

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

DIPLOMOVÁ PRÁCE



MANAGEMENT FIREM

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

NÁZEV DIPLOMOVÉ PRÁCE/TITLE OF THESIS

Analýza možnosti využití automatizovaného testování v T-Mobile a.s.

TERMÍN UKONČENÍ STUDIA A OBHAJOBA (MĚSÍC/ROK)

6/2015

JMÉNO A PŘÍJMENÍ / STUDIJNÍ SKUPINA

Alena Ustohalová /MF13

JMÉNO VEDOUČÍHO DIPLOMOVÉ PRÁCE

Ing. Kamil Hofrichter, MBA

PROHLÁŠENÍ STUDENTA

Odevzdáním této práce prohlašuji, že jsem zadanou diplomovou práci na uvedené téma vypracoval/a samostatně a že jsem ke zpracování této diplomové práce použil/a pouze literární prameny v práci uvedené.

Jsem si vědom/a skutečnosti, že tato práce bude v souladu s § 47b zák. o vysokých školách zveřejněna, a souhlasím s tím, aby k takovému zveřejnění bez ohledu na výsledek obhajoby práce došlo.

Prohlašuji, že informace, které jsem v práci užil/a, pocházejí z legálních zdrojů, tj. že zejména nejde o předmět státního, služebního či obchodního tajemství či o jiné důvěrné informace, k jejichž použití v práci, popř. k jejichž následné publikaci v souvislosti s předpokládanou veřejnou prezentací práce, nemám potřebné oprávnění.

Datum a místo: V Praze dne 18. 4. 2015

PODĚKOVÁNÍ

Ráda bych tímto poděkovala vedoucímu diplomové práce, za metodické vedení a odborné konzultace, které mi poskytl/a při zpracování mé diplomové práce.

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

SOUHRN

1. Cíl práce:

Cílem této práce je vybrat konkrétní IT systémy a jejich regresní testovací scénáře, které je možné doporučit pro automatizaci. Sekundárním cílem je pak zhodnocení vybraných systémů z pohledu finanční výhodnosti pro společnost T-Mobile.

2. Výzkumné metody:

Byla provedena analýza s pomocí několika sad kritérií, nastavené vyhodnocovací logiky a indikátorů. Jedná se o kvalitativní výzkum a pro metodu usuzování byla použita induktivní metoda.

3. Výsledky výzkumu/práce:

Jako vhodné pro automatizaci byly analýzou vybrány čtyři IT systémy: e-shop/Portál, VCC a Siebel. Dohromady celkem 125 testovacích scénářů. Sekundárním cílem bylo zhodnotit finanční výhodnost realizace v podobě automatizace testů, výše uvedených systémů. Vzhledem k výsledným hodnotám indikátorů NPV = 416 tis. Kč, Návratnosti investice 2 roky a 5 měsíců a pozitivní EBITDA (851 000 Kč/rok) byl záměr vyhodnocen metrikami společnosti T-Mobile jako pro společnost výhodný.

4. Závěry a doporučení:

Primární i sekundární cíl práce byl naplněn. Analýza byla určena jako vstupní dokument pro realizaci projektu, který byl již v současné chvíli společností T-Mobile spuštěn. Doporučení této práce je zaměřit se na automatizaci regresních testů aplikací: E-shop/Portál, VCC a Siebel.

KLÍČOVÁ SLOVA

Automatizace testů, regresní testy, business case, nástroje pro automatizaci testů

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

SUMMARY

1. Main objective:

The aim of this work is to select a specific IT systems that can be recommended for the automation of regression tests. A secondary goal is to evaluate selected systems from the perspective of a financial benefit for the company as T-Mobile.

2. Research methods:

The analysis was done using several sets of criteria, set evaluation logic and indicators. This is a qualitative research method and reasoning was used inductive method.

3. Result of research:

Suitable for automation was four IT systems: e-shop / portal, VCC and Siebel. Altogether, a total of 125 test scenarios. The secondary objective was to evaluate the financial benefits realization, in the form of test automation systems mentioned above. Due to the resulting values of indicators NPV = 416 thousand. CZK, ROI 2 years and 5 months and positive EBITDA (851 000 CZK / year) was evaluated as a plan for T-Mobile as advantageous.

4. Conclusions and recommendation:

Primary and secondary objective was achieved. The analysis was intended as an input document for the project, which was already at the present time by T-MOBILE running. The recommendations of this work is to focus on automating regression test applications: e-shop / portal, VCC and Siebel.

KEYWORDS

Test automation, Regression tests, Business case, Tools for test automation

JEL CLASSIFICATION

M15 – IT Management, M21 – Business Economics

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

Vysoká škola ekonomie a managementu

Nárožní 2600/9a, 158 00 Praha 5

ZADÁNÍ DIPLOMOVÉ PRÁCE

Jméno a příjmení:	Alena Ustohalová
Studijní program:	Ekonomika a management (Ing.)
Studijní obor:	Management firem
Studijní skupina:	MF 13
Název DP:	Analýza možnosti využití automatizovaného testování v T-Mobile a.s.
Zásady pro vypracování (stručná osnova práce):	<ol style="list-style-type: none">1 Úvod2 Cíl a metodika3 Teoreticko-metodologická část<ol style="list-style-type: none">3.1 Princip automatizovaných testů3.2 Testování aplikací4 Analytická / praktická část<ol style="list-style-type: none">4.1 Analýza vhodnosti využití automatizovaných testů v TMCZ4.2 Náklady na testování a business case5 Doporučení6 Závěr
Seznam literatury: (alespoň 4 zdroje)	<ul style="list-style-type: none">• HEUSSER, M., FEWSTER, G., PAUL, J. <i>How to reduce the cost of software testing: case studies of software test automation</i>. Boca Raton, FL: CRC Press, 2012, xxvii, 312 p. ISBN 14-398-6155-2.• GRAHAM, D., FEWSTER, M., PAUL, J. <i>Effective Software Test Automation Developing an Automated Software Testing Tool: case studies of software test automation</i>. Upper Saddle River, NJ: Addison-Wesley, 2012, lii, 617 p. ISBN 03-217-5406-9.• DUSTIN, E., RASHKA, J., PAUL, J. <i>Automated software testing: introduction, management, and performance</i>. Reading, Mass.: Addison-Wesley, 1999, xxi, 575 p. ISBN 02-014-3287-0.• MOSLEY, D. J., POSEY, B., PAUL, J. <i>Just enough software test automation: case studies of software test automation</i>. Upper Saddle River, NJ: Yourdon Press, 2002, xix, 260 p. ISBN 01-300-8468-9.
Harmonogram	<ul style="list-style-type: none">• Zpracování cílů a metodiky do 30. 11. 2014• Zpracování teoretické části do 31. 12. 2014• Zpracování výsledků do 30.03.2015• Finální verze do 01.05.2015
Vedoucí DP:	Ing. Kamil Hofrichter, MBA

V Praze dne 7. 11. 2014

Prof. Ing. Milan Žák, CSc.
rektor

Milan
Žák

Digitálně podepsal Milan Žák
DN: c=CZ, cn=Milan Žák,
o=Vysoká škola ekonomie a
managementu, s.p.a.,
email=zak@vsem.cz,
serialNumber=KA - 10107655
Datum: 2014.11.07 11:17:25
+0100

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

Obsah

1. Úvod.....	8
1.1. Metodika a cíle práce	2
2. Teoretická část.....	3
2.1. Testování aplikací	3
2.1.1. Chyby v systému	3
2.1.2. Testování aplikací - vývoj SW	5
2.1.3. Druhy softwarových testů	8
2.1.4. Obecné problémy.....	9
2.1.5. Práce testera.....	10
2.2. Princip automatizovaných testů	12
2.2.1. Případy použití automatizovaného testování	13
2.2.2. Stručný přehled automatizovaných nástrojů	14
2.2.3. Výhody automatizovaného testování.....	15
2.2.4. Shrnutí.....	16
3. Praktická část.....	18
3.1. Představení společnosti T-Mobile.....	18
3.1.1. Způsob testování ve společnosti T-Mobile	18
3.1.2. Nástrojová podpora testování.....	22
3.2. Analýza vhodnosti využití automatizovaných testů v T-MOBILE	22
3.2.1. Export systémů z CMDB a prvotní evaluace	24
3.2.2. Analýza systémů – regresně testovaných	26
3.2.3. Výběr testovacích scénářů nevhodných k automatizaci	29
3.2.4. Výběr vhodných testovacích scénářů pro automatizaci.....	34
3.2.5. Shrnutí výsledků analýzy	37
3.2.6. Doporučení	39
3.3. Náklady na testování a business case	40
3.3.1. Přínosy projektu.....	41
3.3.2. Cíle projektu.....	41
3.3.3. Náklady a přínosy projektu	42
3.3.3.1. Přínosy.....	43
3.3.3.2. Náklady.....	44
3.3.4. Výpočet business Case - shrnutí.....	45
4. Závěr	49

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

Seznam Obrázků

Obrázek 1 Příklad množství chyb, úměrně k počtu provedených testů.....	4
Obrázek 2 V-Model - životní cyklus (The V Life Cycle).....	7
Obrázek 3 Příklad vodopádového modelu	9
Obrázek 4 Postup analýzy	23
Obrázek 5 Výsledek analýzy	39

Seznam Tabulek

Tabulka 1 Testování ve společnosti T-Mobile.....	19
Tabulka 2 Závažnost chyb	21
Tabulka 3 Regresně testované systémy	24
Tabulka 4 Kritéria výběru systémů vhodných k automatizaci	26
Tabulka 5 Systémy vhodné k automatizaci.....	28
Tabulka 6 Kritéria výběru testovacích scénářů nevhodných k automatizaci.....	29
Tabulka 7 Doporučení testovacích scénářů k automatizaci	33
Tabulka 8 Kritéria výběru testovacích scénářů vhodných k automatizaci z pohledu efektivnosti	34
Tabulka 9 Doporučení testovacích scénářů k automatizaci z pohledu efektivnosti	36
Tabulka 10 Cíle projektu.....	41
Tabulka 11 Podklady pro výpočet business case.	43
Tabulka 12 zautomatizování manuálního testování – přínos.....	44
Tabulka 13 Správa defektů - přínos.....	44
Tabulka 14 Nákup licencí	45
Tabulka 15 Výpočet CAPEX	46
Tabulka 16 Výpočet OPEX.....	46
Tabulka 17 Business Case.....	48

1. Úvod

Psaní této diplomové práce probíhá v době, kdy se všechny soukromé podniky zaměřují na aktivity, které přináší úsporu, a to nejen finanční, ale i časovou. Tato práce je aplikována na telekomunikační společnost T-Mobile a.s. A telekomunikace ve výše uvedené snaze nejsou výjimkou. Tento obor se za posledních 20 let velmi rozvíjel a namísto nákladového šetření probíhala spíše podpora nových investičních záměrů, které se týkaly nejen rozvoje samotné síťové infrastruktury, ale i investic do nových softwarových řešení. Rozvoj síťové infrastruktury je i nadále podporován nemalými finančními prostředky (výstavba sítí nové generace – LTE), ale softwarových řešení a IT podpory se změny na českém telekomunikačním trhu velmi dotkly, a to nejen společnosti T-Mobile, nad kterou je provedena níže uváděná analýza. IT strategie společností směřuje k vysoké míře automatizace, a to k automatizaci činností, kde automatizace vytváří přidanou hodnotu. Trend posledních let zmiňuje i Williams (2012, str. 22), který uvádí, že v následujících letech se bude informační strategie společností ubírat směrem, který říká: automatizujme takové činnosti, které jsou opakovatelné, snadno definovatelné a ušetříme tak lidské znalosti a dovednosti pro využití, které bude podniku přinášet skutečnou přidanou hodnotu.

Důvod nákladového šetření a optimalizace ve společnosti T-Mobile je založen na faktu, že ceny produktů a služeb v telekomunikacích výrazně klesají (-8,7 % - v meziročním měření), zatímco spotřeba služeb stoupá (objem dat přenesených nejrychlejší českou sítí meziročně vzrostl o 68 %). Společnosti se podařilo dosáhnout v roce 2014 uspokojivých finančních výsledků. Vše však v důsledku toho, že operátor částečně kompenzoval klesající tržby silným důrazem na efektivitu veškerých aktivit, a to výhradně proto, aby nemusel snížit investice do rozvoje technologií a služeb pro zákazníky (T-Press, 2015).

Tato práce se zabývá analýzou možností využití automatizovaného testování ve společnosti T-Mobile. Cílem je nejen navrhnout IT systémy, které je možné testovat automatizovaně, ale i výpočet business case nad projektovým záměrem, který vyplyne z analýzy systémů a testovacích scénářů. Tato práce má tematickou strukturu, rozdělenou do dvou částí - první část je zaměřena na definici pojmů: automatizovaných testů a základní principy testování softwaru. Nechybí ani vymezení výhod a nevýhod automatizovaného testování a popis možností využití pokročilých nástrojů pro automatizování testů. Práce je zaměřena na teoretickou část, která pojednává o výše uvedených principech a na část praktickou, kde je provedena analýza současného stavu regresního testování společnosti T-Mobile (společnost EI Prague (2014, str. 1) definuje regresní testy následujícím způsobem: „*Cílem regresního testování je zajistit, aby změny software, jako je přidávání nových funkcí, nebo úpravy stávajících funkcí, neovlivnily nepříznivě funkcionální ostatních částí aplikace, na které změny nemají mít vliv.*“), z níž budou posléze vyvozena doporučení konkrétních systémů vhodných pro automatizované testování. Na základě výše uvedených výsledků budou shromážděny informace a číselné údaje nutné pro výpočet business case, který bude sloužit k obhájení či neobhájení záměru reálné implementace ve výše zmíněné společnosti.

Tato práce je založena na reálném požadavku IT oddělení společnosti T-Mobile, kde autorka již šest let pracuje. Tato analýza bude využita jako podklad pro operativní rozvojové aktivity naplánované IT oddělením pro období 2015 - 2017. Cílem těchto rozvojových aktivit je nejenom nutná úspora nákladů na personální obsazení, ale i zvýšení kvality vývojové implementace testovaných systémů: zvýšení kvality je poté i

součástí strategického cíle společnosti, který prosazuje implementaci takových řešení, která povedou ke zvýšení hodnoty nabízeného produktu pro zákazníka.

1.1. Metodika a cíle práce

Struktura samotné práce je nastíněna v první kapitole: Úvod. V teoretické části práce bude využito sekundárních a primárních pramenů. Sekundární prameny budou použity pro odborné definování termínů. Z primárních zdrojů bude poté čerpáno pro samotnou obsahovou část práce. Většina primárních pramenů není přeložena do českého jazyka a vyskytují se tak pouze v anglickém originále. Tyto informace z uvedených pramenů jsou zkompileovány a porovnány tak, aby bylo dosaženo uceleného teoretického přehledu o problematice testování a možnosti automatizovaného testování, který bude sloužit pro analytickou část práce. Teoretický oddíl práce je rozdělen na dvě části. První z nich, nazvaná "Princip automatizovaných testů", se zabývá teorií. Najdeme v ní základní principy pro testování software, stručný přehled známých nástrojů pro automatizované testování, výhody a nevýhody automatizovaného testování. Ve druhé části, nazvané "Testování aplikací pomocí automatizace", najdeme také typy testů a jejich popis. V rámci této kapitoly je možné ukázat problémy, ale i výhody automatizace testů.

Praktická část práce bude strukturována do dvou podkapitol. V první podkapitole bude provedena analýza současného stavu regresního testování společnosti T-Mobile. Pro relevantnost výzkumu bude provedeno šetření s konkrétními testery a odborníky na automatizaci testování. Realizace bude spočívat v nastavení kritérií, které po vyplnění testery či na základě objektivních ukazatelů, poukážou na vhodnost či nevhodnost systému k automatizaci. Tato analýza v její druhé a třetí části bude zaměřena již na konkrétní testovací scénáře, kde bude snaha pomocí nastavených kritérií zhodnotit technickou možnost realizace a její případnou efektivnost. Celou tuto podkapitolu budou provázet seznamy nastavených kritérií, indikátorů a vyhodnocovací logiky. Byl proveden kvalitativní výzkum a pro metodu usuzování byla použita induktivní (neúplná) metoda

Na výsledky této části bude navazovat podkapitola s názvem Business Case záměru, kde bude finančně znázorněna nákladová a přínosová stránka případné implementace do reálného prostředí společnosti. Součástí této kapitoly bude vzhledem k silné vazbě na nákladovou stránku záměru proveden průzkum softwarových produktů určených pro automatizaci testů. Tato část bude vypracována na základě konkrétních nabídek dodavatelů SW, kteří budou poptáni pro dodání podkladů v rámci výpočtu reálných nákladů.

Cílem této práce je vybrat konkrétní IT systémy, které je možné doporučit pro automatizaci jejich regresních testů. Sekundárním cílem je pak zhodnocení vybraných systémů z pohledu finanční výhodnosti pro společnost T-Mobile.

2. Teoretická část

Předkládaná práce si klade za cíl seznámit čtenáře s tématem obecného testování software a následně konkrétně se složitou oblastí automatizovaných testů, včetně specifických nástrojů, které jsou určeny pro tvorbu automatizovaných testů.

Teoretický oddíl práce je rozdělen na dvě části. První z nich, nazvaná "Testování aplikací" se zabývá teorií a lze v ní nalézt základní principy pro testování software, jednotlivé typy a druhy testů. Ve druhé části nazvané "Princip automatizovaných testů" je stručně popsán princip automatizovaných testů, výhody a nevýhody automatizovaného testování a přehled známých nástrojů pro automatizované testování. V rámci této kapitoly je poukázáno nejen na problémy, ale i na výhody spojené s automatizací testů – převážně tato část teorie bude použita v analytické části práce.

2.1. Testování aplikací

Programové vybavení počítače nazýváme pojmem software. V informatice jsou jako software označovány veškeré programy zanesené v počítači, které provádí nějaké činnosti a jsou využívány ke konkrétním úkonům. Software můžeme dále dělit na dvě kategorie: zaprvé jde o systémový software (např. jádro a další součásti operačního systému), který slouží k zajišťování provozu počítačové sestavy a jejího nejbližšího okolí, např. připojených zařízení. Za druhé jde o aplikační software, neboli aplikaci (např. textový editor, grafický program atd.), se kterým zpravidla pracují počítačová uživatelé. Proto musí mít aplikace uživatelské rozhraní, ve kterém dochází k interakci mezi člověkem a počítačem (Kolář, 2005). Toto rozhraní Graphical User Interface je označováno anglickou zkratkou GUI.

Pokud má být software dostatečně kvalitní, musíme naplnit jistá kritéria, která jsou na něj kladena. Norma IEEE Std 610.12-1990 popisuje kvalitu jako stupeň, do jaké míry splní systém, komponenta anebo proces specifikované požadavky, případně splní zákaznické nebo uživatelské potřeby a jejich očekávání. (IEEE Standard Glossary of Software Engineering Terminology, 1990, online). Zde tedy vidíme, že správný software bude natolik kvalitní, do jaké míry bude účelný a použitelný pro lidi, kteří s ním budou pracovat. S tímto tvrzením souhlasí i Klimeš, Procházka (2011, str. 70), kteří tvrdí, že kvalita dodávky leží v určitém místě klasického trojimperativu s vrcholy cena, termín a požadavky. Kdy poměr těchto složek určuje výslednou kvalitu, nákladovost a čas dodání produktu.

Pohledy na kvalitu software mohou být velmi rozdílné, protože každý - ať již jde o programátora, testera nebo uživatele - má rozdílné požadavky. Pro zajištění kvality software je nutné jej podrobit procesu testování, ve kterém můžeme odhalit případné chyby v systému (Vymětal, 2009, str. 81). Jako nezbytnou, popisuje fázi testování i Bruckner, Voříšek, Buchalceová a kol. (2012, str. 217).

Se softwarem je neodmyslitelně spojen proces testování, který slouží dle Hýska (2014) zejména pro zajištění důkladné kontroly kvality produktů a také pro nalezení a korekci případných chyb, které by se v něm mohly na některých místech vyskytovat.

2.1.1. Chyby v systému

Samotný pojem "testování" bychom mohli popsat jako komplexní proces, jehož cílem je ověřit nepřítomnost chyb v softwarovém systému.

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

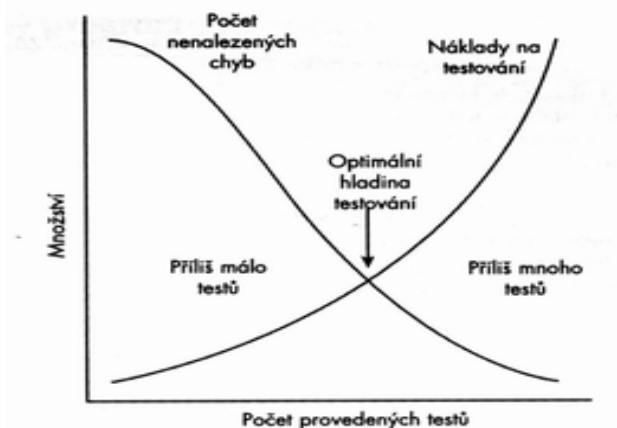
Známý profesor automatizace a inteligentních inženýrských systémů Ron Patton (2002, str. 13) přišel s obsáhlou definicí toho, co podle něj představují chyby v systému. Patton (2002, str. 14) uvádí, že o chybu v systému se jedná tehdy, pokud platí jedna nebo více z následujících skutečností:

- Software nedělá něco, co by podle specifikace dělat měl.
- Software dělá něco, co by podle specifikace produktu dělat neměl, nebo dle Janička, Marka a kol. (2013, str. 202) – chyby ve formulaci specifikace.
- Software dělá něco, o čem se specifikace nezmiňuje.
- Software nedělá něco, o čem se produktová specifikace nezmiňuje, ale měla by se zmiňovat.
- Software je obtížně srozumitelný, těžko se s ním pracuje, je pomalý nebo (podle názoru testera softwaru) jej koncový uživatel nebude považovat za správný (Patton, 2002, str. 14).

Chyby v systému, coby nežádoucí prvky, je třeba co nejrychleji najít a odstranit ze systému. Patton (2002, str. 16) dále uvádí, kde může nejčastěji dojít k chybám a jaké bývají jejich zdroje: Chyby v programovém kódu vznikají nejčastěji proto, že aplikace je příliš složitá a máme k dispozici jen nedostatečnou dokumentaci. Určitou roli může samozřejmě hrát i stres programátorů a nedostatek času. Oba tyto faktory vedou ke vzniku chyb z nepozornosti (Patton, 2002, str. 16). Indičtí autoři Srinivasam a Gopalaswamy (2006, str. 9) navíc dodávají, že chyby nalezené v testovací fázi je nutné opravit nejen v kódu, ale i v celé doprovodné dokumentaci a požadavcích zadavatele, což je nejen nepříjemné, ale i nákladné.

Dle odborných studií, například autorů Janička, Marka a kol. (2013, str. 201), je známo, že mezi nejběžnější zdroje chyb patří především nedostatečná specifikace, či spíše to, že specifikace v programu úplně chybí. Mezi další zdroj častých chyb patří vedle chybějící specifikace příliš vágní návrh systému. Pokud není návrh dostatečně podrobný, může se stát, že nebude dobře srozumitelný ani pro členy týmu, kteří na něm budou pracovat. Potom bude pravděpodobně docházet k častým změnám, které se negativně projeví na výsledku. Následující obrázek číslo 1 ukazuje efektivní bod testování, kdy náklady na odhalování chyb jsou adekvátní počtu chyb skutečně odhalených.

Obrázek 1 Příklad množství chyb, úměrně k počtu provedených testů



Zdroj: <http://www.linuxzone.cz/pictures/chyby.png>

Mohli bychom se však neprávem domnívat, že vysoká kvalita software se odvíjí od nízkého počtu nalezených chyb. To ovšem není pravda. Pokud nejsou testy

provedeny správně, nemusí být odhaleny všechny chyby, nebo jen jejich podstatná část. Na tomto místě vidíme, že i kvalita samotného testování může být velice subjektivní pojem. Bez testování však výsledné produkty mají pouze nahodilou kvalitu (Newton, 2008, str. 226).

Měli bychom mít na paměti, že v praxi není reálně možné provést kontrolu všech vstupních a výstupních dat, protože jich existuje až příliš velké množství (Mařík, 2007, str. 34).

Proto není nikdy možné provést softwarové testování, které by bylo schopné ověřit veškerá chybná místa nacházející se v testovaném systému. Stejně tak je naneštěstí nereálná i průkazná kontrola veškerých cest v programu.

Testování je velmi efektivní cestou, jak můžeme odhalit skryté chyby. Bohužel však nefunguje takovým způsobem, aby bylo schopné prokázat, že žádné chyby neexistují. Pokud nebudou odhaleny žádné chyby, není to ještě žádný důvod, abychom si automaticky mysleli, že žádné chyby ani neexistují. Dijkstra (1972, str. 56) uvádí, že testování programu může být velmi efektivním způsobem, jak prokázat přítomnost chyb, ale je naprosto nevhodné k prokázání jejich nepřítomnosti.

Předpoklad o neexistenci chyb můžeme brát pouze jako nepřímý, formální důkaz o kvalitách softwarového programu. Formální důkaz je však věc označovaná za těžko proveditelnou v běžné praxi. Chyba je cokoli, co snižuje hodnotu programu. Chyba je však také to, jestli se nám nepodaří splnit očekávání uživatele. Jelikož naším cílem by mělo být v první řadě splnit jeho požadavky, pak je právě toto velmi důležité. Čím dříve se nám podaří najít chyby v programu, tím nižší budou i náklady na opravy. S touto myšlenkou souhlasí autoři Desikan a Ramesh (2006, str. 10). Právě proto má softwarové testování v současné době tak velký význam. Zároveň výše uvedení autoři Desikan a Ramesh (2006, str. 26) uvádí, že čím častější a komplexnější je test, tím je větší šance přiblížit produkt požadavkům zákazníka.

2.1.2. Testování aplikací - vývoj SW

Testování je nezbytným elementem životního cyklu vývoje software. Cílem testování obvykle bývá ověřit, že software dělá přesně to, co je uvedeno ve specifikaci, dodávka je kvalitní a neohrožuje ostatní SW prostředky (Burnstein, 2003, str 193). Desikan a Ramesh (2006, str. 27) dodávají, že nejdůležitější je spokojenost zákazníka, která je závislá na jeho požadavcích přetransformovaných v SW dodávku.

Začít se psáním testů je možné prakticky kdykoli, musíme však mít k dispozici dokumenty SRS (Software Requirements Specification). Nutný je seznam funkčních a nefunkčních požadavků, na kterých budou testy založeny. Podle těchto požadavků můžeme vytvořit testovací případy a scénáře pro různé typy testů, včetně testů automatizovaných.

Cyklus softwarového testování má nejčastěji tyto základní části:

- vytvoření testovacích plánů, modifikace relevantní dokumentace, vytvoření testovacích scénářů,
- identifikace regresních testů,
- provedení testů podle již vypracovaných plánů a scénářů,
- vyhodnocení informací a dat získaných z procesu testování,
- a opětovné provedení sporných testů včetně plánů, jak postupovat nadále (Klimeš, Procházka, 2011 str. 51).

V závislosti na typu požadavků lze testování rozdělit na čtyři základní typy: funkční testování, nefunkční testování, testování SW struktury – strukturní testování a regresní testování (Graham, Veendaal, Evans, 2008, str. 46-48). Paclík na své webové stránce bugtracker.cz (2014) uvádí pouze první dva typy, a to funkční a nefunkční testování. Strukturní testování, regresní testování a další testy - jako například zátěžové - uvádí jako jednotlivé druhy testů. Tyto druhy testů jsou pro přehlednost ve speciální kapitole 2.2.1.

Funkční testy jsou určeny pro nalezení rozdílů mezi stávající implementací aplikace a funkčními požadavky, které byly změnou implementovány (Paclík, 2015). Funkční testy můžeme provést manuálně či použít speciální nástroje. V případě, že funkční testy chceme provést manuálním způsobem, vytváří situace a stavy aplikace tester sám (Patton, 2006, str. 198).

Vedle funkčních testů existují i **nefunkční testy**, které mají ověřit, zda je aplikace bezpečná a spolehlivá (Paclík, 2015). V rámci jednotlivých druhů testování se do této části dle Paclíka (2015) zahrnují testy zátěžové a stress testování, které jsou taktéž popsány v kapitole 2.2.1. Autoři Graham, Veenendaal a Evans (2008, str. 218) dodávají, že nefunkční testování se provádí s primárním cílem ověřit spolehlivost, účinnost, využitelnost a udržitelnost.

Cílem funkčních testů je ověřit implementaci (zavedení) funkcí v systému, ale také to, zda je systém vůbec bezpečný. **Funkční testy** se používají na požadavky zaměřené na funkčnost aplikace. Definují způsob, jakým bude aplikace užívána. Funkční požadavky bývají zpracovány do způsobů užití, neboli UseCase. Ze způsobu užití vycházejí testy, které ověřují chování aplikace v různých případech užití. Funkční testy tedy slouží především k odhalení neshod mezi aktuální aplikací proti funkčním požadavkům (Graham, Veendaal, Evans, 2008, str. 58).

Funkční testy můžeme provést manuálně či použít speciální nástroje. V případě, že funkční testy chceme provést manuálním způsobem, vytváří situace a stavy aplikace tester sám (Procházka, Klimeš, 2011, str. 51). V případě, kdy je použit nástroj pro automatizované testování, jsou dle Galina (2004, str. 238) výše uvedené stavy a situace automaticky simulovány. Automatizované je v tomto případě i generování případných chybných stavů.

V oblasti tzv. nefunkčních testů je vliv automatizace ještě větší, protože nefunkční testy by byly bez využití příslušných nástrojů jen obtížně proveditelné, ne-li zcela nemožné. Tyto testy lze bez automatizovaného nástroje jen těžko realizovat.

Dle Burnsteina (2003, str. 199) proces testování prochází několika stádii vývoje. Každá z jednotlivých fází se staví na nové, vylepšené verzi dosavadní aplikace. Testování lze rozdělit do následujících stádií podle toho, v jakém časovém rozmezí od zapsání zdrojového kódu se budeme pohybovat:

- Jednotkové testování – unit testy,
- Integroční testování,
- Systémové testování,
- Akceptační testování.

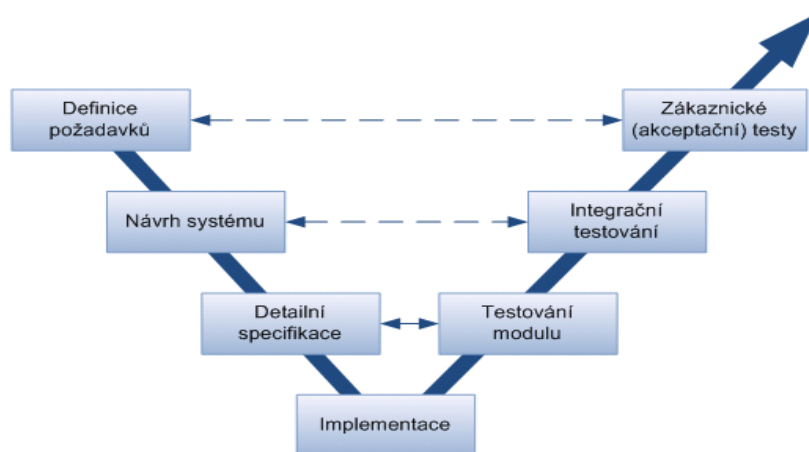
Unit test neboli jednotkové testování je dle autorů Mosley a Posey (2002, str. 101) testování konkrétních ucelených jednotek, kde například v případě objektově

orientovaného programování mluvíme o testování konkrétních tříd a metod. Integrovaní testování pak následuje po úspěšně provedených unit testech a často i po testech systémových, které mají za úkol odhalit případná pochybení v implementaci v rámci komunikace mezi jednotlivými moduly, systémy či hardwarem (Graham, Veendaal, Evans, 2008, str. 216). Procházka, Klimeš (2011, str. 109) dodává, že součástí integrovaního testování je i ověřování správnosti dat posílaných mezi jednotlivými systémy. Portál testovanisoftware.cz uvádí, že kromě systémových testů jsou vhodné k automatizaci i integrovaní testy. S tímto názorem však ne zcela souhlasí Mosley a Posey (2002, str. 100), kteří u integrovaního testování s velkou mírou integrací vidí zásadní problém v nestabilitě testovacích a ověřovacích dat.

Systémové testy ověřují aplikaci z pohledu zákazníka. Na základě předem definovaných testovacích scénářů jsou testovány různé případy užití, které vychází ze specifikace zadané uživatelem (testování softwaru.cz, 2014). Akceptační testování je autory Procházka, Klimeš (2011, str. 52) popsáno následujícím způsobem: „Tato fáze se zaměřuje na zákaznické testování (možné provádět třetí stranou) plně integrované změny v systému a zahrnuje funkční a regresní testy a také testy interoperability (integrovaní).“ Vymětal (2009, str. 95) s výše uvedeným autorem souhlasí a dodává, že akceptační testy jsou realizovány zpravidla uživateli a IT personálem. Důležitou částí akceptačního testování je doporučení či nedoporučení funkcionality k nasazení do ostrého provozu na produkčních systémech.

Havlíčková (2009) navíc přidává ještě jedno stadium, a to pilotní testování, které je dle autorky určeno pro uvěření správnosti implementované funkcionality v reálném prostředí (na produkci). Níže uvedený obrázek číslo 3 poukazuje na vazby jednotlivých druhů testování na jednotlivé fáze softwarového vývoje.

Obrázek 2 V-Model - životní cyklus (The V Life Cycle)



Zdroj: <http://www.umel.feec.vutbr.cz/bdts/images/mmu/modelIV.PNG>

Je nutné pochopit, jaké cíle má software, který tvoříme. Také se musíme zamyslet nad tím, pro jakého koncového uživatele je vlastně software určen (Crispin, Gregory, 2009). Než tedy začneme s procesem vývoje testů, musíme nejprve ustanovit měřítko kvality. Také musíme rozhodnout o přístupu, ze kterého budeme vycházet. To, jaký implementační postup vybereme, bude mít vliv na rozsah testování a cenu testovací fáze.

Proces testování tvoří významnou část procesu ověřování kvality, který lze naplánovat. Testování a jeho způsob je založen na metodice testování dané společnosti a používaných nástrojích pro podporu testování. Dále pak na porozumění produktu a zákazníků, pro které se produkt vypracovává. Testování lze popsat také jako dynamický proces kontroly, zda chování programu odpovídá specifikaci (Abran, 2004).

Z hlediska praxe by bylo nejlepší, kdybychom si hned od začátku ujasnili přesnou definici záběrů testů. Šíří záběru však můžeme během času ještě doladit a upravit dle našich aktuálních potřeb. Test by měl v prvé řadě sloužit k tomu, aby nám mohl pomoci určit cíle aplikace. Svůj účel ale splní až v okamžiku, kdy bude dokončen výsledný softwarový produkt.

V metodice a v metrice kvality musí být zahrnut mimo jiné i jednoznačný popis práce, který je nezbytný pro zpětné vyhodnocení díla. Popis práce lze využít pro zvýšení kvality projektu, ale také do budoucna jako jedno z vodítek pro tvorbu budoucích děl. Proces testování je ovlivněn složením testovacím teamu, který se skládá z vývojářů, analytiků a testerů. Testování je tedy závislé na sesbíraných zkušenostech členů týmu, jejich pracovním nasazení a stupních motivace, které vykazují (Patton, 2002, str. 22).

2.1.3. Druhy softwarových testů

Testy a různé techniky testování obecně rozeznáváme podle toho, na jaké konkrétní věci se zaměřují, a co je vlastně předmětem jejich testů.

Dělit je však můžeme dle Paclíka (2015) například na **testy dynamické**, které se provádí na aplikaci, která je v právě v provozu- jde tedy o testování velmi aktuální - a **testy statické** na neaktivních částech systému. Naproti tomu **regresivní testování** se provádí až po vyhledání a opravách chyb, jde více do hloubky a znovu ověřuje systémové funkce, zatímco **progresivní testování** se zaměřuje na ověření nových funkcí aplikace.

Rozdíl mezi statickým a dynamickým testováním spočívá ve stavu objektu, který je předmětem testů. Statické testování spočívá ve zkoumání a kontrole projektové dokumentace a následné analýze získaných dat. Jde tedy o test, který pracuje s prvky, jež nevykazují prakticky žádnou aktivitu. **Dynamické testy** však zkoumají aktivní části programu. Dynamické testování spočívá v kontrole vstupů a výstupů aplikace, mimo to se však soustřeďuje i na komunikaci klienta s programem, se kterým přichází do styku (Graham, Veendaal, Evans (2008, str. 58).

Dalším druhem testů jsou **regresní testy**. Ty mají ověřit, jestli nedošlo k nějakému zásahu do provozu různých částí programu (např. ve funkčních ohledech nebo přidáním nějakého nového prvku do systému), které mají zůstat stejné a neměnit se (Procházka, Klimeš, 2011, str. 52).

Testy výkonu (performance testy) mají ověřit, jak systém funguje v případě, že je silně zatížen. Protože nefunkční testy v sobě zahrnují testování výkonu, zvládnutí zátěže, zda jsou použitelné a stabilní, jde vlastně o test, který má ověřit, jak celkově systém pracuje (Procházka, Klimeš, 2011, str. 31).

Velice specifickým druhem testů je tzv. "**Smoke test**". Smoke test provádí všechny předem naplánované testovací procesy (př. beta testy, anebo průzkumné testy), aby bylo v průběhu těchto testů možné ověřit, zda všechny podstatné části systému pracují tak, jak správně mají. Cílem smoke testu je zjistit, jestli je aplikace již připravena na provedení další etapy testů Mosley a Posey (2002, str. 101). Graham, Veenendaal,

Evans (2008, str. 221) souhlasí s výše uvedenou definicí smoke testů a navíc dodávají, že by smoke testy (nazvané taktéž sanity testy) měly být prováděny primárně nad hlavními funkcionalitami systému a jejich každodenní provádění je součástí doporučených praktik testování. Mosley a Posey (2002, str. 101) dodávají, že jsou smoke testy vzhledem k jejich časté opakovatelnosti a nízkému počtu změn vhodné pro automatizaci.

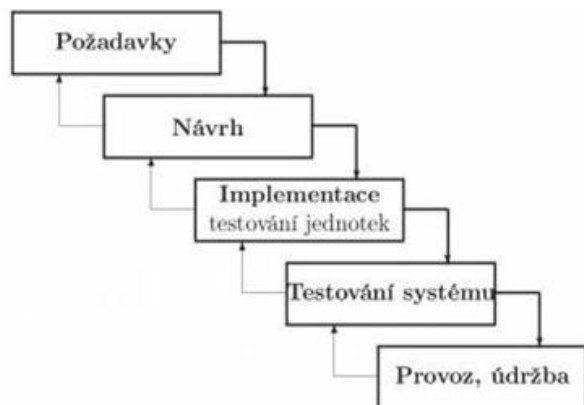
Testy, které jsou postavené na datech (to znamená, že proces testů se nemění, různí se pouze vstupní a výstupní data), jsou vhodné pro automatizaci. Mnoho nástrojů pro automatizované testování se nejlépe využívá právě pro datově řízené testy. Smoke testy jsou další oblastí, která je vhodná pro automatizaci. Typický smoke test totiž zpravidla obnáší jen jedno procházení programem. Protože toto základní procházení se nemění, není nutné kvůli němu vypracovávat stále nové způsoby testování. Proto je možné smoke testy snadno automatizovat. Smoke testy ověřují, že program je dobře nainstalovaný a proběhla v něm správná konfigurace, tedy správné nastavení ještě před začátkem dalšího testování (Graham, Veenendaal, Evans, 2008, str. 221).

Další dělení může být podle způsobu provedení. Můžeme tedy mít buď **testy manuální**, ručně zapsané (např. testy použitelnosti), anebo **testy automatizované**, které nám umožňují práci s velkými objemy dat (např. testy zátěžové). Nelze říci, který způsob testování je nejlepší. Je zřejmé, že každý způsob testování má své opodstatnění v určité fázi vývoje daného systému, jeho složitosti a rozsahu. Do značné míry je volba vhodného způsobu testování ovlivněna i možnostmi a schopnostmi zadavatele a dodavatele testů. Vždy je třeba zvážit pro koho a k čemu bude daná aplikace používána (Galín, 2004, str. 216). Automatizovaným testováním se pak zabývají kapitoly v dalších částech práce.

2.1.4. Obecné problémy

Tým se společně snaží vyvinout kvalitní produkt, který by zákazníkům přinášel prospěch, protože jednou z přirozených motivací lidí je dobrý pocit z dobře vykonané práce. Dosažení vysoké kvality je třeba naplánovat a celý proces pečlivě řídit. Abychom mohli aplikaci testovat, musí napřed existovat jisté specifikace, podmínky, či pravidla, na jejichž základě můžeme přistoupit k provedení kontroly toho, zda jsou v uspokojivé míře splněny. Na obrázku číslo 3 níže jsou znázorněny typické fáze vodopádového modelu vývoje, kde jsou vidět konkrétní fáze, které předchází a následují fází testovací.

Obrázek 3 Příklad vodopádového modelu



Zdroj: http://www.szz.g6.cz/lib/exe/fetch.php?w=400&media=temata:30zivotni_cyklus_softwaru:vodopadovy_model.jpeg, 2012.

Na začátku vývoje projektu musíme nejprve stanovit měřítka kvality, podle kterých se budeme nadále řídit. Tato měřítka se mohou různit, vždy bychom podle nich však měli brát zřetel na přání a potřeby klientů. Poté musíme zvolit vhodný přístup, který budeme využívat a zhodnotit možnosti jeho zavedení (implementace). Musíme mít jistotu, že výsledek vývojového procesu, neboli konečný produkt, dosahuje požadované kvality (Hailpern, 2002).

Jakmile se nám podaří stanovit pokrytí testovacích procesů, můžeme začít se sběrem dat, výběrem testovacích nástrojů a typů testů. Všechny požadavky na aplikaci musí být jednoznačně specifikované tak, abychom později mohli podle předem připravených kritérií ověřit, zda se nám všechny požadavky skutečně podařilo splnit. Tato kontrola je velice důležitá a neměli bychom ji za žádnou cenu vynechat. Výjimkou nejsou ani automatizované testy.

Pro úspěšný proces testování je potřebné stanovení jednoznačně specifikovaných cílů, ke kterým testování povede. Musíme si přesně formulovat to, čeho se snažíme dosáhnout. Proces testování zasahuje do celého vývoje projektu a hraje v něm nezastupitelnou úlohu hned z několika důvodů. Všechny požadavky na produkt musí být specifikovány v takové formě, abychom mohli vyčerpávajícím způsobem zkontrolovat, zda byly správně splněny. Samotné testování je zkoumání na různých úrovních programu a analýza nalezených jevů (testovanisoftware.cz, 2015).

Mezi nejběžnější problémy v oblasti softwarového testování patří následující:

Obecné problémy dle autorů Janíček, Marek (2013, str. 201) a Tilley, Floss (2014, str. 14):

- špatně provedené testy, možná vina lidského faktoru;
- omezení komunikace v týmu;
- testováním se nepodaří pokrýt všechny případy zadání;
- nedostatek času a zdrojů;
- opomenutí požadavků klientů a uživatelů.

Problémy se specifikací:

- chybějící specifikace;
- přílišná nejednoznačnost specifikace;
- chybné údaje a opomenutí ve specifikaci.

Problémy s procesem testování:

- chybná zadání;
- chybná vstupní data;
- použití zastaralých, neaktuálních a neadekvátních testů;
- testy nejsou dostatečně schopné zareagovat na aktuální okolnosti;
- nedostatečné přizpůsobení testů pro konkrétní produkt;
- nepostižitelnost veškerých chyb v procesu testování.

2.1.5. Práce testera

Vývoj software je náročný proces jak z hlediska řízení, tak z hlediska samotného procesu vývoje, který v sobě zahrnuje komplexní procesy, nástroje a techniku. Proto je nutné čelit potencionálním hrozbám. Jakýkoli softwarový produkt by měl projít testy.

Buchalcegová a kolegové (2008, str. 35) definují samotné testování softwaru jako proces, při kterém dochází k ověření, zda softwarový produkt odpovídá definovaným požadavkům. Za tímto účelem je software podroben sérii testů, kterými je softwarový produkt prověřen na několika úrovních - od jednotlivých programových komponent až po programový systém jako celek. Testování softwaru tvoří jen jednu z mnoha složitých částí procesu vývoje. Jednou z předních charakteristik současné doby je velmi rychlý vývoj v oblasti technologií. Informační a komunikační technologie, neboli ICT (Information and Communication Technologies), patří mezi ty nejvíce dynamické obory v dnešním světě.

Tester je osoba, která má za úkol vyhledávat chyby v programech. Cílem práce testera je ověřit, jestli software splnil specifikace a nakolik je ve výsledku kvalitní. Náplní práce testera je především kontrola práce - a to nejen programu, ale přímo celého vývojového procesu a ostatních pracovníků, kteří jsou do něj také zapojeni a podílí se na výsledné podobě produktu (např. programátoři, analytičtí pracovníci atd.) Tester je zodpovědný za provedení testů, kontrolu logů a další neméně důležité činnosti. Musí také vést dokumentaci testů a podávat zprávy o nalezených nesrovnalostech. Chyby však neřeší samotný tester, ale vývojář.

Kromě softwarových testerů se v týmu nachází také řada dalších osob, zodpovědných za hladký průběh práce na projektu. Testovací scénáře a podklady pro přípravu testů tvoří designéři. A konečně nesmíme zapomínat ani na manažery, kteří vypracovávají přesné plány, organizují činnosti a koordinují práci ostatních.

Pokud nalezne tester v programu nějakou chybu, popíše ji a zároveň zaznamená jednotlivé kroky k řešení problému. Úkolem testera je tedy objevit problém, objasnit jeho vliv na softwarový systém a navrhnout nějaké vhodné řešení, které by problém mohlo úplně odstranit ze systému. Gibbs (2006, str. 89) tvrdí, že tester reprezentuje poslední šanci, jak odchytil problémy, než je spatří oko zákazníka. Dle autorů Grahama, Fewstera (2012, str. 5) je však role a zkušenosti testera, který automatizuje testy, poněkud jiné. Naráží na fakt, že vytvoření automatizovaného testu zahrnuje jak znalosti práce testera, tak i práci test architekta, který by měl být schopný test nejen vytvořit, ale i správně navrhnout.

Mezi typické činnosti testera patří zaprvé příprava testovací strategie, kterou by bylo vhodné využít. Nejprve je třeba mít plán, včetně přípravy testovacích scénářů pro manuální nebo automatizovaný způsob testování. Je nutná také příprava testovacích případů, v případě automatizovaného testování musí tester naprogramovat skript (Buchalcegová a kolegové, 2008, str. 36).

Mezi další činnosti, které spadají do práce testera, patří také příprava testovacího prostředí, příprava a správa prostředků pro testování a vytvoření dokumentace se zaznamenanými výsledky testování software. Tester musí samozřejmě komunikovat s vývojovým týmem a podílet se na rozhodování ohledně produktu. Tester software je členem vývojového týmu. S narůstáním složitosti software je to dokonce osoba nadmíru důležitá a svým způsobem nepostradatelná.

Je také nutné si uvědomit, že právě vysoký stupeň kvality software má pochopitelně velký vliv na celý systém, ve kterém pracuje. Vadný software dokáže napáchat velké škody, a pokud nejsou jeho slabá místa včas odhalena a chyby opraveny, mohl by jeho vinou zkolabovat celý systém jako celek, což by se v dnešní přetechnizované době dalo

označit jako katastrofa. Už proto bychom měli testování software přikládat velkou důležitost.

Testování je důležitou součástí i vývoje. Tvoří jeden z nejvýznamnějších aspektů kvality koncového produktu. Pokud chceme odhalit místa, na kterých se vyskytují chyby, musí být tester nadměru pečlivý a důkladně prozkoumat spornou aplikaci. Program musí splnit všechny technické požadavky, ale také musí naplnit očekávání uživatelů, kteří budou s programem pracovat.

Existuje však celá řada důvodů, které mohou do značné míry zkreslit a někdy dokonce úplně zničit výsledky testování. Jedním z těchto důvodů jsou malé či neodpovídající znalosti testerů o procesech jejich práce, počáteční neznalost produktu, anebo nedostatečné informace o různých typech testů, které lze využít. Někteří testeři mohou považovat jisté typy testů za vhodnější než jiné, což může vést k tomu, že je zcela opomenou zrealizovat. Dalším faktorem je nedostatek času. Problém může být i samotné úloze testování. Testování je totiž velmi obtížný úkol. Jsou různé druhy testů, stejně jako různé pohledy na vyvíjený produkt. To vše může přispět k tomu, že se nám nemusí podařit najít všechny chyby v programu. Ideální by bylo, kdyby se všechny tyto problémy podařilo zcela eliminovat. Vhodným řešením by bylo učinit testovací proces více efektivní při použití částečně automatizovaných nástrojů.

2.2. Princip automatizovaných testů

Momentálně se softwarové testování ubírá směrem k automatizaci testů. Dle Graham, Fewster (2012, str. 4) je ale nezbytné sledovat i návratnost investice, která spočívá nejen v samotné automatizaci, ale i v údržbě automatizovaného řešení. Automatizaci jako krok do budoucna vidí i autoři Janiček, Marek a kol. (2013, str. 204), kteří tvrdí, že je v současné chvíli snaha automatizovat celý životní cyklus softwarového vývoje.

Sankar (2013, str. 4) definuje automatizované testy následujícím způsobem: automatizované testy jsou soubor zautomatizovaných manuálních testovacích scénářů a procesů, které jsou automaticky zpracovávány pomocí nástroje pro automatizaci nebo utility. Hlavním cílem automatizace je kratší testovací životní cyklus a zvýšené pokrytí testovaných scénářů bez závislosti na čase.

Jednou z velkých výhod automatizace testů je prostý fakt, že když se nám podaří vytvořit velmi dobrý test, můžeme jej využít i opakovaně. Možnou opakovatelnost popisuje jako výhodu i Moreira (2010, str. 152) a Mařík (2007, str. 5). Znovupoužitelné testy jsou sice velmi přínosné, ale po delším časovém úseku již mohou být poměrně neaktuální. Na to musíme pamatovat. Proto vyžadují systematické úpravy, které jsou přímo úměrné komplexitě testovaného produktu (Mařík, 2007, str. 6). Automatizace je v dnešní době natolik běžným jevem, že se s ní každý z nás setkává téměř na každém kroku. Manuální činnost při softwarovém testování nahrazuje využití specifických nástrojů, které slouží pro práci s testy a jejich tvorbou.

Největšími výhodami nástrojů a automatizovaných testů jsou podle Pattona (2002, str. 14):

- **Rychlost** – automatické testovací nástroje jsou velmi rychlé a efektivní. Dalece překonávají jak lidskou činnost testera, tak pomalejší manuální testy.
- **Efektivita** – automatizované testy šetří náklady a čas.
- **Přesnost** – automatické testy a nástroje jsou přesnější než člověk, který má sklony k děláni chyb.

- **Neúnavnost** – automatické testy pracují non-stop.

Automatizovaně vykonávané činnosti, včetně softwarových testů, s sebou přináší mnohé výhody. Jednou z největších výhod je, že náklady se znatelně sníží, zatímco práce je rychlejší a výkon větší. Automatizované testy se také vyznačují větší přesností a efektivnějším tempem práce. Pro automatizované testování jsou použity nástroje, které umožňují snadnější provedení testů. Jandák (2012, str. 8) zastává názor, že automatizované nástroje slouží k urychlení a zefektivnění práce testera, nedokážou ji však zcela nahradit.

Graham a Fewster (2012, str. 12) uvádí, že skripty pro automatizaci by měly být tvořeny takovou formou, aby bylo možné je znovu použít, což výše uvedení autoři považují za největší benefit automatizovaných testů. Podle autorů Graham, Fewster (2012, str. 13) je nejlepší automatizaci započít na činnostech, které jsou pro testery „nudné“. Velmi často se totiž jedná o výše zmíněné často opakované testovací scénáře.

Ash (2003, str. 3) ve své knize definuje termín testování jako fungování provozu aplikace za určitých kontrolovaných podmínek. Cílem testování je vyhodnotit kvalitu softwaru. Kvalita aplikace se odvíjí od poměru bezproblémových míst a chybných, problémových částí, které byly objeveny během procesu testování. Problém se nejčastěji projevuje jako nečekané chování software proti své specifikaci. Prací testera je najít ty nejzávažnější problémy a zároveň zkontrolovat, zda jsou dodrženy závazné standardy. Tester by měl také celkově posoudit kvalitu daného programu z mnoha různých hledisek. Proto je nutné vypracovat zprávu o výsledku testů.

Přístupy k testování software se liší, stejně jako se od sebe liší jednotlivé metody, využití při softwarovém vývoji. Metody mají rozdílné zaměření a postupy, což se odráží i na oblastech, na které se primárně zaměřují. Každá metoda disponuje řadou technik, proto je každá z nich vhodná pro něco jiného. Proces testování však musí vykazovat určitý stupeň kvality, aby mohl podávat správné, ověřené informace o testované aplikaci.

2.2.1. Případy použití automatizovaného testování

Ne vždy je možné automatizovat všechny manuální testovací scénáře a taktéž není vhodné se o to za každou cenu snažit. Sankar (2013, str. 8) dělí testovací scénáře vhodné k automatizaci na tři kategorie:

- často opakované – sanity (smoke) a regresní testy;
- testovací scénáře využívající komplexní matematické výpočty;
- testovací scénáře, které používají velké množství vstupních dat v různých variantách.

K automatizaci naopak vůbec nedoporučuje následující typy testovacích scénářů:

- jednorázové testy
- a testy vyžadující manuální vstupy do testování.

Navíc Sankar (2013, str. 8) uvádí i testovací případy, u kterých je třeba si dávat pozor na případnou efektivitu automatizace, kde příkladem může být například kontrola fontů, barev nebo automatizace testovacího scénáře. V takovýchto případech je automatizovaná kontrola v porovnání s manuální pomalejší. Na podporu automatizovaných testů je samozřejmě možné použít SW nástroje.

Využití automatizovaného nástroje je nejvhodnější pro programy se stabilním kódem. To znamená, že když nebudeme provádět žádné manuální testy a zvolíme

automatizované metody jako primární formu testování, pak musíme po celou dobu pracovat se stejnými metodami objektů totožné struktury (Graham, Fewster, 2012, str. 12). S tímto názorem i Galin (2004, str. 245), který dodává, že v případě častých změn není automatizace efektivní. Tento typ přístupu se týká pouze regresního testu, kde je z logiky testu předpokládáno, že četnost změn v kódu je minimální a pouze inkrementální v případě rozšiřování rozsahu testování a minoritních úprav v již existujícím kódu.

Složité aplikace bývají oříškem, který je mnohdy velmi těžké rozlousknout. I náklady na koupi a údržbu automatizovaného testovacího nástroje sice nejsou zanedbatelné, ale výhodou je skutečnost, že automatizaci můžeme využít i pro nové, ještě nezrealizované projekty, které teprve přijdou. Vzhledem k tvrzení Ammanna a Offutta (2008, str. 10) tvoří náklady na testování v případě implementace nové funkcionality až 50 % nákladů. Je tedy namístě zaměřit se na možnosti úspor právě ve fázi testování. Každá společnost má možnost v závislosti na typu testovaných aplikací a metodice testování vybrat z komerčních, tak i open source nástrojů. Komerční řešení jsou poměrně dobrá varianta nástroje pro automatizované testy, a to hned z několika důvodů. Zaprvé jde o relativně nízkou cenu nástroje, snadnější možnosti použití a hotová řešení. Proces automatizovaného testování je jiný než u manuálních testů. Implementace automatizovaných nástrojů do testovacího procesu je však složitější než by se na první pohled mohlo zdát. Implementace musí být provedena odbornými pracovníky, kteří již mají zkušenosti v oboru. I přesto bývá nezbytná zpětná vazba ze strany tvůrců nástroje, ať už pro instalování produktu, nastavení (neboli softwarovou konfiguraci), nebo pravidelné udržování. Příprava testovacích nástrojů vyžaduje velké množství času, na rozdíl od méně náročné přípravy manuálního testování systému. Tester musí mít dobré znalosti nástrojů, analýzy požadavků, i zkušenosti s automatizací obecně.

Na otázku, jaký nástroj je nejvhodnější pro automatizaci, neexistuje žádná jednoznačná odpověď. V tomto případě není možné zobecňovat, protože každý problém si vyžaduje odlišné řešení. Každý problém je unikátní. To, jaký nástroj zvolíme, má velký vliv na možnosti, jak automatizovat testování.

Při vybírání vhodných nástrojů bychom měli hledět především na technické požadavky, ze kterých budeme nadále vycházet. Nástroj musí být kompatibilní s platformou, na jaké fungují aplikace, jež zamýšlíme podrobit testování. Nástroj, který vybereme, musí správně pracovat se všemi objekty v aplikaci. Abychom si byli jistí, že jsme při výběru správného nástroje neponechali nic náhodě, měli bychom také provést testy technologie s aplikací, kterou zamýšlíme zautomatizovat. Nástroje pro automatizaci jsou zmíněny v kapitole 2.3.6.

2.2.2. Stručný přehled automatizovaných nástrojů

Nástroje pro automatické testování nám nabízí kompletní řešení pro manuální i automatické testování softwaru a test management. Jde o druh software, jehož úkolem je provést test aplikace, nebo alespoň určitý úsek této činnosti. Zde jsou stručně charakterizovány některé ze známých nástrojů pro automatické testování software:

Selenium je nástroj psaný v jazyku Java (což je objektově orientovaný programovací jazyk) pro tvorbu automatizovaných testů v širokém rozpětí programovacích jazyků. Skládá se z více součástí, které se navzájem doplňují: Selenium IDE, Selenium RC, Selenium WebDriver a Selenium Grid (Selenium HQ, 2013).

SilkTest je komerční nástroj pro testování software od společnosti Borland. Slouží především k provedení funkčních testů GUI. Kromě toho také podporuje automatizaci desktopových a webových aplikací. Tento nástroj je momentálně dostupný ve dvou variantách: SilkTest Classic a SilkTest Work Bench (Borland, Silk Test, 2015).

TestComplete je nástroj pro automatizované testování webových aplikací, stejně dobře se však dá využít i pro testování mobilních platform. Proto existuje ve více variantách různých modulů, z nichž se každý hodí pro odlišnou činnost. Tím pádem vidíme, jak má tento nástroj velmi široké možnosti využití. Navíc nástroj má jednu velmi přínosnou vlastnost - umožňuje osobě testera, aby si mohla vybrat, jaký skriptovací jazyk chce použít (např. JavaScript, C++ Script, C# Script atd) (SmartBear.cz, 2015).

Velmi zajímavý je díky svým možnostem využití také Sankar (2013, str. 68) **HP QuickTest** a **HP UFT (Unified functional Testing)** - nástroj pro testování software umožňující automatizaci funkčních a regresních testů.

Z komerčních nástrojů lze také zmínit nástroj společnosti Globtech a jejich dva produkty: Modul QACEGEN a Modul GUI master. Modul AceGen je určen pro řešení problematiky vytváření testovacích dat a jejich otestování v aplikacích. Modul Gui Master je zaměřen na samotné automatizované testování uživatelských rozhraní a to jak webových, tak i desktopových aplikací.

Nástrojů pro automatické testování software samozřejmě ještě existuje celá řada. Zde je krátký seznam ostatních, které stojí za zmínku:

- IBM Rational Functional Tester;
- TestArchitect.

Následující software je pak vhodný pro automatické reportování chyb odhalených v rámci testování:

- JIRA;
- MANTIS;
- LoadIU;
- TestLink.

2.2.3. Výhody automatizovaného testování

Automatizovaný způsob softwarového testování výrazně sníží náklady a přitom slouží také ke zvýšení rychlosti testování. Zvyšuje kvalitu a spolehlivost webových aplikací. Automatizované testy simulují uživatelskou práci, tím pádem je možné vždy ověřit, jestli jsou klíčové údaje v aplikaci pro klienty správné. Automatizovaný test je také schopen zaznamenat a vyhodnotit různé souvislosti. Údaje změřené při procesu automatizovaného testování, je možné provázat s interními výkonnostními ukazateli. Automatické testy webových aplikací, stejně jako sledování a vyhodnocování dostupnosti aplikací, podporují kontrolní proces manažera kvality. Navíc dle Hoffmana (1999, str. 3) lze díky rychlému běhu a nízkým nárokům na spouštění testů realizovat mnohem častěji. S větším pokrytím testované oblasti souhlasí i Sankar (2013, str. 4), který zároveň zmiňuje další výhodu v podobě „konzistence provedení testu“. Tato výhoda spočívá ve využití vlastnosti automatu, který se nikdy „nenudí“ opakovaním stále stejných testovacích kroků jako lidský jedinec. Tato konzistence zajišťuje, že díky lidské chybě či nedůslednosti nedojde k nedetekování chyby z důvodu přeskokování triviálních či pro testera nedůležitých kroků.

Dle Desikana a Rameshe (2006, str. 21) neexistuje špatná automatizace, ale pouze její špatný účel. U každé formy automatizace tak radí aktivitu řádně naplánovat, správně zhodnotit a nastavit správná očekávání, tak aby se nedostavil pocit zklamání. Trochu negativně k tomuto přistupuje Sankar (2013, str. 4), který tvrdí, že automatizace testů je drahá a navíc nenahrazuje manuální práci testerů. Oblast, kde ji lze z pohledu autora využít, jsou pouze takové činnosti, které lze automatizovat dlouhodobě, čímž má konkrétně na mysli regresní testy.

2.2.4. Shrnutí

Automatizace je momentálně velmi aktuální téma a jako metoda se z pozice „luxusního“ řešení posunula směrem k řešení „nezbytnému“ (Graham, Fewster, 2012, str. 16). Téma automatizace je hojně diskutované v různých souvislostech. Jednu z nich představují ušetřené peníze a čas, což je výhodné především pro řízení a financování mnoha vývojových projektů (Hoffman, 1999, str. 33). Je mnoho definic, které nám vysvětlují, co je to automatizace. Obecně platí to, že proces automatizovaného testování software - nebo minimálně některá z jeho dílčích součástí - probíhá prostřednictvím specifických softwarových nástrojů speciálně určených pro tuto náročnou činnost, a to bez jakýchkoli lidských zásahů do testovacího procesu. Máme k dispozici mnoho programů, které jsou schopné provést automatizované části testovacího procesu, jako jsou např. instalace a konfigurace otestovaných aplikací. Jde o činnosti, které program umí udělat zcela samostatně, tedy bez jakéhokoli zásahu člověka. Jestli se nám podaří získat odpovídající složení nástrojů, pak můžeme zautomatizovat veškerý průběh softwarového testování.

Automatizace testování se musí vyplatit jak po stránce profesní, technické, tak i ekonomické. Dá se říci, že testování má smysl automatizovat tehdy, kdy si dobře promyslíme všechny výhody a nevýhody, které s sebou přináší. A pak společně dospějeme k nějakému optimálnímu řešení, které bude výhodné pro všechny, kteří se na něm podílí. Nejvhodnější jsou testy, které musíme opakovat. Musí tedy být stabilní součástí softwarového programu. Větší množství testů se musí opakovat ve stejné podobě, a to ne nahodile. Množství těchto stabilních, hojně opakovaných testů může být spíše subjektivní. To, zda je již vhodné pro automatizaci, je na uvážení zainteresovaných osob. Musíme vzít také v úvahu, kolik zaměstnanců máme k dispozici a kolik peněz jsme ochotní do zavedení automatizace investovat.

Když tester pečlivě prochází stále stejné testovací scénáře, může se snadno zmýlit a přehlédnout některé chyby. Tomu však můžeme snadno předejít. Zavedení automatizovaných testů je proto o mnoho spolehlivější a navíc šetří jak čas, tak energii odborných pracovníků.

Ve většině případů je vhodné využívat automatizované testy. Mnozí na automatizované testy nahlíží spíše jako na pouhého pomocníka, který jim může do jisté míry usnadnit práci s testováním software. Jejich význam je však mnohem větší. Je-li testování provedeno automaticky, tj. bez práce člověka, pak je testování přesnější, rychlejší a nachází se v něm méně chyb. Automatizace testů usnadňuje postup softwarového testování. Význam automatizace testování softwaru je doslova k nezaplacení. Přináší zvýšení kvality výsledného programu.

Pro automatizaci je nejlépe uzpůsobeno nefunkční testování, smoke testy, komparační testování a regresní testování. To tedy znamená, že testy vhodné pro automatizaci by měly zahrnovat aplikace či jednotlivé oddíly aplikace, které se málo mění a zůstávají

většinou stále stejné. Musíme zajistit určité podmínky pro zavedení automatizovaných testů. Nejvhodnější etapa pro vytváření těchto zkoušek je tedy návrh a vývoj softwaru. Architektura aplikace musí umožnit automatizaci testů. Tím pádem by bylo dobré, kdyby byla již zkonstruována s tímto úmyslem. Budeme-li mít vypracovaný program, bude obtížnější začít s automatizovaným testováním za provozu.

Máme-li aplikaci s kratší životností, počítanou třeba na měsíce či týdny, pak se ji automatizovat ani nevyplatí. Naproti tomu máme-li aplikace s delší životností, např. na roky či více, pak bychom je rozhodně automatizovat měli. Automatizované testy bychom měli zaměřit na problémové části aplikací, kde se mohou vyskytovat chybná místa ve větším počtu.

Výhody a nevýhody automatizace testů softwaru musíme vždy posuzovat v širším kontextu projektu. Každý projekt má odlišný charakter, požadavky a cíle. Tyto faktory hrají zásadní roli při možnosti automatizace. Jak už bylo řečeno, jednou z největších výhod automatizace testů je úspora času. Pokud jsou testy již zcela automatizované, mohou být prováděny prakticky kdykoli, například v nočních hodinách, kdy jsou ostatní činnosti zastaveny. Navíc není nutná ani neustálá přítomnost lidí na pracovišti. Automatizované nástroje ovšem nedokážou zareagovat na změnu. V tomto ohledu je dalece překonává osoba testera. Každá změna v aplikaci si může vyžádat přepsání testovacích skriptů nebo jiné nastavení nástrojů.

3. Praktická část

3.1. Představení společnosti T-Mobile

Společnost T-Mobile Czech Republic působí na českém trhu od roku 1996, od roku 2002 je součástí mezinárodní telekomunikační skupiny Deutsche Telekom. K 31. prosinci 2014 společnost obsluhovala více než 6 milionů zákazníků mobilních i fixních služeb, a zůstává tak jedničkou na českém mobilním trhu co do počtu zákazníků. T-Mobile je integrovaným operátorem. Vedle mobilních a fixních telekomunikačních služeb nabízí i široké portfolio IT služeb (například cloudové služby či nabídku produktů Office 365) a řešení systémové integrace pod značkou T-Systems pro firemní zákazníky a veřejnou správu. V polovině roku 2014 byla provedena akvizice se společností GTS, která je již plně integrována do společnosti T-Mobile a značně tak rozšířila produktové portfolio společnosti T-Mobile. (T-Press, 2015).

Dle interních zdrojů (2015) se T-Mobile dlouhodobě zaměřuje na kvalitu svých služeb a produktů. Nezávislé testy opakovaně potvrzují, že T-Mobile je leaderem v kvalitě jak v oblasti 2G, 3G (UMTS FDD) a LTE dat, tak i v případě hlasových služeb v síti GSM. Společnost získala také řadu odborných ocenění za inovativní ICT řešení – například v anketě IT projekt roku.

Od svého založení společnost klade důraz také na zodpovědný přístup k podnikání a společnosti. T-Mobile je mimo jiné držitelem ocenění Zaměstnavatel roku i desetiletí, Firma roku: Rovné příležitosti či ceny Via Bona za zapojování zaměstnanců do dárcovství a dobrovolnictví. V roce 2012 časopis World Finance vyhlásil T-Mobile Czech Republic nejlepším mobilním operátorem ve východní Evropě (T-Press, 2015).

3.1.1. Způsob testování ve společnosti T-Mobile

Práce je aplikovaná na útvar IT, konkrétně na činnost týmu Integration and Validation, který je zodpovědný za otestování kvality dodávky dodané od vývojové části IT útvaru. Výše zmíněná jednotka se zabývá integračním testováním, jehož obsahová část je popsána v teoretické části práce v kapitole 2.9. Společnost T-Mobile dodává v tzv. releasových dodávkách. Releasem je myšlen soubor jednotlivých dodávek, které jsou spouštěny v jeden okamžik. V současnou chvíli lze rozdělit release na čtyři typy. Prvním z nich je Major Release – hlavní release, který spouští funkcionality zhruba osmkrát do roka (každých 6 týdnů) – v tomto releasu jsou dodávány hlavně businessové funkcionality s největším dopadem na zákazníka. Typicky se jedná o zavedení nových tarifů, balíčků či vylepšení stávajících služeb. Nedílnou součástí jsou i požadavky vzniklé nařízením vlády či Českého telekomunikačního úřadu. Druhý typ releasu je nazýván Minor release. Tento typ releasů je primárně určen pro mezidobí mezi spuštěním dvou hlavních (Major) releasů. Roční frekvence je podobná jako u předchozího typu, tzn. osmkrát až devětkrát ročně. Je určen pro dodávky malého charakteru v rozsahu několika MDs¹, které nejsou závislé na dodávkách v Major releasech. Dalším typem je release Specific – specifický release, který slouží pro dodávky úprav takových systémů, které mají individuální způsob nasazování. Například se jedná systémy čistě technického charakteru, kde příkladem může být technická evidence sítě T-Mobile. U tohoto typu nelze definovat počet. Jejich počet je libovolný a obecně lze ale říci, že se v posledních dvou letech pohyboval mezi patnácti až dvaceti ročně. Posledním typem je release technický, který v zásadě obsahuje dodávky typu

¹ MDs – Jednotka – člověkoden – 8 hodin

úpravy databází, logování nastavení bezpečnostních záplat atp. Platí stejné rozsahové vymezení jako v předchozím případě, tzn. počet těchto typu dodávek není procesně omezen a v současné chvíli se jedná o aktivitu prováděnou jednou cca 14 dní – 25 krát ročně. Release management společně s Change management procesem deklaruje nutnost zařazení všech implementačních dodávek do releasů. Na základě charakteru dodávky se poté zvolí, do jakého typu releasu bude implementace zařazena. (Interní zdroje, Release management proces, 2014).

Téměř každému nasazení nové funkcionality na produkci předchází testování funkcionality – minimálně na uživatelské úrovni. Zároveň je třeba provádět i regresní testy, které ověřují, zda nasazením nových funkcionalit nebyly rozbity ty dříve nasazené, stávající funkcionality. Dalším testem, který je prováděn, je performance (zátěžový) test na produkčním prostředí, který se snaží simulovat vysokou zátěž na systémy. Tento test se snaží odhalit, zda při vysoké zátěži na systém nedojde k výpadku systému. Přehled prováděných systémů včetně provádějících osob a využívaného prostředí jsou popsány níže v Tabulce č. 1. Větší detail tabulky je popsán níže. Obecně lze říci, že společnost T-Mobile provádí testování na několika úrovních:

Tabulka 1 Testování ve společnosti T-Mobile

Typ testování	Testující osoba	Typ prostředí
Unit testy	Vývojáři	Vývojové
Systémové testy	Vývojáři	Vývojové + referenční
Entry Check (Smoke Testy)	Integrační testeři	TEST1/TEST2
Integrační testy	Integrační testeři	TEST1/TEST2
Uživatelské testy	UAT testeři (uživatelský tester)	TEST1/TEST2/Produkční
Regresní testy	Integrační testeři + UAT testeři	TEST1/TEST2/Produkční
Performance testy	Integrační testeři	Produkční

Zdroj: Interní zdroje, 2015 (vlastní zpracování)

Pro další popis je nutné vydefinovat ta prostředí, na kterých se výše uvedené testy realizují a typy testů, které jsou prováděny. Společnost T-Mobile používá 4 druhy prostředí: vývojové, referenční (používané pouze pro některé vyvíjené systémy), testovací prostředí 1 (dále označováno jako TEST1), testovací prostředí 2 (dále označováno jako TEST2) a prostředí samotné produkce, ke kterému již mají přístup samotní zákazníci, infolinky či aktivační týmy.

Vývojové prostředí je určeno pro potřeby vývojových týmů, kde pomocí unit testů a izolovaných systémových testů ověřují, zda jimi navržená funkcionality funguje. Unit a systémové testy jsou prováděny vývojářem, který vyvinul konkrétní funkcionality. Tyto testy jsou realizovány na vývojovém prostředí systému. Tento test je nazván izolovaným z toho důvodu, že není ovlivňován okolními systémy. Data na vstupu jsou v případě front-endové aplikace² plněna přímo do testovaného systému. V případě back-

² Autoři Pavelka, Voříšek (2003, str. 207) definují front-endovou aplikaci jako soubor systémových prostředků na straně klienta. V praxi lze tento pojem aplikovat například na e-shop. Je však třeba říci, že

endové aplikace, která pro svou správnou funkci potřebuje vstupní data od výše uvedených aplikací, je tento test plněn z dat nasimulovaných (přesto však založených na reálných datech).

Smoke testy jsou nadefinovány v teoretické části práce – kapitola 2.2.1. V rámci testovacího týmu jsou obecně nazvány jako Entry Check. Tyto testy předcházejí samotnému integračnímu testování a mají za cíl ověřit, zda byla funkcionální správně nainstalována na prostředí TEST1 či TEST2, zda proběhla nutná konfigurace vývojářem či business administrátory (například v případě úpravy produktů, či číselníků, které administrují lidé z marketingu). Tento test je omezenou množinou, často se opakujících funkcionalit daného systému. Velmi často se v této části testu objevují nekompatibility s daným testovacím prostředím, nenastavené prostupy, vadné či chybějící konfigurace. Všechny tyto výše uvedené chyby jsou důvodem pro nespustění samotných integračních testů. V případě, že většina základní funkcionality není dodána nebo je zcela nefunkční, nepovažujeme aplikaci/projekt/release za překlopený do fáze testování, jedná se o tzv. „ladění“. Testy jsou prováděny vždy na začátku bloku testování určených pro dodávku pro Major release, který je popsán výše. Entry Checky nejsou prováděny v rámci ostatních typů releasů.

Jak je výše zmíněno, samotnému spuštění integračních testů předchází provedení Entry Checků. Integrační testování spočívá v testování hlavní funkcionality aplikace a testování veškerých konektivit do ostatních systémů. Zde by již měla být připravena vstupní data takovým způsobem, jaký vyžadují funkcionality na produkčním prostředí – to znamená, že v této fázi testů není žádoucí jakýmkoliv způsobem vstupní data imitovat či nahrazovat. Naopak cílem je otestovat mezi-systémové vazby na úrovni validity výstupních i vstupních informací tak, aby mohl být na úrovni uživatelských testů proveden end-to-end test (test od začátku až do konce – velmi často je procesně orientován). Součástí je samozřejmě přetestování hlavní funkcionality po odstranění případných chyb. Pro testy integrační testeři využívají obou testovacích prostředí. Nikdy však paralelně. Jedno testovací prostředí je v současné chvíli vždy vyhrazeno pro strategické projekty typu nového CRM nebo ERP systému. Do budoucna je plánováno paralelního využití obou testovacích prostředí.

Uživatelské testy jsou prováděny jiným oddělením než testy integrační. Převážně jde o zaměstnance oddělení péče o zákazníky. Jak již název testů napovídá, jde o testy, které simulují samotnou uživatelskou činnost. Testy jsou prováděny kvůli faktu, že ani sebelepší designér a vývojář nedokáže odhadnout reálné chování uživatelů. V zásadě lze po odstranění nejvýznamnějších chyb v integrační testování testovat současně uživatelsky a integračně. Nemělo by se však stát, že bude k uživatelskému testování předána funkcionální či testovací scénář, který obsahuje neopravenou či integračně nepřetestovanou funkcionální s chybou závažnosti (severity) číslo 1, 2 a 3. V tuto chvíli je paralelní testování neefektivní. Priorita chyby určuje, v jakém pořadí má být chyba odstraněna (vyřešena). Prioritám přiřazujeme číselné hodnoty 1, 2, 3, 4 a 5. Popis závažnosti chyb je pro přehlednost popsán v Tabulce č. 2 níže:

klientem může být nejen zákazník externí, ale i zákazník interního charakteru, jako je například obsluha infolinky.

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

Tabulka 2 Závažnost chyb

Číslo chyby	Označení chyby	Popis chyby - Doba řešení
#1	Show Stopper	Chyba, která je určena k okamžitému řešení a zamezuje dokončení testovacího případu. Velmi často se jedná o chybu na straně instalace či konektivity. Doba řešení se pohybuje v řádech minut až jednotek hodin
#2	Critical	Velmi závažná chyba, která znemožňuje dokončení testovacího scénáře. Může se jednat například o chybějící výběrovou možnost v číselníku (chybná konfigurace). Doba řešení je v řádech jednotek hodin.
#3	Serious	Závažná chyba, která může znamenat ohrožení otestování dané funkcionality. Typicky se jedná o nezobrazení relevantní nabídky například v podobě nových balíčku. Doba řešení je v řádech jednotek dnů.
#4	Minor	Tato severita chyby umožňuje dokončení testovacího scénáře. Upozorňuje však na nevalidní chování testované funkcionality. Může jít například o špatné nastavení práv pro skupiny uživatelů, špatné popisky v rámci statického obsahu stránky či popis nového tarifu. Doba řešení tohoto typu defektu se pohybuje v řádech dnů.
#5	Cosmetic	Jednou se o chybu, jak název napovídá, kosmetického charakteru. Může jít o špatné zobrazení barev, či bannerů. Tato chyba nemá vliv na samotnou funkcionality. Negativně však ovlivňuje vnímání zákazníka a snižuje komfort využívání aplikace. Tyto chyby jsou nejčastěji odhalovány v rámci uživatelských testů. Doba řešení se pohybuje v řádech dnů.

Zdroj: Interní materiály (Testing Process), vlastní úprava, 2014

Ve společnosti panuje domněnka, že jsou prováděny regresní testy na všechny vyvíjené systémy. Tento mylný předpoklad bude osvětlen v rámci detailní analýzy v praktické části práce. Regresní testy jsou v současné chvíli realizovány po nasazení Major releasů. UAT testerů provádí regresní testy častěji, a to v závislosti na stavu produkčního prostředí. Integrované regresní testy jsou prováděny na obou testovacích prostředích (TEST1 a TEST2). Uživatelské regresní testy ověřují stávající funkcionality na testovacím i na produkčním prostředí. Ve strategických cílech společnosti 2015 - 2017 existuje formulace, která říká, že je nutné zvyšovat kvalitu stávajících služeb. Regresní testy tak byly ze strategického důvodu zvoleny jako vhodný nástroj pro dosažení výše uvedeného cíle.

Samostatnou kapitolou jsou performance testy. Performance testy teoreticky definuje kapitola 2.1.3. V T-Mobile jsou performance testy prováděny v pravidelných intervalech pouze na front-endovou aplikaci VCC, která zákazníkům zajišťuje možnost samostatné administrace jejich telefonních čísel a ostatních služeb. V současné chvíli jsou testy realizovány vždy přibližně 14 dní po nasazení dodávek v hlavním (Major) releasu, a to vždy až po stabilizaci produkce. Testy jsou zaměřeny na provolávání jednotlivých funkcí a simulaci velkého, v současné chvíli přístupujícího, množství zákazníků. V rámci testů jsou nejvíce zatěžovány zákaznické procesy jako je například přihlášení, navýšení FUP či aktivace balíčků. Tyto testy se realizují pouze na produkčním prostředí, a to pouze v nočních hodinách, kdy je náhlým výpadkem

ohrožen nejmenší počet zákazníků. V rámci testů se na přibližně 2 hodiny odstavuje produkční prostředí. Performance testy nejsou prováděny pouze ty pravidelné. V T-MOBILE existují projekty, které po nasazení nové funkcionality požadují provedení performance testů tak, aby nebyla narušena nově implementovaná i stávající funkčnost systému. Rozsah a časování performance testování je poté závislé na vydefinovaných požadavcích konkrétního projektu.

Práce se ve větším detailu zabývá možností automatizace regresního testování na TEST1 a TEST 2 a produkčním prostředím. Tato část testování byla vybrána na základě klíčové aktivity společnosti T-Mobile, která si v rámci strategického plánování na roky 2015 – 2017 dala za cíl zvýšení kvality služeb pro koncové zákazníky.

3.1.2. Nástrojová podpora testování

Samotné testování a jeho nástrojová podpora se skládá z několika fází, které lze ve zjednodušené podobě shrnout do tří fází: fáze přípravná, samotná fáze testování a post-implemenční fáze. V rámci přípravné fáze je třeba, aby testéři/test analytici společně s test manažery připravili testovací strategie a vytvořili testovací scénáře. Pro evidenci testovacích scénářů je v rámci T-Mobile společný nástroj HPQC (HP Quality Centre), nástroj od Hewlett Packard ve verzi 10.0. V červnu 2015 bude SW povýšen na verzi 11.00, která již umožňuje v rámci modulu HP Sprinter polo-automatizaci testů. Scénáře jsou zde strukturovány dle názvů relevantních projektů, kde implementace proběhla a releasů, kam nasazení dodávky směřuje. Speciální typy testů jako jsou například regresní či performance testy, mají svou vlastní složku, kam jsou scénáře ukládány.

V rámci samotného testování se práce testerů zaměřuje na exekuci testovacích scénářů, která probíhá opět za podpory výše uvedeného nástroje. Součástí každého typu testování je samozřejmě i evidence chyb/defektů. V současné chvíli je pro tzv. bugfixing ve společnosti T-Mobile vybrán tentýž nástroj od Hewlett Packard – HPQC (Quality Centre) jako pro evidenci testovacích scénářů. Defect management je v rámci výše uvedeného SW nastaven dle platného Testing Procesu. Nástroj umožňuje nejen evidenci scénářů a defektů, ale umožňuje i předávání defektů mezi jednotlivými uživateli HPQC, měření SLA a aktuální reporting stavu (Interní zdroje, 2015).

Post-implemenční fáze začíná okamžikem nasazení funkcionality na produkční prostředí. V tuto chvíli se již chyby evidují na dvou místech – v HPQC – pokud jsou odhaleny v rámci uživatelských testů či HPSM (HP Service Manager), kam mají možnost zadávat Service cally či incidenty ostatní zaměstnanci společnosti, kteří tak typicky jednají na základě stížností zákazníků (Interní zdroje, 2015).

3.2. Analýza vhodnosti využití automatizovaných testů v T-Mobile

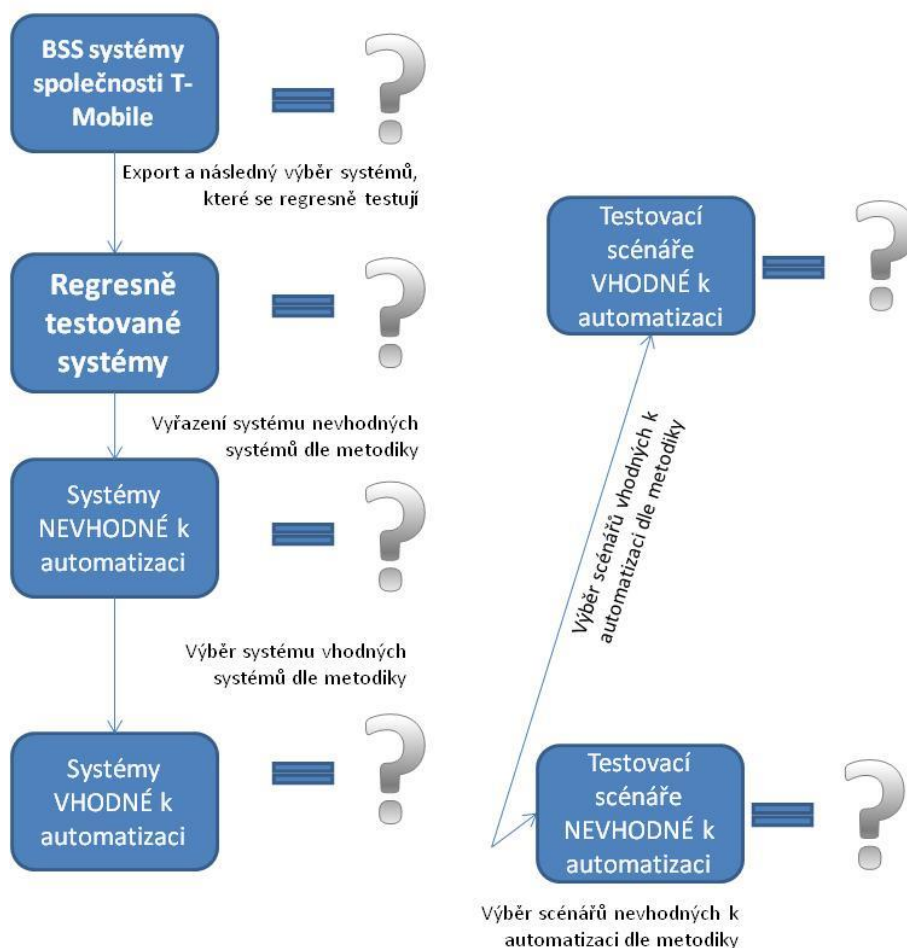
Touto kapitolou lze pomyslně oddělit představení společnosti T-Mobile a vlastní analytickou práci autorky, která ovšem ze současného stavu společnosti vychází. Předmětem analýzy je zhodnocení současného stavu regresních testů ve společnosti T-Mobile a doporučení vhodných systémů pro automatizaci regresního testování. Pro analýzu byly vybrány systémy z T-Mobile BSS³ domény. Jedná se o systémy, kde alespoň z části probíhá integrační a uživatelské testování. Výběr relevantních systémů probíhal následujícím čtyř krokovým způsobem:

³ BSS – Business Support Systems – systémy, které jsou klíčové pro interakci se zákazníkem

1. **Export systémů z DB a prvotní evaluace systémů** - Export seznamu stávajících BSS systémů z konfigurační databáze CMDB z nástroje HPSM. Evaluace systémů společně s architektem společnosti T-Mobile na základě vydefinovaných kritérií. Cílem je zjistit, jaké systémy regresně testujeme a identifikovat technické specifikace systémů.
2. **Analýza systémů – regresně testovaných** - Detailní specifikace systémů, které byly vybrány jako vhodné pro automatizaci. Konkrétní detailní provádění regresních testů (popis systému, rozsah, četnost, ...).
3. **Výběr testovacích scénářů nevhodných k automatizaci** - Výběr testovacích scénářů, které nejsou vhodné k nasazení automatizace dle nastavené metodiky.
4. **Výběr vhodných testovacích scénářů pro automatizaci** dle nastavené metodiky.

Postup analýzy je graficky znázorněn na obrázku 4, kde jsou otazníky označené výstupy, které je třeba v rámci vypracování praktické části naplnit konkrétními hodnotami

Obrázek 4 Postup analýzy



Zdroj: vlastní tvorba (2015)

3.2.1. Export systémů z CMDB a prvotní evaluace

Cílem tohoto kroku je provedení analýzy nad stávajícím výčtem BSS systémů a zjistit, které systémy jsou v současné chvíli regresně testovány. Z konfigurační databáze T-Mobile, umístěné v nástroji HPSM, který je používán pro change management společnosti byl proveden export do excelového souboru. Export z databáze obsahoval celkem 211 položek.

Každá položka obsahuje následující identifikátory. Jedná se o jednoznačný identifikátor – ID, název systému, účel aplikace, zkratka systému, kontaktní osoba v T-Mobile a pole poznámka. Ke zjištění, zda je daný systém regresně testován v zásadě stačilo vyfiltrovat z tabulky systémy testované integračními a uživatelskými testery. Toto mapování bylo provedeno na základě interních materiálů, kde je k dispozici název systému a jméno konkrétního za test zodpovědného testera. V důsledku tohoto omezení byl počet analyzovaných systémů snížen na počet 74. Vyloučené systémy byly vyřazeny z integračního a uživatelského testování z následujících důvodů:

- systém byl již terminován (nekonistence databáze systémů),
- systém již není dále rozvíjen – blíží se terminace systému,
- systém nemá testovací prostředí a na produkci je testován pouze koncovými uživateli (typicky se jedná o aplikace interního charakteru, jako je například aplikace určená pro rezervaci parkovacích míst - ROPA),
- systém, ve kterém dochází pouze k výjimečnému rozvoji – není tak alokován tester s přímou zodpovědností,
- reportingové aplikace, které jsou testovány end-usery
- a aplikace ve správě třetích stran, kde testování probíhá v rámci dodavatelské firmy. Jedná se taktéž o mezinárodní aplikace, kde testování zajišťuje některá z dceřiných společností Deutsche Telecom nebo dedikovaný mezinárodní tým.

Dále proběhla spolupráce s architektem společnosti T-Mobile, (Hynek Ruta), který pomohl vydefinovat další údaje nutné pro další fázi k posouzení, zda je možno u vydefinovaných systémů doporučit automatizaci regresního testu. Architekt doplnil následující informace: bližší účel aplikace, důležitost aplikace (v rozmezí nízká, střední a vysoká), která se odvíjela podle priority řešení výpadku na aplikaci (pod tímto pojmem si lze představit, odlišný přístup k aplikaci s dopadem na 6 milionů zákazníků a k aplikaci s dopadem pouze na interní zaměstnance), typ aplikace (webová, serverová, desktopová, Oracle Forms-reporting), typ aplikace (Front-end; Back-end) a typ uživatele (interní – zaměstnanci, externí – zákazníci).

Seznam 74 systémů byl předložen relevantním testerům a ti měli za úkol doplnit další informace. Ty se týkaly otázek, zda je prováděno regresní testování aplikace (ano; ne), u aplikací, kde byla v předchozí otázce zvolena možnost „Ano“ pak byly doplněny následující informace: Jak často jsou prováděny regresní testy za rok (číselná hodnota), počet modulů aplikace (číselná hodnota, počet koncových uživatelů, počet provázaných systémů (stupeň integrace – počet systémů). V tabulce níže autorka uvádí výpis systémů, kde je prováděno regresní testování a bude na nich provedena další analýza.

Tabulka 3 Regresně testované systémy

Systém	Tester	Popis systému - účel	Důležitost APP	Typ aplikace	Typ aplikace 2	Počet Releaseů / rok	Počet modulů aplikace	Počet uživatelů	Stupeň integrace
Portál -	PQT	Hlavním účelem portálu je	vysoká	webová	Front-end	10	9	800000	12

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

eshop		sdužování informací z několika zdrojů – portletů – do jedné výsledné HTML stránky, konkrétně e-shopu							
e-shop	UAT	Hlavním účelem portálu je sdužování informací z několika zdrojů – portletů – do jedné výsledné HTML stránky konkrétně e-shopu	vysoká	webová	Front-end	26	9	800000	12
VCC	UAT	Systém umožňuje samoobslužné změny a kontrolu využívání služeb.	vysoká	webová	Front-end	26	2	3000000	4
CLF/CSR	UAT	CRM systém - evidence služeb a zákazníků	vysoká	serverová	Front-end	26	6	800	7
IVR	UAT	IVR je systém, který zajišťuje zákaznickou linku a její automatické odpovědi	střední	serverová	Front-end	26	11	2200000	7
Siebel	UAT	nová generace CRM systému - evidence služeb a zákazníků	vysoká	serverová + webová	Front-end	26	4	300	12
CCM	UAT	manipulace s tarify zákazníka, se SIM kartami, nadstavbové balíčky	vysoká	webová	Front-end	16	10	300	10
CCM	PQT	manipulace s tarify zákazníka, se SIM kartami, nadstavbové balíčky	vysoká	webová	Front-end	8	10	300	10
CLF	PQT	CRM systém - evidence služeb a zákazníků	vysoká	serverová	Front-end	8	6	800	7
DARE	PQT	Online mediační billingový systém určený pro roamingové služby	vysoká	serverová	Back-end	3	10	30	5
FNR	PQT	Portační systém pro přenášení fixních telefonních čísel mezi operátory.	vysoká	serverová	Back-end	2	1	20	2
MNR	PQT	Portační systém pro přenášení mobilních telefonních čísel mezi operátory.	vysoká	serverová	Back-end	6	1	20	2
AMX	PQT	Amdocs je billingový systém T-MOBILE	vysoká	serverová	Back-end	12	10	30	5

Zdroj: vlastní tvorba (2015)

V první části analýzy byly po prvotním exportu z konfigurační DB a doplňujících informací ze strany architekta a relevantních testerů vybrány pro další část analýzy následující systémy: Portál-e-shop (za integrační testování - PQT i uživatelské testery - UAT), VCC za uživatelské testy, CLF/CSR (za integrační testování - PQT i uživatelské testery - UAT), IVR za uživatelské testy, Siebel za uživatelské testy, CCM CSR

(za integrační testování - PQT i uživatelské testery - UAT) a DARE, FNR, MNR a AMX za integrační testování. Kritériem výběru byla pouze samotná existence a provádění regresních testů. Ze stávajících 74 systémů je regresně testováno pouze 10 systémů. Takto nízký počet je zarážející, a proto testeři (interní rozhovory, 2015) vysvětlují důvody následujícím způsobem: v rámci testování je T-Mobile zaměřen hlavně na testování nových funkcionalit. Regresní testy jsou požadovány pouze u výše uvedených, strategicky významných aplikací, a to vzhledem k časové náročnosti vytváření a údržby testovacích scénářů a samotného regresního otestování funkcionalit.

Dle interních materiálů, konkrétně operativních cílů na rok 2015, je cílem regresní testy rozšířit tak, aby bylo dosaženo strategického cíle, který je směřován ke zvýšení kvality systémových dodávek. Problémem jsou samozřejmě - v případě navýšení počtu regresních testů - rostoucí finanční a personální problémy. Konkrétní nákladovou a přínosovou částí se bude zabývat separátní kapitola „Business Case“. V další kapitole bude představena metodika výběru vhodných systémů pro automatizaci, která bude pracovat se základnou systémů, kterou již v současné chvíli společnost testuje.

3.2.2. Analýza systémů – regresně testovaných

Předmětem této analýzy jsou v předchozí kapitole vydefinované systémy, kde jsou v současné chvíli prováděny regresní testy. Konkrétně jde o celkem 10 systémů. Cílem je vybrat z množiny deseti systémů takové systémy, které je vhodné doporučit k automatizaci a k detailnější analýze testovacích scénářů. Nejdříve je nutné určit kritéria výběru. Kritéria výběru mají za cíl na základě poskytnutých informací odhalit systém, který je nevhodný pro automatizaci. V níže uvedené tabulce jsou jednotlivá kritéria popsána včetně vyhodnocovací stupnice.

Tabulka 4 Kritéria výběru systémů vhodných k automatizaci

ID kritéria	Název kritéria	Popis – cíl kritéria	Indikátor	Hodnotící stupnice
SA01	Rozvoj systému	Cílem je vybrat takové systémy, u kterých dochází ke kontinuálnímu rozvoji a není plánována terminace systému.	Počet releasů za rok 2014 (+ případná informace o terminaci systému)	Méně než 2 releasové úpravy systému ročně = nesplněno. Terminace systému = kritérium nesplněno
SA02	Důležitost systému	Cílem je vybrat pouze systémy se střední a vyšší důležitostí pro T-MOBILE	Důležitost systému	Důležitost systému = nízká – kritérium nesplněno
SA03	Náročnost testování	Cílem je vybrat pouze ty systémy, kde díky automatizaci může dojít k časové nebo finanční úspoře – měřeno ročně.	Náročnost regresního testu/ MDs/ rok	< 6 Mds/rok – kritérium nesplněno

Zdroj: vlastní zpracování (2015)

Každé kritérium ve výše uvedené tabulce má své opodstatnění. Kritérium Rozvoje systému (SA01) je založeno na architektonické strategii společnosti T-Mobile (CAR_architecture, 2014), která v následujících letech nemá v plánu investovat větší personální a finanční prostředky do zanikajících systémů. Kritérium je zaměřeno čistě na osobní potřebu podniku. Kritérium Důležitost systému (SA02) je zvoleno s ohledem na pozici systému vzhledem k jeho dopadům v případě nefunkčnosti. Systémy s nízkou důležitostí ohrožují primárně interní uživatele T-MOBILE a v případě jejich výpadku není ohrožen business společnosti T-Mobile, ale pouze osobní komfort uživatelů aplikace. Na podporu tohoto tvrzení i autor Regent (2013) zmiňuje, že: *Dobrým výchozím bodem je garance stability funkcionalit z pohledu koncového uživatele, zákazníka, na front-endu*“. Třetí kritérium Náročnost testování vychází z tvrzení z webové stránky Testování softwaru (2012), které říká, že hlavním cílem automatizace testování je časová úspora při následném spouštění. Kritérium má za cíl eliminovat z užšího výběru takové systémy, kde je v současné chvíli časová náročnost příliš nízká (méně než 7 MDs ročně = $7 \cdot 4400 = 30800$ CZK) a společnosti se tak v každém případě nevyplatí investovat do automatizace testů.

Pro vyhodnocení výše uvedených kritérií je nutné zjistit a sumarizovat výše uvedené indikátory:

- počet releasů za rok/systém,
- důležitost systému,
- a náročnost regresního testu za rok v MDs.

Počet Releasů za rok/systém vychází z tvrzení testerů, kolik provádí regresních testů ročně (vzhledem k faktu, že v každém releasu, kde byla systému měněna funkcionalita, byl proveden regresní test). Testeři při určování počtu releasů vycházeli z releasového kalendáře společnosti T-Mobile, ve kterém byly zaznamenány všechny dopady na jednotlivé systémy. Tento detail byl k dispozici za celý rok 2014. Jedinou výjimku zde tvoří UAT testeři, kteří regresně testují všechny výše uvedené systémy každých 14 dní. Kromě hlavních dodávek v rámci releasů testují stávající funkcionalitu systémů po produkčních opravách, které nejsou nasazovány releasově, ale ve speciálních opravných nasazovacích oknech.

Důležitost systému byla již uvedena v první části vyhodnocení systémů. Tato informace pochází od IT architekta společnosti T-Mobile a prošla validací autorky a zainteresovaných testerů.

Náročnost regresního testování – tento údaj bylo nutné zjistit od testerů, kteří testy provádí. Tento údaj je samozřejmě expertním odhadem, vzhledem k situaci, že se nedá předpokládat konstantní doba samotného testu, která je závislá na mnoha ukazatelích. Mezi ně patří například stav testovacích dat a testovacího a produkčního prostředí, chybovost či komplexnost testovacího prostředí. Testeři do přibližného času provedení testu nezapočítávali čas určený pro evidenci a popis chyb a studium dokumentace, a to vzhledem k faktu, že tuto práci budou muset vykonávat i v případě automatizace testů (i když například proces evidence a popis chyb je možno velmi zjednodušit). Do odhadů pracnosti byla započítána i nutnost přetestování v případě nalezení chyby.

Způsob vyhodnocení kritérií byl nastaven následujícím způsobem. Do užší analýzy (již na úrovni konkrétních testovacích scénářů budou zařazeny pouze ty systémy, které

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

budou splňovat všechna výše uvedená kritéria. Níže je uvedena tabulka se zjištěnými indikátory. Pod tabulkou se nachází vyhodnocení jednotlivých kritérií.

Tabulka 5 Systémy vhodné k automatizaci

Systém	Tester	Důležitost aplikace	Počet Releasů/ rok	Náročnost testování/MDs /rok	Doporučení k automatizaci systému
Portál - eshop	PQT	vysoká	10	40	Ano
e-shop	UAT	vysoká	26	5	Ne
VCC	UAT	vysoká	26	13	Ano
CLF/CSR	UAT	vysoká	26 - Terminace systému 2016	13	Ne
IVR	UAT	střední	26	5	Ne
Siebel	UAT	vysoká	26	26	Ano
CCM	UAT	vysoká	16 - Terminace systému 2016	2	Ne
CCM	PQT	vysoká	8 - Terminace systému 2016	96	Ne
CLF	PQT	vysoká	8 - Terminace systému 2016	48	Ne
DARE	PQT	vysoká	3	1,5	Ne
FNR	PQT	vysoká	2	0,5	Ne
MNR	PQT	vysoká	6 - Terminace systému 2016	12	Ne
AMX	PQT	vysoká	12	0 - testy jsou již automatizované	Ne - již automatizováno

Zdroj: Vlastní tvorba (2015)

Zhodnocení kritéria SA01 – Rozvoj systému

Toto kritérium bylo postaveno na počtu releasů, kdy byla funkcionální daného systému měněna. Tento údaj je vztažen k roku 2014. Dle výsledků uvedených v tabulce výše kritérium nesplňují pouze ty systémy, kde je plánovaná terminace v roce 2016. Tyto systémy jsou nahrazovány novým CRM řešením Siebel. Konkrétně se jedná o aplikace CLF/CSR, CCM a MNR. Rozhodnutím managementu je takové aplikace dále nerozvíjet a pouze udržovat v provozuschopném stavu. Vzhledem k tomu, že v letošním a v dalších letech, nebudou tyto systémy spouštět žádné klíčové funkcionality, neexistuje předpoklad pro optimalizaci testovacího procesu v podobě případné automatizace.

Zhodnocení kritéria SA02 – Důležitost systému

Důležitost systémů byla určována podle ukazatele, který byl naplněn expertním odhadem architekta a testerů. Z výše uvedených aplikací jsou všechny aplikace kromě IVR uvedeny s vysokou důležitostí. IVR (zákaznická linka s automatickým výběrem možností pomocí automatu) je uveden s důležitostí střední, a to díky tomu, že v případě možného výpadku je možné jej dočasně nahradit pracovníky na zákaznické lince, osobní návštěvou na prodejně, případně aplikací VCC, která umožňuje samoobslužnou manipulaci se službami. Kritérium tak bylo splněno ve všech zkoumaných případech.

Zhodnocení kritéria SA03 – Náročnost testování

Náročnost současného manuálního testování ohodnocovali jednotliví testéři. Vycházeli z vlastní, minimálně dvouleté zkušenosti. V rámci vytvoření vlastního expertního odhadu vycházeli z vydefinovaných testovacích scénářů a k nim přiřazovali relevantní pracovníci. Výše pracovníci respektuje čas strávený na přípravné části, realizační čas a čas na případné přetestování aplikace po opravě nalezené chyby. Data v tabulce výše jsou agregována na úroveň jednotlivých systémů. Detail po jednotlivých testovacích scénářích je přiložen v příloze v části Business Case. Na základě výše uvedených odhadů pracovníci a četnosti regresního testování za rok byla vypočítána celková hodnota

indikátoru. Kritérium Náročnost testování > 6MDs nesplnily následující aplikace: E-shop (UAT) a IVR s 5 MDs ročně, CCM (UAT) s 2 MDs ročně, systém DARE s pracností 1,5 MDs ročně, FNR s 0,5 MDs ročně. Speciálním případem je billingový systém Amdocs, kde byly v průběhu analýzy testy zautomatizovány a pracnost na automatizovaných testech tak již spočívá jen ve spouštění testů a případné evidenci defektů. Aplikace, které kritérium splnily, jsou následující: Portál-eshop - PQT (40 MDs), VCC (13 MDs), CLF/CSR (13 MDs), Siebel (26 MDs), CCM (PQT – 96 MDs) a systém MNR s 12 MDs ročně.

Celkové zhodnocení

V tabulce číslo 5 je přehledně označeno splnění (zelenou barvou) či nesplnění (červenou barvou) požadovaného kritéria. Výsledný stav dle nastavení celkového kritéria pro výběr systému vhodného k automatizaci je možno vidět výše v poslední sloupci tabulky 5. Podmínka byla nastavena následujícím způsobem. Pokud má být systém doporučen k další analýze (analýze testovacích scénářů), musí být splněna všechna kritéria, a to: Rozvoj systému – u systému dochází ke kontinuálnímu rozvoji a nebude v nejbližších letech terminován, Důležitost systému – systém má v případě výpadku velký nebo střední vliv na business společnosti, a zároveň musí splňovat kritérium Náročnost testování – kdy kritérium splňují pouze ty systémy, kde je pracnost na manuální regresní testování v součtu za rok vyšší než 6 MDs.

Všechna výše uvedená kritéria splnily pouze tři systémy. Jedná se o Portál, e-shop (PQT), VCC a Siebel. Tyto systémy budou vedeny jako vstupní pro další část analýzy, která se bude detailně zabývat jednotlivými systémy, způsobem testování a konkrétními typy testovacích scénářů, které jsou v rámci současných regresních testů validovány. Výsledky této analýzy byly verifikovány zodpovědnými testery a výsledky se shodují s jejich očekáváním. Následující část se zabývá již konkrétní definicí testovacích scénářů a jejich vhodností či případnou nevhodností pro automatizaci.

3.2.3. Výběr testovacích scénářů nevhodných k automatizaci

V rámci této kapitoly budou vydefinovány jednotlivé testovací scénáře, které jsou v rámci vybraných systémů Portál (PQT), VCC a Siebel regresně testovány. Na základě těchto testovacích scénářů budou pomocí níže vydefinovaných kritérií vybrány testovací scénáře, které z podstaty automatizovaných testů nejsou vhodné k automatizaci. Kritéria pro výběr nevhodných scénářů byla nastavena následujícím způsobem:

Tabulka 6 Kritéria výběru testovacích scénářů nevhodných k automatizaci

ID kritéria	Název kritéria	Popis – cíl kritéria	Indikátor	Hodnotící stupnice
TA01	Audiovizuální kontrola	Cílem je vybrat takové test scénáře, u kterých dochází během regresního testování k audiovizuální podobě	Potřeba audiovizuální kontroly 0: kontrola není potřebná; 1: kontrola barev a obrázků; 2: audio kontrola a plovoucí pozice na zobrazení v GUI	Potřeba audiovizuální kontroly = hodnota indikátoru 2 = vylučovací kritérium SPLNĚNO

TA02	Míra integrace	Cílem je vybrat pouze takové scénáře, které pro regresní testy nepotřebují vysokou míru integrace s ostatními systémy	Míra integrace 0: průběh testovacího scénáře je pouze v jednom systému s následnou kontrolou v Datawarehouse 1: průběh testovacích scénářů je závislý na více než 1 systému, ale méně než 4. 2: testovací scénář vyžaduje kontrolu ve více než 3 systémech	Míra integrace = hodnota indikátoru 2 = vylučovací kritérium SPLNĚNO
TA03	Konzistentnost testovacích dat	Cílem je vybrat pouze takové scénáře, kde je možné zajistit validní (konzistentní) podobu zákazníka na celém průběhu testovacího scénáře	Míra konzistentnosti zákazníka 0: je možné v plném rozsahu zajistit konzistenci zákazníka 1: částečná konzistentnost zákazníka 2: není možné, zajistit konzistentnost zákazníka	Míra konzistentnosti zákazníka = hodnota indikátoru 2 = vylučovací kritérium SPLNĚNO

Zdroj: vlastní zpracování (2015)

Pro vyhodnocení výše uvedených kritérií je nutné zjistit a sesumarizovat výše uvedené indikátory:

- Potřeba audiovizuální kontroly
- Míra integrace
- Míra konzistentnosti zákazníka

Nastavení kritéria TA01 – Audiovizuální kontrola

Potřebou audiovizuální kontroly se myslí nutnost kontroly správného zobrazení (interní zdroje - automatizace testů, 2015). Typickým příkladem ve společnosti T-Mobile je vzhledová kontrola webových aplikací, e-shopu či bannerů. Tuto část testování zajišťuje uživatelský tester. Integrační tester nemá vzhledovou či případnou audio kontrolu vymezenou v pracovní náplni. U tohoto kritéria je důležité zkoumat nejenom samotnou potřebu audiovizuální kontroly, ale hlavně míru této potřeby. Hodnota tohoto indikátoru bude zjištěna od testerů, kteří vybranou aplikaci testují. Na výběr je ze tří možností hodnot indikátoru, a to: 0 – audiovizuální kontrola není potřebná (touto hodnotou budou označeny takové testovací scénáře, které pro správné provedené regresního testu nevyžadují žádnou kontrolu audiovizuálního typu), další výběrová hodnota je označena číslem 1 - kontrola barevného nastavení a obrázků (takto budou označeny testovací scénáře, které jistou míru audiovizuální kontroly vyžadují). Tento typ kontroly je možné

alespoň částečně zkontrolovat ověřením požadovaného barevného spektra (RGB⁴ kódu barvy) vůči skutečnému stavu testovaného objektu. Zde je vyžadováno detailnější (a tím pádem pracnější) nastavení automatizovaného testu. Druhou možností, jak problém s kontrolou vyřešit, je rychlá manuální kontrola obrazovek testerem po otestování skriptem určeným pro automatizovaný test. Jako hodnota indikátoru určená k vyloučení testovacího scénáře výše uvedeného kritéria, je možnost s hodnotou 2 - audio kontrola nebo plovoucí pozice objektu v GUI⁵. Již z uvedené hodnoty kritéria je patrné, že audio kontrola je na automatizaci velmi těžká, ne-li nemožná. Pokud toto aplikujeme na systémové prostředí T-Mobile, může se jednat například o poslech hlásek ze systému IVR, který pomocí automatu hlasově navádí zákazníky v rámci položek obsažených v samoobslužných kanálech. Systémově je možné ověřit pouze existenci hlásky, nikoliv však její obsahovou a kvalitativní část. Plovoucí pozicí objektu se v rámci tohoto vyhodnocovacího kritéria předpokládá existence takového objektu na testované obrazovce, která je závislá na pozici ostatních, někdy velmi často variabilních objektů. Typicky lze mluvit o pozicích reklamních či poutacích bannerů na portálu www.t-mobile.cz. Takový typ kontroly bude vždy závislý na současné situaci nastavení objektu a manuální kontrole stavu testerem.

Nastavení kritéria TA02 – Míra integrace

Dalším kritériem byla míra požadované integrace v rámci testovacího scénáře. Tímto kritériem je snaha odhalit takové typy testovacích scénářů, které pro svůj regresní test požadují reakci mnoha systémů. Dle Kanling, Menggi (2006, str. 30) lze efektivně automatizovat integrační test pouze v případě, kdy budou rozhraní mezi systémy nahrazena takzvaným „stubováním“. Kangling a Menggi (2006, str. 242) definují Stub metodu jako tzv. dvojníka, který imituje původní systém. Na volání dokáže reagovat s předpřipravenými výsledky nebo navrženou logikou, a díky tomu docela věrně napodobuje svou předlohu. Důvodem výběru kritéria je analyzovaný typ testů (integrační a uživatelské testy), nikoliv možnost automatizace izolovaných systémových testů, kde je stubování vzhledem k povaze testu žádoucí. Indikátor byl zvolen k samotnému počtu požadované integrace na ostatní systémy při regresním testování a nabývá tři hodnot. První z nich je hodnota 0 – průběh testovacího scénáře probíhá pouze v primárním systému a maximálně s kontrolou v datovém skladu (DWH) nebo na autorizačním serveru. Hodnotu 1 nabývá indikátor v případě, kdy test probíhá v rámci primárního systému (Portál, VCC nebo Siebel) a maximálně dalších tří systémů. Toto testy jsou náročnější k automatizaci, ale vzhledem k vysoké manuální náročnosti kontrol a chybovosti, je stále výhodné testy automatizovat. Vylučovací hodnotou indikátoru je možnost 2 – která indikuje nutnost regresního testu ve spolupráci s více než 4 dalšími systémy. Hodnoty jednotlivých indikátoru naplňuje stejně jako v předchozím případě relevantní tester.

Nastavení kritéria TA03 – Konzistentnost testovacích dat

Posledním kritériem je konzistentnost testovacích dat. Dle Gilfillan (2003, str. 284) konzistence dat představuje stav „*v jakém se nacházejí data, když jsou naplněny určité podmínky. Jedním z pravidel může být to, že každá faktura musí mít nějaký vztah*“

⁴ RGB kód barvy lze definovat například dle Schafer (2009, str. 398) jako RGB funkci, která je používána pro zobrazení barev, která pomocí smíchání určitého množství barev červené, zelené a modré (red-green-blue) dosáhne požadovaného odstínu zvolené barvy.

⁵ GUI – neboli grafické uživatelské rozhraní (Graphic User Interface)

k zákazníkovi v tabulce.“ Klíčovým faktorem pro provedení regresního testu je kvalita testovacích dat napříč celým řetězcem systémů, kterým testovací případ prochází. Tento problém je velmi citlivé téma kvality testovacího prostředí ve společnosti T-Mobile. Testovací prostředí není zrcadlovým obrazem prostředí produkčního. Mezi důvody patří omezená kapacita HW prostředků, z části neexistující testovací prostředí u produktů třetích stran a nedůslednou konfigurací ze strany vývojových a business týmů, kteří prioritně řeší stav produkčního prostředí (Interní materiály - stav testovacího prostředí T-Mobile, 2015). Toto kritérium vychází z předpokladu, že v případě nedostatečné konzistence testovacích dat (zákazníků a jejich služeb) není možné provádět automatizované testy. Automatizované testy jsou přímo závislé na exaktně vydefinovaných kontrolách zákazníků, služeb a jejich parametrů. Indikátor, který je vyplňován testerem, může nabývat stejně jako v předchozích případech tří hodnot (0; 1; 2), přičemž hodnota 0 znamená, že testovací případ pracuje v současnou chvíli s konzistentním zákazníkem, který je obsažen ve všech relevantních, návazných systémech v takové podobě, která je z pohledu funkcionality správná. Hodnotou 1 označují testeři testovací scénáře takového typu, kde je konzistentnost testovaného zákazníka částečná, ale zároveň toto omezení nelimituje použití automatizovaného testu – stále je předpokládán přínos automatizace. Třetí hodnotou – eliminační - je hodnota 2, která hodnotí testovací scénář jako neautomatizovatelný z pohledu velkého množství manuálních zásahů individuálního charakteru.

Vyhodnocení kritérií

Tabulka 7. níže přehledně sumarizuje zjištěné hodnoty a v posledním sloupci vyhodnocuje testovací scénář z celkového pohledu. Vyhodnocovací logika byla postavena na výše uvedených hodnotách indikátorů, kdy pro vyřazení celého testovacího scénáře stačilo, aby pouze jedno kritérium ze tří nabylo hodnoty 2. V důsledku provedené analýzy lze říci, že jediné nedoporučení testovacích scénářů je u modulu Portál. Zde byly k automatizaci zvoleny nevhodné testovací scénáře s názvem Mapa pokrytí a Platba faktury.

Testovací scénář: Mapa pokrytí

Testovací scénář Mapy pokrytí byl nedoporučen k automatizaci z níže uvedených důvodů. Mapa pokrytí je samostatný modul v rámci Portálového řešení veřejně dostupného na <http://coverage-t-mobile.position.cz/>. Cílem této aplikace je pomocí barevných vrstev a filtrů zákazníkovi prozradit, zda v jím vybrané lokalitě existuje pokrytí (v detailu LTE, HSPA, +42, 3G, GSM a DSL služeb). Již samotný účel modulu předjímá nutnost rozsáhlé audiovizuální kontroly, která při kontrole vyžaduje velmi přesnou definici parametrů zvolených zákazníkem. Detail zobrazení je natolik podrobný, že většinu případných nedostatků není možno odhalit bez rozsáhlé analýzy velkého množství vstupních a výstupních dat, které zajišťuje velké množství integrovaných systémů. Konzistentnost testovacích dat je velmi těžké nastavit a liší se velkým množstvím parametrů.

Testovací scénář: Platba faktury

Stejně jako testovací scénář Mapy pokrytí nebyl scénář Platba faktury doporučen k automatizaci. U tohoto scénáře taktéž existuje jistá forma audiovizuální kontroly, která však svým rozsahem neměla dopad na vyloučení testovacího scénáře. Velkým problémem u tohoto testovací scénáře je vysoká míra integrace a konzistentnost zákazníka. Tyto dva zmíněné problémy spolu značně souvisí. Testovací scénář Platba

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

faktury spočívá v otestování možnosti platby faktury pomocí online platební brány České Spořitelny. Míra integrace je v tomto případě přímo závislá na celé řadě billingových a CRM systémů, ale hlavně v také na integraci na samotnou platební bránu. Jak již bylo zmíněno výše, aplikace třetích stran téměř znemožňují validní manuální integrační test, natož automatizovaný, kdy se předpokládá, že vše bude exaktně vydefinováno a společnost bude schopna ovlivnit opravy a úpravy testovaného systému.

Tabulka 7 Doporučení testovacích scénářů k automatizaci

Systém	Tester	Název testovacího scénáře	Audiovizuální kontrola	Míra Integrace	Míra konzistentnosti zákazníků	Doporučení k automatizaci TC
Portál	PQT	Login + Registrace	0	1	0	Ano
	PQT	Můj T-Mobile	0	1	1	Ano
	PQT	E-shop	0	1	1	Ano
	PQT	Diskuzní fórum	0	0	0	Ano
	PQT	Dobíjení kreditu	0	1	1	Ano
	PQT	Vyhledávací modul	0	1	0	Ano
	PQT	Časté dotazy (FAQ)	0	0	0	Ano
	PQT	Kontaktujte nás	0	1	0	Ano
	PQT	Platba faktury	1	2	2	Ne
	PQT	Mapa pokrytí	2	2	2	Ne
Siebel	UAT	Základní testy	0	1	1	Ano
	UAT	Ordering	1	0	1	Ano
	UAT	Migrace	1	1	1	Ano
	UAT	Výměna SIM	0	1	1	Ano
	UAT	Portace	0	0	1	Ano
	UAT	Dobíjení	1	1	1	Ano
	UAT	RTD	1	0	1	Ano
VCC	UAT	A - Dashboard	1	1	1	Ano
	UAT	B - Vyúčtování	1	1	1	Ano
	UAT	C - Změna tarifu	1	1	1	Ano
	UAT	D - Roaming	1	1	1	Ano
	UAT	E - Mých 5	1	1	1	Ano
	UAT	F - Tarify a Zvýhodnění	1	1	1	Ano
	UAT	G - Mobilní internet	1	1	1	Ano
	UAT	H - Změna admina	1	0	1	Ano
	UAT	CH - Online výpis služeb	1	1	1	Ano
	UAT	I - Moje útrata	1	0	1	Ano
	UAT	J - Datový limit	1	0	1	Ano
	UAT	K - Kontrola kreditu	0	0	1	Ano
	UAT	L - Výhody našim	1	0	1	Ano
	UAT	M - Kontrola Datového limitu	1	0	1	Ano
	UAT	N - Mobilní internet	1	0	1	Ano
UAT	O - Online výpis služeb	0	0	1	Ano	
UAT	P - Proklik do iVCC	0	1	1	Ano	
UAT	Q - Náhodné proklikání	1	1	1	Ano	

Zdroj: vlastní zpracování (2015)

Ostatní testovací scénáře budou použity v následující kapitole, která bude hodnotit z pohledu efektivnosti využití ty testovací scénáře, které byly předchozí analýzou identifikovány jako potenciálně vhodné.

3.2.4. Výběr vhodných testovacích scénářů pro automatizaci

Cílem této části analýzy je zhodnocení testovacích scénářů z pohledu efektivnosti automatizace. V zásadě lze říci, že některé testovací scénáře není efektivní automatizovaně testovat z následujících důvodů:

- systém bude v nejbližší době terminován nebo jeho rozvoj není nikterak progresivní,
- systém v případě výpadku nemá dopad na business společnosti,
- provedení manuálního testování je časově nenáročné a tvoří jednotky MDs ročně,
- je třeba velkého množství audiovizuálních kontrol, kde není možné nahradit rozlišovací schopnost lidského jedince,
- existence velkého množství integrovaných systémů – převážně produktů třetích stran,
- nemožnost dosáhnout ani částečné konzistence testovacích dat.

Všechny výše uvedené body se staly vstupem pro vytvoření vylučovacích kritérií v předchozí části analýzy. V této části analýzy budou vybrané testovací scénáře podrobeny zkoumání z pohledu efektivity. Lze tedy v návaznosti na teoretickou část práce obecně říci, že efektivní je zautomatizovat takové scénáře, které jsou:

- často prováděné,
- stabilní – z pohledu proměnlivosti jednotlivých kroků testovacího scénáře,
- manuálně náročné – z pohledu stále se opakující činnosti.

Vzhledem k výše uvedenému byla nastavena následující hodnotící kritéria.

Tabulka 8 Kritéria výběru testovacích scénářů vhodných k automatizaci z pohledu efektivnosti

ID kritéria	Název kritéria	Popis – cíl kritéria	Indikátor	Hodnotící stupnice
TC01	Počet opakování	Cílem je vybrat takové test scénáře, které se často v průběhu roku opakují.	Počet opakování /rok	Počet opakování /rok > 12 x = SPLNĚNO
TC02	Stabilita testovacího scénáře	Testovací případ není potřeba často měnit. Testovaná funkcionality se mění relativně zřídka	Stabilita (vysoká; střední; nízká)	Stabilita = vysoká nebo střední = SPLNĚNO
TC03	Manuální náročnost	Cílem je vybrat takové scénáře, kde automatizace ušetří stále se opakující manuální činnost testera	Manuální náročnost (vysoká, střední, nízká)	Manuální náročnost = vysoká nebo střední = SPLNĚNO

Zdroj: vlastní zpracování (2015)

Pro vyhodnocení výše uvedených kritérií je nutné zjistit a sesumarizovat výše uvedené indikátory:

- počet opakování/rok,
- stabilita
- a manuální náročnost.

Pro celkové splnění podmínky efektivnosti musí každý testovací scénář splnit nejméně dvě ze tří následujících kritérií.

Nastavení kritéria TC01 – Počet opakování

Cílem kritéria je vybrat pouze takové scénáře, které se v rámci roku poměrně často opakují. Zjišťovaný údaj je totožný s počtem regresního testování na systémech. V tomto případě je ale dvojnásobně navýšen na základě faktu z interních zdrojů (2014), které říkají, že každý testovací scénář je v průměru přibližně dvakrát přetestován kvůli otestování funkcionality po opravě chyby nalezené v prvním běhu testu. Vyhodnocení tohoto indikátoru bude probíhat na základě počtu testování testovacího scénáře v průběhu roku. Počet releasů, kdy testovací scénář splňuje podmínku efektivnosti, je nastaven na počet 12.

Nastavení kritéria TC02 – Stabilita testovacího scénáře

Cílem tohoto kritéria je vybrat takové scénáře, u kterých budou požadavky a změny v průběhu roku minoritní záležitostí. Každá změna, byť sebemenší, zvyšuje náklady na samotné otestování funkcionality. Rozdíl pak činí, zda změna testovacího scénáře spočívá v pouhé změně parametru, či je měněna logika testování. Samozřejmě je nutné se změnami počítat zvláště u vybraných rozvíjených systémů, kde se změny předem očekávají. I manuální testování je citlivé na časté změny postupu testování a testeři se s touto potřebnou, leč neoblíbenou aktivitou, denně setkávají. Je tedy předpokladem, že v rámci vlastních regresních testovacích scénářů dokážou zhodnotit „stabilitu testovacího případu“. Proto byla zvolena metoda subjektivního hodnocení testovacích případů testery, kteří měli možnost využití následující stupnice. Vysokou stabilitou se myslí testovací scénář, kde jsou úpravy v průběhu roku minoritní nebo žádné. Typicky se může jednat o přihlašovací proces, který pracuje se stále stejnými identifikátory, jako je například uživatelské jméno a heslo. Střední stabilita by měla být nastavena u testovacích scénářů, kde v průběhu posledního roku (2014) došlo ke změnám parametrického charakteru. Nízkou stabilitou testovacího scénáře má popsat situaci, kdy je testovací scénář opakovaně měněn z pohledu logické funkcionality. Pro splnění kritéria efektivnosti je nutné, aby byl testovací scénář označen vysokou nebo střední stabilitou. Testovací případy, které jsou často měněny z pohledu logické funkcionality, jsou doporučeny k pouhému manuálnímu vykonávání testovacího scénáře.

Nastavení kritéria – TC03 - Manuální náročnost

Toto kritérium má snahu odhalit takové testovací případy, u kterých tester při regresním testu používá velké množství manuálních kroků, jako je například ruční plnění formulářových polí. Tato část testovacího scénáře částečně vychází z datové přípravy a je předpokladem, že při jednoznačném namapování obsahu formulářových polí a jejich pozice ve formuláři, lze testerům velmi ulehčit práci. Výhodou je také jednoduchost automatizace tohoto kroku v testovacím případě. Hodnotící stupnice je vysoká, střední a nízká. Stejně jako v předchozích kritériích bude testovací scénář hodnotit tester, který běžně provádí regresní testy u dané aplikace.

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

V níže uvedené tabulce je přehled výsledků analýzy:

Tabulka 9 Doporučení testovacích scénářů k automatizaci z pohledu efektivity

Systém	Tester	Název testovacího scénáře	Počet opakování/rok	Stabilita testovacího případu	Manuální náročnost	Doporučení k automatizaci TC - efektivnost
Portál	PQT	Login + Registrace	20	Vysoká	vysoká	Ano
	PQT	Můj T-Mobile	20	Střední	vysoká	Ano
	PQT	E-shop	20	Nízká	střední	Ano
	PQT	Diskuzní fórum	20	Vysoká	vysoká	Ano
	PQT	Dobíjení kreditu	20	Vysoká	střední	Ano
	PQT	Vyhledávací modul	20	Vysoká	vysoká	Ano
	PQT	Časté dotazy (FAQ)	20	Vysoká	vysoká	Ano
	PQT	Kontaktujte nás	20	Vysoká	střední	Ano
Siebel	UAT	Základní testy	52	Vysoká	vysoká	Ano
	UAT	Ordering	52	Střední	vysoká	Ano
	UAT	Migrace	52	Střední	vysoká	Ano
	UAT	Výměna SIM	52	Střední	vysoká	Ano
	UAT	Portace	16	Střední	vysoká	Ano
	UAT	Dobíjení	16	Střední	střední	Ano
	UAT	RTD	16	Střední	střední	Ano
VCC	UAT	A - Dashboard	52	Nízká	střední	Ano
	UAT	B - Vyúčtování	52	Střední	střední	Ano
	UAT	C - Změna tarifu	52	Střední	střední	Ano
	UAT	D - Roaming	52	Střední	střední	Ano
	UAT	E - Mých 5	52	Střední	střední	Ano
	UAT	F - Tarify a Zvýhodnění	52	Střední	vysoká	Ano
	UAT	G - Mobilní internet	52	Střední	střední	Ano
	UAT	H - Změna admina	52	Vysoká	střední	Ano
	UAT	CH - Online výpis služeb	52	Střední	střední	Ano
	UAT	I - Moje útrata	52	Střední	střední	Ano
	UAT	J - Datový limit	52	Střední	nízká	Ano
	UAT	K - Kontrola kreditu	52	Střední	nízká	Ano
	UAT	L - Výhody našim	52	Střední	vysoká	Ano
	UAT	M - Kontrola Datového limitu	52	Střední	střední	Ano
	UAT	N - Mobilní internet	52	Střední	vysoká	Ano
	UAT	O - Online výpis služeb	52	Střední	střední	Ano
UAT	P - Proklik do iVCC	52	Vysoká	střední	Ano	
UAT	Q - Náhodné proklikání	52	Střední	nízká	Ano	

Zdroj: vlastní úprava (2015)

Vyhodnocení kritérií

Z výše uvedených výsledků je patrné, že všechny zvolené oblasti testovacích scénářů je vhodné a efektivní automatizovat. Vyhodnocení kritérií bylo nastaveno tak, že postačí splňovat dvě ze tří nastavených kritérií. Všechny testovací scénáře jsou velmi často opakovány, a tak splnily podmínku minimálně 12 opakování ročně.

Jako nestabilní označili testeři testovací scénáře Portál – e-shop a VCC (Dashboard). Nestabilitu výše uvedených testovacích případů v čase lze vysvětlit následujícím způsobem. Ve společnost T-Mobile v posledních letech probíhaly časté implementační změny poměrně zásadního charakteru na Portálu a e-shopu. Typickou změnou byla optimalizace pro mobilní zařízení, kompletní re-desing stránek, zeštíhlení portfolia

prodáváných produktů, změny nákupních procesů a logistiky, možnost splátkového kalendáře u zakoupených zařízení, pojištění HW, navyšování FUP u datových tarifů a dobíjecí akce pro majitele předplacených karet. Všechny tyto změny je samozřejmě nutné otestovat v rámci nasazení na produkční prostředí, ale postupně i zařazovat do obsahu regresních testů. I v budoucnu je předpokládána nižší stabilita těchto testovacích případů, vzhledem k tomu, že jsou to funkcionality, které jsou nejčastěji inovovány (Interní zdroje, 2015). U výše zmíněných testovacích scénářů je jen třeba při případné automatizaci počítat s tím, že náklady na úpravy skriptů provádějících automatizované testování, budou častější a tím pádem i nákladnější. Vzhledem ale ke strategické povaze systému VCC a Portál lze předpokládat, že tato investice bude vykoupena vyšší kvalitou a přínosy v podobě spokojenosti zákazníka a interní obsluhy aplikací.

Posledním vyhodnocovaným kritériem bylo: Kritérium manuální náročnosti, které sledovalo, jak velkou manuální práci pro testery představuje vykonávání výše uvedených souborů testovacích scénářů. V negativním pojetí kritéria zde byly označeny soubory následujících testovacích scénářů: VCC - Datový limit, VCC - Kontrola kreditu a VCC – Náhodné proklikání. Tyto scénáře jsou specifické tím, že téměř nevyžadují manuální práci testera v podobě například vyplňování formulářů. Tyto kroky obstarávají předchozí scénáře a výše uvedené na ně pouze navazují.

3.2.5. Shrnutí výsledků analýzy

Cílem této podkapitoly je shrnutí výsledků analýzy a doporučení. Základní soubor analyzovaných dat činil 211 systémů. Tyto systémy jsou obsaženy v konfigurační databázi CMDDB společnosti T-Mobile, konkrétně poté v doméně BSS. Z 211 systémů bylo za pomoci IT architekta a testerů vyspecifikováno 7 systémů, které se regresně testují oddělením integračních testerů (PQT), a 6 systémů, které regresně testují uživatelští testéři (UAT). Na základě vydefinovaných kritérií definovaných v kapitole 3.4.2 jako je rozvoj systému, důležitost systému a náročnost regresního testování, byl vyspecifikován konečný počet systémů, kde nedává automatizace smysl převážně ze strategického pohledu firmy a účelu aplikace. Převážně se jednalo o systémy, kde se jejich životní cyklus blížil k brzkému konci (rok 2016 - 2017), nebo jsou regresní testy prováděny pouze sporadicky. Systémy, které byly vyřazeny z další analýzy: Amdocs - AMX (v průběhu analýzy byly testy zautomatizovány), FNR, MNR, CLF/CSR, CCM a IVR.

Do další části analýzy takto postoupil systém Portál za regresní testování integračního charakteru a systémy Siebel a VCC za testy uživatelské. V rámci těchto systémů byly shromážděny relevantní soubory testovacích scénářů, které byly rozebrány v další části analýzy. Samotná analýza testovacích scénářů byla rozdělena na dvě části. Cílem první části bylo eliminovat takové testovací scénáře, které není technicky možné zautomatizovat. Metodika tohoto výběru byla postavena na třech kritériích – nutnost audiovizuální kontroly, míra integrace systému a konzistentnost testovacích dat. Tímto výběrem byly eliminovány dva soubory testovacích scénářů. Oba dva byly součástí souboru testovacích scénářů systému Portál. Konkrétně se jednalo o scénáře týkající se regresního testování Map pokrytí a Platba faktur. Scénáře pro testování Mapy pokrytí není technicky možné zautomatizovat vzhledem k povaze testu, který je zaměřen na silnou kontrolu audiovizuálního charakteru, vysoké míry integrace na systémy, ze kterých mapa pokrytí čerpá pro vlastní validní zobrazení, a zároveň z důvodu neskutečně velké variability vstupních a testovacích dat. Scénáře souboru Platba faktur

byly z další analýzy vyloučeny z důvodu velkého problému s vysokou mírou integrace a konzistentnosti zákazníka. Namodelovat testovací data vyžaduje velmi individuální přístup a znalosti testera.

V poslední části analýzy bylo cílem vybrat takové testovací scénáře, které jsou vhodné k automatizaci nejen po technické stránce, ale zároveň je jejich vytvoření efektivní a má přínos pro společnost. Kromě výše zmíněných testovacích scénářů na Mapy pokrytí a Platby faktury byly všechny testovací scénáře shledány jako více či méně efektivní pro automatizaci. Kritéria byla nastavena na základě tří indikátorů, a to počet opakování testovacího scénáře v průběhu roku, stabilita testovacího scénáře a manuální náročnost testování. Samotné vyčíslení přínosů a nákladů ve finančních jednotkách bude vyčísleno až v následující kapitole Business Case.

Ověření správnosti provedené analýzy je možné provést ve dvou rovinách. Jednou z možností je porovnat zjištěné údaje (samozřejmě obecného charakteru) s primárními zdroji literatury. Druhou možností je pak provedení post implementační analýzy, která bude porovnávat skutečné údaje z reálné implementace vůči výsledkům analýzy. Druhá možnost je o mnoho přesnější, ale vyžaduje data, která v současnou chvíli nejsou k dispozici a bude možné s nimi pracovat až po realizaci projektu. V rámci shrnutí analytické části práce je tak možné ověřit správnost výsledků pouze porovnáním se zjištěným teoretickým základem. Na základě výše uvedených údajů lze výsledek shrnout do následujících bodů:

- Regresní testy lze automatizovat.
- Je efektivní automatizovat pouze takové testovací scénáře, kde je testovací scénář stabilní a často se opakující. Automatizace testů se vyplatí, pokud je aplikace rozvíjena a udržována.
- Nelze automatizovat (a to z pohledu technického nebo z pohledu efektivnosti) takové testovací scénáře, kde dochází k častým manuálním kontrolám audiovizuálního typu.

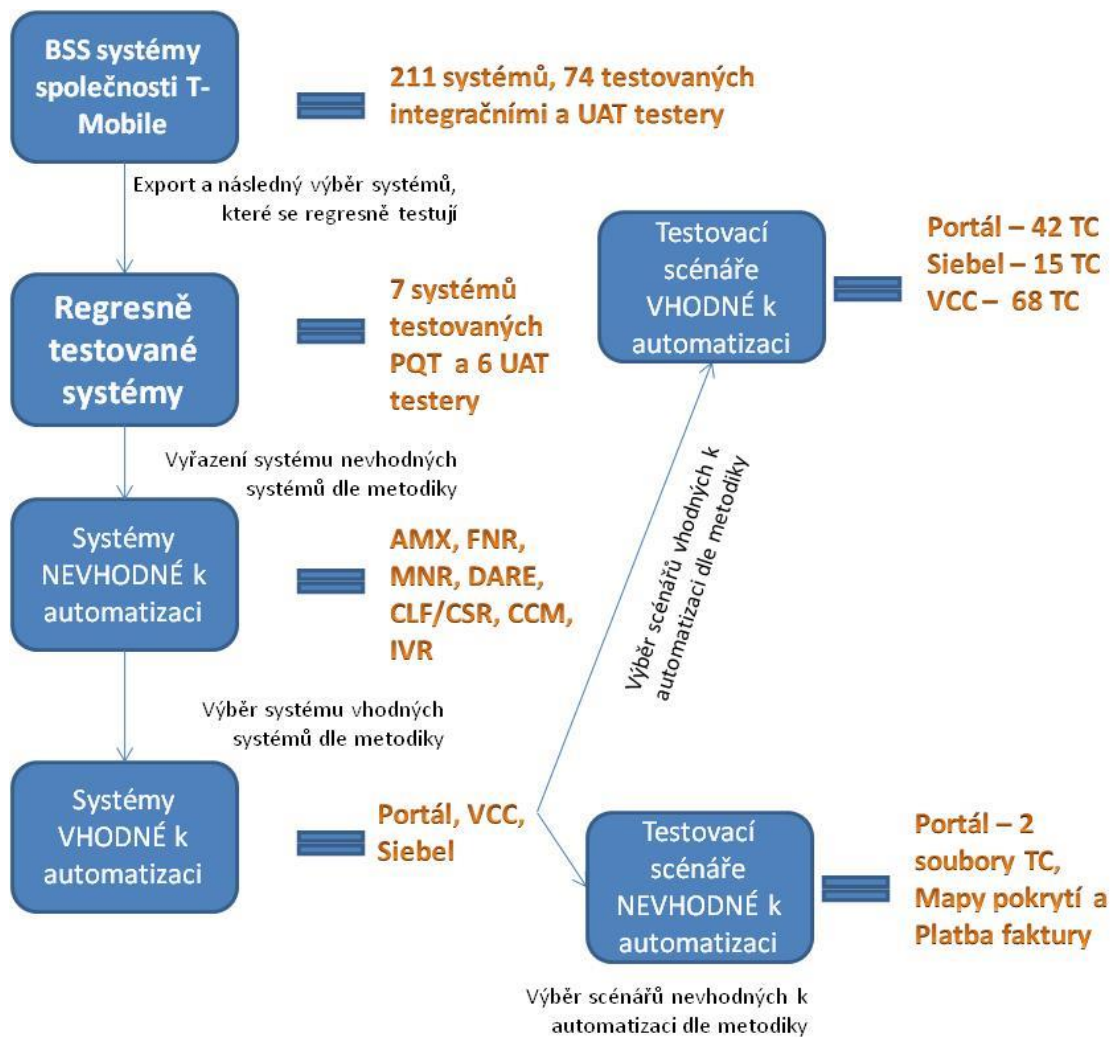
Tyto tři výše uvedené údaje lze dokázat pomocí komparace s primárními zdroji. Na prvním tvrzení, že regresní testy lze automatizovat a jsou to jedny z nejvhodnějších testů pro automatizaci, se shodnou například autoři Goucher a Riley (2009, str. 23), kteří tvrdí, že regresní testy jsou vzhledem ke své pomalosti, procentu vzniklých chyb, opakovatelnosti a finanční nákladovosti vhodným kandidátem pro automatizaci. Dalšími autory, kteří toto tvrzení podporují, jsou autoři Dustin, Rashka a Paul (1999, str. 45), kteří kromě výše uvedených regresních testů doporučují automatizovat i smoke testy. Graham a Fewster (2012, str. 449) dokonce uvádí, že by automatizace regresních testů měla být takzvaným pomyslným startovacím bodem, kdy se organizace rozhodne pro automatizaci testů. Stejní autoři pak tvrdí (2012, str. 451), že opravdovou schopnost (moc) automatizovaných testů lze objevit až po aplikaci automatizovaného řešení na další typy testů, jako jsou například testy zátěžové nebo smoke testy. Provedená analýza vylučovala z automatizace takové aplikace, které nejsou rozvíjeny či udržovány. S tímto eliminačním faktorem souhlasí i Regent (2013). Mosley a Posey (2012, str. 21) s tímto tvrzením souhlasí jen částečně.

Autoři souhlasí, že není vhodné automatizovat testovací scénáře u aplikací, které budou v nejbližší době terminovány, ale nabádá ke zvážení automatizace testů u aplikací, které mají být i v budoucnu fungující, i když nerozvíjené. Autor tvrdí, že takové aplikace, pokud jsou klíčové, mohou být z pohledu automatizace testů efektivní právě kvůli vysoké stabilitě testovacího scénáře. Na základě vyhodnocovacích kritérií a rozhovorů s

testery byly k automatizaci nedoporučeny testovací scénáře, u kterých jejich převážnou část tvořila audiovizuální kontrola testera. Toto tvrzení podporují i autoři Desikan, Ramesh (2006, str. 388), kteří ve své publikaci zmiňují fakt, že automatizované testování nemůže nahradit lidské smysly, jako je zrak či sluch a lidskou intuici. Automatizovaný tak test otestuje pouze to, co má naprogramováno.

Na níže uvedeném obrázku, který navazuje na úvodní obr. č. 4 Postup analýzy, jsou přehledně a stručně znázorněny výsledky jednotlivých dílčích částí analýzy.

Obrázek 5 Výsledek analýzy



Zdroj: vlastní zpracování (2015).

3.2.6. Doporučení

Závěrem této analýzy lze doporučit celkem 125 testovacích scénářů stávajících regresních testů k automatizaci. K tomu, aby mohla proběhnout samotná realizace, je třeba provést několik následujících kroků. Doporučení jsou dělena na obecná, která jsou platná pro jakékoliv zavádění automatizovaných testů, a na doporučení specifická pro společnost T-Mobile a její právní rámec.

Obecná doporučení znějí:

- Testovací scénáře, které byly určeny jako vhodné pro automatizaci, je třeba konkrétně zadokumentovat – konkrétní kroky TC,
- Je třeba vydefinovat vstupní a výstupní data a výlučnosti,
- Zvolit nástroj určený k automatizaci,
- Vytvořit automatizované testy,
- Nastavit proces správy testovacích scénářů a jejich úprav.

Doporučení pro případnou realizaci ve společnosti T-Mobile:

- Otevření projektu či jiné formy implementační aktivity,
- Výpočet business casu (návratnost, náklady, přínosy),
- Vytvoření a alokace týmu (dále pak je k aktivitě přístupováno klasicky v rámci projektové řízení společnosti T-Mobile),
- Výběr dodavatele (sepsání business a technických požadavků) – RFP.

S výše uvedenými body souvisí doporučení, která vznikla na základě informací od testerů a výše uvedené analýze. V první řadě je vzhledem k požadavkům společnosti nutné zaměřit se na automatizaci systémů, které mají strategický potenciál, a to jak vzhledem k životnímu cyklu systémům, tak i k jejich dopadům na business společnosti. Dalším doporučením je zaměřovat se spíše na testovací scénáře, které mají menší míru integrace. Testovací scénáře je pak možno i v automatizované podobě udržet v konzistentní podobě a snížit tak potřebu neustálých úprav. Konkrétně v prostředí společnosti T-Mobile je nutné zavést i důkladnou dokumentaci k jednotlivým testovacím scénářům, kde budou jasně popsány typy jednotlivých testovacích scénářů, termíny, způsoby jejich úprav a zodpovědné osoby. Doporučení pro společnost T-Mobile je v případě zavádění automatizovaných testů postupovat pomalu po jednotlivých systémech. Většina osob nemá s automatizací testů žádné nebo jen malé zkušenosti a na tak významnou změnu je třeba zaměstnance upozornit.

Vzhledem k povaze testovacích scénářů a osob, které regresní testy v současné chvíli vykonávají, by bylo vhodné, aby testéři (a to i UAT), měli možnost testy nejenom spouštět, ale i vytvářet. To ale předpokládá výběr takového nástroje pro automatizaci testů, pro který není nutná vysoká znalost programovacích jazyků.

Doporučení nad rámec této práce je následující. Při nasazování nového systému zároveň s implementací je dobré začít dokumentovat možné regresní testovací scénáře, které mohou být po zkušebním provozu a ladění zautomatizovány. Analýza testovacích scénářů by tak probíhala ve chvíli, kdy se celý tým zaměřuje na funkční stránku a jádro systému, a testovací scénáře musí být stejně vzhledem k testování vytvořeny. Výhodou tohoto přístupu je i fakt, že náklady na vytvoření automatizovaných testovacích scénářů budou součástí implementovaného projektu, a nebude tak nutné zatěžovat rozpočty jednotlivých oddělení, které jsou určeny pro operativní provoz.

3.3.Náklady na testování a business case

Celá diplomová práce je postavena na faktu, že se společnost T-Mobile (konkrétně oddělení Integration and Validation) rozhodla investovat čas a prostředky k zavedení automatizovaných regresních testů. Proto je třeba provedenou analýzu obhájit nejen pomocí kvalitativních přínosů, které jistě tato aktivita má, ale i po finanční stránce. Společnost T-Mobile pro zhodnocení investice používá metodu Business Case, která

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

spočívá ve zjištění následujících ukazatelů. Ty pomáhají managementu rozhodnout, zda do dané aktivity investovat či nikoliv.

- EBITDA vs. status Quo (v celkové, diskontované i celkové podobě),
- NPV – Výpočet čisté současné hodnoty,
- PAYBACK PERIOD – Návratnost investice,
- Celkové náklady na projekt,
- CASH FLOW (volné celkové cashflow, diskontované cashflow a kumulativní diskontované cashflow).

Výpočet provádí controllingové oddělení společnosti T-Mobile, které k výpočtu požaduje od Business ownera (vlastníka projektu) následujících sadu informací:

- přínosy projektu;
- definici cílů projektu včetně jeho měření a vyhodnocení v postimplementační fázi projektu;
- náklady a přínosy projektu:
 - o CAPEX (nákup licencí, kapitalizovatelné náklady v MDs);
 - o OPEX (nekapitalizovatelné náklady);
 - o náklady na údržbu a nákup HW;
 - o přínosy;
- délku implementace projektu – spuštění projektu.

Cílem této analýzy je dodat potřebné vstupy pro výpočet business casu a získat tak výstupy od controllingu.

3.3.1. Přínosy projektu

Cílem této části projektu je jednoznačně a stručně vydefinovat jeho přínosy. Všechny níže uvedené přínosy samozřejmě vychází z výhod samotné automatizace, která se aplikuje na prostředí společnosti T-Mobile, tak, aby bylo možné přínosy v následujících krocích vyčíslit. Zvolené přínosy jsou popsány níže. Zároveň proběhlo jejich schválení managementem společnosti. Hlavním přínosem projektu je zvýšení kvality testovacího a produkčního prostředí. Automatizace testovacích případů povede k optimalizaci personálních kapacit na oddělení testování tak, aby bylo možné přebírat další znalosti a požadavky závislé na strategickém projektu NG CRM, který zavádí nový CRM systém. Přínosem a zároveň důsledkem automatizace bude zjednodušení a zrychlení regresního testování a synchronizačních procesů mezi jednotlivým testovacími a produkčními prostředími. A samozřejmě je nutné zmínit nižší náklady na regresní otestování.

3.3.2. Cíle projektu

Cíle projektů musí být nadefinované metodou SMART, což dle Procházky, Klimeše (2011, str. 187) odpovídá následující specifikaci: konkrétní, měřitelný, naplňující očekávání zákazníka, realistický a ohraničený v čase. Na základě těchto kritérií byly zvoleny tři následující cíle, které se staly vstupem pro výpočet business casu:

Tabulka 10 Cíle projektu

Cíl	0%	100%	150%	Měřené období
Snížení počtu incidentů na produkci o 10 % u systémů PORTAL, SIEBEL a VCC	Více než 170 incidentů	135 incidentů	120 incidentů	3 měsíce po nasazení

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

Optimalizace kapacit testovacího týmu – uvolnění FTE pro další aktivity týmu	0 FTE ⁶	0,5 FTE	1 FTE	6 měsíců po nasazení
Relevantní Testovací scénáře budou prováděny automatizovaně	0 testovacích scénářů	125 testovacích scénářů	150 testovacích scénářů	3 měsíce po nasazení

Zdroj: vlastní zpracování

První cíl se zabývá počtem incidentů na produkci, což je chybový stav, který nebyl nalezen během testování. Cílem je snížení takových chybových stavů. Nejde jen o pouhé počty, ale každý incident je řešen s vysokou prioritou a dle interních zdrojů (2015) znamená průměrně 1 MDs na jeho opravu a přetestování. Ve chvíli, kdy je dosaženo zmenšení počtu incidentů o 10 %, se v reálných číslech jedná o úsporu 15 MDs každé tři měsíce. Tento výpočet vychází z analýzy provedené oddělením péče o zákazníky (interní zdroje, 2015), která pro potřeby této práce poskytla celkový počet incidentů na systémy SIEBEL, PORTAL a VCC, který činil za poslední 3 měsíce 150 incidentů.

U druhého cíle je třeba zmínit potřebu společnosti T-Mobile, která poměrně zásadním způsobem mění výnosovou část projektu. Vzhledem k probíhajícím aktivitám (nasazení nového CRM systému v letech 2015, 2016, 2017) bude nutné podstatně zvýšit počet regresních testů. V případě manuálních testů by se jednalo o navýšení stávajícího počtu lidí nebo nákup externího pracovníka. Stoprocentní splnění cíle představuje ušetření pomocí automatizovaných testů přibližně 0,5 FTE, což v rámci celého roku znamená přibližně 100 efektivních MDs, což v případě sazby interního člověka znamená úsporu 440 000 Kč každý rok.

Třetí cíl se zabývá ušetřením personální kapacity na oddělení testingu. Cíl byl nastaven společně s integračními i UAT testery. Nastavení cíle a jeho vyhodnocovací škály bylo nastaveno dle skutečné pracnosti, která je v současné chvíli věnována regresním testům. Dle výsledků analýzy uvedených v kapitole 3.6 je efektivní automatizovat celkem 125 testovacích scénářů. Cíl je nastaven mírněji, vzhledem k nízkému detailu testovacích scénářů, a to následujícím způsobem. Stoprocentní dosažení cíle představuje automatizace 120 testovacích scénářů. Úspora je celkem 85 MDs ročně, které jsou dnes vynakládány na manuální regresní testy. Detaily výpočtů budou uvedeny v následující podkapitole nákladů a přínosů projektu.

Nikoliv cílem, ale důsledkem automatizace je úspora nejen na straně samotného testování, ale vzhledem k možnostem nástrojů na automatizované testování byl identifikován i dopad na snížení pracnosti v rámci správy defektů. V průměru tato pracnost dosáhla až 20 % z celkové pracnosti na jeden testovací scénář. Tento odhad byl založen na zkušenosti testerů jednotlivých aplikací (PORTAL, e-shop, VCC a Siebel).

3.3.3. Náklady a přínosy projektu

Pro výpočet business casu je třeba zjistit kompletní nákladovou a přínosovou stránku záměru. Níže je uvedena přehledová tabulka s konkrétními položkami a odpovídajícími částkami. Nákladová a přínosová část projektu byla počítána ve variantě, kde by mělo dojít k plné automatizaci všech 125 testovacích scénářů (za vybrané systémy, Portál,

⁶ FTE – Full-time equivalent – jednotka indikující práci jednoho člověka zaměstnaného na plný úvazek 40h/týden (BusinessDictionary, 2012)

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

e-shop – PQT testy, VCC a Siebel). Konkrétní položky jsou vysvětleny v části cíle projektu. Pod tabulkou 11 je uveden způsob výpočtu.

Tabulka 11 Podklady pro výpočet business case.

Popis		MDs/ks/ročně	částka v Kč/Mds	Částka celkem (v Kč)
Přínosy	Snížení počtu incidentů	60 Mds	3 000	180 000
	Úspora 1/2 FTE	100 Mds	4 400	440 000
	Zautomatizování manuálního testování	84,97 Mds	4 400	373 868
	Úspora na správě defektů	21,3, Mds	4 400	93 772
	Celkem			1 087 640
Náklady	Nákup licencí	10 ks	120 400	1 204 000
	Poplatek za podporu nástroje	241000Kč/ročně	241 000	241 000
	Náklady na analýzu interní	20 MDs	4 400	88 000
	Náklady na analýzu externí	21MDs	2 800	58 800
	Náklady vytvoření a otestování skriptů interní	60,8 MDs	4 400	267 520
	Náklady vytvoření a otestování skriptů externí	71Mds	2 800	198 800
	Celkem			2 058 120

Zdroj: vlastní zpracování (2015)

Výpočet výše uvedených položek probíhal následujícím způsobem.

3.3.3.1. Přínosy

Snížení počtu incidentů: Dle výše uvedených údajů je možné snížit počet incidentů o 10 %. V důsledku tohoto snížení poklesnou náklady na opravu a testování nevzniklých incidentů. Vzhledem k současnému počtu incidentů se jedná o pokles přibližně o 60 incidentů ročně. To vzhledem k provedené interní analýze (interní zdroje, 2015) představuje úsporu zhruba 60 MDs (1 MDs na 1 incident). Tyto zdroje jsou primárně čerpány z týmu uživatelského testování. Pro výpočet tak bude použita částka za MDs platná pro jejich oddělení, což činí 3 000 Kč za MDs. Celkově tak uspořena částka činí 180 000 Kč ročně.

Úspora 0,5 FTE: Jak už bylo výše vysvětleno, tato úspora představuje náklad, který by musel být vynaložen na nákup interního či externího testera pro nově vzniklé aktivity. Cílová úspora 440 000 Kč je vypočítána následujícím způsobem:

$$\begin{aligned}
 \text{ÚSPORA} &= \left(\text{počet prac.} \cdot \frac{\text{dny}}{\text{rok}} - \text{počet dní nároku dovolené} - \text{dny na školení} - \text{operativa zaměstnance} \right) \times \text{úspora FTE} \times \text{MD částka} \\
 440\,000 &= (250 - 25 - 5 - 20) \times 0,5 \times 4\,400
 \end{aligned}$$

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

Zautomatizování manuálního testování: Tento typ úspory je spočítán na základě uvedené manuální pracnosti jednotlivých testovacích scénářů. Do pracnosti byly započteny pouze časy potřebné pro samotnou exekuci, nikoliv správu defektů, která je uvedena v přínosech níže. Pro výpočet je použita pracnost a počet opakování testovacího scénáře během jednoho roku. Vzhledem k tomu, že jsou testy vykonávány interními lidmi, bude pro výpočet cílové částky použita interní rate = 4 400Kč/MD. Konkrétní čísla prezentuje níže uvedená tabulka.

Tabulka 12 zautomatizování manuálního testování – přínos

Systém	MDs/rok	Čistá exekuce testů	částka v Kč/MDs	Částka celkem (v Kč)
Portál	40	25,6	4 400	112 640
Siebel	18,76	15,008	4 400	66 035
VCC	52,86	44,2	4 400	195 242
Celkem				373 917

Zdroj: vlastní vypracování (2015)

Úspora na správě defektů: Tato úspora je založena na předpokládané (a požadované) funkcionalitě nástroje pro automatizaci testů. Značnou část testování tvoří popis chybových hlášení a logů (v tabulce níže se jedná o sloupec Správa defektů). Dle informací od testerů tvoří práce s logy a evidence defektů přibližně 20 % práce na každém defektu. Tuto činnost může nástroj velmi ulehčit například vytvořením automatického reportu chyby – jejím popisem, ve kterém kroku vznikla, případně vytvořením printscreenů nevalidních obrazovek. Částka, kterou je možné ušetřit, je vypočítána následujícím způsobem:

Tabulka 13 Správa defektů - přínos

Systém	MDs/rok	Správa defektů	Částka v Kč/MDs	Částka celkem (v Kč)
Portal	40	0,2	4 400	35 200
Siebel	18,76	0,2	4 400	16 509
VCC	52,86	0,2	4 400	46 517
Celkem				98 226

Zdroj: vlastní vypracování (2015)

3.3.3.2. Náklady

Nákup licencí: V rámci výpočtu business casu bylo nutné zjistit licenční politiku a ceny nástrojů určených pro automatizaci. V rámci neoficiální poptávky poskytla (taktéž neoficiální) nabídku firma HP a firma GlobTech. Obě firmy mají bohatou zkušenost s automatizací testů. Rozdíl mezi jednotlivými SW byl způsob manipulace s nástrojem. GlobTech byl uživatelsky více komfortní a nevyžadoval téměř žádnou znalost programovacího jazyka. Výhodou nástroje od HP je jeho nativní podpora systému HPQC, který společnost T-Mobile používá pro správu defektů.

Cílem bylo zjistit následující ceny:

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

- Cena za licenci
 - o pro člověka, který testy vytváří a udržuje
 - o pro člověka, který testy spouští
- Cena za maintenance (údržbu a podporu /ročně)
- Konzultace a školení v rámci projektu

Výsledné neoficiální nabídky si byly cenově relativně podobné. Pro výpočet business casu bylo nakonec použito nabídky od HP, která byla dražší, a byla tak vytvořena případná rezerva při výběru jiného, levnějšího dodavatele. Kompletní výběrové řízení je součástí feasibility study (studie proveditelnosti) fáze projektu.

Tabulka 14 Nákup licencí

Položka		HP	GlobTech
Licence editor (v Kč)	10 ks	1 120 000	838 500
Licence „launcher“ (v Kč)	40 ks	0 (neomezený počet)	0(40 uživatelů)
Školení (v Kč)	2 ks	0	48 360
Technická podpora a údržba (v Kč)	ročně	241 000	193 500
Celkem		1 361 000	1 080 360

Zdroj: Interní materiály – nabídky od dodavatelů, vlastní zpracování (2015)

Náklady na analýzu interní a externí: Vzhledem k rozsahu projektu, je samozřejmě nutné důkladně zmapovat testovací scénáře k automatizaci a vybrat dodavatele nástroje. V rámci této aktivity byly naplánovány zdroje na analýzu. Tým by měl být složen z testerů, kteří regresně systém testují, systémových analytiků, kteří by měli být k dispozici pro případné technické dotazy, solution designer, který povede výběrové řízení na dodavatele z technického pohledu, a projektového manažera, který celou aktivitu bude řídit. Na všechny tyto lidi bylo naplánováno přibližně 41 MDs (20 MDs interních se sazbou 4 400 Kč a 20 MDs externích se sazbou 2 800 Kč). Celkově tak částka činí 88 000 Kč za interní zdroje a 58 800 Kč za zdroje externí.

Náklady na vytvoření automatizovaných testů: Tento údaj samozřejmě vychází z odhadu možné pracnosti a je založen na ukázkách firmy GlobTech a HP, které zkušebně automatizovaly vybrané testovací scénáře. Na základě těchto odhadů byla spočítána celková náročnost na automatizaci testů, která činila 132 MDs (466 320 Kč).

3.3.4. Výpočet business Case - shrnutí

Cílovými ukazateli pro posouzení projektového záměru jsou ve společnosti T-Mobile indikátory: čistá současná hodnota (NPV) a doba návratnosti. Níže uvedená tabulka je součástí oficiálního dokumentu, kterým se ve společnosti projektu hlásí k posouzení záměru. Business Case je počítán na 4 roky.

Do implementačních nákladů vstupují všechny nákladové položky, které jsou shrnuté v tabulce číslo 11. kromě nákladů na podporu nástroje (které spadají do operačních nákladů společnosti a přínosů, které jsou v našem případě také součástí operačních nákladů (s opačným znaménkem než již zmíněná podpora nástroje). Jednotlivé položky jsou vysvětleny níže:

CAPEX výpočet

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

Do výpočtu CAPEX výpočtu vstupují položky jako: náklady na vývoj – interní počty MDs, jednorázové položky typu nákup licencí a externí počty MDs nutné pro realizaci projektu. Vzhledem k tomu, že projekt bude dokončen do konce roku 2015, objevují se nákladové položky výše uvedeného charakteru pouze v tomto roce.

Tabulka 15 Výpočet CAPEX

		2015
CAPEX	Částka	
Interní MDs – Celkový počet (rate 4400)	80,8	-355 520
<i>Interní kapitalizovaná práce</i>		-355 520
Externí náklad – nákup licencí	10 licencí	-1 204 000
External MDs – celkový počet (rate 2800)	90,2	-256 800
<i>Ostatní CAPEX</i>		-1 460 800
Celkem CAPEX		-1 816 320

Zdroj: Interní materiály (výpočet business case, 2015), vlastní úprava

OPEX výpočet

Do výpočtu operativních nákladů v našem případě vstupují položky jak z přínosové, tak i nákladové stránky. Nákladovou stránku zde představuje poplatek za pravidelnou údržbu nakoupeného nástroje. Přínosová stránka je zde postavena na splnění cílů definovaných a vyčíslených v kapitole 3.3.3.1. Operativní náklady a přínosy jsou počítány na celkem 4 roky. Všechny OPEX náklady a přínosy vzniknou až od roku 2016. Výpočet celkového OPEXU pak probíhá sečtením nákladů (minusová položka) a přínosů.

Tabulka 16 Výpočet OPEX

	2015	2016	2017	2018	SUM
OPEX	Částka				
Poplatek za údržbu nástroje		-241 000	-241 000	-241 000	-723 000
Snížení počtu incidentů (60 MDs x 3000)		180 000	180 000	180 000	540 000
Úspora ½ FTE (100 MDs x 4400)		440 000	440 000	440 000	1 320 000
Zautomatizování manuálního testování (85 MDs x 4400)		378 000	378 000	378 000	1 134 000
Úspora na správě defektů (21,3 MDs x 4400)		93 700	93 700	93 700	281 100
Celkem OPEX	0	850 700	850 700	850 700	2 552 100

Zdroj: Interní materiály (výpočet business case, 2015), vlastní úprava

Výsledný výpočet

Pro celkový výpočet všech požadovaných indikátorů je třeba CAPEX a OPEX položky sumarizovat do jedné tabulky, konkrétně do tabulky č. 17. Na základě výše uvedených položek byly vypočítány následující indikátory: EBITDA, Cashflow (diskontované i kumulované), Čistá současná hodnota (NPV) a návratnost investice. V rámci excelového souboru určeného pro výpočet business casu jsou jednotlivé položky

počítány automaticky. Pro úplnost je však níže uvedeno vysvětlení jednotlivých indikátorů.

EBITDA

Dle ebitda.cz (2015) je EBITDA jedním z indikátorů, který ukazuje provozní výkonnost společnosti. V praxi to znamená zisk před odčtením úroků, daní, odpisů a amortizace. Na výše uvedeném projektu EBITDA odpovídá hodnotě přínosové části OPEX položek. EBITDA činí v prvním roce 2015 0 Kč, a to vzhledem ke skutečnosti, že právě v roce 2015 vznikají pouze náklady, nikoliv ziskové položky. V následujících letech je pak EBITDA rovna 851 000 Kč.

CASHFLOW

Cashflow, neboli peněžní toky, si lze podle serveru faf.cz (2015) vysvětlit jako hotovostní toky firmy za konkrétní období (nejčastěji rok, čtvrtletí, měsíc). Volné cashflow (Total free Cash Flow) Režňáková (2012, str. 17) definuje jako: „*provozní cashflow ponížené o investice do provozně nutného kapitálu. Provozní cashflow je součtem provozního zisku a odpisů*“. Na konkrétně uvedeném příkladu jsou peněžní toky jednoznačné. V prvním roce obsahují záporné cashflow v podobě investice do vývoje automatizovaných testů a nákupu licencí. V dalších letech pak cashflow dosahuje kladné hodnoty v podobě přínosů za položky, které vedou k ušetření peněz (zautomatizování testů, nižší počet incidentů, ...). Diskontované cashflow (Discounted Free Cash Flow) je definována jako hodnota podniku, která je dána současnou hodnotou cash flow po zdanění, kterou očekávají vlastníci a investoři v následujících letech (kensington.cz, 2015). Kumulované diskontované cash flow (Cummulative discounted Free Cash Flow) je pak pouhý kumulovaný výsledek toku hotovosti k posuzovanému roku - je to tedy průběžný součet všech toků hotovosti od nultého roku (tedy rok 2015) do konce daného roku (2018) (Režňáková, 2012, str. 18).

Čistá současná hodnota (NPV – Net Present Value)

Svozilová (2006, str. 92) uvádí, že metoda pro výpočet čisté současné hodnoty slouží k porovnání aktuální hodnoty peněz vzhledem k předpokládané ceně v budoucím okamžiku. V našem případě se jedná o hodnotu kumulovaného diskontovaného volného cashflow v roce 2018, kdy hodnota čisté současné hodnoty činila 416 000 Kč.

Návratnost (Payback Period)

Návratnost opět vychází z údajů o velikosti investice a diskontovaného cashflow. Svozilová (2006, str. 93) definuje návratnost jako: „*doba (počet let), za kterou peněžní příjmy z investice vyrovnají počáteční kapitálový výdaj na investici*“. V případě T-Mobile je návratnost počítána jako poměr investičních nákladů a diskontovaného volného cashflow. Při aplikaci konkrétních číselných údajů byl výsledný údaj pro návratnost 2 roky a 5 měsíců.

Z tabulky níže je vidět hodnota indikátoru NPV i doba návratnosti. Čistá současná hodnota je na uvedeném projektu 416 000Kč a doba návratnosti investice je Kč2 roky a 5 měsíců. To bylo shledáno i z pohledu controllingového oddělení jako pozitivní a

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

projekt byl podpořen ze strany finančního oddělení. Tento projekt byl následně obhájen před managementem a nyní je otevřen do fáze určené k popsání požadavků na budoucího dodavatele a interní a externí zdroje společnosti.

Velmi důležitý je i časový plán. Analýzy včetně finálního výběru dodavatele by měly proběhnout do konce června 2015. Nasazení pilotní verze systému Siebel je plánováno k 30. 9. 2015 a ostatních systémů k 31. 10. 2015. Pilotní provoz bude probíhat do konce ledna 2016, kde bude finálně vyhodnoceno, zda se projekt vyplatil a bude rozhodnuto, zda li bude aplikován v širším měřítku na větším počtu systémů a testovacích scénářů.

Tabulka 17 Business Case (v tis. Kč)

BC		2015	2016	2017	2018	SUM
Project cost		-1 816	851	851	851	736
Implementation cost	ths CZK	-1 816	0	0	0	-1 816
CAPEX	ths CZK	-1 816	0	0	0	-1 816
Interní kapitalizovaná práce	ths CZK	-356	0	0	0	-356
Operational cost	ths CZK	0	851	851	851	2 552
EBITDA	ths CZK	0	851	851	851	2 552
Total Free Cash Flow	ths CZK	-1 816	851	851	851	736
Discounted Free Cash Flow	ths CZK	-1 816	795	743	694	416
Cummulative discounted Free Cash Flow	ths CZK	-1 816	-1 021	-278	416	416

Net Present Value	416 ths CZK
Payback Period	2 years 5 months

Zdroj: Interní materiály (výpočet business case, 2015), vlastní úprava

4. Závěr

Práce byla vypracována na reálné potřebě společnosti T-Mobile, konkrétně IT oddělení, které se rozhodlo zautomatizovat stávající regresní testy tak, aby bylo dosaženo nejen časové úspory, ale hlavně, aby byla zvýšena kvalita služeb pro zákazníky. Cílem této práce bylo vybrat konkrétní IT systémy, které je možné doporučit pro automatizaci jejich regresních testů. Sekundárním cílem poté bylo zhodnocení vybraných systémů z pohledu finanční výhodnosti. Nejprve bylo nutné v teoretické části nadefinovat proces testování, jak proces probíhá, co obsahuje, jaké druhy testování se běžně provádí a jak se od sebe vzájemně liší. Zmíněny byly i zodpovědnosti a role testera. Druhá část teoretické části se zabývala samotnými automatizovanými testy, kde bylo snahou postihnout současné možnosti automatizovaného testování, výhody a nevýhody a nástroje používané pro automatizované testování.

V analytické části práce byla pro výběr vhodných systémů, provedena rozsáhlá analýza, která se skládala z velkého množství kvantifikovaných i nekvantifikovaných údajů získaných z primárních databází T-Mobile a kvalifikovaných informací a odhadů zaměstnanců T-Mobile (primárně zodpovědných za testování zkoumaných aplikací). Bylo použito velké množství vylučovacích kritérií, která měla za úkol vyřadit takové systémy a testovací scénáře, které není vhodné automatizovat. Postupným vyřazováním z původních 211 analyzovaných systémů bylo za pomoci IT architekta a testerů vyspecifikováno 7 systémů, které se regresně testují oddělením integračních testerů (PQT) a 6 systémů, které regresně testují uživatelští testéři (UAT). Na základě vydefinovaných kritérií v kapitole 3.4.2 jako je rozvoj systému, důležitost systému a náročnost regresního testování, byl vyspecifikován konečný počet systémů, kde nedává automatizace smysl převážně ze strategického pohledu firmy a účelu aplikace.

Pro další detailnější, analýzu testovacích scénářů pak byly vybrány systémy **Portál za regresní testování integračního charakteru a systémy Siebel a VCC za testy uživatelské**. V detailnější analýze testovacích případů bylo provedeno rozdělení testovacích scénářů na dvě skupiny: na scénáře nevhodné k automatizaci z technických důvodů a na scénáře vhodné k automatizaci. Testovací scénáře vybrané dle kritérií jako vhodné pro automatizaci byly dále zkoumány z pohledu efektivnosti jejich automatizace. Cílem budoucího projektu společnosti T-Mobile je automatizovat opravdu jen takové testovací scénáře, kde je automatizace přidanou hodnotou, nikoliv plošně zavést automatizované testování. Po výše uvedených eliminačních výběrech bylo vybráno celkem 125 testovacích scénářů, které jsou dle analýzy pro automatizaci vhodné.

Primární cíl práce, který spočíval ve výběru IT systémů a jejich testovacích scénářů regresního typu vhodných pro automatizaci, byl tak naplněn. Ověření správnosti provedené analýzy bylo provedeno v kapitole 3.2.5 Shrnutí a to komparací tvrzení s primárními zdroji literatury. Pro přesné ověření validity provedené analýzy je třeba ověřit konkrétní informace a data z provedené implementace vůči výše uvedené analýze. Vzhledem k faktu, že pro toto ověření není v současné (úvodní fázi projektu) dostatek dat, byla pro ověření použita pouze komparace s literaturou.

Sekundárním cílem bylo zhodnocení vybraných systémů z pohledu finanční výhodnosti. Zde byl na základě příjmových a nákladových položek vypočítán business case záměru na čtyřleté období 2015 - 2018, ve kterém byly zjištěny klíčové ukazatele určené pro

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

rozhodnutí managementu o budoucnosti projektu. Celkové implementační náklady včetně pořízení licencí se pro výše uvedené vybrané systémy pohybují okolo 1,8 mil. Kč. Příjmová část v podobě úspory lidských zdrojů a nákladů na opravy defektů a incidentů se pohybuje okolo 851 000 Kč ročně. Na základě těchto položek (ve větším detailu popsanych v části Business case) byly vypočítány ukazatele EBITDA, NPV a cashflow (diskontované a kumulované), které společnost T-Mobile používá pro vyhodnocení investičního záměru. EBITDA v jednotlivých letech dosahuje výše 851 000 Kč a je shodná s výší příjmové části projektu (tento fakt je možný vzhledem k tomu, že příjmová část je u projektů počítána v částkách bez daně, odpisů i amortizace). Hodnota čisté současné hodnoty dosahuje 416 tis. Kč a je shodná s výší kumulované diskontované cashflow v roce 2018. Návratnost investice byla vypočítána na 2 roky a 5 měsíců. Na základě výše uvedených údajů byl projekt shledán controllingovým oddělením a managementem jako smysluplným pro další realizaci. Sekundární cíl byl tímto krokem taktéž naplněn.

Vypracování této práce společně s nově nabytými poznatky z literárních pramenů, diskusí s testery a představením nástrojů od případných dodavatelů přineslo autorce mnoho informací, které budou následně využity pro realizaci záměru ve společnosti T-Mobile. Provedená analýza dopomohla k vyjasnění, které systémy a testovací scénáře jsou vhodné pro automatizaci a které pro automatizaci vhodné nejsou. Analýzu je zároveň v budoucnu možné použít pro nově vznikající aplikace a testovací scénáře. Důležitým přínosem této práce je i vyčíslení odhadovaných nákladů na realizaci včetně specifikace a kvantifikace příjmové stránky záměru.

Seznam použité literatury

AMMANN, Paul a Jeff OFFUTT. *Introduction to software testing*. New York: Cambridge University Press, 2008, xxii, 322 s. ISBN 978-0-521-88038-1.

ASH, Lydia. *The Web testing companion: the insider's guide to efficient and effective tests*. Indianapolis, Ind.: Wiley Pub., 2003, xix, 554 str. ISBN 04-714-3021-8.

BRUCKNER, Tomáš, VOŘÍŠEK, Jiří, BUCHALCEVOVÁ, Alena a kol., *Tvorba informačních systémů: principy, metodiky, architektury*. 1. vyd. Praha: Grada, 2012, 357 s. Management v informační společnosti. ISBN 978-80-247-4153-6.

BUCHALCEVOVÁ, Alena, KUČERA, Jan. *Hodnocení metodik vývoje informačních systémů z pohledu testování*. Systémová integrace, 2008, roč. 15, č. 2, s. 42–54. ISSN 1210-9479.

BURNSTEIN, I. *Practical software testing*. Vyd. 1. New York: Springer, 2003, 709 s. ISBN 03-879-5131-8.

CRISPIN Lisa, Janet Gregory GREGORY Janet, 2009 - *Agile Testing: A Practical Guide for Testers and Agile Teams*, Adobe Reader, 2009 Addison-Wesley Professional, ISBN-10: 0-321-53446-8.

DIJKSTRA, Edsger W. The humble programmer. [Online], 1972 Turing Award Lecture; Publisher as EWD:EWD340pub. Dostupné z WWW: <http://www.cs.utexas.edu/users/EWD/ewd03xx/EWD340.PDF>

DUSTIN, Elfriede, Jeff RASHKA a John PAUL. *Automated software testing: introduction, management, and performance*. Reading, Mass.: Addison-Wesley, 1999, xxi, 575 strp. ISBN 02-014-3287-0.

GALIN, Daniel. *Software quality assurance*. New York: Pearson Education Limited, 2004, xxvi, 590 p. ISBN 02-017-0945-7.

GOUCHER, Adam a Tim RILEY. *Beautiful Testing Leading Professionals Reveal How They Improve Software*. Sebastopol: O'Reilly Media, Inc, 2009. ISBN 978-144-9388-683.

GRAHAM, Dorothy, Erik VEENENDAAL a Isabel EVANS. *Foundations of software testing: ISTQB certification*. Rev. ed. Australia: Course Technology Cengage Learning, 2008. ISBN 978-184-4809-899.

GRAHAM, Dorothy, Mark FEWSTER a John PAUL. *Effective Software Test Automation Developing an Automated Software Testing Tool: case studies of software test automation*. Upper Saddle River, ANJ: Addison-Wesley, c2012, lii, 617 str.p. ISBN 03-217-5406-9.

HEUSSER, Matthew, Govind FEWSTER a John PAUL. *How to reduce the cost of software testing: case studies of software test automation*. Boca Raton, FL: CRC Press, c2012, xxvii, 312 strp. ISBN 14-398-6155-2.

JANÍČEK, Přemysl a Jiří MAREK. *Expertní inženýrství v systémovém pojetí*. 1. vyd. Praha: Grada, 2013, 592 s. Expert (Grada). ISBN 978-80-247-4127-7.

KANLING, Li a Wu MENGQI. *Effective GUI Testing Automation Developing an Automated GUI Testing Tool*. Hoboken: John Wiley, 2006. ISBN 978-078-2150-674.

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

KANLING, Li a Wu MENGQI. *Effective Software Test Automation Developing an Automated Software Testing Tool*. Hoboken: John Wiley, 2006. ISBN 978-078-2151-015.

MOREIRA, Mario E. *Adapting configuration management for Agile teams: balancing sustainability and speed*. West Sussex: Wiley, 2010, xxiv, 277 str.

MOSLEY, Daniel J, Bruce A POSEY a John PAUL. *Just enough software test automation: case studies of software test automation*. Upper Saddle River, NJ: Yourdon Press, 2002, xix, 260 strp. ISBN 01-300-8468-9.

NEWTON, Richard. *Úspěšný projektový manažer: [jak se stát mistrem projektového managementu]*. 1. vyd. Praha: Grada, 2008, 255 s. ISBN 978-80-247-2544-4.

PATTON, John. *Testování softwaru*. Vyd. 1. Praha: Computer Press, 2002, xiv, 313 s. Programování. ISBN 80-722-6636-5.

SANKAR, Kiran. *QUICK TEST PROFESSIONAL*. Tata McGraw-Hill Education, 2013. ISBN 9780071333009.

SCHAFER, Steven M. *HTML, XHTML a CSS: bible [pro tvorbu WWW stránek] : 4. vydání*. 1. vyd. Praha: Grada, 2009, 647 s. ISBN 978-80-247-2850-6.

SLAVÍK, Jakub. *Finanční průvodce nefinančního manažera: jak se rychle zorientovat v podnikových a projektových financích*. 1. vyd. Praha: Grada, 2013, 175 s. ISBN 978-80-247-4593-0.

TILLEY, Scott a Brianna FLOSS. *Hard Problems in Software Testing: Solutions Using Testing as a Service (TaaS)*. Morgan & Claypool Publishers, 2014. ISBN 9781627055246.

VOŘÍŠEK, Jiří a Jan PAVELKA. *Aplikační služby IS/ICT formou ASP: proč a jak pronajímat infromatické služby*. 1. vyd. Praha: Grada, 2003, 213 s. ISBN 9788024760551.

VYMĚTAL, Dominik. *Informační systémy v podnicích: teorie a praxe projektování*. 1. vyd. Praha: Grada, 2009, 142 s. Průvodce (Grada). ISBN 978-80-247-3046-2.

WILLIAMS, John. *Jak dělat, co vás baví, a dostat za to zaplaceno*. 1. vyd. Praha: Grada, 2012, 199 s. ISBN 978-80-247-4164-2.

Elektronické zdroje

Borland: *Silk Test* [online]. [cit. 2015-04-18]. Dostupné z: <http://www.borland.com/Products/Software-Testing/Automated-Testing/Silk-Test>

Business dictionary: *FTE* [online]. [cit. 2015-04-18]. Dostupné z: <http://www.businessdictionary.com/definition/full-time-equivalent-FTE.html>

REGENT, Jan. Computer World: *Jak automatizovat regresní testy* [online]. [cit. 2015-04-18]. Dostupné z: <http://computerworld.cz/ness-up-ideas/jak-automatizovat-regresni-testy-50428>

EBITDA: *Výpočet EBITDA* [online]. [cit. 2015-04-18]. Dostupné z: <http://www.ebitda.czhttps://books.google.cz/books?isbn=8024781611>

FAF: *Faf* [online]. [cit. 2015-04-18]. Dostupné z: <http://www.faf.cz>

EI Prague: *Regresní testy*. EI Prague [online]. 2014 [cit. 2015-04-18]. Dostupné z: http://www.eiprague.cz/index.php?option=com_content&view=article&id=84&Itemid=95&lang=cz

HAILPERN, Brent T.; Santhanam, Padmanabhan: *Software debugging, testing and verification*. IBM Systems Journal, 2002, roč. 41, č. 1. Dostupné z WWW: "<http://www.research.ibm.com/journal/sj/411/hailpern.pdf>"

HAVLÍČKOVÁ, Alena. *Typy testů* [online]. Dostupné z: <http://www.bugtracker.cz/>

HOFFMAN, Dustin. *Cost Benefits Analysis of Test Automation* [online]. Dostupné z: <http://www.softwarequalitymethods.com/Slides/AutoCost%20Slides.pdf>

HÝSEK, Jiří. *Co jsou to ty testy?* [online]. Dostupné z: <http://www.jirihysek.cz/co-jsou-ty-testy>

JANDÁK, Tomáš. *Možnosti automatizovaného testování aplikací* [online]. Plzeň, 2012 [cit. 2015-04-18]. Dostupné z: https://otik.uk.zcu.cz/bitstream/handle/11025/2828/BC_Tomas_Jandak_2012.pdf?sequence=1. Bakalářská práce. Západočeská univerzita v Plzni.

Kensington: *Valuace* [online]. [cit. 2015-04-18]. Dostupné z: <http://www.kensington.cz/Valuace.pdf>

KOLÁŘ, Petr. *Operační systémy* [online]. Liberec: 2005-02-01, [cit. 2015-03-21]. S. 2. Dostupné z: <http://www.nti.tul.cz/~kolar/os/os-s.pdf>

MAŘÍK, Roman. *Automatizované testy* [online]. [cit. 2015-04-18]. Dostupné z: http://labe.felk.cvut.cz/~marikr/teaching/Y33TSW_10/12.automatizaceTestovani.pdf

Selenium HQ: *Selenium HQ* [online]. [cit. 2015-04-18]. Dostupné z: http://www.seleniumhq.org/docs/01_introducing_selenium.jsp

Smartbear: *TestComplete – overview* [online]. [cit. 2015-04-18]. Dostupné z: <http://smartbear.com/product/testcomplete/overview/>

Testování softwaru: *Základy testování* [online]. [cit. 2015-04-18]. Dostupné z: <http://testovanisoftwaru.blogspot.cz/2009/08/zaklady-testovani.html>

Testování softwaru: *Automatizované testování* [online]. [cit. 2015-04-18]. Dostupné z: <http://testovanisoftwaru.cz/automatizovane-testovani/>

T-MOBILE. *T-press* [online]. [cit. 2015-04-18]. Dostupné z: <http://t-press.cz/cs/tiskove-materialy/tiskove-zpravy-T-Mobile/q3-2014-T-Mobile-potvrzuje-svou-vedouci-pozici.html>

Interní zdroje společnosti T-Mobile

Architektura: *Architektura CAR* [online]. [cit. 2015-04-18]. Dostupné z http://sharepoint.cz.tmo/technology/SaS/Architecture/CAR/SitePages/CAR_home.aspx

T-MOBILE. *Automatizace testů*. Praha, 2015.

T-MOBILE. *HR proces: Výběr zaměstnanců*. Praha, 2014.

T-MOBILE. *Release management proces*. Praha, 2014.

T-MOBILE. *Testing management proces*. Praha, 2014.

T-MOBILE. *Stav testovacího prostředí*. Praha, 2015