Login").WinButton("OK").Click
- 5. Window("Flight Reservation").Close



Obrázek 11 - test pro zalogování po úpravě, zdroj: autor z nástroje HP QTP

V této podobě spuštěný skript provádí všechny požadované kroky a úspěšně se dokončí.

5.5 Spuštění testu a analýza výsledků

Spuštění hotového testu se provádí stiskem tlačítka Run na nástrojovém panelu Automation.



Obrázek 12 - spuštění testu, tlačítko Run, zdroj: autor z nástroje HP QTP

Po spuštění testu se objeví dialogové okno s výběrem cílového adresáře, kam se ukládají výsledky testu. Jsou dvě možnosti volby buď dočasné uložení v *Temporary files* nebo uživatelem definovaný adresář.

Pro zkoušku stačí uložení do dočasné složky. Po potvrzení tlačítkem OK se již spustí test.

Run 🛛
Results Location Input Parameters
Write run results to:
C New run results folder
• Temporary run results folder (overwrites any existing temporary results)
C:\DOCUME~1\MICHAL~1.MAL\LOCALS~1\Temp\TempR
OK Cancel Help



V probíhajícím testu lze sledovat, který krok právě probíhá prostřednictvím žluté šipky, která průběh testu sleduje.

🔊 fli	ght_login
	Action2
1: 2: 3: 4: 5: 6:	SystemUtil.Run "C\Program Files\HP\QuickTest Professional\samples\flight\app\flight4a.exe","","C: Dialog("Login").WinEdit("Agent Name:").Set "Michal" Dialog("Login").WinEdit("Password:").SetSecure "5316001fde242a3509c355dde651c27151b429a8" Dialog("Login").WinButton("OK").Click ♥Window("Flight Reservation").Close

Obrázek 14 - šipka označující právě probíhající krok v testu, zdroj: autor z nástroje HP QTP

Po ukončení testu jsou prostřednictvím aplikace *HP Run Result Viewer*, zobrazeny výsledky průběhu testu.

🕌 HP Run Results Viewer							
File View Tools Help							
i 🗁 📼 🝸 🦍 🕼 🧟 🕼 🛛 🗢 🛛 🔚 🛛 🎅							
flight_login - TempResults_10 🛛 👻 🖟 🗙	Result Details						▲ İ ×
Search for:	Executive Summ	nary - flight_login - TempResult	ts_10 ✓ Passed				
Test flight_login Summary Section 1 (Row 1)	Test name:	flight_login		Product name:	QuickTest Professional		
Action2 Summary	Results name:	TempResults_10		Product version:	11.00		
Systemutii Systemutii Run "C:\Program Files\HP\QuickTest Profe	Time zone:	Central Europe Standard Time		Host name:	QTP00		
⊿ <mark>≣</mark> Login	Run started:	3/10/2014 - 11:11:17		Operating system:	Windows XP		
Agent Name:.SetText	Run ended:	3/10/2014 - 11:11:31					
OK.Click Iflight Reservation	Total time:	00:00:14					
Flight Reservation.Close	Statistics					Passed 📕 Failed	📕 Warning 📕 Done
		Threading	Current Run ∢ Passed	Chana		Pre flight_login -	vious Run Passed TempResults_11
		rterations		Steps		Iterations	Steps
		0 Failed		0 Failed		0 Failed 0 Warnings	0 Failed 0 Warnings
		0 Warnings		0 Warnings		1 Passed	1 Passed
		1 Passed		1 Passed			Open

Obrázek 15 - výsledek dokončeného testu zobrazený v aplikaci *HP Run Results Viewer,* zdroj: autor z nástroje HP QTP

Na levé straně je panel, který hierarchicky znázorňuje průběh testu podobně jako v Expert view. Zde je možné procházet jednotlivé kroky a kontrolovat, zda proběhly bez problému a vykonaly požadované akce s příslušnými parametry.

Na pravé straně jsou zobrazeny detaily výsledku testu a to buď celého testu, nebo jednotlivých kroků, podle toho co je aktuálně vybráno ve stromu akcí v levém panelu.

flight_login - TempResults_10 📃 💌 A 🗙	Result Details			
Search for:	Step Name: Agent Name:.SetTe	xt		
A Cubic Summary A GasternUtil Gas	Object	Details	Result	Time
Agent Name.SetText The Reservation General Content of the Reservation	Agent Name: SetText	"Michal"	Done	3/10/2014-11:11:22
Flight Reservation.Close				

Obrázek 16 - detail výsledku konkrétního kroku v testu, zdroj: autor z nástroje HP QTP

Do výsledku lze zahrnout i screenshoty⁹, případně videa z průběhu testu v případě, že je nastaveno v menu *Tools/Options/Run/Screen* capture snímání obrazovky nebo nahrávání videa z průběhu testu. Ve výsledcích se pak mnohem lépe analyzují chyby, za pomocí přiložených screenshotů či videa, které se pak zobrazují v dalším panelu ve výsledcích.

⁹ Snímky aktuálního zobrazení na monitoru.

21 UD D D							
Re New Task Value							
Hie view Tools Help							
i 🔚 👦 🔽 🕼 🗑 🔕 🕼 < 🗢 🗄 🛛 ? 🅛							
flight_login - TempResults_10 🔹 🕸 🗙	Result Details				- † ×	Captured Data	- † X
Search for: P × V					<u> </u>		4
🖌 🖌 🎇 Test flight_login Summary	Step Name	: OK.Click				Login	
 Fight_login Iteration 1 (Row 1) A Action2 Summary 	Step Done				_	Agent Name:	
⊳ 🛹 🚺 SystemUtil ⊿ 🔢 Login	Object	Details	Result	Time		Michal	
Agent Name:.SetText	OK.Click		Done	3/10/2014-13:13:23		Password:	
Flight Reservation							
						OK Cancel Help	

Obrázek 17 - zobrazení screenshotu konkrétní akce provedené v označeném kroku testu, zdroj: autor z nástroje HP QTP

Nastavení snímání je však kvůli výkonnosti lepší ponechat na nastavení snímání pouze v případě chyb.

Options	×
 General Text Recognition Folders Active Screen Run Screen Capture Windows Applications Advanced Web Page/Frame Options Advanced 	Run > Screen Captures Save still image captures to results: Save movie to results: Save movie to results: Save movie to results: Save movie of entire run Select the options you want to apply to all Screen Recorder movie clips. These options affect the movie file size and appearance. Record sound Set plain wallpaper Do not show window contents when dragging windows Capture Driver The Screen Recorder Capture Driver cannot be installed or uninstalled when working with QuickTest via a remote desktop connection. For more information, click Help.
	OK Cancel Apply Help

Obrázek 18 - nastavení snímání obrazovky a videa během testu, zdroj: autor z nástroje HP QTP

Výsledky je dále možné exportovat či tisknout v různých formátech HTML, Word či PDF. Samozřejmostí je i export do *HP Quality Center*¹⁰.

5.6 Parametrizace

V následující části se popisuje, jakým způsobem lze v QTP parametrizovat. Využití parametrů zjednodušuje testy a šetří čas a úsilí při jejich vytváření. Díky parametrizaci lze jednoduše spustit jeden test pro různá vstupní data.

Parametrizaci lze vyzkoušet na prvním testu, kde jsme se logovali do aplikace pro rezervaci letenek, pouze zde přidáme nový požadavek, aby test proběhnul pro 3 varianty loginu a hesel.

Nejjednodušším způsobem, jak v tomto případě nastavit heslo a login jako parametr je přepnout na Keyword view, jít na příslušný řádek (zde "Agent Name" a "Password") a ve sloupci *Value* kliknout na tlačítko *Configure the value* (viz Obrázek 19).

\$	flight_login*			
	Retion2	Gack Chow		
	Item	Operation	Value	Documentation
	👻 🎇 Action2			
	🖳 📮 SystemUtil	Run	"C:\Program Files\HP\QuickTest Profe	Open the "C:\Program F
	👻 🔚 Login			
	Inst Agent Name:	Set	"tester"	Enter "tester" in the "Ag
	- FARE Password:	Set	Text ^{ury"}	Configure the value e "
	<u></u> OK	Click		<ctrl+f11> n.</ctrl+f11>
	Flight Reservation	Close		Close the "Flight Reserv

Obrázek 19 - nahrazení nastavované hodnoty objektu parametrem, zdroj: autor z nástroje HP QTP

Po kliknutí se otevře dialogové okno pro výběr zdroje dat pro parametr. Zde se klikne na radiobutton¹¹ *Parameter*. Jako zdroj se nastaví "DataTable" a poté se pojmenuje sloupec, do kterého se v *Data Table* budou ukládat vstupní data pro tento parametr.

¹⁰ Nástroj pro test management od společnosti Hewlett-Packard

¹¹ Tlačítko pro kroužkovací volbu v aplikaci.

¥alue Cor	nfigurat	ion Opt	ions			×
C Cons	tant	tester				
Paran	neter	DataTa	ole		<u>-</u>]
Name	e: Logir			•		
⊢ Loo	ation in l	a Tab	le			
	Global s	heet				
0	Current	action sh	neet (loca	Ŋ		
<u> </u>						
	OK		Cancel		Help	



Stejným způsobem se nastaví i parametr pro zadání hesla a nakonec se zadají vstupní data do *Data Table*.

Data Tab	le		43		
A	.1 🔤	ester			
	Login	Heslo	С	D	
1	Tester	mercury			
2	Tester99	Mercury			
3	9999	MERCURY			
4					
5					
√ Î∙	Global 📈	Action2 /			
To D	o 🛛 🛅 Infor	mation 🖂 🎢 Ad	tive Screen	🏭 Data Tab	le



Takto upravený test se již může spustit. Ten následně probíhá ve třech iteracích pro každý řádek z *Data Table* a všechny proběhnou v pořádku, protože byly splněny všechny podmínky pro login a heslo (zde je login jakýkoli řetězec delší než čtyři znaky a heslo "mercury", přičemž se nerozlišují velká a malá písmena).

5.7 Vkládání checkpointů

Checkpointy slouží k porovnání aktuálního průběhu testu s očekávanými výsledky. Navazují se na konkrétní objekty a porovnávají aktuální zobrazované hodnoty s těmi, jaké jsou zadány v checkpointu. Lze například kontrolovat přítomnost tlačítek, hodnoty v polích nebo šířku aktivního okna. Po porovnání hodnot se kontrola checkpointu zapíše do výsledku jako krok, který má buď hodnotu "passed" a pokud proběhne v pořádku nebo "failed", pokud neproběhne podle očekávání. Checkpointy jsou tak výborným nástrojem pro kontrolu jednotlivých kroků v průběhu testu.

Pro ukázku fungování checkpointu lze použít předchozí test, kde se kontroluje, zda se zobrazilo hlavní okno aplikace pro rezervaci letenek. Pro přidání stačí pravým tlačítkem kliknout na řádek v testu, kde se již pracuje s objektem, který představuje hlavní okno aplikace pro rezervaci letenek. Potom se v menu vybere položka *Insert standard checkpoint*.

1:	SystemUtil.Run "C:\Program Files\HP\QuickTest Professional\samples\flight\app\flight4a.exe"						
2. 3:	Dialog("Login").WinEdit("Password:").Set Data Table("Heslo", dtGlobalSheet) Dialog("Login").WinEdit("Password:").Set Data Table("Heslo", dtGlobalSheet)						
4:	Dialog("Login").WinButton("OK").Click Window("Elight Becometion").Close						
5: 6:	window("Flight Reservation"). Close		Object Properties				
		1	Insert Standard Checkpoint				
			Insert <u>O</u> utput Value ^{VS}				
			Insert <u>S</u> tep				
			Go to Fu <u>n</u> ction Definition				
			Action				
			Insert/Remo <u>v</u> e Breakpoint				
		<i>3</i>	Enable/Disable <u>B</u> reakpoint				
		M	Run <u>f</u> rom Step				
			Debu <u>a</u> from Step				
			<u>R</u> un to Step				
			Add to <u>W</u> atch				
		χ	Cu <u>t</u>				
		Ep.	<u>С</u> ору				
		62	<u>P</u> aste				
		\times	<u>D</u> elete				
H 4 🕨	Keyword View Expert View		Select A <u>l</u> l				
)ata Table			Co <u>m</u> ment Block				
A1	Tester	Ŕ	Uncomm <u>e</u> nt Block				

Obrázek 22 - vložení checkpointu, zdroj: autor z nástroje HP QTP

Následně se již otevře okno s vlastnostmi checkpointu, který je navázaný na objekt, z jehož řádku se checkpoint vytvářel (v tomto případě hlavní okno rezervace letenek). Nastavení chekpointu se může ponechat ve výchozích hodnotách, protože už tak se bude ověřovat, jestli je okno aplikace dostupné, zdá má správný popisek a zda má nastavenou výchozí výšku a šířku okna, což pro potvrzení, že se okno otevřelo, bohatě postačuje.

* *	The	kpoin	t Properties		x
Na	ame:	Fligh	nt Reservation		
Cl	ass:	Wind	wot		
	_				
		Туре	Property	Value 🔺	
		ABC	enabled	True	
		ABC	focused	False	
		ABC	height	461	
		ABC	text	Flight Reservation	
		ABC	width	609	
	Ē	nsc.		A1C	
	- Co	infigure	value		
	6	Consi	tant True	T	
			uni pride		
	0) Parar	neter		
		Data	Table("Flight Becervatio	n enabled" dtGlobalS	
		In are	n abiel i light_neservatio		
Cł	heck	point tin	neout: 10 seconds		
0.			-		
In	sert :	stateme	nt: 💿 Before current s	tep 🔘 After current step	
				Cancel Help	1

Obrázek 23 - nastavení vlastností checkpointu, zdroj: autor z nástroje HP QTP

Po potvrzení checkpointu a spuštění testu se do výsledků testu přidá jeden krok s hodnotou "Passed" pro každou iteraci testu

5.8 Ukládání výstupních hodnot

Uložení výstupních hodnot je krok, během kterého je jedna nebo více hodnot zaznamenána a uložena na dobu trvání průběhu aktuální běhu testu.

Tyto hodnoty mohou být použity jako vstupní data pro další navazující testy

Pro demonstraci ukládání jsou přepoužity první čtyři kroky předchozího testu. K těmto krokům se přes tlačítko *Record* přidá zobrazení chybové zprávy, uložení zprávy, uzavření okna se zprávou a logovacího okna.

Pro nahrání těchto kroků je potřeba zapnout aplikaci Flight reservation, poté v QTP stisknout *Record* pro nahrávání a přihlásit se bez přihlašovacích údajů. Poté se objeví chybová

zpráva, zde je potřeba kliknout na tlačítko *OK*, následně na *Cancel* opět v aplikaci. Poté se ukončí nahrávání a odstraní se zbytečné (aktivace okna) nebo duplicitní řádky, které při něm vznikly.

Zbývá přidat krok pro uložení chybové zprávy. Označí se pátý řádek a přepne se na záložku *Active screen*, kde by se mělo zobrazit dialogové okno s chybovou zprávou. V dialogovém okně se klikne pravým tlačítkem myši na text chybové zprávy a z menu se vybere možnost "Insert output value".

Active Screen	
Flight Reservations	×
Please enter agent na	me
ок	Insert Standard <u>⊂</u> heckpoint
	View / <u>A</u> dd Object
To Do 🖬 Information 🚟 Ac	Step Generator Insert Bitmap Checkpoint Insert Text Checkpoint

Obrázek 24 - vložení výstupní hodnoty z okna "Active screen", zdroj: autor z nástroje HP QTP

Potvrdí se výběr vlastnosti objektu.

Object Selection - Output Value Properties				
The location you clicked is associated with several objects. Select the required object from the tree below.				
Dialog : Flight Reservations Static : Please enter agent name				
OK Cancel Help				

Obrázek 25 - výběr vlastnosti objektu pro výstup, zdroj: autor z nástroje HP QTP

Po potvrzení se objeví vlastnosti výstupní hodnoty, kde se zaškrtne hodnota, která se má do výstupu uložit, v tomto případě text chybové zprávy. V poli "Name" lze ještě výstupní hodnotu pojmenovat a toto pojmenování pak případně upravit kliknutím na tlačítko "Modify" opět v poli "Name". Pojmenování se poté přenese do jména sloupce v *Data table*, kam se výstupní hodnoty budou ukládat.

📲 Outpu	it Value Proj	perties		×
Name:	Chybové zp	rávy		
Class:	Static			
	[upo	Property	Malua	
	i ype I enabler		Тпе	
		4	Falce	
	eight		15	
	text		<please agent<="" enter="" td=""><td>na</td></please>	na
Ë,	ABC width		122	
			CR	
Con	figure value—			
Outj Outj Loc	put Type: put Name: ation:	DataTable Please_enter_a Global Sheet	agent_nan Modify	
◄			▼ ▶	
Insert st	atement: 💽	Before current ste	epi i O After current step	2
		<u> </u>	Cancel	Help

Obrázek 26 - vlastnosti výstupní hodnoty, zdroj: autor z nástroje HP QTP

Po vložení tohoto kroku je test již kompletní a vypadá takto:

🔬 flighl	t_login_negativ*
1	Action2
1:	SystemUtil.Run "C\Program Files\HP\QuickTest Professional\samples\flight\app\flight\a.exe","","C\Program Files\HP\
2:	Dialog("Login").WinEdit("Agent Name:").Set DataTable("Login", dtGlobalSheet)
3:	Dialog("Login").WinEdit("Password:").Set DataTable("Heslo", dtGlobalSheet)
4:	Dialog("Login").WinButton("OK").Click
5:	Dialog("Login").Dialog("Flight Reservations").Static("Please enter agent name").Output CheckPoint("Chybova zprava")
6:	Dialog("Login").Dialog("Flight Reservations").WinButton("OK").Click
7:	Dialog("Login").WinButton("Cancel").Click
8:	

Obrázek 27 - skript pro ukládání výstupních hodnot, zdroj: autor z nástroje HP QTP

Pro vyzkoušení ukládání chybových zpráv je potřeba vyplnit *Data Table* nevyhovujícími loginy a hesly, tak aby nesplňovali některou z podmínek:

- Login musí být libovolný řetězec delší než tři znaky.
- Heslo je Mercury, přičemž se nerozlišují velká a malá písmena. I zde platí podmínka, že zadané heslo musí být delší než 3 znaky.

Loginy se mohou vyplnit například následovně:

	Login	Heslo	Chybova	_zprava		
1	Tester					
2	Tes	Mercury				
3	Tester	asdf				
4	Tester	Mer				
5		Mercury				
6						
7						
. 8						
Global / Action2 /						
🛐 To D	📴 To Do 🛛 🚺 Information 🛛 📆 Active Screen 🛛 🎹 Data Table					

Obrázek 28 - vzorové vyplnění vstupních hodnot v *Data Table* pro potřeby testu na ukládání výstupních hodnot, zdroj: autor z nástroje HP QTP

Po spuštění testu je ve výsledcích vidět, že se do Data table zapsaly všechny chybové

zprávy.

Data	Table		▼ ‡
	Login	Heslo	Chybova_zprava
1	Tester		Please enter password
2	Tes	Mercury	Agent name must be at least 4 characters long.
3	Tester	asdf	Incorrect password. Please try again
4	Tester	Mer	Password must be at least 4 characters long
5		Mercury	Please enter agent name

Obrázek 29 - zapsání výstupních hodnot do výsledků testu

5.9 Podmínky

Podmínky mají typickou konstrukci *If-Then-Else*. Podmínka se uzavírá zápisem "End If". Součástí "If" podmínky musí být i část "Else", která určuje, co se má provést při nesplnění podmínky.

Z předchozích testů jsou již k dispozici pokryté scénáře úspěšného a neúspěšného zalogování. Aby byl test komplexní, je dobré tyto dva scénáře integrovat do jednoho. Pro toto spojení se používá podmínková funkce "If", která test rozděluje na negativní nebo pozitivní větev na základě toho, zda se objeví nebo neobjeví chybová zpráva po zadání loginu.

První 4 kroky v testu jsou ponechány stejné (tedy spuštění aplikace, zadání loginu a hesla a potvrzení tlačítkem OK). Za tyto kroky se již zařadí "If" funkce, kde v případě splnění podmínky, že se zobrazí dialogové okno s chybovou zprávou, bude scénář pokračovat negativní větví a uloží chybovou zprávu do výstupních hodnot. Pokud podmínka splněna nebude, proběhne pozitivní varianta, kdy scénář prostřednictvím přidaného checkpointu ověří, že se uživatel úspěšně zalogoval do aplikace.

Skript pro zapsání podmínky je popsán níže. Pro ověření, zda se zobrazilo okno s chybovou zprávou, se použije metoda "Exist".

- 1. If Dialog("Login").Dialog("Flight Reservations").Exist Then
- 2. Dialog("Login").Dialog("Flight Reservations").Static("Please enter agent name").Output CheckPoint("Chybova zprava")
- 3. Dialog("Login").Dialog("Flight
 Reservations").WinButton("OK").Click
- 4. Dialog("Login").WinButton("Cancel").Click
- 5. Else
- 6. Window("Flight Reservation").Check CheckPoint("Flight
 Reservation")
- 7. Window("Flight Reservation").Close

8. End If

S ukončením podmínky zároveň končí i samotný scénář a jeho výsledná podoba je následující (viz Obrázek 30). Pokud se do *Data table* zadají pro login a heslo různé kombinace dat pro pozitivní i negativní variantu scénáře, tak scénář vždy proběhne úspěšně. Důvodem je to, že se buď úspěšně ověří checkpoint pro pozitivní variantu, nebo se zapíše výstupní hodnota v podobě chybové zprávy při negativní variantě.

oova zprava") -
ł

Obrázek 30 - výsledný skript pro test na If funkci, zdroj: autor z nástroje HP QTP

5.10 Reportování událostí pomocí metody Reporter.ReportEvent

V Předchozím testu se spojil negativní a positivní scénář. Do výsledků se ovšem obě varianty zapíší se statusem "Passed", což není žádoucí a v případě neúspěšného zalogování do aplikace je na místě status "Failed". K dosažení tohoto výsledku se používá metoda Reporter.ReportEvent, která reportuje informace o událostech v testu do výsledku testu.

Syntaxe je následná:

Reporter.ReportEvent EventStatus(jaký status se má reportovat viz níže), ReportStepName(název události), Detail(popis události)

EventStatus může nabývat následujících hodnot:

- 0 nebo micPass odesílá status "Passed"
- 1 nebo micFail odesílá status "Failed"
- 2 nebo micDone odesílá zprávu do výsledku bez ovlivnění Failed/Passed statusu
- 3 nebo micWarning odesílá zprávu, která se ve výsledcích zobrazí jako varování

Pokud je potřeba reportovat pro negativní scénář z předchozího testu "Failed" status,

vloží se krok ReportEvent na konec negativní části scénáře.

- 5: If Dialog("Login").Dialog("Flight Reservations").Exist Then
- 6: Dialog("Login").Dialog("Flight Reservations").Static("Please enter agent name").Output CheckPoint("Chybova zprava")
- 7: Dialog("Login").Dialog("Flight Reservations").WinButton("OK").Click
- 8: Dialog("Login").WinButton("Cancel").Click
- 9: Reporter.ReportEvent micFail, "Neúspěšné přihlášení", "špatný login nebo heslo"

Obrázek 31 - vložení kroku Reporter.ReportEvent, zdroj: autor z nástroje HP QTP

Následně po seběhnutí testu (pokud jsou v *Data table* logovací údaje pro negativní scénář) je již ve výsledcích vidět, že všechny iterace, které používali špatná data pro přihlášení, jsou nyní ve statusu "Failed".

🖬 HP Run Results Viewer						_		×
File View Tools Help								
i 🔚 🖻 Y 🕼 🕷 🐼 🕻 🗢 🗮 🛛 ? 📑								
flight_login_IFELSE - TempResults_14 🛛 🔍 🖟 🗙	R	Result Details				*	џ	×
Search for: 🖉 🗴 🗸					-		-	<u></u>
Test flight_login_IFELSE Summary	1	Step Name: N	Step Name: Neúspěšné příhlášení					
Fight_login_IFELSE Iteration 1 (Row 1)		Step Failed						
A standard SystemUtil		-					1	
⊳ 📰 Login		Object	Details	Result	Time			
Ri Neúspěšné přihlášení Fight_login_IFELSE Iteration 2 (Row 2)		Neúspěšné přihlášen	í špatný login nebo heslo	Failed	3/13/2014-12:10):13		
								-
For help, press F1				R	leady			1.

Obrázek 32 - interpretace reportované události ve výsledcích, zdroj: autor z nástroje HP QTP

5.11 Akce

Při testování jedné aplikace často dochází k situacím, kdy se jedna akce musí použít ve více testech přesto, že není předmětem testu, ale je to nutná podmínka pro otestování další části, například zalogování se do aplikace. Pro tyto případy lze v QTP akce ukládat a později je jednoduchým příkazem zavolat aniž by se musela psát delší část kódu.

V QTP lze vytvořit knihovnu akcí, které zahrnují standardní kroky v aplikaci a testy následně skládat pouze z jednotlivých akcí a částečných úprav, což zvyšuje modularitu skriptu.

Vytvářením znovupoužitelných akcí se lze také vyhnout náročným úpravám v testech, v případech, že dojde například ke změně v logování – změní se pouze skript v konkrétní akci a změna se projeví ve všech testech, které akci využívají.

Vytvoření akcí bude předvedeno na základním scénáři přihlášení a odhlášení z aplikace pro rezervaci letenek nahraným v předchozích kapitolách.

🚮 flig	ht_login
	& Action2
1: 2: 3: 4: 5: 6:	SystemUtil.Run "C\Program Files\HP\QuickTest Professional\samples\flight\app\flight\a.exe","","C\Program Files\HP\QuickTest Professional\samples\flight\app\","open" Dialog("Login"),WinEdit("Agent Name:"), Set ("Tester") Dialog("Login"),WinButton("OK"), Click Window("Flight Reservation"), Close

Obrázek 33 - skript pro zalogování a odlogovaní z aplikace Flight reservation, zdroj: autor z nástroje HP QTP

Tento test zahrnuje akci přihlášení na 1. - 4. Řádku a odhlášení na 5. Řádku. Skript se může rozdělit na 2 akce kliknutím na řádek, kde je potřeba akci rozdělit a stiskem tlačítka "Split action".



Obrázek 34 - rozdělení akcí tlačítkem "Split action", zdroj: autor z nástroje HP QTP

Následně se objeví dialogové okno s nastavením výsledných akcí. Zde se může nastavit, zda jsou na sobě akce nezávislé ("Independent of each other") nebo jedna závisí na druhé ("Nested"). V tomto případě vybereme možnost "Nested", protože odlogování se provádí, až poté co se dokončí zalogování. Akce je možné následně pojmenovat a popsat.

Split Action											
This will split the current action into two actions. The second action will start on the current step.											
The actions are: Independent of each other Nested (the second action is called by the first)	1st action Name: Login Description: Zalogovani										
E⊷ 🌮 BuyABook1	2nd action Name: Logout Description: Odlogovani										
	OK Cancel Help										

Obrázek 35 - nastavení popisu nově vzniklých akcí, zdroj: autor z nástroje HP QTP

Po potvrzení je v okně *Test flow* vidět, že se vytvořily dvě akce, přičemž akce "Logout" závisí na akci "Login" a v akci "Login" se v posledním řádku volá akce "Logout". Tento test si pro pozdější využití a lepší identifikaci uložíme jako Test1.





Tyto akce lze později znovupoužít v dalších testech a mezi akcemi mohou probíhat další operace. Pro demonstraci se vytvoří test na otevření hotových rezervací.

Otevře se nový test přes *File\New\Test*. Poté se spustí aplikace Flight reservation, provede se přihlášení do aplikace a poté se spustí nahrávání tlačítkem *Record*. V aplikaci se klikne na *File/OpenOrder* a do *Order No*. se vyplní hodnota "1" a stiskne se tlačítko *OK*. Nyní se může nahrávání vypnout. V nahraném testu se parametrizuje zadání (*Order No*.) na 1,4 a například 5.

Stejně jako v předchozím případě parametrizace se přepne do Keyword view a u akce *Order No./edit* se vloží jako zdroj dat parametr. Parametr se pojmenuje a zvolí se, zda má uložit uložit do *Data table* v *Global sheet* nebo *Local sheet*.

- Global sheet Unikátní pro cely test. Jakákoliv akce může používat a editovat tyto data.
- Local sheet Obsahuje pouze data vytvořené v dané akci. Každá akce může užívat a editovat data ve svém vlastním *Datasheet*.

Vzhledem k tomu, že parametr je potřeba pouze pro akci vyhledání objednávky, stačí ho vytvořit pouze v *Local sheet*.

👻 📰 Open Order				
🗁 🗹 Order No.	Set	0"	N''	
IABC Edit	Set	"1"		
	Click			
	Value Configurati	on Options		×
	Parameter Name: Cislo o	DataTable objednavky	• •	
	C Global sł C Global sł C Current a	rata Table neet action sheet (local)		
Table	ок	Cancel	Help	

Obrázek 37 - nastavení parametru, zdroj: autor z nástroje HP QTP

Do parametru se poté vloží požadovaná čísla objednávek.

🔬 test2*	
2	Login
1: 2: 3: 4: 5:	Window("Flight Reservation").WinMenu("Menu"). Select "File;Open Order" Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No."). Set "ON" Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit"). Set DataTable("Cislo_objednavky", dtLocalSheet) Window("Flight Reservation").Dialog("Open Order").WinButton("OK").Click

Obrázek 38 - Skript po přidání parametru Cislo_objednavky, zdroj: autor z nástroje HP QTP

Nakonec se upraví vlastnosti výsledné akce kliknutím pravým tlačítkem myši na akci v *Test flow* a zvolením *Action properties*. Akce se pojmenuje, vyplní se popis a zaškrtne se políčko *Reusable action*.

- Reusable action (zaškrtnuto jako výchozí hodnota) pokud je zaškrtnuto, akce může být použita i v jiných testech a může být použita vícekrát v jednom testu. Akci lze volat metodou *Call to Copy* a *Call to Existing action* (popsáno níže).
- Non-Reusable action akce nemůže být použita ve více testech ani vícekrát v jednom testu. Volá se pouze metodou Call to Copy of an action.

Jak bylo zmíněno, metody pro volání akcí jsou následující dvě:

- Call Copy of an Action zavolá se kompletní kopie akce včetně checkpointů, parametrů a odpovídajícími záznamy v Data table. Pokud se akce zavolá touto metodou, mohou se v akci provádět změny a ty se projeví pouze v této její kopii a nebudou mít vliv na ostatní testy, kde byla akce použita. Touto metodou lze volat Reusable i Non-reusable akce.
- *Call to Existing action* touto metodou lze volat pouze *Reusable* akce. Tyto akce nelze modifikovat v testu, kam jsou volány, ale pouze v testu, kde byly vytvořeny.

Action Properti	es	X
General Para	meters Associated Repositories Used By	_
Name :	VyhledaniObjednavek (Reusable action)	
Location :	C:\Program Files\HP\QuickTest Professional\Tests\MMA\test	
Description :	Vyhleda objednavky dle zadanych parametru 📃	
	Y	
Reusable	action	
	OK Cancel Help	

Obrázek 39 - nastavení vlastností akce, zdroj: autor z nástroje HP QTP

Celý test se pro lepší orientaci následně uloží jako Test2. V tuto chvíli je uložen Test1, kde jsou uloženy akce "Login" a "Logout" a Test2, kde je uložena akce pro vyhledávání objednávek. Pro zkompletování testu se bude volat akce z Testu1 do Testu2.

Aby se vyzkoušely obě metody volání funkcí, nastaví se v Testu1 akci login jako *non-Reusable* a akci "Logout" jako *Reusable* stejným způsobem jako v předchozím případě u akce pro vyhledávání objednávek.

Nyní se otevře Test2 a klikne se na menu volání akcí, kde se vybere metoda *Call to copy* of an action.

🚽 🔒 🔝 i 🛪 🕀 i 🕫		: 🚰 🗸	💓 - 🕅	7	4) 🛓 🗐 🖇	🖓 🐮 🖕	
<mark>₀}test2</mark>			🐲 Call t	:o <u>N</u> ew	Action		
VyhledaniObiednavek	Q Show	Call to Copy of <u>A</u> ction					
ltem → ₩ VuhledaniΩbiednavek	Operation	Value	Call t	Call to Existing Action "			
Flight Reservation	Generation						
🚽 📅 Menu	Select	"File;Open Order"			Select item "File;0)	pen Order'' from the "Menu" menu.	
🗸 📰 Open Order							
🛛 🔽 Order No.	Set	"ON"			Set the state of the	"Order No." check box to "ON".	
Edit	Set	DataTable("Cislo_obje	ednavky", dtLocal		Enter <the 'cislo_objednavky'="" column="" data="" of="" table="" the="" value=""> in th</the>		
0K	Click				Click the "OK" but	ton.	

Obrázek 40 - zavolání akce metodou Call to Copy of Action , zdroj: autor z nástroje HP QTP

Otevře se dialogové okno pro výběr testu, ze kterého chceme volat akci a pro výběr konkrétní akce. Zvolíme Test1 a zobrazí se nabídka *Reusable* a i *Non-reusable* akcí v tomto testu.

Select Action 🛛 🛛 🛛 🛛
From test:
C:\Program Files\HP\QuickTest Professional\T 💌
Action:
🥩 Login 📃 💌
💋 Login 😤 Logout
Action description:
V
Edit new action properties Location At the end of the test
After the current step
OK Cancel Help



Vybereme akci "Login" a vložíme v části *Location* volbu *At the end of the test*. Akce Login se tak přidá nakonec testu a do *Data table* se přidá její vlastní editovatelný *Datasheet*.

Test Flow 🔻 🕈 🗙	J.	est2*												
⊡…∭ Test2 …∕∕ VyhledaniObjednavek		Test Flow				✓ ⇐ Back Q Show								
⊡ 🥩 Copy of Login	l n	Item				Operation	Value			Documentation				
		e 🥩 1	/yhledaniObj	ednavek										
		- 🥔 1	Copy of Logir	1 I										
				/										
	14 4	• •	Keywor		Expert VI	iew_/								
	Data	Table												
		A1												
			A	В	С	D	E	F	G	Н		J	K	L
	1													
	2	_												
	3	_												
	4													
	6													
	7													
	1	GI	bal 🗸 Wr	ledaniOt	ojednavel	k 入 Copy of	Login /							•

Obrázek 42 - Přidání kopie akce Login s vlastním datasheetem, zdroj: autor z nástroje HP QTP

Nyní ještě přidáme akci "Logout" metodou *Call to Existing action*. V nabídce si všimněte, že již není v nabídce akce Login, protože není nastavena jako *Reusable*.
Select Action 🛛 🛛 🛛 🛛
From test:
C:\Program Files\HP\QuickTest Professional\T 💌 🛄
Action:
🚓 Logout 💽
🚓 Logout
Action description:
Odlogovani z aplikace
Location
 At the end of the test
C After the current step
OK Cancel Help

Obrázek 43 - výběr akce Logout, zdroj: autor z nástroje HP QTP

Potvrdí se tedy přidání akce "Logout" opět na konec aktuálního testu. Akce se opět přidá s vlastním *DataSheet*, který ovšem již nelze editovat, stejně jako nelze editovat samotnou akci, protože je akce volána metodou *Call to existing action*, takže upravovat ji lze pouze v testu, kde byla vytvořena.

Test Flow 🔻 🕈 🗙	Б	est2*									
⊡ Test2		To al Flave			(D Back	O Show					
VyhledaniObjednavek		I est How				C onow					
Copy of Login		item) bie de suele	Upe	ration	Value			Documentati	on	
Logout [Test1]		- Sonu of Lo	oin Jujeunavek								
		Logout [Te	st11						Call the Logo	ut [Test1] ac	tion.
		•		:	1				1		
	I I I	Keyw	ord View 📈	Expert View	7						
	Data	Table									
		A1									
		A	В	С	D	E	F	G	Н	I	J
	1										
	2										
	4										
	5										
	6										
	7										
		\ Global 🖌 \	/yhledaniOt	ojednavek 🗡	Copy of L	.ogin <u>入 Log</u>	out [Test1] /	/			

Obrázek 44 - přidání akce "Logout", zdroj: autor z nástroje HP QTP

Nakonec se pouze pomocí drag&drop¹² uspořádá akce, aby test začal akcí "Login", pokračoval akcí pro vyhledání objednávek a končil. Zbývá ještě v akci "VyhledavaniObjednavek" přes volbu *Action call properties* nastavit tabu *Run* možnost *Run on all rows*. Tím bude zajištěno, že test seběhne pro všechna čísla objednávek zadaných jako parametr v *Data table*.

¹² Označuje metodu přesouvání objektů v aplikaci myší,

Test2*	
💐 Test Flow	✓ Back Q Show
Item	Operation Value
👘 🌮 Copy of Login	
🖉 🥩 VyhledaniObjednavek	
Logout [Test1]	Action Propert <u>i</u> es
	Action Cal <u>l</u> Properties
	Object Repository
	Insert Call to Copy of Action
	Insert Call to Existing Action

Obrázek 45 - otevření vlastností volání akce, zdroj: autor z nástroje HP QTP

Action Call Properties	×
Run Parameter Values	
Data Table iterations	
C. Bun one iteration only	
Bun on all rows	
S Harrion Town 1	
OK Cancel Help	

Obrázek 46 - vlastnosti volání akce, zdroj: autor z nástroje HP QTP

V tuto chvíli je test možné spustit. Ve výsledcích je vidět, že test proběhl úspěšně a v akci "VyhledaniObjednavek" došlo ke třem iteracím.

HP Run Results Viewer		
File View Tools Help		
i 🔁 🖻 🏹 in in 🎯 🕒 < ⇒ 🚍 ? 📄		
Test2 - TempResults_10 Image: Constraint of the second	Result Details Copy of Login Results Action: Copy of Login Run started: 3/14/2014-12:34:02 Run ended: 3/14/2014-12:34:03 Result: Passed	• ₽ × ≤ Summary
	Status Passed Failed Warnings	Times 1 0 0

Obrázek 47 - výsledky akce "VyhledaniObjednavek", zdroj: autor z nástroje HP QTP

5.12 Object repository (Objektový repositář)

Součástí QTP je *Object repository*, který slouží jako úložiště informací o objektech, které QTP dokáže rozpoznat a pracovat s nimi. *Object repository* se rozšiřuje v průběhu nahráváním práce v testované aplikaci případně ručním přidáváním objektu, kdy se označí objekt v testované aplikaci a ten se poté přidá do repositáře.

Aby bylo možné pracovat v testu s objekty, je nutné, aby se nacházely v *Object repository*. Každý nový test má repositář prázdný a je proto nutné objekty nahrát buď přes *Record* nebo ručně přidat funkcí *Add object to local*.

Object repository se dělí na 2 typy:

- Local object repository (Lokální objektový repositář)
- Shared object repository (Sdílený objektový repositář)

5.12.1 Local object repository (Lokální objektový repositář)

Local object repository je definovaný jako výchozí repositář objektů. Je vázaný na konkrétní akci a objekty uložené v *Local object repository* lze využívat pouze v této akci.

Nevýhodou tohoto repositáře je, že ho nelze znovu použít v jiné akci. Využívá se proto hlavně v aplikacích, které se v průběhu času nemění.



Obrázek 48 - Local object repository, zdroj: autor z nástroje HP QTP

V Local object repository lze využívat množství funkcí, jejichž výběr sr předvede na aplikaci pro rezervaci letenek.

 Highlight in application – Tato funkce zvýrazní zvolený objekt repositáře v testované aplikaci a je tak snadné objekt identifikovat. Předpokladem je, že testovaná aplikace je spuštěná.

💡 Object Repository - All Object Repositories				×
Eile Edit Object View Tools Help				
🔣 🗠 🖓 🗈 🖻 🗙 🖓 🖗 🤪	۲ 🖓 🗄 الگر 🔍	🟅 Filter: 🛛 🔒 Al	ll Objects 🔻	
Action: Action2	• •	Object Prope	erties	
		Name: Class: Repository: Test object de	OK WinButton Local	+ × 0
Fight Reservation		Name		Value
WinObject Window Start Menu Start Checkpoint and Output Objects	Login Agent Name: Password:		Droberties	on ne. Click to add] e

Obrázek 49 - funkce Highlight in application, zdroj: autor z nástroje HP QTP

 Locate in repository – tato funkce oproti předchozí naopak identifikuje, zda je objekt aplikace již zahrnut v repositáři. Výběrem ikonky pod kurzorem se zobrazí testovaná aplikace, kde kliknutím na objekt provedeme ověření přítomnosti v repositáři, v tomto případě tlačítko *Cancel*. Pokud se objekt v repositáři nachází, otevře se dialogové okno s jeho vlastnostmi, v opačném případě se zobrazí pouze zpráva, že objekt nebyl nalezen.

🔋 Object Repository - All Object I	tepositories	x
Eile Edit Object View Tools	telp	
💽 🗠 🗠 💃 🗈 🖪 🗙 🎼	🕯 🎲 🚯 🄪 🔍 🔩 🖞 🚱 🏹 Filter: 🔋 All Objects 🛛 👻	
Action: Action2	🔨 💌 Object Properties	
🖃 🔒 Test Objects	Name: Cancel	
Cancel	Object Selection	
	The location you clicked is associated with several objects.	
Agent Name.	Select the required object from the tree below.	+ × ດ
Flight Reservation	WinButton : Cancel	Value
🖉 🦉 WinObject		Canaal
Energy Window ⊡⊡⊡ Start Menu		Button
🕞 🚱 Start		
📔 🔤 🖁 Checkpoint and Output Obje	s	[None. Click to add]
		None
	OK Cancel Help	NOILE
	Enable official raonalitication	False
	Comment	
<u> </u>		

Obrázek 50 - funkce Locate in repository, zdroj: autor z nástroje HP QTP

 V repositáři je možné provádět s objekty podobné operace jako se soubory, tzn.: vyjmout, kopírovat, vložit, přejmenovat a smazat.



Obrázek 51 - operace s objekty v objektovém repositáři

Pokud se v aplikaci nachází více podobných objektů (jako například tlačítko OK v různých místech aplikace), je možné upravit vlastnosti objektů, tak aby se dali lépe rozlišit, například upravením popisku objektu. To je možné výběrem objektu a následnou editací jeho vlastností v pravé části objektového repositáře. Pokud dojde k nechtěné úpravě objektu, lze vrátit původní vlastnosti funkcí Update form application, která objektu vrátí původní vlastnosti z aplikace.



Obrázek 52 - úprava vlastností objektů, zdroj: autor z nástroje HP QTP

 Add object to local – vkládání objektů do repositáře je velmi důležitou funkcí pro vytváření testů. Pomocí této funkce se do repositáře přidávají chybějící objekty, se kterými potřebujeme v testu operovat. Bez jejich vložení by skript nedokázal s objekty pracovat. Tlačítkem Add objects to local se aktivuje testovaná aplikace a tam už se jen vybere požadovaný objekt a kliknutím se přidá do repositáře, kam se přidá i se všemi výchozími vlastnostmi.

📔 Obj	ject Reposito	ory - All (Object	Repositorie	:5				
: <u>F</u> ile	<u>E</u> dit <u>O</u> bject	<u>V</u> iew	<u>T</u> ools	Help					
	503		×I	44 👰 🚯	ی 🔘 💕	🛃 🖏	😽 Filter:	🔒 All Objects	-
Action:	Action1			Add	Objects to Loca		•		
	Test Object Checkpoint	and Out	put Ob	jects			The sele This noc object re the root	ected node is the de contains all the epository. You car Object Repository	root Ol test o nnot pe node.

Obrázek 53 - vložení objektu tlačítkem Add objects to local, zdroj: autor z nástroje HP QTP

5.12.2 Shared object repository (Sdílený objektový repositář)

Shared object repository se využívá především v aplikacích, které se dynamicky mění. V projektech automatizovaného testování je to častěji využívané řešení, ale přináší s sebou větší nároky na údržbu a administraci. V následujících krocích je popsáno vytvoření Shared object repository.

Vzhledem k tomu, že jako výchozí repositář objektů je nastaven *Local object repository*, otevře se nejprve *Local object repositury*, například v testu pro přihlášení do aplikace pro rezervaci letenek.

Zde se v menu zvolí položka *Export local object*. Následně se objeví dialogové okno pro uložení *Shared object repository*. Vybere se umístění a pod vhodným jménem se uloží, tím je sdílený repositář vytvořen.

6	Obj	ject Repository - All Object Reposito	ries		X
:[File	<u>E</u> dit <u>O</u> bject <u>Vi</u> ew <u>T</u> ools <u>H</u> elp			
		Export Local Objects	s 😜 💿 🕾 i 🗄 🗞 i S	🟅 Filter: 🔒 All Objects 🔹	
А		Export and Replace Local Objects	•	Object Properties	
R	×	<u>Cl</u> ose		Name: Login	
			1	Class: Dialog	
				Repository: Local	
		Agent Name:		Tool allow date:	
				l est object details	TXU
	Ē	- Login		Name	Value
		🦢 🎯 WinObject		 Description properties 	
	Ē	🛄 Window		text	Login
		🗄 🛄 Start Menu		nativeclass	#32770
	0	G Start		is owned window	False
•	🥫	Checkpoint and Uutput Ubjects		is child window	False
				⊟ Visual relation identifier	
				Visual relation identifier settings	[None. Click to add]
L				⊟ Ordinal identifier	
				Type , Value	None
L				⊟ Additional details	
L				Enable Smart Identification	False
L				Comment	
L				1	:

Obrázek 54 - vytvoření Shared object repository volbou Export to local, zdroj: autor z nástroje HP QTP

Pokud je již zcela vytvořený *Shared object repository*, je potřeba ho přiřazovat k akcím, které jsou vytvořeny v testu. Toho lze dosáhnout tak, že se v aktuálně otevřeném testu zvolí v menu *Resources* možnost *Associate Repositories*.



Obrázek 55 - přiřazení Shared object repository k akci, zdroj: autor z nástroje HP QTP

Objeví se dialogové okno pro, kde se přes zelené tlačítko se symbolem "+" přiřadí požadované *Shared object repository*, kterých může být více než jeden. A Následně se k vybraným repositářům přiřadí požadované akce.

Associate Repositories	X
This dialog box lists the object repository files associated with all the actions in the current test. You can change the associations and perform other administrative tasks.	
Repositories + >	۲
C:\Program Files\HP\QuickTest Professional\Tests\MMA\Test re	
Available Actions Action3 Action2 <	
OK Cancel Help	

Obrázek 56 - přiřazení Shared object repository k akci, zdroj: autor z nástroje HP QTP

Pokud se *Shared object repository* asociuje k nově vytvořené akci, je vidět, že v repositáři objektů jsou objekty zobrazeny jako read only. V případě potřeby editace *Shared object repository*, je nutné provést editaci opět v menu *Resources*, kde se nachází *Object repository manager*. Po Spuštění *Object repository manager* se vybere požadovaný *Shared object repository* a následně ještě na možnost *Enable editing* (viz Obrázek 57) a poté lze již objekty v *Shared object repository* volně editovat.

📸 Object Repository Manager		
: <u>File E</u> dit <u>O</u> bject <u>Vi</u> ew <u>T</u> ools <u>Wi</u> ndow <u>H</u> elp		
: 🞦 🗁 🗐 🔣 🗠 ా X 🖻 🖹 🗙 AA 🎲 🗞 🤪 👀 🗞 🤇	🕂 🛃	
🔒 C:\Program 🎦es\HP\QuickTest Professional\Tests\MMA\Test reposit	ory.tsr (Read only)	
Test Object	Object Properties	
	Name: Login	
- OK	Login	
Agent Name:	Class: Dialog	
Flight Reservation	Test object details	+ × 0
E- Login	Name	Value
Window Geochapint and Output Objects	Description properties	
Greekpoint and output objects	text	Login
	nativeclass	#32770
	is owned window	False
	is child window	False
	Visual relation identifier	
	Visual relation identifier settings	[None]
	Ordinal identifier	
	Type , Value	None
	Additional details	
	Enable Smart Identification	False
	Comment	
J		

Obrázek 57 - editace Shared object repository, zdroj: autor z nástroje HP QTP

5.13 Typy nahrávání v QTP

Velkou výhodou QTP je vytváření testů pomocí přímého nahrávání akcí uživatele. Díky tomu jsou i uživatelé bez zkušeností z programování schopni vytvářet funkční testy. V QTP lze nahrávat třemi způsoby.

5.13.1 Kontextové nahrávání (Context sensitive recording)

V QTP je tento mód používán jako výchozí. Využívá v testech identifikaci podle objektů. Rozpoznává objekty bez ohledu na jejich aktuální poloze na obrazovce. Nahrává objekty a operace, které se s nimi provádí.

5.13.2 Analogové nahrávání (Analog recording)

Nahrávací mód, který zaznamenává veškeré pohyby myši na obrazovce. Používá se případech, kdy nelze využít kontextového nahrávání, například při zaznamenání podpisu myší.

Na analogové nahrávání lze přepnout, pouze během spuštěného nahrávání. Příklad s podpisem je možné předvést na aplikaci pro rezervaci letenek. Po zapnutí aplikace, se otevře libovolná objednávka a poté se v menu *File* vybere možnost *Fax order*. Následně se objeví dialogové okno s podepisovacím apletem.

ax			.
Name: John Doe		Urder: Flight: 1 6232	Date: 03/01/14
From: Denver	Departure: 10:24 AM	To: Los Angeles	Arrival: 01:24 PM
Class: First	# Tickets:	Ticket Price: 312.00	Total: 312.00
Fax Number:		Agent Signature:	
<u> </u>			
🔲 Send Signature v	vith order		

Obrázek 58 - detail objednávky v aplikaci Flight reservation, zdroj: autor z Flight Reservation

V QTP se zapne nahrávání a tlačítkem *Analog recording* se přepne na analogové nahrávání. Objeví se dialogové okno se dvěma možnostmi nahrávání.

- *Relative to the screen* používá se, pokud je nahráváno ve více oknech.
- *Relative to the window* používá se, pokud se nahrává pouze v jednom okně.

🚾 QuickTest Profession	al - [Test*]		
i 🚮 <u>F</u> ile <u>E</u> dit <u>V</u> iew	Insert <u>A</u> utomati	on <u>R</u> esou	urces <u>D</u> ebug <u>T</u> o
🔍 Record 🕨 Run 🔻	🔳 Stop 🛃 👯		Ŧ
🚰 • 🐲 • 🖗 🗷 🗍	📣 斗 🚽 🛃	og Recordir	na (Shift+Alt+F3)
🔬 Test*		-	
Retion1	•	🗢 Back	Q Show
Item	Op	peration	Value
🛛 🤣 Action1			

Obrázek 59 - přepnutí na analogové nahrávání, zdroj: autor z nástroje HP QTP

Pro naše potřeby se vybere možnost *Relative to the window* a ikonkou s ukazatelem ruky se vybere okno, které se bude nahrávat, tzn. aplikaci Flight reservation. Poté se už spustí nahrávání tlačítkem *Start analog recording*.

Obrázek 60 - spuštění analogového nahrávání, zdroj: autor z nástroje HP QTP

Následně se zapíše text na podepisovací applet a ukončí se nahrávání. Pokud se skript opět spustí, zopakuje opět náš podpis.

Fax Order No. 1	
Fax Name: John Doe	Order: Flight: Date: 1 6232 03/01/14
From: Departur Denver 10:24 A	e: To: Arrival: M Los Angeles 01:24 PM
Class: # Tickel	s: Ticket Price: Total: 312.00 312.00
Fax Number:	Agent Signature:
[] ☐ Send Signature with order	Tester
Preview Fax Send	Cancel Clear Signature



5.13.3 Low level recording

Umožňuje nahrávání jakýchkoliv objektů bez ohledu na to jestli QTP dokáže tyto objekty a operaci s nimi rozpoznat. Při nahrávání se všechny objekty rozlišují jako *Window* nebo *WinObject*.

Pro ilustraci je možné nahrát jednoduchý test na vyhledání objednávky. Při nahrávání se přepne na *Low level recording* a v aplikaci rezervací se provede vyhledání objednávky.



Obrázek 62 - přepnutí na Low level recording, zdroj: autor z nástroje HP QTP

Výsledný skript vypadá následovně:

🕅 QuickTest Professional - [Test*]					
i 🚮 Eile 😼 dit ⊻iew Insert	: <u>A</u> uto <mark>mation <u>R</u>esources <u>D</u>ebug <u>T</u>ools <u>W</u>indow <u>H</u>elp</mark>				
i 🏹 New 🕶 🖙 Open 🕶 🔚 🖓 🍜 🐒 📭 🟝 🗹 🏢 🔍 🖕 i 🗠 🖂 🍳 🖓 🚧 🎝 📮 i 🎼 💕 🔐 💭					
🔋 🔍 Record 🕨 Run 🔻 🔳 Sta	00 🗶 🕼 🗧 🖕 👘 🚺 🕴 🐉 🖓 🐼 🖓 🐼 🖓 🐼 🖓 🐼				
Test Flow 🔻 🗣 🗙	Test*				
⊡∰ Test	Action1 1: Window("Flight Reservation").WinObject("Button").Click 11,9 2: Window("Flight Reservation").Window("Open Order").WinObject("Order No.").Click 31,10 3: Window("Flight Reservation").Window("Open Order").WinObject("Edit").Click 25,6 4: Window("Flight Reservation").Window("Open Order").WinObject("Edit").Type "1" 5: Window("Flight Reservation").Window("Open Order").WinObject("Click 19,4 6: 7:				

Obrázek 63 - skript pro vyhledání objednávky nahraný v low level módu, zdroj: autor z nástroje HP QTP

Ze skriptu je vidět, že všechny objekty se zaznamenají jako *Window* a *WinObject*. Taktéž kliknutí nejsou zaznamenány v kontextu ale jako přesné souřadnice, kde ke kliknutí došlo.

Analog a Low level nahrávání zabírá více místa na disku a je nutné, aby při spouštění testu bylo vždy stejné uspořádání oken. Používá se výhradně v případech, kdy nelze použít kontextové nahrávání.

5.14 Funkce

Během definování testů dochází k případům, kdy je potřeba část kódu použít vícekrát v jednom testu. V takových případech je na místě vydefinovat tuto část jako funkci, kterou lze během testu opětovně volat. Díky funkcím jsou skripty kratší, přehlednější a snadněji se na nich provádí údržba.

Na jednoduchém příkladu se vyzkouší, jakým způsobem zle používat funkce a funkční knihovny. Otevře se nový test a nová knihova funkcí přes tlačítko *New* a volbu *Function library*.



Obrázek 64 - vytvoření nové knihovny, zdroj: autor z nástroje HP QTP

Vytvoří se nám nová knihovna funkcí a zde zapíšeme funkci pro zobrazení okna se zprávou, která se pojmenuje například "MsgTest". Skript vypadá následovně:

Function MsgTest (začátek funkce a její pojmenování)

MsgBox "Testovací funkce" (funkce MsgBox bude obsahovat zprávu v uvozovkách)

End Function (konec funkce)

Takto je funkce kompletní a knihovna se může uložit. V dalším kroku se přiřadí uložená knihovna k novému testu. Přes menu *File/Settings* se v záložce *Resources* klikne na zelené tlačítko "+" a přiřadí se požadovanou knihovna.

Test Settings
Properties Run Parameters Environment XML Warehouse Web Recovery Local System Monitor Log Tracking Resources Enter and prioritize the function libraries you want to associate with your test. Associated function libraries: Iterary1.qff Adds a new file to the file list Set as Default Check Syntax Click to check the syntax of all accessible files Data Table Iterary1.qff Default location (under test directory) Chere location: Iterary1.qff Check Syntax Click to check the syntax of all accessible files Data Table Iterary1.qfff Iterary1.qff Iterary1.qff Iterary1.
OK Cancel Apply Help

Obrázek 65 - přiřazení nové knihovny k testu, zdroj: autor z nástroje HP QTP

Volba se potvrdí a v tuto chvíli se již v testu mohou volat funkce obsažené v knihovně.

Funkce v testu se zavolá pouze vypsáním jména funkce do testovacího skriptu, v tomto případě "MsgTest". Pokud se test spustí, funkce se zavolá a zobrazí okno s vydefinovanou zprávou.

_										
OT	Q	Jick1	est I	Profe	ession	nal - [Te	st*]			
i ş		<u>F</u> ile	<u>E</u> d	it y	<u>V</u> iew	Insert	<u>A</u> utom	nation	<u>R</u> esou	irces _
1	0	Reco	rd	► R	lun 🔻	🔳 Sto	p 🗃	17	i (Ţ
4) T	est*	۱ 🕥	ibra.	ry1.qfi					
		*	Actio	n1			~			
	1: 2:	¢	•Msg	;Te:	st	'estovaci OK	í funkce			

Obrázek 66 - spuštění funkce, která zobrazuje vydefinovanou zprávu, zdroj: autor z nástroje HP QTP

5.15 Deskriptivní programovaní

Během nahrávání testů se do *Object repository* v QTP ukládá každý objekt, se kterým se v testované aplikaci pracuje. Při běhu testu se poté objekty identifikují v *Object repository* a lze s nimi dále manipulovat a provádět požadované kroky v testu. Pokud ale objekt není v *Object repository* nalezen, nelze metody nad objekty provést a skript selže. To může nastat, pokud objekt z nějakého důvodu nelze nahrát nebo se objekt mění a tyto změny nelze nijak podchytit. Tyto případy se dají vyřešit deskriptivním programováním.

Deskriptivní programování dokáže metodám určit, se kterým objektem mají pracovat bez toho, aby se museli odkazovat do objektového repositáře.

Deskriptivní programování lze provést následujícími dvěma metodami.

5.15.1 Statická metoda deskriptivního programování

Do QTP se zadá vlastnost objektu s jeho hodnotou a QTP pak objekt v aplikaci díky těmto hodnotám může identifikovat.

Statickou metodu deskriptivního programování si lze opět vyzkoušet na aplikaci pro rezervaci letenek. Do nového testu se nahraje pouze krok vyplnění pole login, potom se ukončí nahrávání a otevře *Object repository*.



Obrázek 67 - editace objektu pole pro login v Object repository, zdroj: autor z nástroje HP QTP

V Object repository je vidět, že pole pro login, představuje objekt, který je pojmenovaný "Agent Name" a v pravé dolní části je vidět jaké má objekt vlastnosti a jejich hodnoty. To představuje dostatek informací, aby bylo možné objekt popsat deskriptivně. Použijí se tedy vlastnosti objektu z Description properties k popisu objektu a ty pak nahradí ve skriptu jméno objektu "Agent Name".

Při statické metodě se vlastnost a její hodnota zapisuje ve tvaru *Property:=Value*. Pokud je vlastností více oddělují se čárkou. Výsledný zápis tedy vypadá následovně:

Dialog("Login").WinEdit("nativeclass:=edit","attached
text:=Agent Name:").Set "Tester"

Při tomto zápisu skriptu se po spuštění testu login vyplní, i když se objekt "Agent Name" odstraní z repositáře.

5.15.2 Dynamická metoda deskriptivního programování

Dynamické deskriptivní programování se používá, pokud se při deskriptivním programování využívá stejný objekt vícekrát.

Tato metoda umožňuje uložit vlastnosti pod vlastní popis a následně volat tyto vlastnosti tímto popisem. Dynamický zápis předchozího příkladu vypadá následovně:

Set Popis = Description.Create()

Popis("nativeclass").Value = "Edit"

Popis("attached text").Value = "Agent Name:"

Tím se definovali vlastnosti objektu pro login jako "Popis" a následně je stačí ve skriptu tímto heslem zavolat.

Dialog("Login").WinEdit(Popis).Set "Tester"

5.16 Doporučení

Tato příručka zdaleka nepokrývá veškerou funkcionalitu a možnosti HP QTP, ale poskytuje základní znalosti a orientaci v programu tak, aby bylo možné začít s nástrojem pracovat a nabyté zkušenosti dále rozvíjet.

Příručka by měla postačovat na tvorbu testů základní úrovně a testeři by se postupně podle potřeby měli učit nové postupy z některého z internetových zdrojů, které umožňují výuku s tímto nástrojem a znalosti si sami rozšiřovat. Zaměřit by se přitom měli především na výuku VBSkriptu, který jim umožní využívat velké množství užitečných funkcí a metod, které jsou ve VBSkriptu zahrnuty.

Základní přehled možností a funkcí toho nástroje má zároveň pomoci lépe a přesněji určit strategii pro automatizaci vybraných testů při zavádění automatizace.

6 Závěr

Za hlavní cíl této práce jsem v úvodu určil posouzení vhodnosti automatizace na reálném projektu pomocí nástroje HP QTP 11 a určení, v jakých fázích testovacího procesu se případná automatizace vyplatí. Tento cíl byl splněn ve čtvrté kapitole, kde jsem vydefinoval kritéria, která musí být splněna, aby se automatizace vyplatila. Tato kritéria jsem poté porovnal s podmínkami na reálném projektu a určil, že projekt je na automatizaci připraven a automatizace se zde vyplatí. Na základě tohoto porovnání byla poté identifikována nejvhodnější místa pro automatizaci v rámci testovacího procesu na projektu.

Porovnáním výše uvedených kritérií a podmínek na reálném projektu bylo určeno, že automatizace by se měla zaměřit především na tvorbu a údržbu sady regresních testů a potom na podporu testování v podobě přípravy dat pro systémové, integrační a regresní testy, tam kde náročnost přípravy bude tak vysoká, že se ji vyplatí automatizovat.

Dílčími cíli bylo představení nástroje HP QTP a vytvoření příručky k tomuto nástroji, protože právě s jeho využitím je automatizace na řešeném projektu realizována. Tyto cíle byly splněny ve třetí kapitole, kde byl nejprve nástroj HP QTP představen a poté v páté kapitole, kde byla vytvořena příručka popisující základní funkcionalitu HP QTP včetně příkladů.

Za hlavní přínosy teoretické části práce považuji dokázání, že zavedení automatizace testování se na nejmenovaném existujícím projektu vzhledem k vydefinovaným kritériím vyplatí a dále, že byla identifikována nejvhodnější místa pro automatizaci v rámci testovacího procesu na tomto projektu a s odůvodněním vyloučena místa, kde se automatizace nevyplatí.

Přínosem praktické části je vytvoření příručky k nástroji HP QTP, která by měla testerům pomoci rychleji zvládnout obsluhu tohoto nástroje a lépe naplánovat automatizaci testů, díky přehledu o možnostech tohoto nástroje. Tato příručka je zároveň výstupem práce a bude poskytnuta testovacímu týmu na řešeném projektu, kde je automatizace zaváděna.

Za přínos také považuji úvodní část práce, kde je zpracována teorie z oblasti testování vztahující se k tématu práce a porovnán automatizovaný a manuální přístup k testování.

Zavedení automatizace na projektu, kde pracuji, je významnou událostí nejen pro testovací tým, ale i celý projekt. Očekává se, že by s sebou automatizace prostřednictvím nástroje HP QTP měla přinést úsporu času a testovacích kapacit. Tato práce měla za cíl určit, zda se automatizace vyplatí, určit nejvhodnější místa vhodná k automatizaci a poskytnout informace, které by pomohli pracovníkům testovacího týmu, kteří ještě nemají zkušenosti s tímto nástrojem, k rychlejšímu zvládnutí obsluhy nástroje a efektivnějšímu naplánovaní automatizace testů.

Vytyčené cíle byly splněny a automatizaci tak lze doporučit. Doporučená automatizace regresních testů a přípravy dat by měla zefektivnit proces testování a ušetřit požadované zdroje, které lze alokovat na důležitější úkoly. Tomu by měla přispět i příručka, jež je výstupem této práce.

Seznam použitých zkratek a termínů

Termín	Zkratka	Význam [zdroj]
Architecture	ASD	Dokument popisující řešení požadovaných změn v softwaru.
Solution		[vlastní definice autora]
Document		
Automatizované		Jakékoliv testování, k jehož provedení je využíván software. Ten
testování		vykonává některé úlohy místo lidských testerů a do jisté míry je
		tím nahrazuje. [21]
Best practice		Obecně uznávané a používané nejlepší postupy. [vlastní definice
		autora]
Customer	CRM	CRM označuje systémy pro řízení vztahů se zákazníky [vlastní
Relationship		definice autora]
Management		
Detail Design	DD	Dokument popisující detailní návrh řešení požadovaných změn
		v softwaru. [vlastní definice autora]
Drag&Drop		Označuje metodu přesouvání objektů myší. [vlastní definice
		autora]
End-to-end test		Testování procesu od jeho počátku až do konce napříč celou
		integrační cestou. [vlastní definice autora]
FURPS	FURPS	Metoda definující, jak poznat a ověřit kvalitu dodávaného
		software. Je to akronym slov Functionality, Usability, Reliability,
		Perrformance a Supportability označujících 5 dimenzí ze kterých
		je na kvalitu softwaru nahlíženo. [3]
HP Quality	HP QC	Nástroj pro test management od společnosti Hewlett-Packard.
center		[vlastní definice autora]
HP QuickTest	HP QTP	Nástroj pro automatizované testování od společnosti Hewlett-
Professional		Packard. [vlastní definice autora]

Checkbox	Okénko pro zaškrtávací volbu v aplikaci. [vlastní definice autora]
Checkpoint	Kontrolní body v průběhu testu. [vlastní definice autora]
Integrační	Testování, jehož cílem je otestování integrace samostatných
testování	jednotek do stabilního systému. [21]
Login	Obecně používané označení pro přihlašovací jméno do aplikace. [vlastní definice autora]
Manuální	Testování prováděné lidskými testery, kteří obstarávají jak
testování	samotnou exekuci testů, tak případné hlášení chyb. [vlastní
	definice autora]
Oracel Siebel	Systém pro řízení vztahu se zákazníkem vyvíjený společností
	Oracle. [vlastní definice autora]
Outsourcing	Zajištění vybraných činností firmy externími zdroji. [vlastní
	definice autora]
Radiobutton	Tlačítko pro kroužkovací volbu v aplikaci. [vlastní definice
	autora]
Regresní	Regresní testování, že po změně určité části systému zůstávají
testování	ostatní součásti funkční a ve stejném stavu. [21]
Release	Release je označení pro soubor dílčích projektů, jež zahrnují
	změny softwaru, které mají být implementovány v rámci
	jednoho období. [vlastní definice autora]
Screenshot	Snímek aktuálního zobrazení na monitoru. [vlastní definice
	autora]
Softwarová	Jakákoliv odchylka od specifikace softwarového produktu, kvůli
chyba	které produkt neplní svojí funkci nebo nepracuje tak, jak se od
	něj očekává. [vlastní definice autora]
Softwarová	Míra splnění požadavků a schopnost plnit funkci, pro kterou byl
kvalita	software vytvořen. [vlastní definice autora]
Systémové	Testování již finálního softwarového produktu, které má
Systemore	

		specifikovaných zákazníkem. [21]
Testování		Proces zahrnující jakékoliv aktivity zaměřené na vyhodnocení
		vlastností nebo schopností programu nebo systému a určení,
		zda odpovídají požadovaným výsledkům. 89[4]
Testování		Jednotky jsou chápány jako nejmenší testovatelné součásti
jednotek		programu. Cílem jejich testování, je otestovat každou jednotku
		nezávisle na ostatních a prokázat, že její chování je správné. [21]
User	UAT	Testování uživateli na straně zákazníka, jehož smyslem je zjistit,
Acceptance		zda produkt splňuje akceptační kritéria. Ty jsou stanovena
Testing		zákazníkem jako měřitelné a ověřitelné podmínky pro přijetí
		produktu. [21]

Seznam použité literatury

- Boehm, Barry W. A Spiral Model of Software Development and Enhancement. TRW
 Defense Syst. Group, Redondo Beach, CA, roč. 21, č. 5, August 1988. ISSN: 0018-9162
- [2] Faustová, Tereza. Nástroje na podporu testování. Praha, 2008. Diplomová práce.
 Vysoká škola ekonomická. Vedoucí práce Ing. Alena Buchalcevová, Ph.D.
- [3] FURPS. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [vid. 2014-03-25]. Dostupné z: <u>http://cs.wikipedia.org/wiki/FURPS</u>
- [4] Hetzel, William C., *The Complete Guide to Software Testing, 2nd ed.* Publication info: Wellesley, Mass. : QED Information Sciences, 1988. ISBN: 0894352423.Physical description: ix, 280 p. : ill ; 24 cm.
- [5] HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P. HP QuickTest Professional Software Version: 11.00: User Guide. 2010, 1854 s. Dostupné z: http://www.asitest.com/Docs/QTPv11/QTUsersGuide.pdf
- [6] Hlava, Tomáš. Automatizované testování. *Testování softwaru* [online]. [vid. 4. duben
 2014]. Dostupné z: <u>http://testovanisoftwaru.cz/automatizovane-testovani/</u>
- [7] Hlava, Tomáš. Fáze a úrovně provádění testů [online]. 21. srpen 2011 [vid. 2. duben 2014]. Dostupné z: http://testovanisoftwaru.cz/tag/urovne-testovani/
- [8] Holdberh, Raman. Automatizované testování webových aplikací. Praha, 2011.
 Bakalářská práce. Vysoká škola ekonomická. Vedoucí práce Ing. Alena Buchalcevová,
 Ph.D.
- [9] HP QuickTest Professional. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [vid. 31. březen 2014]. Dostupné z: <u>http://en.wikipedia.org/wiki/HP_QuickTest_Professional</u>

- [10] IBM Rational Unified Process. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [vid. 2014-03-27]. Dostupné z: <u>http://en.wikipedia.org/wiki/IBM_Rational_Unified_Process</u>
- [11] Kaner, Cem. Developing Skills as an Exploratory Tester[online]. *Quality Assurance Institute Worldwide Annual Software Testing Conference*, Orlando, FL, November 2006. Dostupné z WWW: <u>http://www.kaner.com/pdfs/ETatQAI.pdf</u>
- [12] Kaner, Cem. Lessons learned in software testing: a context-driven approach. New York, c2002, xxvii, 286 s. :. ISBN 04-710-8112-4.
- [13] Levels of Software Testing. tutorialspoint [online]. [vid. 2. duben 2014]. Dostupné z: http://www.tutorialspoint.com/software_testing/levels_of_testing.htm
- [14] Modely životního cyklu softwaru. Testování softwaru [online]. 2. únor 2011 [vid. 1. duben 2014]. Dostupné z: <u>http://testovanisoftwaru.cz/manualni-testovani/modely-</u> zivotniho-cyklu-softwaru/
- [15] Pan, Jiantao.: Software Testing [online]. Carnegie Mellon University. 1999 [vid. 24-03-2014]. Dostupné z WWW: http://www.ece.cmu.edu/~koopman/des_s99/sw_testing/
- [16] Patton, Ron. Testování softwaru. Praha : Computer press, 2002. 80-7226-636-5.
- [17] Pouzar, Lukáš. Automatizované testování. Praha, 2007. Bakalářská práce. Vysoká škola ekonomická. Vedoucí práce Ing. Alena Buchalcevová, Ph.D.
- [18] Proč používat verifikaci softwaru? Control Engineering Česko [online]. 9. červen 2009 [vid. 1. duben 2014]. Dostupné z: <u>http://www.controlengcesko.com/hlavni-</u> <u>menu/artykuly/artykul/article/proc-pouzivat-verifikaci-softwaru/</u>
- [19] QTP Tutorials 25+ HP QuickTest Professional (QTP) Training Tutorials Software Testing Help. Software testing help [online]. [vid. 18. duben 2014]. Dostupné z: http://www.softwaretestinghelp.com/qtp-quicktest-professional-tutorial-1/
- [20] QTP Tutorials & Interview Questions. *qtp.blogspot.cz* [online]. [vid. 18. duben 2014]. Dostupné z: http://qtp.blogspot.cz/

- [21] Roudenský, Petr a Havlíčková, Anna. Řízení kvality softwaru: průvodce testováním. 1.vyd. Brno: Computer Press, 2013. 208 s. ISBN 978-80-251-3816-8.
- [22] Software development process. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [vid. 2014-04-01]. Dostupné z: <u>http://en.wikipedia.org/wiki/Software_development_process</u>
- [23] Software Testing Levels. Software Testing Fundamentals [online]. [vid. 2. duben 2014]. Dostupné z: <u>http://softwaretestingfundamentals.com/software-testing-levels/</u>
- [24] Štolc, Robin. Porovnání komerčních a open source nástrojů pro testování softwaru. Praha, 2010. Diplomová práce. Vysoká škola ekonomická. Vedoucí práce Ing. Alena Buchalcevová, Ph.D.
- [25] The Spiral Model. Expertiza Wiki [online]. 27. říjen 2012 [vid. 1. duben 2014]. Dostupné z: <u>http://wiki.expertiza.ncsu.edu/index.php/CSC/ECE_517_Fall_2012/ch2a_2w3_sm</u>
- [26] tutorialspoint Simply easy learning [online]. [vid. 18. duben 2014]. Dostupné z: <u>http://www.tutorialspoint.com/index.htm</u>
- [27] Vývojové modely. Diagnostika a testování elektronických sytémů [online]. 2012 [vid. 1. duben 2014]. Dostupné z: <u>http://www.umel.feec.vutbr.cz/bdts/index.php/embeddedsystemy/vyvojove-modely</u>

Seznam obrázků a tabulek

Obrázky

Obrázek 1 - vodopádový model, zdroj: [18]7
Obrázek 2 - spirálový model, zdroj: [25] 8
Obrázek 3 - V-model vývoje softwaru, zdroj: [21]9
Obrázek 4 - náhled na Expert view, zdroj: autor z nástroje HP QTP 19
Obrázek 5 - náhled na Keyword view, zdroj: autor z nástroje HP QTP 19
Obrázek 6 - základní obrazovka aplikace Flight reservation, zdroj: autor z aplikace Flight
Reservation
Obrázek 7 - volba pluginů v QTP, zdroj: autor z nástroje HP QTP 34
Obrázek 8 - startovní obrazovka QTP, zdroj: autor z nástroje HP QTP 35
Obrázek 9 - hlavní okno QTP, zdroj: autor z nástroje HP QTP 37
Obrázek 10 - nastavení nahrávání, zdroj: autor z nástroje HP QTP 38
Obrázek 11 - test pro zalogování po úpravě, zdroj: autor z nástroje HP QTP 40
Obrázek 12 - spuštění testu, tlačítko Run, zdroj: autor z nástroje HP QTP 40
Obrázek 13 - dialogové okno s výběrem cesty pro uložení výsledků, zdroj: autor z nástroje HP
QTP
Obrázek 14 - šipka označující právě probíhající krok v testu, zdroj: autor z nástroje HP QTP 41
Obrázek 15 - výsledek dokončeného testu zobrazený v aplikaci HP Run Results Viewer, zdroj:
autor z nástroje HP QTP 42
Obrázek 16 - detail výsledku konkrétního kroku v testu, zdroj: autor z nástroje HP QTP 42
Obrázek 17 - zobrazení screenshotu konkrétní akce provedené v označeném kroku testu, zdroj:
autor z nástroje HP QTP 43
Obrázek 18 - nastavení snímání obrazovky a videa během testu, zdroj: autor z nástroje HP QTP
Obrázek 19 - nahrazení nastavované hodnoty objektu parametrem, zdroj: autor z nástroje HP
QTP 44
Obrázek 20 - konfigurace parametru, zdroj: autor z nástroje HP QTP 45

Obrázek 21 - vkládání dat do <i>Data Table,</i> zdroj: autor z nástroje HP QTP
Obrázek 22 - vložení checkpointu, zdroj: autor z nástroje HP QTP
Obrázek 23 - nastavení vlastností checkpointu, zdroj: autor z nástroje HP QTP
Obrázek 24 - vložení výstupní hodnoty z okna "Active screen", zdroj: autor z nástroje HP QTP . 49
Obrázek 25 - výběr vlastnosti objektu pro výstup, zdroj: autor z nástroje HP QTP
Obrázek 26 - vlastnosti výstupní hodnoty, zdroj: autor z nástroje HP QTP
Obrázek 27 - skript pro ukládání výstupních hodnot, zdroj: autor z nástroje HP QTP 50
Obrázek 28 - vzorové vyplnění vstupních hodnot v <i>Data Table</i> pro potřeby testu na ukládání
výstupních hodnot, zdroj: autor z nástroje HP QTP51
Obrázek 29 - zapsání výstupních hodnot do výsledků testu
Obrázek 30 - výsledný skript pro test na If funkci, zdroj: autor z nástroje HP QTP53
Obrázek 31 - vložení kroku Reporter.ReportEvent, zdroj: autor z nástroje HP QTP54
Obrázek 32 - interpretace reportované události ve výsledcích, zdroj: autor z nástroje HP QTP 54
Obrázek 33 - skript pro zalogování a odlogovaní z aplikace Flight reservation, zdroj: autor
z nástroje HP QTP 55
Obrázek 34 - rozdělení akcí tlačítkem "Split action", zdroj: autor z nástroje HP QTP55
Obrázek 35 - nastavení popisu nově vzniklých akcí, zdroj: autor z nástroje HP QTP 56
Obrázek 36 - závislost nově vytvořených akcí, zdroj: autor z nástroje HP QTP
Obrázek 37 - nastavení parametru, zdroj: autor z nástroje HP QTP
Obrázek 38 - Skript po přidání parametru Cislo_objednavky, zdroj: autor z nástroje HP QTP 58
Obrázek 39 - nastavení vlastností akce, zdroj: autor z nástroje HP QTP
Obrázek 40 - zavolání akce metodou <i>Call to Copy of Action</i> , zdroj: autor z nástroje HP QTP 61
Obrázek 41 - výběr akce "Login", která se bude volat, zdroj: autor z nástroje HP QTP61
Obrázek 42 - Přidání kopie akce Login s vlastním datasheetem, zdroj: autor z nástroje HP QTP 62
Obrázek 43 - výběr akce Logout, zdroj: autor z nástroje HP QTP
Obrázek 44 - přidání akce "Logout", zdroj: autor z nástroje HP QTP
Obrázek 45 - otevření vlastností volání akce, zdroj: autor z nástroje HP QTP65
Obrázek 46 - vlastnosti volání akce, zdroj: autor z nástroje HP QTP
Obrázek 47 - výsledky akce "VyhledaniObjednavek", zdroj: autor z nástroje HP QTP 66
93

Obrázek 48 - Local object repository, zdroj: autor z nástroje HP QTP
Obrázek 49 - funkce Highlight in application, zdroj: autor z nástroje HP QTP
Obrázek 50 - funkce Locate in repository, zdroj: autor z nástroje HP QTP
Obrázek 51 - operace s objekty v objektovém repositáři 69
Obrázek 52 - úprava vlastností objektů, zdroj: autor z nástroje HP QTP 70
Obrázek 53 - vložení objektu tlačítkem Add objects to local, zdroj: autor z nástroje HP QTP 71
Obrázek 54 - vytvoření Shared object repository volbou Export to local, zdroj: autor z nástroje HP
QTP
Obrázek 55 - přiřazení Shared object repository k akci, zdroj: autor z nástroje HP QTP
Obrázek 56 - přiřazení Shared object repository k akci, zdroj: autor z nástroje HP QTP
Obrázek 57 - editace Shared object repository, zdroj: autor z nástroje HP QTP
Obrázek 58 - detail objednávky v aplikaci Flight reservation, zdroj: autor z Flight Reservation 75
Obrázek 59 - přepnutí na analogové nahrávání, zdroj: autor z nástroje HP QTP
Obrázek 60 - spuštění analogového nahrávání, zdroj: autor z nástroje HP QTP
Obrázek 61 - provedení podpisu na podepiposovacím apletu, zdroj: autor z nástroje HP QTP 77
Obrázek 62 - přepnutí na Low level recording, zdroj: autor z nástroje HP QTP
Obrázek 63 - skript pro vyhledání objednávky nahraný v low level módu, zdroj: autor z nástroje
HP QTP
Obrázek 64 - vytvoření nové knihovny, zdroj: autor z nástroje HP QTP 79
Obrázek 65 - přiřazení nové knihovny k testu, zdroj: autor z nástroje HP QTP 80
Obrázek 66 - spuštění funkce, která zobrazuje vydefinovanou zprávu, zdroj: autor z nástroje HP
QTP
Obrázek 67 - editace objektu pole pro login v Object repository, zdroj: autor z nástroje HP QTP82

Tabulky

Tabulka 1 - porovnání výhod a nevýhod manuálního a automatizovaného přístupu	15
Tabulka 2 - Přehled vydaných verzí HP QTP (později HP UFT), zdroj: [9]	17
Tabulka 3 - vhodnost použití HP QTP v jednotlivých fázích testovacího procesu na řešene	ém
projektu	28