

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Katedra informačních technologií

Studijní program: Aplikovaná informatika

Obor: Informační systémy a technologie

Nejčastější problémy s testovacími daty a možnosti jejich řešení

DIPLOMOVÁ PRÁCE

Student : Bc. Kamila Langrová

Vedoucí : doc. Ing. Alena Buchalcevová, Ph.D.

Oponent : Ing. et Ing. Michal Doležel

2015

Prohlášení

Prohlašuji, že jsem diplomovou práci zpracovala samostatně a že jsem uvedla všechny použité prameny a literaturu, ze které jsem čerpala.

V Praze dne 10. 12. 2015

.....

Bc. Kamila Langrová

Poděkování

Na tomto místě bych ráda poděkovala vedoucí mé diplomové práce doc. Ing. Aleně Buchalcekové, Ph.D., za její podnětné připomínky, cenné rady a čas, který mi věnovala.

Abstrakt

V této diplomové práci jsem se zabývala testováním, testovacími daty, nejčastějšími problémy s testovacími daty a možnostmi jejich řešení.

Teoretická část práce vysvětluje pojmy testování, testovací data, test data management a zaměřuje se na kategorizaci testování podle typů, úrovně a způsobu testování a kategorizaci testovacích dat. Také představuje rozdíly mezi manuálním a automatizovaným testováním.

V praktické části práce je popsán dotazník a realizace průzkumu nejčastějších problémů s testovacími daty a také možností řešení těchto problémů. Představení dotazníku i průzkumu je popsáno včetně cílů průzkumu a shrnutí vyhodnocení dotazníkového šetření.

Přínosem této diplomové práce je ucelený pohled na testování, testovací data a jejich význam v celé oblasti testování. Jsou popsány překážky, se kterými se pracovníci v oblasti testování setkávají. Tato práce přináší shrnutí možností řešení identifikovaných problémů s testovacími daty a různé způsoby, jak těmto problémům předcházet, případně je řešit.

Klíčová slova

Testování, testovací data, test data management, problémy s testovacími daty, řešení problémů s testovacími daty, průzkum.

Abstract

This thesis is focused on testing, test data, most frequent test data issues and solutions of these issues.

The theoretical part of thesis explains testing, test data, test data management. This thesis focuses on categorizing of testing by type, class and a way of testing and categorizing test data. There are also introduced differences between manual and automating testing.

The practical part of thesis introduces the survey questionnaire and realization of most frequent test data issues and solutions of these issues survey. In this part of thesis is survey description, goals formulation and evaluation of survey included.

The contribution of this thesis is integrated view to testing, test data and their importance at whole testing domain and obstacles, which testing workers have to deal with. This thesis also contributes to a summary of test data issues solutions and ways to prevent or handle these test data issues.

Keywords

Testing, test data, test data management, test data issues, test data issues solutions, survey.

Obsah

1	Úvod.....	1
1.1	Vymezení tématu práce a důvod výběru tématu	1
1.2	Cíle práce a očekávané přínosy	2
1.3	Předpoklady a omezení práce	3
1.4	Struktura práce.....	3
2	Charakteristika současného stavu problémové oblasti.....	5
2.1	Rešerše prací	8
3	Testování a testovací data	12
3.1	Co je testování.....	12
3.2	Kategorie testování	15
3.2.1	Úrovně testování	15
3.2.2	Způsoby testování.....	20
3.2.3	Typy testů	21
3.2.4	Testování dle obsáhlosti	22
3.2.5	Manuální vs. automatizované testy	23
3.3	Co jsou testovací data	26
3.4	Typy testovacích dat.....	31
3.5	Management testovacích dat (Test data management).....	34
4	Průzkum nejčastějších problémů s testovacími daty a možností jejich řešení	37
4.1	Cíle průzkumu.....	37

4.2	Metoda sběru dat.....	38
4.3	Dotazník a výběr otázek	41
5	Vyhodnocení průzkumu.....	43
5.1	Analýza odpovědí	43
5.2	Charakteristika respondentů.....	43
5.3	Problémy s testovacími daty	46
5.4	Možnosti řešení identifikovaných problémů	51
5.5	Závěry vyplývající z průzkumu.....	58
6	Závěr	61
7	Seznam literatury.....	64
8	Seznam obrázků.....	67
9	Seznam tabulek	69
10	Přílohy.....	70

1 ÚVOD

Tato diplomová práce se zaměřuje na problematiku testování softwaru a testovacích dat, které jsou nezbytnou součástí každého provedeného testu. Testování softwaru není sice novou, ale za to rychle se rozvíjející, nezbytnou součástí vývoje softwaru. Vzhledem k rychlým změnám v oblasti technologií se neustále vyvíjí nový software, který je vždy potřeba otestovat. Od počátku rozvoje informačních technologií, tedy zhruba od padesátých let dvacátého století po dnešní dobu se celé odvětví informačních technologií velice rozrostlo, existuje nepřehledné množství programovacích jazyků, různých platforem a hardwaru. Tento rozvoj od velkého počítače, který zabíral celý sklad, po dnešní malé chytré telefony přináší komplikovanější hardware i software. Zvyšující se komplikovanost nově vyvíjeného software přináší i vyšší složitost i obsáhlost testů daného software. A s rostoucí obsáhlostí testů souvisí i množství a složitost testovacích dat, která jsou pro testy nezbytná. V dnešní době vznikají aplikace pro chytré telefony, které jsou tak složité, že jejich testování zabere i měsíce a je na ně potřeba stovky různých testovacích dat. To s sebou přináší i překážky a problémy, se kterými se testéři musí dnes a denně setkávat.

1.1 VYMEZENÍ TÉMATU PRÁCE A DŮVOD VÝBĚRU TÉMATU

Téma diplomové práce se zaměřuje na testovací data, která jsou k testům nezbytná, jejich definici a kategorizaci a na nejčastější problémy s testovacími daty, se kterými se můžeme v praxi setkat. V souvislosti s problémy s testovacími daty jsou rozebrány možnosti řešení těchto problémů. Dále tato diplomová práce také definuje a kategorizuje testování, různé typy testů včetně dnes již poměrně rozšířených automatizovaných testů.

Téma jsem si vybrala na základě své pětileté praxe v oblasti testování. Vzhledem ke svým zkušenostem s testováním jsem si uvědomovala každodenní překážky, na které jsem při výkonu práce narážela a rozhodla jsem se zjistit, jak časté tyto problémy jsou a zda se již podařilo v jiných týmech či firmách tyto problémy vyřešit nebo jim předcházet.

Dalším důvodem volby tohoto tématu bylo, že jsem k dané problematice nenašla ucelený průzkum, proto jsem se rozhodla touto problematikou zabývat blíže.

Ráda bych se dostala k ucelenému obrazu toho, jaké problémy jsou nejčastější a zda je nějaké vyzkoušené řešení, které pomáhá tyto problémy řešit nebo jim předcházet tak, aby se tyto problémy eliminovaly. Výsledkem by mělo být zefektivnění i zrychlení tvorby i používání testovacích dat nebo možná nové možnosti získávání testovacích dat, které je velmi složité získat.

1.2 CÍLE PRÁCE A OČEKÁVANÉ PŘÍNOSY

Jako hlavní cíl diplomové práce jsem si stanovila definovat, identifikovat a kategorizovat nejčastější problémy s testovacími daty, se kterými se testeři, test analytici a ostatní pracovníci v oblasti testování nejčastěji setkávají a zjistit možnosti jejich řešení, které se uplatňují v praxi v České republice. Tento cíl práce naplním pomocí rešerší a dotazníkového průzkumu.

K naplnění hlavního cíle práce musím splnit několik dílčích cílů:

- definovat a kategorizovat testování jako samostatnou oblast
- definovat a kategorizovat samotná testovací data
- identifikovat jednotlivé oblasti problémů s testovacími daty
- realizovat a vyhodnotit průzkum ohledně problémů s testovacími daty
- vyhodnotit, se kterými problémy se pracovníci setkávají nejčastěji

Výše zmíněný průzkum se také zaměřuje na to, zda některé již identifikované problémy s testovacími daty se daří efektivně řešit, nebo jim dokonce předcházet. Je možné, že z dotazníků vyplyne i více možných způsobů prevence nebo řešení určitého problému.

Nakonec v dotazníku zjišťuji, zda zmíněná řešení problémů nepřinášejí jiné návazné problémy, či překážky.

Ráda bych, aby práce přinesla ucelený pohled na testovací data, jejich význam v celé oblasti testování, a na překážky, se kterými se pracovníci v oblasti testování setkávají. Dále bych chtěla touto prací přinést shrnutí možností řešení identifikovaných problémů s testovacími daty, případně způsoby, jak těmto problémům předcházet. Vzhledem k tomu, že jsem při řádné rešerši neobjevila žádnou práci zabývající se testovacími daty tímto způsobem,

pak tuto práci považuji za velmi přínosnou, protože zmapuje nový úhel pohledu na testovací data.

1.3 PŘEDPOKLADY A OMEZENÍ PRÁCE

Pro úspěšné napsání této diplomové práce, včetně splnění hlavního i všech dílčích cílů, je předpokladem nejen jasné definování pojmů, ale zároveň důkladná rešerše zdrojů a prací k vybranému tématu. Tato práce bude po dokončení zdrojem informací o problémech s testovacími daty a možnostech jejich řešení nejen pro studenty či pracovníky zabývající se o testování softwaru, ale také pro pracovníky z daného oboru, kteří, stejně jako já, v praxi naráží na různé překážky při tvorbě nebo používání testovacích dat.

Největším omezením této práce je dotazník, respektive závislost na respondentech. Naštěstí pouze dva respondenti nebyli ochotni vyplnit otevřené otázky v dotazníku, tedy celých 93% respondentů odpovědělo v dotazníku na všechny otázky.

1.4 STRUKTURA PRÁCE

Tato diplomová práce se skládá z několika částí. V úvodní části této diplomové práce je vymezeno téma práce. Přibližuji zde důvod výběru tématu, dále si definuji hlavní a dílčí cíle práce a k tomu i související omezení této práce. Za úvodní částí následuje v kapitole 2 část rešeršní, která kromě rešerše zdrojů se stejným nebo podobným tématem jako je téma této diplomové práce obsahuje charakteristiku problémové oblasti, v tomto případě stav testování v současné době v České republice a ve světě.

Celá kapitola 3 je teoretickou částí práce. Teoretická část práce definuje pojmy, které v této diplomové práci používám. Na začátku kapitoly 3 definuji testování z různých zdrojů, jelikož pro pojem testování neexistuje jednotná definice. Přibližně polovinu této kapitoly věnuji kategorizaci testování z různých úhlů pohledu. V druhé polovině teoretické části se věnuji právě testovacím datům, a to jejich definici a kategorizaci, opět dle různých kritérií.

Další částí práce je část praktická, které se věnuji v kapitolách 4 a 5. V kapitole 4 popisuji základ k dotazníkovému šetření, které jsem musela provést pro splnění cílů této práce. Vysvětluji zde cíle průzkumu, metodiku sběru dat a s tím pak související vznik dotazníku a výběr

otázek pro dotazník. V kapitole 5 pak vyhodnocuji výsledky průzkumu a to jak slovně, tak graficky.

V poslední části práce jsou definovány závěry vyplývající z provedeného průzkumu a vyhodnocení cílů této diplomové práce.

2 CHARAKTERISTIKA SOUČASNÉHO STAVU PROBLÉMOVÉ OBLASTI

Při zkoumání současného stavu testování v České republice jsem čerpala z nedávného průzkumu Ing. Borovcové, který prováděla v letech 2010-2012 v rámci doktorského studia na Vysoké škole ekonomické v Praze (Borovcová, 2011). Tento průzkum byl zaměřený na úroveň testování a řízení kvality v softwarových firmách v České republice. Výsledky tohoto průzkumu ukazují, jaký je stav testování v České republice oproti zahraničí.

„Jako základ pro hodnocení výsledků průzkumu stavu testování v České republice byl použit obdobný průzkum v zahraničí. Počet a charakteristika účastníků tohoto průzkumu odpovídala počtu a charakteristice účastníků českého průzkumu, je však nutné zmínit dva rozdíly, které mohly v menší míře ovlivnit výsledky průzkumů:

1. Zatímco průzkum v České republice probíhal na konci roku 2010, průzkum v zahraničí proběhl o rok a čtvrt později, v lednu 2012. Tento rozdíl by však měl být zanedbatelný, jelikož se stav testování v České republice dle zkušenosti autorky za tuto dobu nijak nezměnil.
2. Zatímco v České republice byla oslovena velká část firem s žádostí o vyplnění dotazníku, zahraniční průzkum žádal o vyplnění dotazníku pouze na zahraničních fórech a diskuzních skupinách (náhodní čeští respondenti nebyli do výsledků zahrnuti). Dá se tedy předpokládat, že větší část zahraničních respondentů má větší zájem o získávání znalostí v oblasti testování, než by tak mohlo být u průměrného českého vzorku.

Tyto rozdíly a jejich dopady byly vzaty při vyhodnocení a analýze do úvahy. Z výsledků průzkumů bylo zjištěno, že testování softwaru je v České republice *věnována menší pozornost než v zahraničí*, což se odráží zejména ve vzdělávání pracovníků a v používání moderních technik a nástrojů.“ (Roudenský a Havlíčková, 2013, s. 50)

„Průzkum ukázal, že zaměstnanci přicházející do styku s testováním, si uvědomují jeho význam a komplexnost, nicméně často neumí správně definovat, o čem řízení kvality je. Mají problém určit slabé a silné stránky testovacích technik, vhodnost použití nástrojů.

Porovnáme-li odpovědi menších a větších firem, nelze říci, že se jedni nebo druzí více shodují s názory zahraničních odborníků, nebo že by měli lépe zvládnutý proces testování. Větší a menší firmy využívají jiný přístup k testování, takový, jaký jim více vyhovuje.

Domnívám se, že problémem nejen v České republice je, že lidé neumí přizpůsobit metodiku a procesy testování a zajišťování kvality konkrétní situaci. Kvalita se v České republice je dosahována jen za cenu poměrně velkých nákladů, a to z toho důvodu, že jsou na testování najímáni lidé s nedostatečnými technickými a analytickými schopnostmi a nejsou jim dány potřebné pravomoci a respekt. Výsledkem je, že testování sice zastává významnou pozici v procesu vývoje, ale řada firem se nesnaží zajistit kvalitu produktu, pouze dokázat, že testování proběhlo a zabránit velkým problémům při předání a nasazení softwaru.

Tato situace má dle mého názoru několik příčin, z nichž za nejdůležitější považuji:

- nedostatek dostupné literatury, výsledkem je, že téměř všichni publikující v České republice v této oblasti vycházejí ze stejné literatury, a tedy nedochází ke konfrontaci názorů a k hlubšímu zamyšlení nad předkládanými fakty,
- nedostatek zkušených expertů na testování a řízení kvality, kteří by u nás vychovávali mladší testery a prozkoumávali oblast testování softwaru,
- ostatní aktéři účastníci se procesu testování nemají dostatek příležitostí se setkat se schopnými testery, test analytiky a test manažery, což negativně ovlivňuje jejich poznání této oblasti,
- neochotu studovat cizojazyčné materiály ze zahraničních zdrojů či se zcela vzdělávat v oboru mimo rozsah pracovních povinností.

Bohužel na tyto příčiny neexistuje žádné elegantní rychlé řešení, ale pouze dlouhodobá snaha o zlepšování situace všemi zúčastněnými.“ (Borovcová, 2011, s. 106-107)

„Pro zlepšení tohoto stavu byly navrženy následující kroky:

1. Rozšíření nabídky vysokoškolských a odborných kurzů zejména s důrazem na preventivní metody řízení kvality a na seznámení manažerů s ekonomickými aspekty investování do kvality vývoje softwaru.

2. Rozšíření nabídky kurzů zacílených na použití teoretických znalostí o řízení kvality v praxi.
3. Rozšířit nabídku školení ve firmách pro zaměstnance aktivně se podílejících na utváření kvality softwaru, případně motivovat zaměstnance k řízenému samostudiu s doporučeným seznamem kvalitních zdrojů.“ (Roudenský a Havlíčková, 2013, s. 51)

Hlavním zjištěním celého průzkumu je fakt, že v České republice je testování softwaru věnována mnohem menší pozornost než v zahraničí a tento fakt lze pozorovat, jak v používání technik a nástrojů, tak i ve vzdělávání pracovníků.

Dle Roudenského a Havlíčkové (2013) jsou obecně využívány následující zdroje a způsoby získávání znalostí a zkušeností:

- Vysokoškolské studium
- Zaměstnání
- Samostudium
- Certifikace a rekvalifikace
- Školení a kurzy
- Konference
- Internet (blogy, odborné články, diskuzní fóra)
- Knihy
- Praxe

Nejčastějším zdrojem informací ohledně testování je v České republice „využíván internet, jako primární zdroj svých znalostí ho uvedlo 47 % respondentů. Na druhém místě se umístili kolegové, od kterých se učí 20 % dotazovaných, těsně pak následují knihy a školení s 18 % a 14 % hlasů. Pouze jeden respondent (1,2 %) uvedl, že ho čerpání znalostí o této oblasti nezajímá. Kupodivu na otázku spokojenosti s dostupností a rozsahem české odborné literatury o testování projevilo nezájem 46,5 % respondentů. Tedy zajímá je čerpání znalostí, ale nemají zájem o to, aby jim tyto vědomosti byly poskytovány ve formě české odborné literatury.

Nespokojenost vyjádřilo 44 % dotazovaných a pouhých 9,5 % je s dosavadní nabídkou spokojeno.“ (Borovcová, 2011, s. 106)

„Zatímco samostudium je neřízená forma vzdělávání, jehož kvalita závisí na zvoleném způsobu získávání znalostí, absolvování speciálně zaměřených kurzů v rámci vysokoškolského studia by mělo poskytovat dostatečnou záruku kvality vzdělání. V České republice však testování a řízení kvality zatím není plně integrovanou součástí výuky informatiky na všech vysokých školách, které tak i dnes produkují budoucí IT manažery, softwarové inženýry a analytiky, kterým chybí znalosti o utváření kvality a praktické dovednosti s nástroji pro její zajištění.“ (Roudenský a Havlíčková, 2013, s. 52)

Ze svých zkušeností mohu říci, že během pěti let, kdy jsem pracovala v oblasti testování ve velkých bankovních institucích, se pohled na testování postupně měnil a byl na něj kladen čím dál větší důraz. Přesto však je třeba říci, že generace testerů, test analytiků a test manažerů, kteří by se této oblasti věnovali více a hlouběji, pravděpodobně začala z vysokých škol vycházet teprve před pár lety.

Doufám tedy, že v následujících letech se tyto rozdíly mezi Českou republikou a zahraničím, co se týče jak informovanosti, ale i vzdělávání v oblasti testování, budou postupně smazávat. Věřím, že této změně pomohou velké i menší zahraniční firmy na českém trhu, které budou své zaměstnance školit stejně jako zaměstnance v jiných zemích.

2.1 REŠERŠE PRACÍ

Rešerši zdrojů pro tuto diplomovou práci jsem se rozhodla provést souhrnně pro české i zahraniční zdroje.

Hned na úvod této kapitoly bych zmínila existenci normy ISO/IEC 29119 (ISO/IEC/IEEE 29119 Software testing, 2013), která je normou pro testování softwaru. První tři části této normy byly vydány v září 2013. První část se věnuje konceptům a definicím, druhá testovacím procesům a třetí dokumentaci testování. V přípravě jsou další dvě části této normy: Testovací techniky a Testování řízené klíčovými slovy. ISO IEC 29119 nahrazuje existující standardy IEEE

829 Testovací dokumentace, IEEE 1008 Testování jednotek, BS 7925-1 Slovník výrazů v testování softwaru a BS 7925-2 Komponenty standardů testování softwaru.

V první fázi rešerše jsem se zaměřila na zdroje, které se přímo zabývají problémy s testovacími daty a možnostmi jejich řešení. Jak jsem již zmínila v kapitole 1.1 *Vymezení tématu práce a důvod výběru tématu*, právě fakt, že jsem nenašla žádný průzkum o tom, které problémy s testovacími daty jsou ty zásadní, se kterými se pracovníci denně setkávají a které jsou na druhou stranu minoritní, byl pro mě důvodem výběru tématu této diplomové práce.

Problémům s testovacími daty se věnuje web EuroSTAR software testing conference, na jejichž webu <https://conference.eurostarsoftwaretesting.com/> jsou dva články týkající se problémů s testovacími daty. První z nich, „How To Streamline Your Test Data Management,, (Eriksson, 2014) se věnuje přímo problémům s testovacími daty a nabízí „univerzální“ řešení, jak těmto problémům předejít. Slovo univerzální mám v uvozovkách, protože je v článku jasně upozorněno, že toto řešení nemusí fungovat ve všech firmách, přesto je zmíněné řešení poměrně obecné. Toto řešení je velmi zajímavé, ale je velmi náročné na všechny zdroje, tedy jak finanční, tak i lidské a časové. Druhý článek na webu EuroSTARu „Test Data Management – Security Compliance whilst performing production Data Extraction / Data Anonymisation using Tools,, (Rahman, 2013) se věnuje konkrétně jedné problematice oblasti testovacích dat, a to je anonymizace dat při extrakci dat z produkčního prostředí. Všechny problémy, které jsou popsány v těchto dvou příspěvcích na webu EuroSTAR se objevily i v seznamu problémů s testovacími daty při mém průzkumu na 23. setkání [pro]Testlu v Praze na téma Test Data Management (více o tomto průzkumu v kapitolách 4.2 a 4.3). Což naznačuje, že tyto problémy s testovacími daty nejsou záležitostí jen České republiky, ale minimálně Evropy. Další pramen o problémech s testovacími daty je od Khurany a Bindala (2014), kde v rámci definování Test data managementu identifikují osm problémů, které jsou důvodem pro zavedení a udržování test data managementu v organizacích. Více o těchto problémech v kapitole 3.5 *Test data management*.

Další zdroj, který se věnoval problémům s testovacími daty, nikoliv však jejich řešení ani hodnocení jejich závažnosti, je dokument vydaný firmou Hewlett-Packard (2010) s názvem *How to address top problems in test data management: Data reuse, sub-setting and masking*.

Tento dokument je spíše příručkou pro správu testovacích dat v aplikaci HP Test Data Management.

V další fázi rešerše jsem se zaměřila na literaturu a jiné zdroje o samotném testování, kde jsem hledala kapitoly věnující se přímo testovacím datům. Obecně se dá říci, že píše-li se v literatuře jak v české, tak i zahraniční o testování, pak jsou sice testovací data zmíněna, ale pouze v souvislosti se samotným testováním a o problémech a překážkách testovacích dat zde žádný relevantní zdroj informací nenalezneme. Nicméně mnoho zdrojů obsahuje velmi mnoho informací alespoň o testování.

Nejdříve jsem se zaměřila na akademické práce o testování, kterých je 109 jen na Vysoké škole ekonomické v Praze. Tyto práce se věnovaly testování například v agilních metodikách, metodice RUP, automatickému testování, testování různých systémů, nástrojům na podporu testování či na automatické testování softwaru, modelům zlepšování procesů testování softwaru a zajištění kvality atp. Nicméně ani jedna z těchto prací se nevěnuje konkrétně testovacím datům. Zvláště bych tu zmínila bakalářskou práci Analýza projektových rolí v banko-pojišťovacím sektoru České republiky (Langrová, 2013), kterou jsem použila jako zdroj v kapitolách o testování, protože jsem zde věnovala celou jednu kapitolu úvodu do testování.

Jako zdroj pro teoretickou část práce jsem použila několik publikací zabývajících se testováním, převážně takové, které se věnují testování obecně, nikoli jen jedné konkrétní oblasti testování. Z českých zdrojů je to hlavně kniha Řízení kvality softwaru: průvodce testováním (Roudenský a Havlíčková, 2013). Tato kniha se věnuje velmi podrobně testování, vysvětlení jednotlivých pojmů, rozdělení do jednotlivých úrovní a dle typů a také je zde věnována kapitola současnému stavu testování v České republice. Dále jsem čerpala z knihy Testování softwaru (Patton, 2002), ve které se autor podrobně věnuje jednotlivým způsobům testování, aplikaci postupu testování, automatizovanému testování i nástrojům pro testování. Jako cizojazyčný zdroj informací jsem zvolila knihu Introduction to software testing (Ammann a Offutt, 2008). Tato kniha se podobně jako předchozí publikace věnuje jednotlivým úrovním testování i nástrojům k testování.

Dále jsem jako zdroje pro tuto práci použila weby a blogy o testování. Z českých blogů bych zmínila tři nejvýznamnější a to <http://testovanisoftwaru.cz> a

<http://testovanisoftwaru.blogspot.com/>, jež jsou od stejné autorky a ještě <http://cz-testing.blogspot.com> a dále také web skupiny [pro]Test! <http://www.pro-test.info/>, která je otevřenou skupinou pro lidi pohybující se v oblasti testování nebo se zájmem o testování. Na jejich Google+ stránce <https://plus.google.com/u/0/communities/104358028405626727302> jsou dostupné veškeré podklady z již uskutečněných setkání a prezentací. Co se týče zahraničních blogů, tak kvalitním zdrojem informací je již výše zmíněný web EuroSTAR software testing konference <https://conference.eurostarsoftwaretesting.com/>, a také blog odborníka v dané oblasti Jamese Bacha <http://www.satisfice.com/blog/> a web C. Kanera <http://kaner.com/>. Všechny výše zmíněné blogy a weby obsahují informace o testování, o nástrojích k testování, o možnostech certifikace v dané oblasti, ale také různé zápisy, reporty či hodnocení konferencí o testování, například Czech testu. Jako zdroj je také třeba uvést podklady pro certifikaci International Software Testing Qualification Board (ISTQB®, 2013), které obsahují informace o testování od základů testování, přes typy testů, techniky tvorby testů až po řízení testování a nástroje pro testování.

3 TESTOVÁNÍ A TESTOVACÍ DATA

3.1 CO JE TESTOVÁNÍ

Neexistuje jedna definice popisující a vysvětlující, co je to testování. Díky dynamičnosti vývoje této oblasti za posledních deset let se celý proces testování od prvních definic velmi změnil.

Testování samo o sobě je proces zjišťující kvalitu nejen testovaného softwaru. Obecně se dá říci, že testuji pokaždé, když daný software či produkt používám. Na základě této činnosti je člověk schopen vyhodnotit, zda je produkt funkční, případně nakolik je funkční. Také je schopen vyhodnotit, zda produkt splnil požadavky, kvůli kterým byl vyvíjen a nakolik tyto požadavky splnil. Díky všem zjištěným informacím nakonec lze vyhodnotit kvalitu produktu a případně doporučit další postup, který může být například oprava nalezených chyb, tedy defektů, nebo mohu doporučit vydání produktu na trh. Řečeno jednou větou je testování činnost prováděná za účelem získání informací o produktu, jejich třídění a následné vyhodnocení kvality produktu.

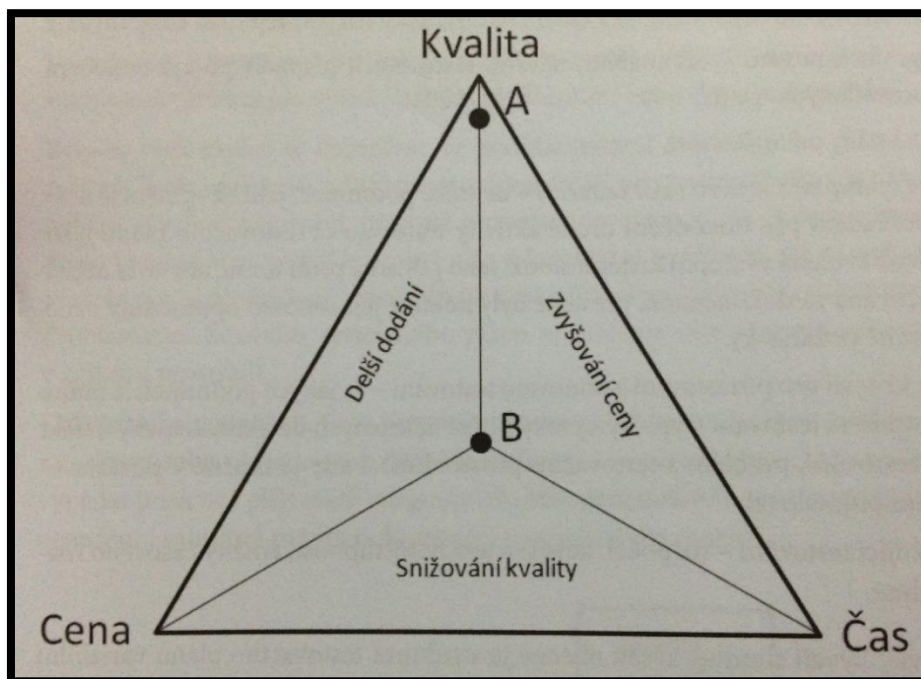
Testování jako činnost může být jednorázová či opakovaná. Záleží na úrovni testu, na zvolené testovací metodě, na zkušenostech testera a dalších podmínkách jako jsou funkčnost prostředí, množství času na test, případně i cíle testu. Hlavní princip testování však zůstává stejný.

Zprv je třeba začít produkt používat, či alespoň vizuálně hodnotit (i pouhé zhlédnutí, zda barva odpovídá požadavkům, je test). Dále je třeba tento produkt používat opakovaně, ať už na různé činnosti či stejné. Je potřeba vyzkoušet všechny možné varianty (cesty), jak dosáhnout stejného výsledku. Zjednodušeně řečeno je veškeré hodnocení při používání i vizuální hodnocení testováním, ze kterého vyhodnotíme, nakolik produkt splňuje předem stanovené požadavky.

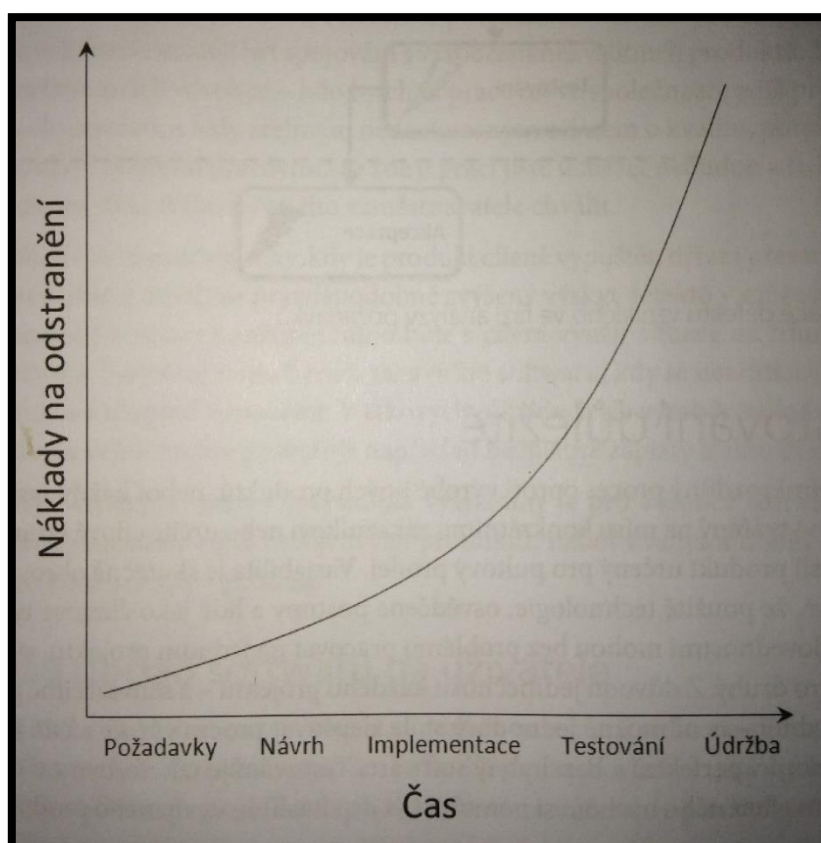
Mezi uznávané kapacity v oblasti testování patří v dnešní době James Bach a Cem Kaner. Bach definuje testování jako proces hodnocení produktu tak, že se s ním naučíme pracovat zkoumáním, objevováním, porovnáváním, pozorováním atd., doslova říká, že „Testing is the

process of evaluating a product by learning about it through exploration and experimentation, which includes to some degree: questioning, study, modeling, observation, inference, etc.” (Bach, 2013), zatímco Kanerova definice tvrdí, že testování softwaru je „An empirical, technical investigation conducted to provide stakeholders with information about the quality of the product or service under test.” (Kaner, 1999), tedy, že testování softwaru je empirický technický výzkum kvality testovaného produktu nebo služby, prováděný za účelem poskytnutí těchto informací stakeholderům. Lze se také setkat s definicí testování jako s “...Testing is the primary method that the industry uses to evaluate software under development” (Anmann, 2008, s.3), tedy, že testování je primární metodou, kterou firmy používají k ohodnocení vyvíjeného softwaru. A Anna Havlíčková s Petrem Roudenským definují testování jako „proces řízeného spouštění softwarového produktu s cílem zjistit, zda splňuje specifikované či implicitní potřeby uživatelů. (...) Testování je řízený proces, neboť musí být prováděno vždy s určitým záměrem, předchází mu plánování ve formě návrhu testů a po něm následuje jejich vyhodnocení. Z uvedené definice tak vyplývá, že testování zkoumá softwarový produkt, čímž získává informace o jeho kvalitě, kterou chápeme jako stupeň naplnění požadavků a přání uživatelů. Obsahem testování je tedy především sběr a analýza informací, přičemž nalezené defekty jsou z tohoto pohledu vedlejším produktem této činnosti.” (Roudenský a Havlíčková, 2013, s. 45)

„Testování je potencionálně nekonečný proces, a o jeho zastavení tak musí být rozhodnuto na základě uvážení různých faktorů, jako je jeho postupně se snižující efektivita, časové a rozpočtové omezení, provedení všech klíčových testů a podobně. Testování a případné ostatní aktivity řízení kvality by měly začínat, co nejdříve je to možné, tedy již ve fázi specifikace požadavků. Zatímco v této fázi vyžaduje odstranění problému minimální úsilí, ten samý problém zjištěný až po nasazení systému do prostředí zákazníka může způsobit velké ztráty. Platí tedy, že čím dříve je defekt odhalen, tím nižší jsou náklady na jeho odstranění. Testování je kompromisem mezi kvalitou a časovým a rozpočtovým omezením – upřednostňování jednoho cíle má pochopitelně negativní dopad na zbylé dva.” (Roudenský a Havlíčková, 2013, s. 47). Tento kompromis mezi cíli se nazývá Trojúhelník kvality, který znázorňuje Obrázek 1. Vztah mezi fází, ve které je defekt identifikován a cenou za jeho odstranění je vidět na Obrázku 2.



Obrázek 1: Trojúhelník kvality. Bod A představuje drahý, velmi kvalitní systém s dlouhou dobou dodání. V bodě B se sice každému atributu věnujeme stejně, avšak ve výsledku všechny zbytečně ztrácí. [Zdroj: Roudenský a Havlíčková, 2013, s. 129]



Obrázek 2: Vztah mezi fází, kdy je defekt nalezen a náklady na jeho odstranění [Zdroj: Roudenský a Havlíčková, 2013, s. 47]

3.2 KATEGORIE TESTOVÁNÍ

Kategorií testování je nepřeberné množství a záleží jen na konkrétním člověku, jaká zvolí kritéria kategorizace. V této kapitole se pokusím shrnout ty nejčastěji používané varianty dělení testů do různých kategorií. Testování se může dělit podle úrovně, způsobu, typu a obsáhlosti. Dále se také testy dělí na manuální a automatizované.

V dnešní době je velmi nestandardní, aby se používala pouze jedna testovací metoda. Často se používá kombinace těchto metod, na různé části systému a různé typy testů se používají různé typy testovacích metod (například manuální testování dle scénářů, manuální testování freetesty a k tomu ještě automatizované testy na pár funkcionalit). Kombinací je nepřeberné množství a vše záleží na konkrétním produktu, požadavcích, finančních a časových možnostech a také zkušenostech test analytika, test koordinátora a hlavně samotných testerů.

Kombinace různých metod testů se nazývají Testovací mix.

3.2.1 ÚROVNĚ TESTOVÁNÍ

Roudenský a Havlíčková (2013) dělí testování dle úrovně do pěti následujících kategorií.

- Testování jednotek
- Integrační testování
- Systémové testování
- Akceptační testování

A. Testování jednotek

Někdy bývá před testováním jednotek uváděna ještě úroveň testování programátorem, ale dle dělení podle Roudenského a Havlíčkové (2013) je testování programátorem součástí testování jednotek.

Při testování programátorem se nejedná o komplexní testování produktu. Jsou to testy na té úplně nejzákladnější úrovni těch nejmenších částí. Když například programátor programuje celou novou aplikaci, tak kdykoliv naprogramuje nějakou malou dílčí část, tak si ji hned

otestuje. Jedná se o testování, které slouží jen pro programátora, že danou část naprogramoval správně. (Langrová, 2013)

„V praxi jsou tyto testy označovány jako „Assembly tests“. Většinou si však programátor netestuje svoji část kódu, ale realizuje se tzv. „test čtyř očí“. To znamená, že kód testuje jiný programátor, než ten, který jej napsal. Program je v tomto stupni kontrolován na úrovni zdrojového kódu. V praxi je bohužel tento stupeň testování často podceňován. Přitom opravy chyby v této části testování softwaru je nejméně nákladná. Sám jsem byl svědkem toho, jak vývojář, aniž by si po sobě jakkoliv zkontroloval kód, předal program k dalšímu testování. Tester se připravil k vlastnímu testování (nastuduje si dokumentaci, připraví testovací data apod.) a začal provádět testy. Záhy však zjistil, že program nelze spustit a nemůže tak ani začít samotné testování. Nahlásil tedy chybu tvůrci programu a ten ji posléze začal opravovat. Tester se tak zbytečně připravoval a začínal provádění testů. Kdyby si po sobě programátor zkontroloval kód dříve, zjistil by chybu mnohem rychleji než tester. Samotná oprava chyby by tak zabrala výrazně méně času. Takovéto postupy zbytečně zabírají čas i ostatním členům týmu.“ (Hlava, 2011)

Testování jednotek také ještě není komplexním testem celé aplikace. Jsou to testy jednotek tzv. unit testy. Jednotka se definuje podle typu a rozsáhlosti aplikace. V případě nějaké aplikace obsahující například deset různých modulů, bude unit testem test jednoho z těchto modulů. Test jednotky by měl být nezávislý na tom, zda jsou už další jednotky naprogramovány. Jednotkové testy testují pouze funkčnost té dané části, nikoliv návazností či souvislostí mezi jednotlivými jednotkami. (Langrová, 2013)

„Testy jednotek se velmi špatně aplikují na již zaběhlých projektech. U již vytvořených aplikací se většinou musí provést kompletní refaktoring kódu či dokonce mnohem hlubší úpravy. Takováto časová investice se u menších projektů většinou nevyplatí, ale ani u velkých projektů takovýto zásah není příliš šťastný a často se neseťká s podporou u vedoucího projektu. Proto je vhodné zabývat se těmito testy již v etapě návrhu aplikace a v té době se rozhodnout, zda tyto testy budeme využívat. Jelikož obecně platí, že čím dříve (v rámci životního cyklu softwaru) chybu nalezneme a opravíme, tím méně času nad touto opravou

strávíme. Proto bych doporučoval této úrovni věnovat maximální pozornost a to již před samotným vývojem aplikace.“ (Hlava, 2011)

B. Integrační a systémové testování

Integrační testy jsou testy celé funkčnosti dané aplikace, hlavně tedy komplexní funkčnosti všech modulů a návazností mezi nimi. K integračním testům dochází ještě u dodavatele. Tyto testy by měly podchytit veškeré chyby, aby dodavatel dodal zákazníkovi produkt bez defektů. Tyto testy ověřují správnou integraci projektu. (Langrová, 2013)

„Musí být ověřena bezchybná komunikace mezi jednotlivými komponentami uvnitř aplikace. Integraci však lze ověřovat nejen mezi komponentami, ale také mezi komponentou a operačním systémem, hardwarem či rozhraním různých systémů. V této fázi se tak testuje integrace dosud jednotlivě ověřených částí. Postupně začínáme testovat integraci mezi dvěma komponentami a postupně přidáváme další. Integrační testy mohou být jak manuální, tak i automatizované. Úroveň integračního testování je svým způsobem obsažena ve většině testovacích postupů softwaru. U menších projektů je však na tyto testy kladen velmi malý důraz. Má to své logické odůvodnění. Integrační testování lze v testovacím cyklu zcela vynechat. Na výslednou bezporuchovost softwaru to přitom nebude mít žádný vliv, tedy alespoň za předpokladu, že korektně provedeme následující úroveň testování. Chyba, kterou bychom odhalili během integračních testů, se zcela jistě projeví v průběhu dalších úrovní testování. Jak již bylo zmíněno dříve, během testování však platí: „čím dříve chybu objevíme, tím méně úsilí nás stojí její oprava“. Proto integrační testy mají svůj význam, nicméně nelze jejich použití nikterak přeceňovat.“ (Hlava, 2011)

Systémové testování je vlastně forma integračních testů. Často se používá zkratka SIT (= System Integration Testing), což naznačuje, že právě systémové testování bývá spojeno s tím integračním. Systémové testy jsou v podstatě poslední úroveň testů před předáním produktu zákazníkovi. (Langrová, 2013)

„Během těchto testů je aplikace ověřována jako funkční celek. Tyto testy jsou používány v pozdějších fázích vývoje. Ověřují aplikaci z pohledu zákazníka. Podle připravených scénářů se simulují různé kroky, které v praxi mohou nastat. Obvykle probíhají v několika kolech.

Nalezené chyby jsou opraveny a v dalších kolech jsou tyto opravy opět otestovány. Součástí této úrovně jsou jak funkční tak nefunkční testy. Poslední úroveň testů, které se provádějí před předáním produktu zákazníkovi, jsou tedy systémové testy. Tato úroveň testů tak většinou slouží jako výstupní kontrola softwaru. Systémové testování je obsaženo prakticky v každém procesu testování. Bez této úrovně by celé testování softwaru nemělo žádný význam. Bezporuchovost výsledného produktu by byla významně ohrožena. Proto tuto úroveň testů považuji za stěžejní v celém postupu testování softwaru. Na realizaci těchto testů by se mělo myslet již v raném stádiu návrhu postupu testování, tak aby bylo možné obsahu testů co možná nejvíce přizpůsobit očekávanému softwaru.“ (Hlava, 2011)

Během systémového testování se testuje nejen funkčnost systému, ale také bezpečnost a stabilita systému. K tomu slouží bezpečnostní a zátěžové testy.

Bezpečnostní testy jsou částečně pokryty v testech klasických. Bývají to bezpečnostní techniky jako elektronický podpis, přihlašování pomocí přihlašovacích údajů, odhlašování atp. Existují ale další možnosti jak otestovat, zda je aplikace bezpečná. Většinou tyto další druhy testů test analytiky při psaní testovacích scénářů ani nenapadnou. Jedná se o testy, kdy tester například zkouší do textového pole v aplikaci vložit nějaký řetězec kódu (zde pak hrozí bezpečnostní riziko napadení systému pomocí SQL injection). Možností, co vše se dá v této speciální kategorii otestovat, je nekonečné množství, takže je zde kladen velký důraz na „fantazii“ testera.

Zátěžové testy jsou velmi důležité vždy, ale obzvláště jedná-li se o větší společnost, která vyvíjí aplikaci, kterou budou používat všichni její klienti (například banky). Bez zátěžových testů by se totiž mohlo stát, že daná firma pustí aplikaci mezi klienty a po přihlášení desátého klienta aplikace přestane odpovídat a zbylé tisíce klientů, kteří se chtěli do nové aplikace také přihlásit, se tam ani nedostanou. Zde je velmi těžké odhadnout, jak velkou zátěž by vlastně aplikace měla vydržet, vzhledem k tomu, že zátěž na žádnou aplikaci není soustavně rovnoměrná, ale je závislá na mnoha faktorech jako den/noc, roční období (například před Vánocemi je mnohem větší zátěž na elektronické bankovníctví než uprostřed roku) atp. Také je třeba se rozhodnout, zda chceme mít aplikaci funkční jen při standardní zátěži nebo i při nadstandardní zátěži. Tyto zátěžové testy se však dělají automatizovaně, kdy se strojově

vytvoří například 5000 klientů, kteří se naráz přihlásí do aplikace a poté se měří odezva systému a další důležité ukazatele. (Langrová, 2013)

„Zátěžové testování by nemělo být prováděno v posledních fázích vývoje produktu, přestože jde o poměrně rozšířený koncept. Identifikace problematických míst v kódu a jejich následná optimalizace by naopak měla probíhat průběžně během vývoje produktu.“ (Roudenský a Havlíčková, 2013, s. 47)

C. Akceptační testování

Akceptační testování provádí zákazník, aby si zkontroloval, že mu dodavatel dodal produkt bez chyb. Povolené množství chyb záleží na dohodě (většinou smlouvě) mezi dodavatelem a zákazníkem. Většinou se však předpokládá naprostá absence závažných chyb. V případě nalezení velkého množství chyb či velmi závažných chyb během této fáze testování je velká pravděpodobnost zpoždění nasazení projektu, což je nežádoucí situace. (Langrová, 2013)

„Tyto testy jsou často prováděny podle připravených scénářů, které společně připravil zákazník s dodavatelem. Testy probíhají na testovacím prostředí u zákazníka. Nalezené nesrovnalosti mezi aplikací a specifikací, jsou reportovány zpět vývojovému týmu. Opravené chyby jsou nasazeny na prostředí u zákazníka. V této úrovni je zřejmě nejdůležitější, definovat si předem jakou formou bude probíhat oznamování chyb od zákazníka a jak zabezpečit opravení těchto chyb v co možná nejkratší době. Z vlastních zkušeností mohu konstatovat, že zákazník zpravidla očekává určitou chybovost software a je spokojen, když jeho testovací tým nějakou chybu objeví. Velmi nespokojen je však ve chvíli, pokud je takovýchto chyb mnoho nebo jsou to chyby, jež mají zásadní dopad na funkčnost celé aplikace. Pokud jsou již nějaké chyby v úrovni akceptačních testů objeveny, je nutné v co nejkratším čase tyto chyby opravit a předat zákazníkovi k dalším testům.“ (Hlava, 2011)

Osobně mám zkušenosti takové, že často, pokud mají vývojáři k dispozici testery od začátku projektu, tak pokud mohou, tak testy jednotek přeskočí a rovnou dají nově vyvinutou funkcionalitu na otestování testerům. To může být v určitých situacích sice urychlení procesu vývoje a testování, v opačném případě to ale zbytečně zdržuje jak vývojáře, tak testery. Mám na mysli například situaci, kdy vývojář vyvine novou část aplikace, a protože má k dispozici

testery, tak bez vlastního otestování předá aplikaci k testům testerům. Pokud je však v aplikaci závažnější chyba, například se nelze vůbec do aplikace přihlásit, tak předání testerům a zpětné předání vývojářům k opravě zabere zbytečně mnoho času, za který by vývojář mohl aplikaci sám otestovat alespoň na této základní úrovni i danou chybu opravit.

Dále jsem se ve své praxi setkala, převážně na malých projektech, se situacemi, že k testům na integračním prostředí vůbec nedocházelo, ale aplikace/systém se testoval rovnou na akceptačním prostředí. K těmto situacím docházelo buď proto, že duplicitní tvorba testovacích dat jak pro integrační, tak pro akceptační prostředí by byla zbytečně časově náročná a pravděpodobně by zabrala více času než samotné otestování aplikace, nebo proto, že jednoduše daná organizace neměla k dispozici integrační prostředí (buď ho vůbec neměla, nebo nebylo udržované v takovém stavu, aby testy na tomto prostředí byly vypovídající o stavu aplikace).

Nejčastější zkušenost však mám s již výše uvedeným spojováním systémového a integračního testování.

3.2.2 ZPŮSOBY TESTOVÁNÍ

A. Formalizované

Testy pomocí testovacích scénářů. Od testera není vyžadována velká znalost aplikace, stačí, když postupuje krok za krokem podle scénáře. (Langrová, 2013)

B. Neformalizované

Testování podle test nápadů. Tester potřebuje určitou úroveň znalosti aplikace. Nejsou napsány jednotlivé testovací scénáře, jsou sepsány pouze test-nápady (= co je třeba otestovat a jakými možnými způsoby) a tester postupuje podle nich. (Langrová, 2013)

C. Badatelské

Tzv. freetesty, nebo exploratory testing - to znamená, že si tester většinou přečte pouze analýzy k dané aplikaci, aby zjistil, jaké jsou požadavky na aplikaci, a poté volně testuje. V praxi to znamená, že zkouší všemožné kombinace testů, které by pravděpodobně ani dobrý testovací scénář nebyl schopen podchytit. Tester tak není fixovaný pouze na to, co mu píše

scénář, ale soustředí se komplexně na celou aplikaci, nejen na funkčnost, kterou má testovat. (Langrová, 2013)

Také se badatelské testy používají v případě, kdy v procesu vývoje chybí podrobná specifikace. Což není pro softwarového testera ideální situace, ale právě v tomto případě jsou badatelské testy schůdným řešením. Za specifikaci se v tomto případě považuje samotný software, který je třeba metodicky prozkoumat, funkci za funkcí, je třeba poznamenat, co software dělá a zmapovat všechny jeho funkce. Takový software nebude nikdy možné otestovat tak důkladně, jako když je specifikace k dispozici. Když například nějaká funkce chybí, pak se to během testů nemusí vůbec zjistit. Přesto jej lze ale otestovat systematicky. Navíc, v takovéto situaci je nalezení jakýchkoli chyb vždy přínosem. (Patton, 2002, s. 57)

3.2.3 TYPY TESTŮ

Toto dělení testů je třeba dodržovat a nelze se jednomu typu testů během testování úplně vyhnout. Všechny tři typy testů se navzájem doplňují, a jakmile by se jeden typ testů vynechal, pak by byla aplikace nebo systém nedostatečně otestován a neměl by být akceptován k nasazení do produkčního prostředí. Jedinou výjimku tvoří situace, kdy je aplikace nebo systém vyvíjen zcela nový, protože v takové situaci není stávající systém, na kterém by při vyvinutí nových funkcionalit měly být prováděny regresní testy. V takovém případě jsou všechny testy inkrementální, do chvíle, než se začnou opravovat první chyby. Pak se k inkrementálním testům přidají i retesty opravených chyb.

A. Inkrementální

Inkrementální testování je testování nových funkcí tzn. buď celé nové aplikace, nebo v případě nové funkčnosti do stávající aplikace, je to testování této nové funkce. (Langrová, 2013)

B. Regresní

Regresní testování jsou opakované testy funkcí, které již dříve byly funkční. Nejčastěji se provádějí regresní testy v okamžiku, kdy je do aplikace nasazena nová funkce a je třeba otestovat, že nasazení nové funkce nemělo vliv na stávající funkce a vlastnosti aplikace nebo systému (nebylo-li to předmětem změny aplikace). Oblasti, které byly předmětem změny, testují inkrementální testy.

C. Retesty

Retesty se provádějí na základě zjištění chyb při testech. Netestuje-li se, nelze provádět retesty chyb, jelikož žádné chyby nejsou zadány. Od okamžiku zadání defektu, projde defekt „pod rukama“ analytika, poté je předán dodavateli, ten zajistí opravu. Tuto opravu si dodavatel otestuje a až poté je oprava chyby předána zákazníkovi. Po předání opravy zákazníkovi, dochází k retestu chyby, jako kontrola toho, že chyba byla skutečně opravena. Zároveň je však třeba otestovat, zda oprava chyby nezaslela novou chybu, tudíž se testují i navazující oblasti aplikace. (Langrová, 2013)

3.2.4 TESTOVÁNÍ DLE OBSÁHLOSTI

A. Testy splněním

Při testu splněním se dle případu užití určí základní cesta (příp. cesty) aplikace, která je testována, zda je funkční od začátku až po úspěšné ukončení.

Testování splněním je ve skutečnosti jen kontrola, zda systém funguje, alespoň na základní úrovni. Rozhodně to není snaha o dostání systému na hranice možností, případně překročení těchto hranic a vyvolání pádu systému. Je to jen jednoduché opatrné testování, v zásadě se testují jen základní pozitivní cesty. (Patton, 2002, s. 57)

Při návrhu a provádění testů se vždy začíná testy splněním, protože je důležité nejprve ověřit, zda software funguje a až poté jej testovat na hranice možností.

B. Testy selháním

U testů selháním, neboli vynucení chyb, na rozdíl od testu splněním, tyto testy nekontrolují, zda funguje hlavní tok případu užití, ale naopak se při těchto testech cíleně

zadávat chybné vstupy. Cílem je zjistit, zda aplikace hlásí při problémech chyby a zda neprovede činnost, pro kterou například nemá informace. (Langrová, 2013)

3.2.5 MANUÁLNÍ VS. AUTOMATIZOVANÉ TESTY

Závěrem této kapitoly bych ráda ještě zmínila dělení testů na manuální a automatizované. V dnešní době jsou velké snahy o zautomatizování testů. Ne všechny testy je však možné automatizovat. A ne u všech, které automatizovat lze, je to efektivní.

Základní rozdíl mezi manuálním a automatizovaným testováním je myšlení a vnímání. Automatické testy testují přesně to, na co jsou naprogramovány a vyhodnocují data pouze z daného pole, ze kterého je požadováno vyhodnocení. Na rozdíl od toho, člověk za počítačem (který „kliká“) je sice několikanásobně pomalejší, na druhou stranu vnímá vždy obrazovku jako celek, a i když se zrovna věnuje testu konkrétního pole, všimne si, pokud jinde na obrazovce chybí tlačítko, je špatná barva pozadí atp. Druhým velmi silným argumentem proti automatizovanému testování je údržba. Obecně platí, že automatizované testy se vyplatí dělat na takové procesy, které jsou stálé a do kterých se téměř nikdy nezasahuje. Pokud se tedy zautomatizuje test funkcionality, která se mění každý měsíc, pak jsou testéři-lidé mnohem efektivnější, protože jim stačí říct, k jaké změně došlo, a testy hned proběhnou podle nových pravidel. Kdežto u automatického testu by bylo třeba naprogramování testu změnit. Tyto změny představují obrovské množství práce, která zabere mnoho času. V takových případech automatizované testy přinášejí naopak více práce, než ulehčení (Langrová, 2013). Velmi výstižně to popsali A. Havlíčková a P. Roudenský (2013, s. 177-178) ve své knize Řízení kvality softwaru: „Ani ve své nejlepší a nejsofistikovanější podobě totiž automatizované testy nejsou skutečně inteligentní, nemluvě o vnímání použitelnosti a jiných mimofunkčních požadavků. Vykonávají, třeba i s jistou mírou variability a teoreticky (spíše na úrovni experimentální) i učeníivosti, nakonec jen to, co člověk při jejich návrhu určí. Nelze proto o automatizovaných testech uvažovat jako o testech manuálních, neboť jejich hodnoty jsou zcela odlišné – automatizovaný test vykonává stejnou činnost opakovaně, rychle a bezchybně. Oproti tomu manuální testování má zmíněnou kvalitu, jíž žádné výkonnostní parametry nemohou konkurovat – myšlení, kreativitu a aplikaci znalostí. Automatizovaný skript sice ověří ve zlomku sekundy, zda všechny řetězce na stránce zobrazují správná data, ale nedokáže říct, že písmo

je pro člověka nečitelné nebo že část některého textu je mimo stránku. A automatizace této činnosti by pravděpodobně byla náročnější než prosté provedení manuálních testů.“

Z podstaty automatizovaného testování vyplývá, že největší uplatnění najde převážně při regresních testech. Přesto, ale může být využito i například při výkonostním testování, kdy se využije k simulaci zátěže generované stovkami až tisíci uživateli. Dále se využívá i u funkčního testování, ale i při testech jednotek. A v neposlední řadě i při bezpečnostních testech, kdy se tímto způsobem ale dá ověřovat pouze odolnost vůči známým napadením či jejich variantám. (Roudenský a Havlíčková, 2013, s. 178)

K provádění funkčních automatizovaných testů se využívají čtyři techniky zmíněné níže. Nejčastěji je však v praxi využíván kombinovaný přístup, což přináší využití předností jednotlivých technik.

A. Zachycení a přehrávání aktivity uživatele

Tato technika je nejznámější formou automatizace, protože umožňuje za určitých okolností dosáhnout uspokojivých výsledků i zcela bez technických dovedností. V principu nástroj zachycuje aktivity uživatele v daném rozhraní a umožní jejich přehrávání.

Na druhou stranu je tato technika velmi citlivá na jakékoliv změny prostředí či aplikace a neumožňuje ošetření chybových stavů. (Roudenský a Havlíčková, 2013, s. 179)

B. Modifikace vygenerovaných skriptů

Tato technika také vychází ze zaznamenaných skriptů, nicméně jsme v ní schopni provést úpravy řešící omezení. Lze také rozšířit obsah testu i variabilitu proměnných. (Roudenský a Havlíčková, 2013, s. 180)

C. Testování řízené daty

Testování řízené daty je vhodné, je-li třeba provést opakovaně testy, které využívají stejnou logiku, ale velký objem vstupů a výstupů. Ty jsou pak udržovány odděleně v souboru CSV nebo XLS. Testovací skript poté načítá řádek po řádku dat ze souboru a provádí potřebné operace. (Roudenský a Havlíčková, 2013, s. 180)

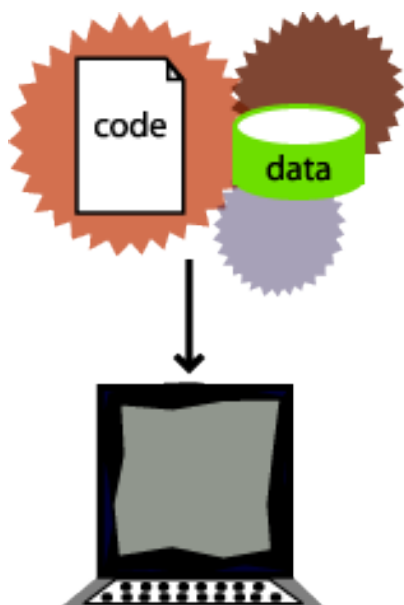
„Řídíte-li test daty, používá skript proměnné pro klíčová vstupní pole aplikace a programy namísto literálových hodnot, aby bylo možné použít externí data při řízení testované aplikace daty.

Testování řízené daty používá data z externího souboru, tj. z datového fondu, jako vstup pro určitý test. Datový fond je kolekce souvisejících datových záznamů. Řídíte-li test daty, datové fondy poskytují datové hodnoty proměnným v testovacích skriptech při jejich přehrávání.

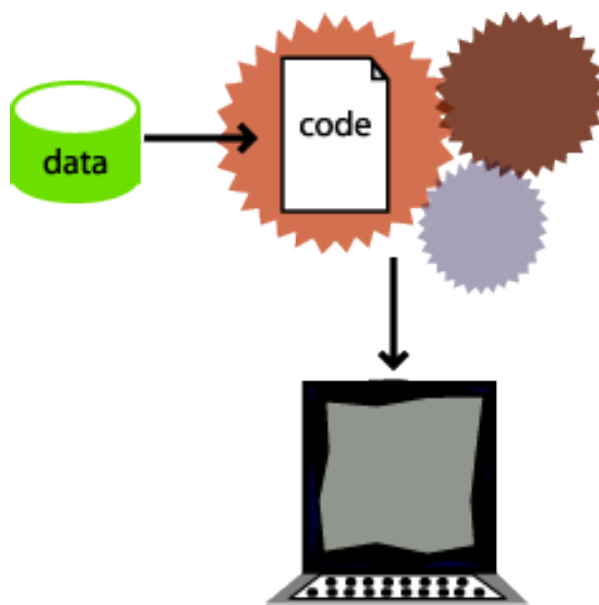
Protože jsou data oddělena od testovacího skriptu, můžete:

- Upravit testovací data bez ovlivnění testovacího skriptu.
- Přidat nové testovací případy úpravou dat, nikoli testovacího skriptu.
- Sdílet testovací data s mnoha testovacími skripty.

Obrázek 3 zobrazuje testovací skript, který používá data s pevně naprogramovanými literálovými odkazy v testovacím skriptu. Obrázek 4 zobrazuje testovací skript řízený daty, který používá data z externího souboru, datového fondu.



Obrázek 3: Pevně naprogramovaný testovací skript s literálovými odkazy [Zdroj: Přehled testů řízených daty]



Obrázek 4: Testovací skript řízený daty s datovým fondem [Zdroj: Přehled testů řízených daty]

Zde je několik případů problémů, které testy řízené daty řeší:

Problém: Během záznamu vytvoříte osobní evidenční kartu nového zaměstnance s použitím jedinečného identifikačního čísla zaměstnance. Při každém spuštění testu bez řízení daty je proveden pokus o vytvoření stejné evidenční karty a poskytnutí stejného identifikačního čísla. Aplikace duplicitní požadavky odmítne.

Řešení: Můžete řídit testovací skript daty, aby byla na server při každém spuštění testu odeslána odlišná data zaměstnance, včetně identifikačních čísel.

Problém: Během provádění záznamu odstraníte záznam. Spustíte-li test bez řízení daty, produkt Rational Functional Tester se pokusí odstranit stejný záznam a zobrazí se chyba „Záznam nebyl nalezen“.

Řešení: Můžete testovací skript řídit daty a při každém požadavku na odstranění při přehrávání skriptu použít odkaz na jiný záznam.

Řízení testů daty poskytuje přesnější pohled na způsob, jakým testovaná aplikace pracuje v reálném prostředí s reálnými daty.“ (Přehled testů řízených daty)

D. Testování řízené klíčovými slovy

Podobají se testům řízeným daty, ale zdrojové tabulky neobsahují jen data, ale také jednotlivé příkazy pro testovací skript. Tím je během procesu vytvářena samotná logika procedury testu. (Roudenský a Havlíčková, 2013, s. 180)

3.3 CO JSOU TESTOVACÍ DATA

Když se řekne testování, tak jako nezbytné součásti si každý představí lidské zdroje – testery, test analytiku, pak různé dokumenty, které jsou k testům potřeba – specifikace produktu, testovací scénáře, ale ne vždy dojde řeč na testovací data. Přesto jsou testovací data nezbytnou součástí testování, bez nich by k žádným testům dojít nemohlo. Forma i množství těchto dat je velmi individuální a závislá na testovaném softwaru. Požadavky na testovací data by měly být definovány už při plánování testů, tedy ve fázi plánování projektu.

Co se týče definice testovacích dat, tak se zdroje naprosto shodují. Tvrdí, že „Test data is the data that is used in tests of a software system.“ (What is Test Data, 2015 ; Software and testing training, 2013), že testovací data jsou taková data, která jsou použita pro testy softwaru. Totéž tvrdí i Collinsův výkladový slovník, doslova: „data that is used in the testing of a computer program“ (Collins English Dictionary, c2015) i různé blogy věnující se testování (Software Testing Class).

Testovací data nemusí nutně být jen elektronická (například databáze klientů), jak název „data“ může evokovat, ale mohou být i fyzická – například SIM karty. Vždy je třeba na testy připravit dostatek dat. Tato data se nevytváří náhodně, ale už při plánování testů by měla být specifikována množina konkrétních dat a jejich vlastností, které je třeba zařídit, a to na základě očekávaných budoucích uživatelů a také dle specifik daného produktu. Sloveso zařídit je v tomto směru vybráno cíleně, protože je několik možností, jak získat množinu testovacích dat. Také je vždy potřeba zařídit několikanásobně větší množství dat než pro ideální případ, že by při testech nebyly nalezeny žádné chyby, protože vždy jsou nalezeny nějaké chyby. Pak je potřeba mít konkrétní data pro první test, případně pro nasimulování chyby, a ještě pak pro retest.

Zajistit data pro testy lze čtyřmi způsoby:

1. Použití již dříve používaných testovacích dat
2. Kopie dat z produkčního prostředí a jejich anonymizace
3. Generování testovacích dat pomocí generátorů
4. Tvorba syntetických dat na testovacím prostředí

První možností je použít data, která byla již dříve používána pro nějaké testy, pokud tato data již používána nejsou a mají přesně ty vlastnosti, které pro své testy potřebujeme. To je nejjednodušší varianta, jak získat testovací data, ne vždy tato možnost ale přichází v úvahu. Nicméně i kdyby jen část dat měla být opakovaně použita tak je to výrazné ušetření času, který bychom strávili získáváním nových dat. Další výhoda těchto opakovaně použitých testovacích dat je, že máme data s určitou historií, což je někdy potřeba a u nově vytvořených dat nikdy

nezískáme například pohyby na účtu klienta či jiné dlouhodobě náročnější data na tvorbu a údržbu.

Druhým a nejčastěji využívaným způsobem získání testovacích dat je varianta překopírování dat z produkčního prostředí (s nutností anonymizace citlivých údajů). Tento proces je mnohem složitější hlavně na přípravu, protože se musí naplánovat nejen přesný datum a čas, kdy se vytvoří „otisk“ dat z produkčního prostředí. Dále zde nastává problém, že často jsou data uložena ve více systémech zároveň a jsou na sebe navázána. Což může samo o sobě způsobovat problémy, ale pokud to nezpůsobí problémy při tvorbě „otisku“ dat z produkčního prostředí, pak to zcela jistě způsobí problémy a některá data zničí ve chvíli, kdy se začnou data anonymizovat (často se data anonymizují promícháním určených polí s citlivými údaji). V tu chvíli k sobě přestanou sedět jednotlivá data a může se stát, že budeme mít nakonec 80-letého muže, který má status studenta a pobírá plat na manažerské pozici. Taková data pro nás jsou bezcenná, protože absolutně neodpovídají reálným datům. Nicméně stále je tento postup velmi vhodný při potřebě velkého množství dat pro testování, aby nebylo nutné data tvořit manuálně. Dále je nutné rozdělit zodpovědnost za jednotlivé akce tvorby dat z produkčního prostředí a s tím souvisí i nutnost určení, která konkrétní data budou převedena, protože testovací prostředí má často mnohem menší kapacitu, než prostředí produkční. (Škrobák., 2011b)

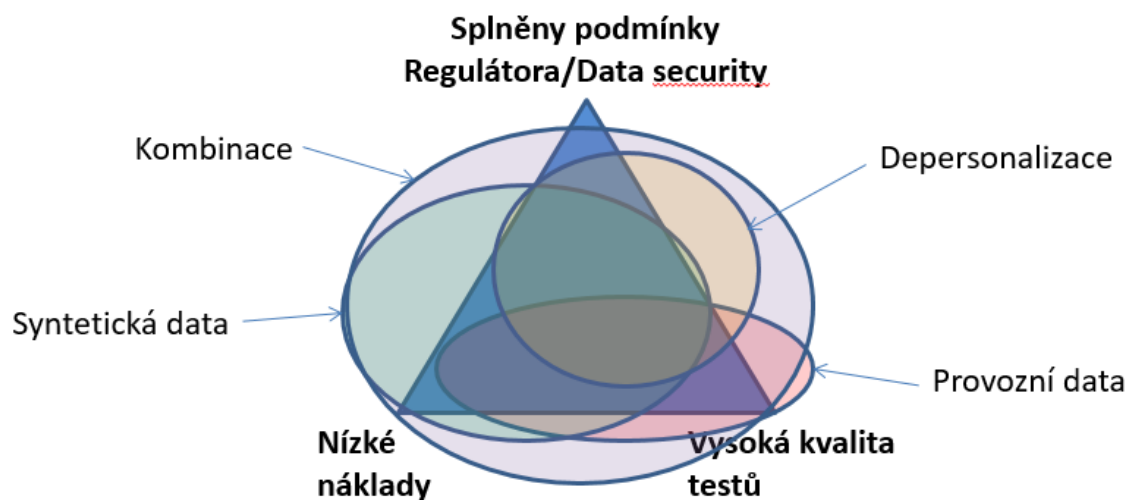
Třetí možností je generování dat. Různé přístupy ke generování dat velmi dobře popsal M. Šimůnek (2013) v časopisu *Systémová integrace*. „Nejjednodušším postupem přípravy umělých dat je generování generátorem náhodných čísel, bez jakýchkoliv dalších požadavků na ně, viz (Ross, 2006 ; Hamuro a kol., 2002). Jde o velmi jednoduchou a rychlou metodu. Opačný postup je pečlivě nastudovat vlastnosti reálných dat a zkonstruovat jejich statistický model, viz (Abowd a Lane, 2004). Jde o pracnou činnost, která může být částečně (nikoliv úplně) automatizovaná, viz (Žabkar a kol., 2013). Umělá data jsou spočtena na základě zkonstruovaného modelu, případně doplněna náhodným šumem pro zvýšení dojmu realističnosti. Problém je složitost statistického modelu, pokud by měl zahrnout všechny vlastnosti a výjimky reálných dat. V praxi se pak setkáváme spíše se zjednodušenými modely, které obvykle ignorují zajímavé výjimky. Druhým problémem je případně dodaná náhodná složka, která musí být opatrně nastavena tak, aby zakryla umělou povahu dat, přitom v nich však

nesmí přebít chtěné vztahy. Za tímto účelem bylo vyvinuto mnoho metod zajišťujících, že vybrané podstatné vlastnosti jsou v datech zachovány, např. vztahy mezi vysvětlujícími a závislými proměnnými, viz (Potharst a Wezel, 2005).“

Čtvrtou a poslední možností získání testovacích dat je jejich vytvoření přímo na testovacím prostředí. Nejčastěji se k tomuto vytvoření používá přímo testovaný produkt, což může už v začátcích pomoci odhalit zásadní chyby produktu. Zcela zásadní nevýhodou této varianty je její časová náročnost. Potřebujeme-li například pár klientů, abychom otestovali, že jde založit běžný účet, pak za dvě hodiny máme založeno deset klientů, na kterých danou funkcionalitu otestujeme. Máme-li ale komplikovanější zadání, větší projekt a potřebujeme-li více dat, pak manuální tvorbou testovacích dat se mohou strávit nejen dny ale dokonce týdny. (Škrobák, 2011a)

Aby taková manuálně vytvořená testovací data byla kvalitní a pro testy použitelná, je třeba dodržovat několik pravidel. Zprv je třeba, aby tato vytvořená data se co nejvíce blížila reálným datům na produkčním prostředí, tedy je vhodné klientům vymýšlet nějaká existující jména, adresy, výplatu, aby systém byl otestován právě na takových datech, pro která je navržen a se kterými bude pracovat. Druhým pravidlem je, že je třeba data obměňovat. Nelze totiž stále testovat na stejné množině testovacích dat. Zprv data stárnou a to opravdu velmi rychle, zadruhé při používání tato data ztrácí své vlastnosti (které ale často ztrácí i při nepoužívání) a zatřetí musí data odpovídat aplikaci tak, v jakém je současném stavu. Nelze testovat nové funkcionality daného produktu na datech, která byla vytvořena pro několik let starou verzi daného produktu. Na druhou stranu je vhodné mít i takováto historická data, aby se dalo otestovat, jak na ně produkt reaguje. Nicméně většinu testů je vhodné ověřovat na datech odpovídajících nejnovější verzi produktu. A jako třetí pravidlo je dobré se vyhnout vytváření dat tzv. „natvrdo“. Myšleno je tím, když například vytvoření klienta skrze aplikaci trvá půl hodiny, ale vytvoření takového klienta přímo do databáze zabere pět minut. Což na první pohled ušetří velké množství času, nicméně se také může stát, že při tvorbě dat přímo do databáze daná data neprošla všemi aplikacemi a databázemi, kterými měla a pak na testera při zadání klienta může vyskočit nepříjemná hláška: „Aplikace nemůže načíst tento záznam“. (Škrobák, 2011a)

Jednotlivé možnosti získání testovacích dat a jejich vliv na regulační podmínky, náklady projektu a kvalitu testů znázorňuje Obrázek 5, kde je vidět kterou část trojúhelníku kvality a jaké typy testovacích dat pokrývají. Každý typ testovacích dat přináší jiné výhody a nevýhody. Samozřejmě ideální kombinací by byla data, která splňují podmínky regulátora, zajistí vysokou kvalitu testů a zároveň náklady na vytvoření nebo získání těchto dat budou minimální. Což je v praxi možné získat jen kombinací všech uvedených typů testovacích dat, rozhodně všechny tyto podmínky zároveň nesplňuje jen jeden typ testovacích dat. Provozní data je velmi jednoduché získat a náklady na jejich získání jsou opravdu minimální. Vzhledem k tomu, že tato data kopírují reálná provozní data, pak zajistí vysokou kvalitu testů, protože testy proběhnou na takových datech, která se reálně vyskytují v provozu. Ale tato data je třeba depersonalizovat, protože bez depersonalizace je protizákonné taková data používat a jde to proti veškerým nařízením regulátora. Depersonalizovaná data naopak splňují bezpečnostní opatření regulátora, nicméně, díky depersonalizaci už nemohou zajistit takovou kvalitu testů, protože se vytratí určité návaznosti mezi jednotlivými daty. Navíc tato depersonalizace provozních dat je velmi finančně i časově náročná. Další variantou je tvorba syntetických dat. Tento typ dat je nejlevnější variantou získání testovacích dat, nicméně syntetická data nejsou schopny zajistit takovou kvalitu testů jako data provozní, protože jsou uměle vytvořená, tudíž nepřinášejí reálné kombinace dat vyskytující se v provozu. Navíc při tvorbě syntetických dat mohou být vytvořena nesmyslná data (například když tester pojmenuje klienta Test Test), která mohou při testování přinášet problémy (v tomto příkladu nerozeznání jména od příjmení klienta). Proto je nejvhodnější variantou kombinace všech těchto typů testovacích dat, ale to je náročné časově a dodavateli se kvůli malým projektům či krátkodobému testování nevyplatí. Na Obrázku 6 je tento vliv zobrazen v tabulce pro větší přehlednost.



Obrázek 5: Vliv testovacích dat na trojúhelník kvality [Zdroj: Gabriel, 2014]

	Regulátor	Náklady	Testing
Provozní data	↓	→	↑
Syntetická data	↑	↑	↓
Depersonalizace	→	↓	↑
Kombinace	↑	→	↑

Obrázek 6: Vliv testovacích dat na trojúhelník kvality v tabulce [Zdroj: Gabriel, 2014]

3.4 TYPY TESTOVACÍCH DAT

Testovací data se dělí dle typu do tří nebo čtyř kategorií v závislosti na zdroji, ze kterého je čerpáno. První varianta kategorizace testovacích dat je dle Software and testing training (2013):

A. Vstupní nebo výstupní data

Vstupní data jsou ta, která se při testování zadávají do systému, a výstupní data jsou data, která očekáváme, že nám systém poskytne. Výstupní data se pak porovnávají s výstupem

systému, zda splňují požadavky na systém. Tato data bývají nejčastěji udržována v textových dokumentech, excelech nebo databázích. Nejčastěji to bývají uživatelská data.

B. Systémová konfigurační data

Nejčastěji bývají uložena v .ini souborech nebo databázových tabulkách. Tato data obsahují některá systémová nastavení jako například propojení databázových řetězců, informace pro připojení k databázi, ale také uživatelská jména a hesla nebo třeba informace o emailovému serveru.

C. Systémová data

Taková data mohou být uložena samostatně ale i v konfiguračních souborech. Jsou to data, která nám ukazují, jak je systém zobrazován uživateli. Příklad systémových dat uložených v konfiguračních souborech může být například menu a systémová data uložená samostatně mohou být třeba produkty v e-shopu.

D. Transakční data

Transakční data vygenerovaná systémem. Jako příklad takových dat vytvořených systémem může být profil uživatele vytvořený na základě vložených dat, nebo objednávka v e-shopu vytvořená dle vstupů, které uživatel zadal.

Druhý způsob kategorizace testovacích dat nabízí blog Testing the new system (1999). Toto dělení je znázorněné na Obrázku 7:

A. Normální datové hodnoty

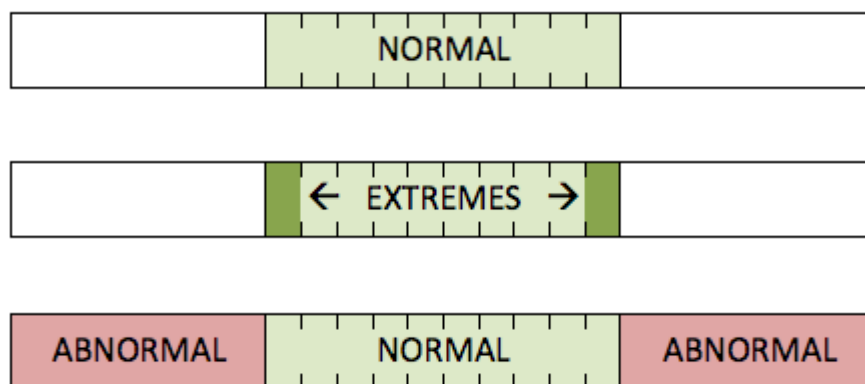
To jsou takové, které za normálních okolností budou zadány do systému. Systém by je měl akceptovat, zpracovat a poté bychom měli být schopni zkontrolovat výsledky a porovnat je s očekávaným výsledkem.

B. Extrémní datové hodnoty

Ty jsou stále ještě normálními datovými hodnotami, avšak jsou to absolutní hraniční hodnoty. Extrémní datové hodnoty se v testování využívají pro kontrolu, že systém bude akceptovat všechny normální datové hodnoty, a že je zpracuje korektně.

C. Abnormální datové hodnoty

Tyto hodnoty by systémem vůbec neměly být přijaty, protože jsou neplatné. Systém by měl všechny tyto abnormální datové hodnoty zamítnout. V testování se používají tato data pro ujištění, že nesprávná data nenaruší funkčnost systému.



Obrázek 7: 3 typy testovacích dat [Zdroj: Testing the new system, 1999]

První dělení testovacích dat je z pohledu toho, jak jsou data vytvořena, zda systémem, nebo testerem/uživatelem, a jaké jsou jejich vlastnosti. Toto dělení ukazuje způsob, jak se k daným datům dostat, kde je nalezneme a také, jak se o daná data dle typu starat. Na druhou stranu, druhý způsob dělení dat je spíše vhodný pro analýzu zadání před tvorbou dat pro testy, abychom si mohli vytvořit vhodné množiny testovacích dat dle hraničních hodnot. Toto druhé dělení považuji pro testy, potažmo pro testery, za přínosnější z pohledu tvorby testovacích dat.

Druhé dělení na normální, abnormální a extrémní hodnoty je možné pouze u dat, která jsou vytvořena testerem, myšleno taková, která jsou vytvořena pro testery pro účely testování. Dle prvního dělení se tedy jedná o vstupní a výstupní data a o transakční data. Dle prvního dělení se dají tato data rozdělit do dvou nadřazených kategorií. První kategorií jsou data vytvořená testery pro testery na účely testování (myšleno tedy i data generovaná nebo

překopírovaná z produkčního prostředí) a do této kategorie by patřila již zmíněná vstupní a výstupní data a transakční data. A tato data se pak dále dají dělit na normální, extrémní a abnormální testovací data. Druhou kategorií pak jsou systémová data, která jsou vytvořena systémem nebo vývojářem, nicméně jsou to data, při jejichž změně dojde ke změně chování aplikace nebo zobrazení výstupu aplikace a nemají žádné normální, abnormální a extrémní hodnoty.

3.5 MANAGEMENT TESTOVACÍCH DAT (TEST DATA MANAGEMENT)

Khurana a Bindal (2014) definují test data management jako vývoj, provedení a kontrolu plánů, politiky, programů a postupů, které kontrolují, chrání, dodávají a zvyšují hodnotu testovacích dat a informačních aktiv. Více formálně je test data management definován jako proces řízení identifikace, získávání, podmínkování, naplnění a udržování testovacích dat během plánování, přípravy a provedení testů. Je to proces, který definuje standardy, úkoly, vlastnictví, role a odpovědnosti k zajištění přesnosti, úplnosti a integrity testovacích dat. Test data management doporučují pro všechny fáze a typy testování: testování jednotek, systémové testování, integrační testování, akceptační testování, testování na produkčním prostředí, regresní testy i automatizované testování.

Dále Khurana a Bindal (2014) identifikovali osm důvodů, proč potřebujeme test data management:

1. Testeři ztrácejí více času přípravou testovacích dat než testováním samotné aplikace. Příprava dat pro testování je velmi náročná na lidské zdroje a ty mohou spotřebovat více než 50 % celkového času na testování a dalších zdrojů.
2. Testeři jsou velmi závislí na business analyticích, že jim poskytnou požadovaná testovací data.
3. Koncové termíny se několikrát posunují kvůli zpoždění v dodávce nebo obnově testovacích dat.

4. Mnoho chybných defektů je způsobeno problémy způsobenými chybnými testovacími daty: Mnoho defektů je zadáno a poté jsou odloženy z důvodů chybných testovacích dat.
5. Testovací data se nepoužívají znovu. Stará testovací data by měla být odložena, když nové testy vyžadují nová testovací data. Je třeba tato data obnovovat.
6. Je spousta závislostí na navazujících systémech, pro které musí být vytvořena testovací data. Čekání na testovací data pro jiný systém způsobuje zpoždění.
7. Je třeba centrálně kontrolovat testovací data. Mnoho aplikací může používat stejná testovací data současně, takže musí být zařízen simultánní přístup a musí se zajistit konzistence dat.
8. Se zvětšující se velikostí projektu je těžší spravovat testovací data.

Techopedia zase definuje test data management (Test Data Management) jako proces plánování, projektování, ukládání a správu procesů a metodologie kvality testování softwaru. To umožňuje týmu testování a kvality softwaru mít kontrolu nad daty, soubory, pravidly a politikami vytvořenými během celého životního cyklu testování softwaru. Test data management je také známý jako software test data management. Primárním účelem test data managementu je vytvářet, spravovat a udržovat zdrojové kódy aplikace nebo softwaru pro účely testování. Tyto zdrojové kódy jsou odlišné od klíčových zdrojových kódů vývoje. Test management dat umožňuje oddělení testovacích a produkčních dat, udržování verze testovaného softwaru, sledování chyb a provádění dalších procesů testování softwaru. Jedním z klíčových cílů správy testovacích dat je minimalizovat a optimalizovat velikost testovacích dat, jakož i shromažďování a centralizování testovací dokumentace a zdrojů.

Osobně se ztotožňuji s formálnější definicí managementu testovacích dat od Khurany a Bindala (2014). Jejich důvody, proč potřebujeme test data management, se většinou překrývají s identifikovanými problémy s testovacími daty. Blíže tyto problémy a jejich souvislost s výše uvedenými důvody popisují v kapitole 4.2 *Metoda sběru dat*. Je však evidentní, že identifikovaným problémům s testovacími daty se dá alespoň částečně předcházet zavedením managementu testovacích dat do organizace. Samozřejmě jen

samotné zavedení test data managementu nevyřeší žádné problémy, je třeba dodržovat stanovené postupy a metodiky a i přesto se nepodaří eliminovat veškeré problémy s testovacími daty, které pravděpodobně před, během nebo i po testech nastanou. Nicméně některým jako častému ničení testovacích dat, špatné specifikaci množiny dat, případně nekonzistenci testovacích dat napříč aplikacemi nebo prostředími se dá alespoň částečně předcházet, což zefektivní a zrychlí testovací fázi projektu a tím i celou dodávku projektu.

4 PRŮZKUM NEJČASTĚJŠÍCH PROBLÉMŮ S TESTOVACÍMI DATY A MOŽNOSTÍ JEJICH ŘEŠENÍ

V této kapitole představím průzkum nejčastějších problémů s testovacími daty a možností jejich řešení, který jsem realizovala pro splnění hlavního cíle této diplomové práce. Tento průzkum byl veden pomocí dotazníkového šetření. Nejdříve představím cíle provedeného průzkumu a poté metody sběru dat. V další části této kapitoly představím dotazník, jeho tvorbu, výběr otázek a jejich návaznost na cíle průzkumu potažmo na cíle této diplomové práce.

4.1 CÍLE PRŮZKUMU

Motivací pro realizaci průzkumu byly zaprvé nedostačující informace o problémech s testovacími daty v České republice a chybějící realizovaný průzkum týkající se této problematiky a zadruhé vlastní zkušenost s těmito problémy a absence informací k řešení těchto problémů, se kterými jsme se se já i mí kolegové testeři, test analytici a test manažeři denně setkávali při provádění testů.

Hlavním cílem dotazníkového průzkumu je vyhodnotit a analyzovat problémy s testovacími daty, se kterými se pracovníci v oblasti testování setkávají nejčastěji. Dalším návazným cílem je zjistit, zda jsou nějaké způsoby, jak tyto problémy řešit, případně jim předcházet, a jaká ta řešení jsou. V rámci zjišťování možností řešení problémů je cílem eliminovat ta řešení, která přinášejí další návazné problémy, tedy nejsou řešeními zcela efektivními.

V rámci kvalitativního výzkumu formou nestrukturovaného rozhovoru, který předchází dotazníkovému šetření, je hlavním cílem identifikovat problémy s testovacími daty, se kterými se pracovníci v oblasti testování setkávají, nezávisle na četnosti či závažnosti těchto problémů. Tyto problémy pak kategorizovat do ucelených oblastí, které pak mohu použít pro dotazníkové šetření o četnosti výskytu těchto problémů s testovacími daty a možnostech jejich řešení.

4.2 METODA SBĚRU DAT

V rámci průzkumu nejčastějších problémů s testovacími daty a možností jejich řešení jsem využila dvou metod sběru dat.

Hlavní použitou metodou bylo dotazníkové šetření, které je jednou z nejčastějších forem sběru dat v dnešní době. Respondenti mají jasně definované otázky, které je nutné nebo možné zodpovědět a další otázky se nemění dle předchozích odpovědí. V dotazníku jsem zvolila otázky uzavřené i otevřené. Uzavřené otázky v dotazníku jsou s výběrem jediné odpovědi, ale i vícehodnotové, a to převážně u otázek týkajících se výběru problémů s testovacími daty, se kterými se respondent setkal, protože je velmi pravděpodobné, že se setkal s více problémy než jen s jedním. Otevřené otázky jsou využity u otázek týkajících se možností řešení. Všechny otázky v tomto dotazníku jsou nepovinné. (Řezanková, 2011, s. 14) Dotazníkový průzkum byl realizován elektronickou cestou.

Druhou použitou metodou sběru dat, v tomto případě jako podpůrnou k dotazníku, byl kvalitativní výzkum formou nestrukturovaného rozhovoru. Protože jsem pro své dotazníkové šetření chtěla mít již definované oblasti problémů s testovacími daty tak, aby respondenti tyto problémy nemuseli vymýšlet, ale aby pouze vybrali ty, se kterými se setkali, potřebovala jsem identifikovat tyto oblasti problémů. V tomto případě jsem se rozhodla pro nestrukturovaný rozhovor, resp. diskusi v kolektivu odborníků z praxe. (Strauss a Corbin, 1999, s. 19) Navštívila jsem 23. setkání [pro]Test! v Praze dne 25. 11. 2014, které mělo jako hlavní téma Test data management. Zde jsem v krátké prezentaci uvedla téma své diplomové práce a pod mým vedením jsem nechala kolektiv přibližně 30 test analytiků, test manažerů a testerů diskutovat na mnou položenou otázku: „S jakými problémy s testovacími daty se v praxi setkáváte?“ Zápis z tohoto setkání včetně diskuze na položenou otázku je v Příloze 2. Výsledky tohoto šetření jsem nadále použila ve výše zmíněném dotazníku. (Langrová, 2014)

Definované oblasti problémů s testovacími daty uvádím v Tabulce 1. Oblasti problémů s testovacími daty z článku od Khurana a Bindala (2014) uvádím v Tabulce 2.

Tabulka 1: Oblasti problémů s testovacími daty definovány na [pro]Test! [Zdroj: autorka]

Oblast problémů s testovacími daty z [pro]Test!
Protichůdné požadavky na data - Při špatné specifikaci, nebo velkém projektu, může být problematické vyhovět s tvorbou dat všem požadavkům IT a business oddělení. Příkladem protichůdného požadavku na data může být používání Data warehouse jako online zdroje.
Problematika anonymizace dat - Při získávání dat z produkčního prostředí je třeba data anonymizovat. To je velmi náročné na zdroje i na provedení.
(Ne)konzistence dat napříč různými aplikacemi nebo prostředími - Například při použití anonymizovaných dat z produkčního prostředí pak je kopie těchto dat na testovacím prostředí. To může způsobovat duplicity. Také při manuální tvorbě dat na testovacím prostředí nebo v nové aplikaci nemusí dojít k propsání všech dat do všech systémů a poté při testech data padají na chyby, právě proto, že nejsou někde propsaná.
Nemožnost tvorby historických dat (například starších typů produktů) - Jedná se o data, která je třeba tvořit dlouhodobě. Jako typický příklad je klient s obraty na účtu, nebo s vygenerovanými výpisy pohybů na účtu za poslední rok. Takového klienta nelze vytvořit ihned, je třeba používat daného klienta dlouhodobě, aby se tato data vytvořila.
Špatná specifikace množiny dat - Při špatném zadání, nebo změnách zadání může dojít k tomu, že jsou špatně definovány požadavky na data. Tak dojde k tomu, že nejsou otestovány všechny případy, které by otestovány měly být.
Časté ničení dat (malým používáním, sdílením dat mezi testery, atp.) - Při sdílení dat mezi testery může docházet k ničení dat, kdy například jeden tester na daném klientovi vytváří dlouhodobá data a druhý pak tohoto klienta použije pro test zrušení účtu, nebo uzavření klienta. Některá data se naopak můžou ničit tím, že se delší dobu nepoužívají.
Časová náročnost přípravy některých specifických dat (dlouhodobá data nebo data zpětně) - Souvisí s předchozími zmíněnými problémy. Dlouhodobá data jsou velmi náročná na tvorbu a obecně udržování dat ve funkčním použitelném stavu je časově velmi náročné a ne vždy jsou na takové činnosti lidské zdroje.

Tabulka 2: Oblast problémů s testovacími daty ze zahraničních zdrojů [Zdroj:Khurana a Bindal, 2014]

Oblast problémů s testovacími daty ze zahraničních zdrojů
Testeři ztrácejí více času přípravou testovacích dat než testováním samotné aplikace: Příprava dat pro testování je velmi náročná na lidské zdroje a ty mohou spotřebovat více než 50 % celkového času na testování a dalších zdrojů.
Testeři jsou velmi závislí na business analyticích, že jim poskytnou požadovaná testovací data.
Koncové termíny se několikrát posunují, kvůli zpoždění v dodávce nebo obnově testovacích dat.
Mnoho chybných defektů je způsobeno problémy způsobenými chybnými testovacími daty: Mnoho defektů je zadáno a poté jsou odloženy z důvody chybných testovacích dat.
Testovací data se nepoužívají znovu: Stará testovací data by měla být odložena, když nové testy vyžadují nová testovací data. Je třeba tato data obnovovat.
Je spousta závislostí na navazující systémy, pro které musí být vytvořena testovací data: Čekání na testovací data pro jiný systém způsobuje zpoždění.
Je třeba centrálně kontrolovat testovací data: Mnoho aplikací může používat stejná testovací data současně, takže musí být zařízen simultánní přístup a musí se zajistit konzistence dat.
Se zvětšující se velikostí projektu je těžší spravovat testovací data.

Z výše uvedených tabulek vyplývá, že problémy tak jak je definovali Khurana a Bindal (2014) jsou víceméně jinými slovy popsány i ve výstupu z diskuze na [pro]Test!u. Problém závislostí na navazující systémy, pro které musí být vytvořena testovací data a zpoždění způsobené čekáním na testovací data je ze setkání [pro]Test!u identifikován jako časová náročnost přípravy některých specifických dat a také (ne)konzistence dat napříč různými aplikacemi. Hned první důvod pro zavedení test data managementu, tedy že testeři ztrácejí více času přípravou testovacích dat než testováním samotné aplikace a příprava dat pro testování je velmi náročné na lidské zdroje a ty mohou spotřebovat více než 50 procent celkového času na testování a dalších zdrojů je také výše zmíněná časová náročnost přípravy některých specifických dat. Nepoužívání testovacích dat znovu, tedy že stará testovací data by měla být odložena, když nové testy vyžadují nová testovací data, a že je třeba tato data

obnovovat dle Khurany a Bindala (2014) je problém častého ničení dat. Důvod mnoha chybných defektů způsobených chybnými testovacími daty, kdy tyto defekty jsou zadány a poté jsou odloženy z důvodu chybných testovacích dat, může být způsobeno buď špatnou specifikací dat, protichůdnými požadavky na testovací data nebo nekonzistencí testovacích dat napříč různými aplikacemi a prostředími.

Rozhodla jsem se tedy, že pro účely mého dotazníkového šetření použiji výstup ze setkání [pro]Test!u, protože tento seznam dali dohromady odborníci v dané oblasti České republiky.

4.3 DOTAZNÍK A VÝBĚR OTÁZEK

Dotazník (viz Příloha 1) jsem se snažila navrhnout tak, aby jeho zodpovězení respondentům nezabralo mnoho času a zároveň, aby tam, kde to je možné, byly otázky uzavřené, nikoli otevřené. To vše mělo respondentům usnadnit a zrychlit vyplňování dotazníku. Otázky pro dotazník byly zvoleny tak, aby došlo k naplnění cílů průzkumu a tím i k naplnění cíle této diplomové práce.

Výběru a definování otázek pro dotazníkové šetření předcházely kvalitativní výzkum ohledně oblastí problémů s testovacími daty. Na základě kategorizace oblastí problémů s testovacími daty jsem mohla vytvořit dotazník. Díky provedenému kvalitativnímu průzkumu a použití výsledků tohoto průzkumu pro dotazník jsem splnila tyto dílčí cíle, které jsem si stanovila v úvodu této práce, a to: „realizovat a vyhodnotit průzkum ohledně problémů s testovacími daty“ a „vyhodnotit, se kterými problémy se pracovníci setkávají nejčastěji“.

Dotazník k průzkumu obsahuje 6 otázek, z toho první dvě jsou informativní charakterizující anonymního respondenta do jaké míry má zkušenost s testováním, jak dlouho se testování věnuje, případně zda prošel jen jedním či více projekty. Pravděpodobnost setkání se s různými problémy s testovacími daty se zvyšuje při práci na různých projektech a s přibývajícím zkušenostmi. Tuto závislost v dotaznících sleduji. Také jsou tyto otázky v dotazníku jako kontrolní, aby dotazník nevyplňoval někdo, kdo se v oblasti testování nepohybuje. Pak by odpovědi takového respondenta mohly zkreslit výsledky zkoumání.

Třetí otázka se už dotazuje na konkrétní oblasti problémů s testovacími daty. Díky odpovědím na tuto otázku zjistím, které problémy s testovacími daty jsou ty nejčastější a tím splnit druhou část výše zmíněného dílčího cíle této diplomové práce. Vyhodnocení této otázky je čistě statistické, mohu díky ní stanovit ty nejčastější problémy se kterými se pracovníci v oblasti testování setkávají, a případně, opět na základě výsledků dotazníku, vyhodnotit některý z problémů jako minoritní, protože se s ním respondenti nesetkávali, nebo jen velmi málo.

Poslední tři otázky se týkají možností řešení výše uvedených problémů. Čtvrtá otázka přináší jasný přehled o tom, zda vůbec k daným problémům bylo nalezeno konkrétní řešení. Vzhledem k tomu, že otázky v dotazníku nejsou povinné, může se stát, že respondent ví, že našel určité řešení daného problému, ale z nějakého důvodu se o dané neřešení nechce podělit. Proto díky otázce 4 uvidím, že řešení na daný problém je, i když nebudu znát konkrétní postup. Na což přímo navazuje otázka 5, která je otevřená a kde dávám respondentům prostor na to, aby se mohli podělit o řešení problémů s testovacími daty, která označili jako ta, ke kterým našli řešení. Tyto dvě otázky mi poskytnou možnost splnit další dílčí cíl: „Výše zmíněný dotazník se také zaměří na to, zda některé již identifikované problémy s testovacími daty se daří efektivně řešit, nebo jim dokonce předcházet. Je možné, že z dotazníků vyplyne i více možných způsobů prevence nebo řešení určitého problému.“

Odpovědi na poslední otázku vyřeší poslední cíl, který jsem si stanovila: „Nakonec se v dotazníku pokusím podchytit, zda zmíněná řešení problémů nepřinášejí jiné, návazné problémy, či překážky.“ Jelikož z vlastní praxe vím, že některá řešení konkrétního problému mohou způsobit jiný problém. Ale samozřejmě i takové řešení může být v určitých situacích efektivní.

5 VYHODNOCENÍ PRŮZKUMU

V této kapitole je vyhodnoceno dotazníkové šetření problémů s testovacími daty a možnosti jejich řešení. Toto dotazníkové šetření je podrobně popsáno v kapitole 4 *Průzkum nejčastějších problémů s testovacími daty a možností jejich řešení*. V kapitole 5.1 *Analýza odpovědí* představím metodu analýzy odpovědí na dotazníkové otázky a v dalších kapitolách jsou představeny výsledky z průzkumu. Tyto výsledky jsou v kapitole 5.5 *Závěry vyplývající z průzkumu* shrnuty a jsou vyhodnoceny cíle průzkumu.

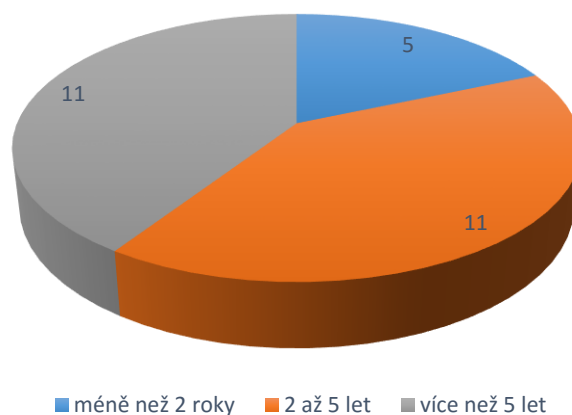
5.1 ANALÝZA ODPOVĚDÍ

Odpovědi na jednotlivé dotazníkové otázky představené v kapitole 4.3 *Dotazník a výběr otázek* jsou analyzovány a poté zpracovány graficky do koláčových nebo pruhových grafů, přehledných tabulek a poté okomentovány. Pro odpovědi na otázky s možností více odpovědí jsem zvolila pruhový graf, který je vhodný právě pro tento typ otázek. Pro ostatní uzavřené otázky jsem volila grafy koláčové, hlavně pro znázornění vlastností vzorku respondentů u otázek s jednou možností odpovědi. Pro shrnutí tyto údaje uvádím také v přehledných tabulkách. Všechny grafy, tabulky a závěry z toho vyplývající jsou poté zhodnoceny a okomentovány.

5.2 CHARAKTERISTIKA RESPONDENTŮ

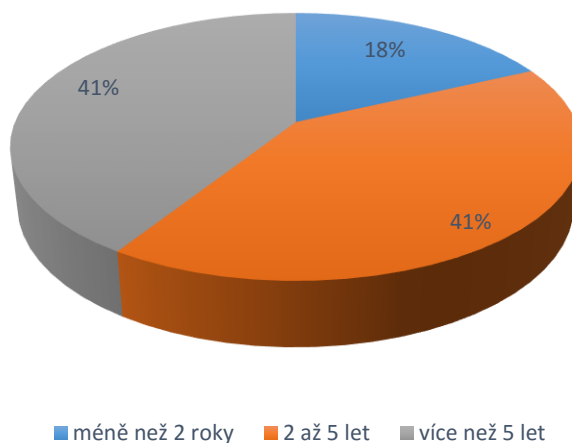
Se žádostí o vyplnění dotazníku pro účely této diplomové práce bylo osloveno celkem 50 respondentů z různých firem, kteří byli osloveni na základě naší předchozí spolupráce v předchozích pěti letech mé praxe v oblasti testování. Dotazníkového šetření se vyplněním dotazníku zúčastnilo celkem 27 respondentů, což dělá dotazníkovou návratnost 54 %. Otázky charakterizující respondenty byly v dotazníku položeny dvě. První otázka se dotazovala na dobu působení v oblasti testování. Druhá otázka se ptala na počet absolvovaných projektů v rámci testování. Na obě otázky odpovědělo 100 % respondentů. Rozdělení respondentů dle délky praxe v oblasti testování uvádím v absolutních hodnotách na Obrázku 8 i v procentech na Obrázku 9.

Rozdělení respondentů dle délky praxe oblasti testování



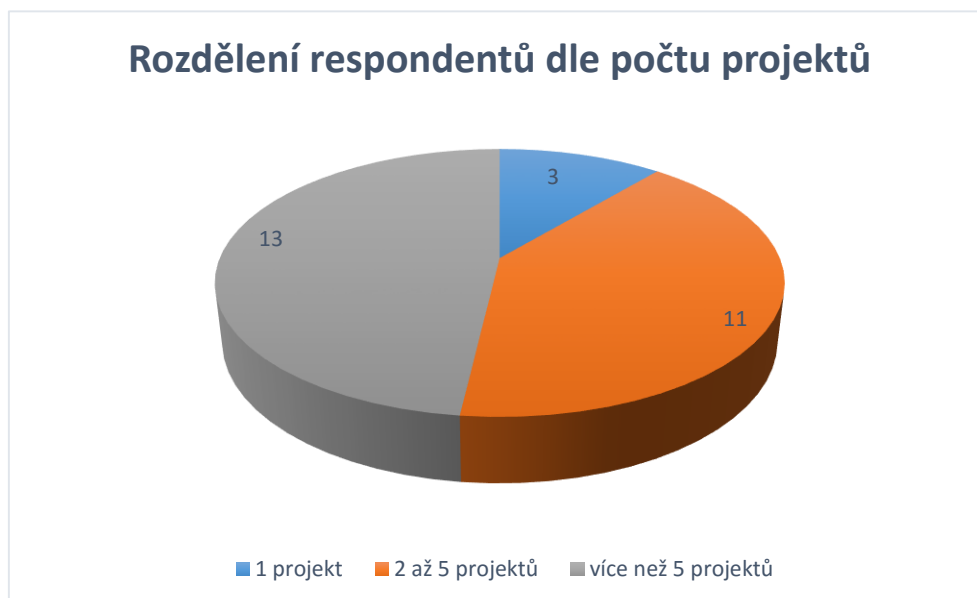
Obrázek 8: Graf rozdělení respondentů dle délky praxe v oblasti testování [Zdroj: autorka]

Rozdělení respondentů dle délky praxe oblasti testování (v %)

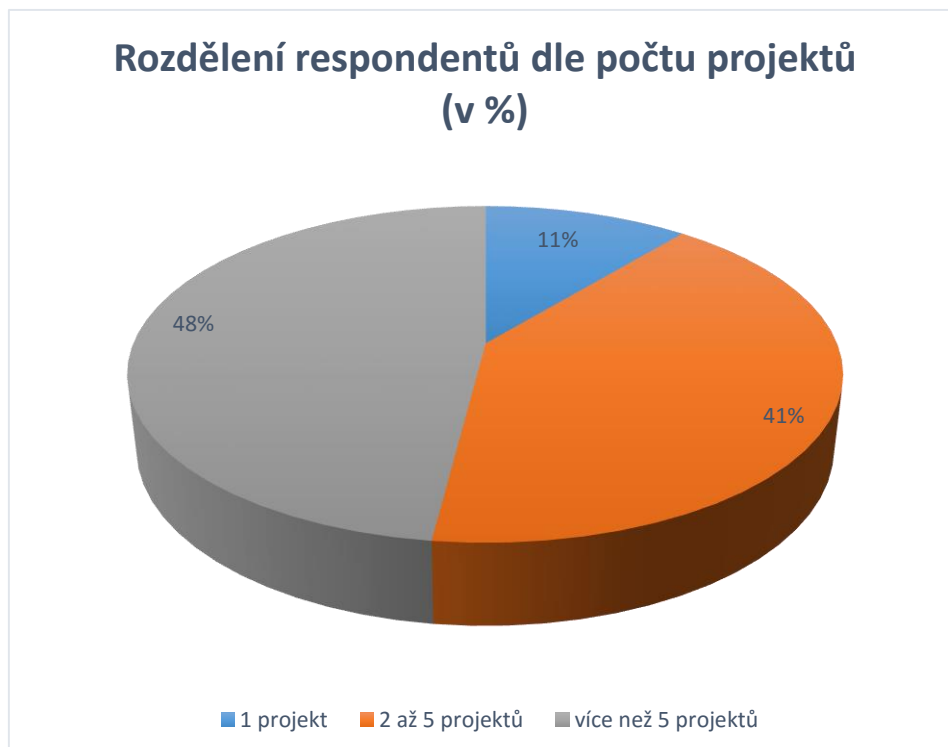


Obrázek 9: Graf rozdělení respondentů dle délky praxe v oblasti testování (v %) [Zdroj: autorka]

Rozdělení respondentů dle počtu absolvovaných projektů je zobrazeno v absolutních hodnotách na Obrázku 10 a v procentech na Obrázku 11.



Obrázek 10: Graf rozdělení respondentů dle počtu absolvovaných projektů v testování [Zdroj: autorka]



Obrázek 11: Graf rozdělení respondentů dle počtu absolvovaných projektů v testování (v %) [Zdroj: autorka]

Shrnutí vlastností vzorku respondentů uvádím níže v Tabulce 3 a Tabulce 4.

Tabulka 3: Doba praxe respondentů v oblasti testování [Zdroj: autorka]

Praxe v oblasti testování	Počet respondentů	Počet respondentů (v %)
<i>Méně než 2 roky</i>	5	18 %
<i>2 až 5 let</i>	11	41 %
<i>Více než 5 let</i>	11	41 %
Celkem	27	100 %

Tabulka 4: Počet absolvovaných projektů respondenty [Zdroj: autorka]

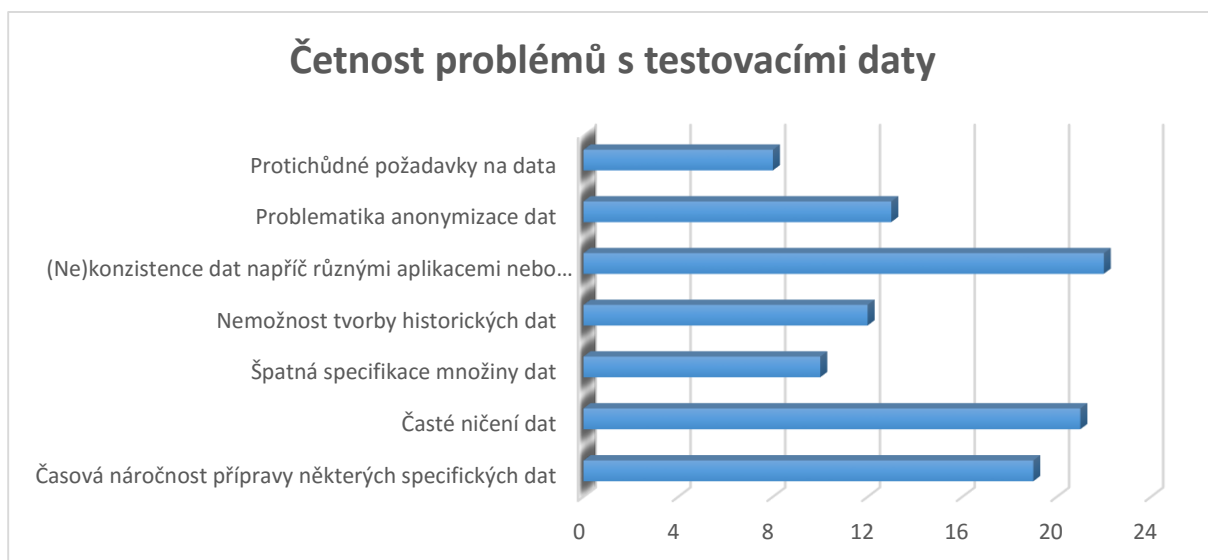
Počet absolvovaných projektů	Počet respondentů	Počet respondentů (v %)
<i>1 projekt</i>	3	11 %
<i>2 až 5 projektů</i>	11	41 %
<i>Více než 5 projektů</i>	13	48 %
Celkem	27	100 %

Z výše uvedených grafů a tabulek je patrné, že oslovený vzorek respondentů je velmi zkušený v oblasti testování. Celých 82 % respondentů pracuje v oblasti testování déle než dva roky, z toho polovina (41 % z celkového počtu) déle než pět let. Co se týče počtu absolvovaných projektů, tak téměř polovina respondentů (48 %) uvádí, že v testování pracovala na více než pěti projektech. Je tedy zřejmé, že data získaná dotazníkovým šetřením jsou získána od respondentů s mnohaletými zkušenostmi z velkého množství projektů, proto se jejich odpovědi dají považovat za komplexní.

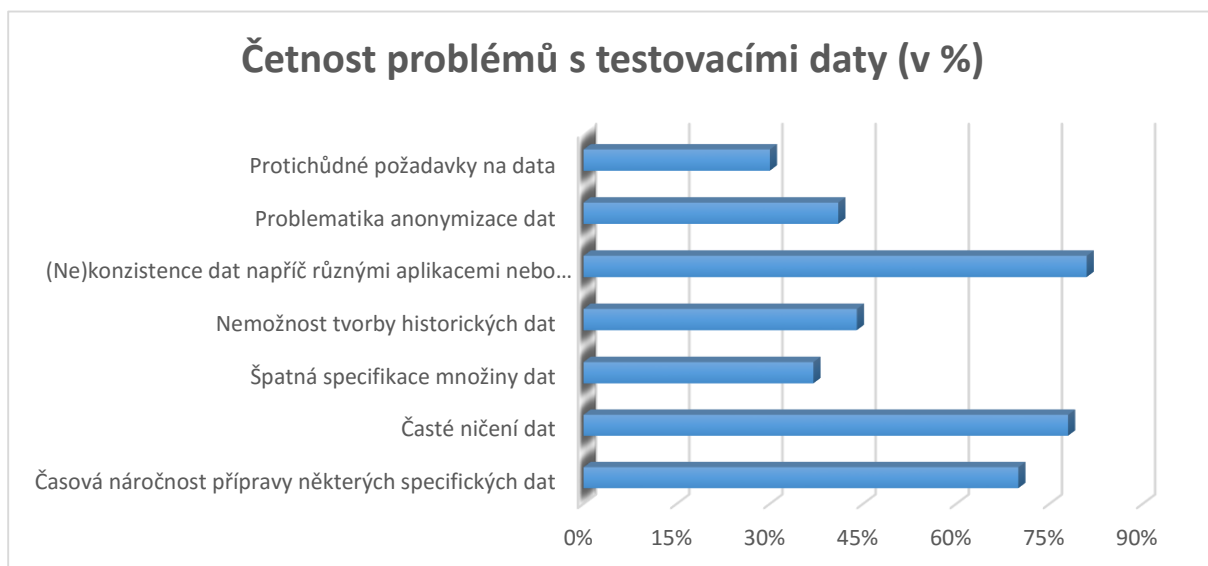
5.3 PROBLÉMY S TESTOVACÍMI DATY

Abych získala informace o tom, které z problémů s testovacími daty jsou ty nejčastější, použila jsem uzavřenou otázku s více možnostmi odpovědí. Z celkových 27 respondentů tuto otázku zodpovědělo všech 27, tedy 100 % všech zúčastněných respondentů. Seznam problémů s testovacími daty, na které jsem se dotazovala, jsem definovala v kapitole 4.2 *Metoda sběru dat* v Tabulce 1. Na Obrázku 12 je v pruhovém grafu graficky znázorněna četnost výskytu daných problémů s testovacími daty, tedy v kolika případech se respondenti s danými

problémy setkávají. Na Obrázku 13 je procentuální vyjádření, kolik procent respondentů se s danými problémy s testovacími daty setkává.



Obrázek 12: Graf četnosti výskytu problémů s testovacími daty [Zdroj: autorka]



Obrázek 13: Graf četnosti výskytu problémů s testovacími daty (v %) [Zdroj: autorka]

Výše zjištěné četnosti výskytu problémů s testovacími daty, jak v absolutních hodnotách, tak i v procentech uvádím dohromady Tabulce 5. V Tabulce 6 uvádím seznam problémů s testovacími daty seřazený dle četnosti výskytu.

Tabulka 5: Přehled četností výskytu jednotlivých problémů s testovacími daty [Zdroj: autorka]

Problém	Výskyt problému	Výskyt problému (v %)
<i>Protichůdné požadavky na data</i>	8	30 %
<i>Problematika anonymizace dat</i>	13	41 %
<i>(Ne)konzistence dat napříč různými aplikacemi nebo prostředím</i>	22	81 %
<i>Nemožnost tvorby historických dat</i>	12	44 %
<i>Špatná specifikace množiny dat</i>	10	37 %
<i>Časté ničení dat</i>	21	78 %
<i>Časová náročnost přípravy některých specifických dat</i>	19	70 %

Tabulka 6: Seznam problémů s testovacími daty dle četnosti výskytu [Zdroj: autorka]

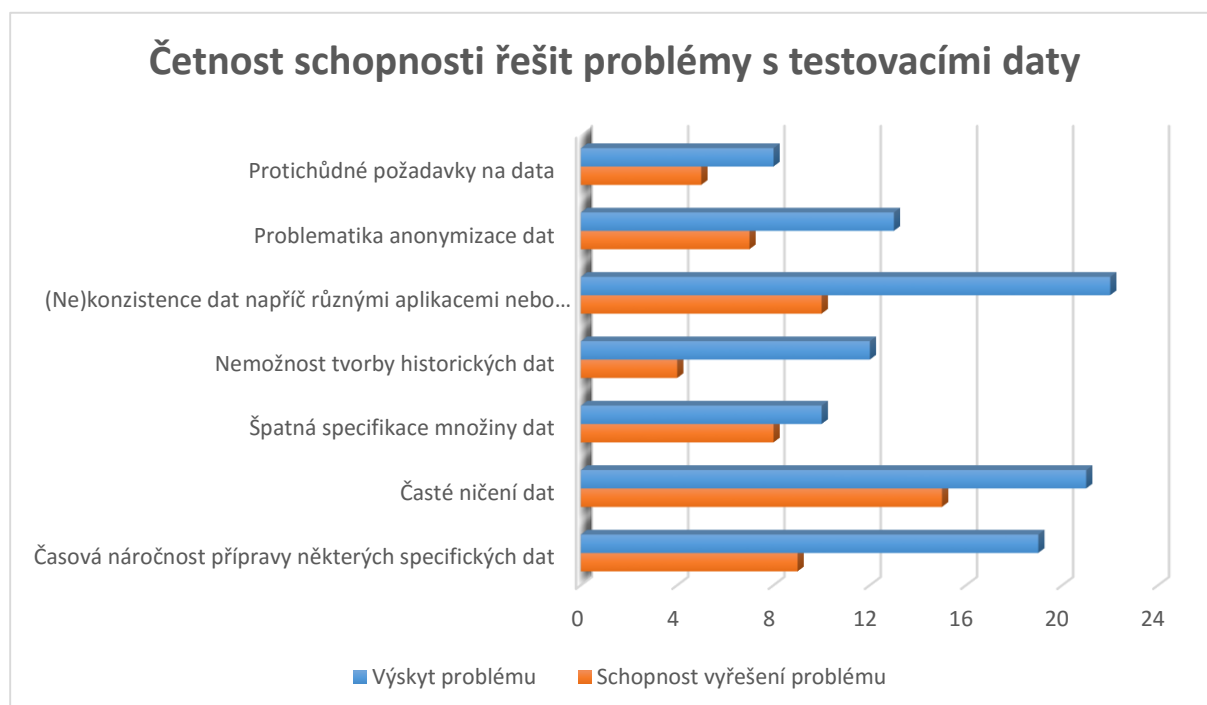
Problém	Výskyt problému (v %)
<i>(Ne)konzistence dat napříč různými aplikacemi nebo prostředím</i>	81 %
<i>Časté ničení dat</i>	78 %
<i>Časová náročnost přípravy některých specifických dat</i>	70 %
<i>Nemožnost tvorby historických dat</i>	44 %
<i>Problematika anonymizace dat</i>	41 %
<i>Špatná specifikace množiny dat</i>	37 %
<i>Protichůdné požadavky na data</i>	30 %

Na otázku výskytu problémů s testovacími daty navazovala čtvrtá otázka „Nalezl(a) jste způsob, jak předejít nebo vyřešit některý z výše uvedených problémů?“ Zde formou výběru ANO-NE měli respondenti označit u jednotlivých problémů, zda v praxi našli způsob, jak daným problémům předcházet nebo je alespoň eliminovat. Je logické, že ne všichni

respondenti u problémů, se kterými se setkávají, našli způsob jak danému problému předcházet. Výsledky odpovědí na tuto otázku zobrazuje Tabulka 7 a také jsem výsledky zanesla graficky do pruhového grafu (Obrázek 14), kde jsem počet odpovědí ANO u daného problému porovnávala s odpověďmi, že se respondenti s daným problémem setkávají.

Tabulka 7: Poměr výskytu problémů a schopnosti řešení problémů s testovacími daty (v %) [Zdroj: autorka]

Problém	Výskyt problému	Schopnost vyřešení problému	Poměr výskytu problému a schopnosti vyřešení problému (v %)
<i>Protichůdné požadavky na data</i>	8	5	62 %
<i>Problematika anonymizace dat</i>	13	7	54 %
<i>(Ne)konzistence dat napříč různými aplikacemi nebo prostředími</i>	22	10	45 %
<i>Nemožnost tvorby historických dat</i>	12	4	33 %
<i>Špatná specifikace množiny dat</i>	10	8	80 %
<i>Časté ničení dat</i>	21	15	71 %
<i>Časová náročnost přípravy některých specifických dat</i>	19	9	47 %



Obrázek 14: Poměr schopnosti řešit problémy s testovacími daty s jejich výskytem [Zdroj: autorka]

Z výše uvedené Tabulky 7 a grafu na Obrázku 14 jasně vyplývá, že četnost výskytu daného problému s testovacími daty nemusí být přímo úměrná jeho možnostem vyřešení a naopak méně častý problém může být mnohem náročnější na řešení, protože respondenti uvedli, že nejsou schopni ho efektivně vyřešit. Například jako druhý nejčastější problém s testovacími daty respondenti uváděli časté ničení dat. Ale tento problém jsou schopni v 71 % případů vyřešit, zatímco problém tvorby historických dat je problém s nízkou četností výskytu, ale pokud nastane, pak jsou schopni ho vyřešit pouze v 33 % případů. Nicméně (ne)konzistence dat napříč různými aplikacemi a prostředími je nejčastější problém (81 %), se kterým se pracovníci v oblasti testování setkávají, ale umí ho řešit jen ve 45 % případů, což z něj dělá problém s druhou nejmenší schopností vyřešení problému. Jen o trochu lépe je na tom časová náročnost přípravy specifických dat, která je třetím nejčastějším problémem (70 %) a zároveň třetím nejzávažnějším (47 %).

Definují-li si schopnost vyřešení problému s testovacími daty jako poměr existence řešení problému s testovacími daty s jejich výskytem, pak čím nižší je tento poměr, tím vyšší je schopnost vyřešení problému. Po seřazení těchto problémů dle této schopnosti vyřešení problému dostaneme seznam zobrazený v Tabulce 8.

Tabulka 8: Seznam problémů s testovacími daty dle schopnosti vyřešení problému [Zdroj: autorka]

Problém	Poměr výskytu problému a existence řešení problému (v %)
<i>Nemožnost tvorby historických dat</i>	33%
<i>(Ne)konzistence dat napříč různými aplikacemi nebo prostředími</i>	45%
<i>Časová náročnost přípravy některých specifických dat</i>	47%
<i>Problematika anonymizace dat</i>	54%
<i>Protichůdné požadavky na data</i>	62%
<i>Časté ničení dat</i>	71%
<i>Špatná specifikace množiny dat</i>	80%

Z tabulky je patrné, že ze všech sedmi problémů mají tři problémy s nejnižší schopností vyřešení problému vyhodnocenou schopnost vyřešení menší než poloviční. Zbylé čtyři problémy mají sice schopnost vyřešení větší než 50 %, ale pouze jeden, a to špatná specifikace dat má vyšší schopnost vyřešení než třičtvrtinovou, konkrétně 80 %.

5.4 MOŽNOSTI ŘEŠENÍ IDENTIFIKOVANÝCH PROBLÉMŮ

V této kapitole analyzuji odpovědi na poslední dvě otázky z dotazníku. Tyto otázky byly otevřené a nepovinné. Vzhledem k tomu, že byly nepovinné, tak na ně odpovědělo 25 z celkového počtu respondentů, což je 93% respondentů. Způsoby řešení výše definovaných problémů s testovacími daty se budu zabývat postupně, vždy ke každému problému zvlášť, a to v pořadí dle četnosti výskytu (viz Tabulka 6) od nejčastěji se vyskytujících problémů s testovacími daty. U každého řešení uvedu i návazné problémy, které toto řešení přináší, byly-li nějaké návazné problémy identifikovány. K identifikování jednotlivých řešení použiji přímé citace z odpovědí respondentů dotazníkového šetření.

A. (Ne)konzistence dat napříč různými aplikacemi nebo prostředími

U odpovědí na otázku ohledně nekonzistence dat napříč různými aplikacemi nebo prostředími byly zmíněny jak možnosti řešení nastalého problému, tak i možnosti předcházení tomuto problému. Co se týče předcházení nekonzistence dat, tak cituji z dotazníku: „někdy se na takové chyby přijde až na ostrém provozu. Ať už na testu, či na LIVE, je nutné zanalyzovat příčiny nekonzistence dat, upozornit na to vlastníky dotčených systémů a společně najít takové řešení, které by zamezilo vytváření takovýchto dat.“ Možnost domluvy s vývojáři aplikací a snahu o sjednocení rozdílů, jako předcházení tomuto problému zmínili tři respondenti, nicméně bylo dále poznamenáno, že toto řešení lze jednodušeji uplatnit ve firmách s malým množstvím interních aplikací, protože u větších firem je takový stav neudržitelný.

Dále bylo v dotazníku k problému nekonzistence dat napříč aplikacemi a prostředími zmíněno několik možností řešení tohoto problému, pokud již nastane. První možností je ruční oprava dat. To s sebou ale přináší vysoké požadavky na zdroje, a to jak lidské, tak časové. Další možností řešení je manuální tvorba dat na front-endu (tedy té části aplikace, kterou vidí

koncový uživatel), což by mělo přinést větší konzistenci dat, než tvorba dat v databázi (insert dat do databáze). Respondenti tuto možnost zmínili čtyřikrát a to: „Snažím se založit si data pomocí front-endů existujících aplikací – tím je značná kompatibilita založena. Pokud to není možné, pak je potřeba důkladná příprava pro inserty dat.“, „Nekonzistenci dat napříč celým prostředím je možné řešit buďto čistou (kompletní) kopií provozních dat nebo vytváření testovacích dat uměle z front-endů.“ a „Tvorba vlastních data, u kterých jsem si jist, že budou konzistentní.“. Všichni respondenti zmiňující tento způsob řešení problému s nekonzistentností dat však jako vedlejší problém zmínili prodražení projektu a náročnost na existující aplikace. Jak bylo výše zmíněno, lze nekonzistenci dat předcházet také kopií produkčních (provozních) dat na testovací prostředí. Tuto možnost zmínili dva respondenti z celkových deseti zmíněných řešení daného problému. Jeden respondent navrhl toto řešení jako „nalítí všech dat z produkčního prostředí, a tam kde není možné nahrát celý objem databáze je do online data storu dodán seznam např. RČ, aby se vybírala pouze konzistentní data.“. Dvakrát bylo také vzpomenueno na použití automatických testů pro tvorbu dat, protože „automatické testy si vždy potřebná data (pokud lze) dopředu připraví a pak po sobě uklidí, nespolehají na to, že na prostředí něco je. Pro to ale bylo nutné mít vytváření potřebných dat pečlivé a úklid po testu udělat maximálně spolehlivě.“ Posledním zmiňovaným řešením nekonzistence dat byl „manuální zásah v daném systému pro sjednocení dat. To však není pohodlné pro velké množství dat. Navíc to přináší riziko dalšího (i nevratného) poškození dat nestandardním zásahem, následné narovnání je velice složité a zvládne je udělat jen několik hodně vytížených lidí.“, „Jsou-li data nekonzistentní, ale jsou pro naše testy nejvhodnější, tak je možné si data narovnat\opravit zásahem do samotné databáze, ovšem ví-li tester co dělá a uvědomuje si všechny možné vedlejší dopady na další dotčené systémy. Nemá-li ovšem tester takovou znalost daného databázového schématu, je lepší zničená data odepsat a vytvořit si jiná.“

Nakonec zmíním jeden velmi specifický příklad nekonzistence testovacích dat, která „byla způsobena nesouladem časových období, za která byla připravována data pro jednotlivé aplikace v testovacím prostředí. Řešením byla obnova testovacího prostředí ze záloh jednotlivých aplikací ke stejnému datu. To však přineslo prodloužení doby testování.“

B. Časté ničení dat

Co se týče řešení problému častého ničení testovacích dat, byly odpovědi až na pár výjimek poměrně jednoznačné. Řešení byla dvě. Zaprvé „příprava velkého množství (základních) testovacích dat nejlépe pro každý release, specifická konkrétní data nechat založit s množstevními rezervami. Když se počítá každý release se zakládáním dat, je pak větší šance, že se vytvoří i solidní historie a tím pádem variabilita historických dat. V opačném případě se budou data (klienti, produkty) stále přepoužívat až nakonec nezbyde z dat nic.“ a s tím související vlastnictví dat jednotlivými testery. Toto nejčastější řešení bylo respondenty zmíněno šestkrát v různých obměnách jako „Řízená distribuce testovacích dat mezi analytiky a testery a přehledové tabulky s evidencí informací o testovacích dat včetně přidělených osob. Ani tento přístup však není 100%. Vyžaduje vyšší režie.“, „podařilo se většinou vyřešit všestrannou dohodu mezi testery. Každý tester si eviduje a používá jen svá data. Před použitím jiných dat se nejprve dotáže jejich vlastníka.“, „Gentlemanské dohody mezi testery o nepoužívání cizích dat. Svá data si pojmenujeme podle projektového klíče, aby bylo jasné, že jsou naše.“, „Tvorba více stejných dat a domluva o nepoužívání jiných než uvolněných dat k dalšímu použití. Nicméně potřeba tvorby více dat vyžaduje vyšší náklady na zdroje.“, „Každý tester má svou sadu dat, kterou si sám vytváří a spravuje. Nevýhodou je možná duplicita testovacích dat, ale většinou si každý tester připraví data různým způsobem, což napomáhá pokrýt větší množinu možností reálných dat. To však není možné využít u každého testování. Záleží od aplikace, akou variaci vstupních dat umožňuje.“.

Byl zmíněn i návrh na „Vytvoření procesu/nástroje na sdílení informací o „majiteli“ daného testovacího data. Např. pokud si vytvořím (vyhledám) konkrétního vhodného testovacího klienta v testovacích datech, tak si jej např. „zarezeruju“ v rezervačním systému. Testeři tento systém využívají rádi, protože sami nechtějí být obětí „zcizení/zneužití“ jejich opečovávaných dat. Nebo v případě tvorby vlastního klienta synteticky si např. jméno toho klienta nazvu dle svého kódového označení.“ a jeden respondent se vyjádřil ke sdílení testovacích dat velmi striktně: „Nesdílím testovací data s ostatními testery, pokud to není nezbytně nutné, raději jim sama ukážu jak data vytvořit pokud to situace umožňuje. Pokud přeci jen dojde ke sdílení, dohlížím na kolegovu práci a po skončení užívání těchto dat mu dále neumožňuji je využívat a ani sdílet (např. změním heslo). Někdy toto řešení vede

k nesamostatnosti kolegů a jejich časté obracení na mou osobu s dojmem, že zásobování daty je mou pracovní náplní a povinností.“

Jako poslední byly jednou opět zmíněny automatické testy pro tvorbu testovacích dat. Data pro automatické testy jsou vždy vytvořena na začátku testu přímo automatem, proto se tato data neničí, protože se vždy vytvoří nová. „Automatické testy používají jiný účet než pro manuální testování, většinou sdílení účtu a dat mezi testery nevadí. Ale vždy bylo potřeba i spolupracovat s vývojáři, aby byl systém odolný na vícenásobné přihlášení a např. při testování e-shopu si testeři mezi sebou nekradli obsah košíku apod.“

Dá se tedy říci, že ničení dat se dá vcelku efektivně předcházet ať rezervací dat testery, řízenou distribucí dat mezi testery, tvorbou soukromých testovacích dat jednotlivými testery či tvorbou dat automatickými testy, nicméně všechna tato řešení přinesou prodloužení doby přípravy testovacích dat a vyšší náklady na zdroje a to jak lidské, tak časové. Taková zvýšení nákladů však často nebývají schválena, jako neopodstatněná.

C. Časová náročnost přípravy některých specifických dat

Časová náročnost přípravy některých specifických dat byla v dotazníkovém šetření zmiňována, jako třetí nejčastější problém také je třetím problémem s nejnižší schopností vyřešení, který dle výsledků šetření jsou schopni řešit v méně než polovině případů (47 %). Z těch respondentů, kteří tento problém řešit umí, pouze dvě odpovědi zmínily způsob předcházení tomuto problému dříve, než vůbec nastane. První cestou je „řešit tento problém už ve fázi estimací (fázi odhadů zdrojů na projekt, pozn. autorky). Případně, jakmile zjistím potřebu další náročnosti, informuju projektového manažera, aby si alokoval potřebné zdroje na přípravu. To ale může způsobit prodražení projektu, případně určitou naštvanost v týmu.“. Druhý způsob je tvorba dat do zálohy, cituji: „Máme k dispozici automat, který se neustále vyvíjí. Tento nástroj nám ušetří práci zakládáním různých typů dat - klientů, produktů apod. Problém však nastane, změní-li se proces resp. kód aplikace, kterou automat používá. V takovém případě výsledky automatu nejsou uspokojivé, ne-li žádné. Stává se tak celkem čas, a jelikož o vyvíjené funkcionalitě či opravě nemáme často ani povědomí, tak obvykle přicházíme na nutnost přizpůsobit danému vývoji i automat až těsně před samotným provedením testů. V takovém případě si nezbytnou část dat vytvoříme ručně a zbytek

automatem, po jeho úpravě.“ Částečná automatizace však je velmi časově náročná, obzvláště, je-li třeba automat opravovat.

Dále byly v dotaznících zmiňovány řešení tohoto problému, když je tedy data nutné vytvořit a je jisté, že to zabere mnoho času. Proto jednou z možností je použití dat z produkce, které je třeba anonymizovat. „Řešením je využití anonymizovaných dat z produkce a tvorba falešných dat (tzv. fake data) dat na úrovni rozhraní mezi aplikacemi. Tvorba falešných dat může způsobit další problémy kvůli různé konvergenci mezi systémy. Architektura IT větších společností je velice komplexní a všechny dopady není možné mnohdy obsáhnout.“ „Pokud identifikuju nutnost tvorby testovacích dat, jejichž tvorba či příprava zabere velké množství času, tak volím mezi kopii produkce nebo sice bezpečnější, ale delší variantou syntetické tvorby dat. Rozhodnutí je nutné vždy dělat společně s vyšším managementem (z důvodu uvědomění si rizik úniku provozních dat).“

Jako poslední varianta řešení tohoto problému bylo zmíněno přímé vkládání dat do databáze, udržování dlouho běžící instance, která akumuluje data a hlavně specifické generátory dat. „Časovou náročnost je možné zčásti řešit pomocí generátorů dat volně dostupných na Internetu, a dále pak ručním zpracováním – ideálním nástrojem je MS Excel a vzorce s VLOOKUP – ale to je jen částečné řešení. Případně lze použít testovací nástroje, které nutné datové sady generují ze vzorových běhů (např. Tricentis), ty však řeší jen datasety (množiny dat, pozn. autorky) na straně „uživatelé“, nikoliv testovací data v systému.“

D. Nemožnost tvorby historických dat

Tento problém „lze řešit opět kopií provozních dat, zde do určité úrovně je možné se spokojit s nekonzistencí dat při anonymizaci, popř. vytvořit speciální nástroje na úpravu historických dat, či nápravu v případě využití a znehodnocení dat testovacím scénářem“. „Test data se nenaklikávají „ručně“ (simulace reálného chování uživatele), ale některé dílčí parametry testovacích dat se upraví v jiných systémech (back-endy, databáze, jiný způsob ohnutí). Největší překážkou je bezpečnostní politika. Díky ní většinou není možné přepsat údaje v databázi, nicméně na některých prostředích to lze. Takže někdy toto řešení zabere, někdy nikoliv.“

Druhou možností je prohledáním databáze nebo zásahem do databáze. To ale přináší další návazné problémy. Při řešení „přímým vytvořením dat v databázi (šlo o vytvoření transakční historie) ve finále data stejně stoprocentně neodpovídala potřebám“. „Prohledání databáze a nalezení vhodného klienta a unifikace dat, aby byla data ve všech systémech, přináší riziko dalšího (i nevratného) poškození dat nestandardním zásahem. Následné narovnání je velice složité a zvládne je udělat jen několik hodně vytížených lidí. Navíc unifikace dat není vždy spolehlivá.“

E. Problematika anonymizace dat

Problematika anonymizace dat z produkčního prostředí je velkou bezpečnostní otázkou. Navíc je tento postup získávání dat velmi náročný a často způsobuje nekonzistenci dat napříč prostředími a aplikacemi, jak zmínilo několik respondentů. Nejlepší nástin na tuto problematiku byl popsán v dotazníku takto: „Čím větší je integrace systémů a dat, tím nižší je šance na uspokojivé vyřešení anonymizace dat. Anonymizaci má smysl dělat jen pro určité snížení rizika zneužití dat. Plně předejít toto riziko anonymizací často nelze. My jsme tento problém vyřešili kombinací několika faktorů, kopií a pouze částečnou anonymizací produkčních dat, vytvořením nástroje na tvorbu syntetických dat a úpravu/simulaci dat historických. Hlavní problém je ale vždy rozhodnutí o využití provozních dat, jelikož se jedná o velmi vysoké bezpečnostní riziko. Toto riziko je vždy nutné nějak ošetřit (možností je několik). V případě anonymizace dat je nutné vždy řešit stránku nákladů na vytvoření nástroje na anonymizaci dat oproti rizikům z využití dat provozních, popř. rizikům z nedostatečné otestovanosti SW (díky nekvalitním datům).“

„Anonymizace dat je nejlépe řešitelná integrací testera mezi důvěryhodné osoby (podepsání NDA – dohodu o mlčenlivosti) a následné používání omezené produkční sady dat. Používané na 2 projektech v různých bankovních institucích. Nakonec různé projekty na anonymizaci dat se ukázaly být příliš pracné a proto nevhodné nebo dokonce (v případě více druhů databázových prostředí) jako nerealizovatelné, protože vždy vedly k nekonzistenci dat. Anonymizace dat je v zásadě schůdná pouze v monolitickém systému s jednou databází.“

Je však vždy nutné „nastavení striktního oprávnění do systémů a zamezení přesunu produkčních obchodních dat do jiných prostředí. To zvyšuje náročnost oprav chyb v systémech, způsobuje větší rozdíly mezi prostředími a k tomu nutné navyšování rozpočtu“.

Jeden respondent uvedl, že v jeho zaměstnání si nechali dodat automatizovaný anonymizační nástroj pro přenos dat z produkčního do akceptačního prostředí. Bohužel toto řešení není univerzální, funguje jen pro tento konkrétní typ aplikace.

F. Špatná specifikace množiny dat

Tento problém byl jeden z nejméně se vyskytujícím v dotazníkovém šetření a má 80% úspěšnost řešení. Nejčastější odpovědí pro předcházení tomuto problému bylo zlepšení komunikace („Zlepšení komunikace s oddělením podpory pro zákazníky.“, „Včasná identifikace množiny ve spolupráci s analytiky“) a „důkladná příprava testovací strategie, ze které bylo jasné, jak, kdo a co bude testovat. Díky strategii byla data vyspecifikována a připraveno bylo dostatek dat i s rezervou“. Také je doporučováno „pečlivě číst business\sysémovou\procesní specifikaci a najdeme-li nějakou nesrovnalost, upozornit na to vlastníka analýzy“. Po upozornění vlastníka analýzy pak pravděpodobně dojde k úpravě původní specifikace a vytvoření nové sady dat. To však je časově náročné a může způsobit nedodržení termínů dodávky projektu.

„Tak jak neexistuje 100% odhad pracnosti, tak neexistuje 100% specifikace jakých variací a množství testovacích dat bude potřeba. Před testy nevíme, zdali daný testovací scénář budeme nuceni spouštět jednou nebo desetkrát, to závisí na kvalitě dosavadního vývoje. Tudíž je vždy nutné se připravit na horší varianty, tzn. kopie z produkce či anonymizace dat ve větším množství než stávající testovací scénáře vyžadují, nebo vytvoření nástroje na rychlou tvorbu syntetických testovacích dat.“

G. Protichůdné požadavky na data

Jednoznačnou odpovědí z dotazníků na problematiku protichůdných požadavků na data je komunikace, doslova „Ujasňování, upřesňování zadání, komunikace se všemi, koho se to týká“. Je potřeba „vyjasnenie požiadaviek na testovacia dáta s test leadrom a ich presné vymedzenie podľa analýzy. To je náročnejšie na čas, nie vždy je to možné.“ Tato komunikace,

ujasňování a upřesňování zadání ale zabírá čas a je zde riziko překročení odhadů na práci a překročení rozpočtu. Pokud však selže snaha o narovnání protichůdných požadavků, pak byla zmíněna možnost řešit tento problém takto: „Pokud to jde, použiju dvě sady dat – každá sada testuje jeden aspekt testů. Následně se snažím najít průnik, abych mohl testovat na datech co nejblíže produkčním.“.

5.5 ZÁVĚRY VYPLÝVAJÍCÍ Z PRŮZKUMU

V této kapitole znovu shrnu závěry, které vyplynuly z provedeného kvalitativního průzkumu a dotazníkového šetření a vyhodnotím cíle, které jsem si pro tento průzkum a dotazníkové šetření stanovila v kapitole 4.1 *Cíle průzkumu*.

Kvalitativního průzkumu na 23. setkání [pro]Test!u, pro identifikování problémových oblastí testovacích dat se zúčastnilo 28 účastníků. Tento průzkum předcházel dotazníkovému šetření, kterého se zúčastnilo 27 respondentů, jiných než předcházejícího průzkumu. Tento průzkum je blíže popsán v kapitole 4.2 *Metoda sběru dat* a Příloze 2. Na základě tohoto průzkumu se podařilo identifikovat sedm hlavních problémových oblastí. Těmito oblastmi jsou:

- Protichůdné požadavky na data
- Problematika anonymizace dat
- (Ne)konzistence dat napříč různými aplikacemi nebo prostředími
- Nemožnost tvorby historických dat (například starších typů produktů)
- Špatná specifikace množiny dat
- Časté ničení dat (malým používáním, sdílením dat mezi testery atp.)
- Časová náročnost přípravy některých specifických dat (dlouhodobá data nebo data zpětně)

Tyto oblasti problémů s testovacími daty byly poté dále dotazovány v dotazníkovém šetření, které je popsáno v kapitole 4.2 *Metoda sběru dat* a 4.3 *Dotazník a výběr otázek*. Cílem bylo vyhodnotit a zanalyzovat, se kterými problémy s testovacími daty se pracovníci v oblasti testování setkávají nejčastěji. Výsledky jsou podrobně popsány v kapitole 5.3 *Problémy*

s *testovacími daty* a znázorněny v grafech na Obrázku 12, Obrázku 13 a Obrázku 14 a v Tabulce 5, Tabulce 6 a Tabulce 7.

Tři nejčastější problémy s testovacími daty, se kterými se respondenti setkávají, jsou tyto:

- (Ne)konzistence dat napříč různými aplikacemi nebo prostředími (81 % respondentů)
- Časté ničení dat (78 %)
- Časová náročnost přípravy některých specifických dat (70 %)

Dle schopnosti vyřešení problému, pak tři s nejnižší schopností vyřešení jsou:

- Nemožnost tvorby historických dat (zjištěná schopnost vyřešení problému je 33 %)
- (Ne)konzistence dat napříč různými aplikacemi nebo prostředími (45 %)
- Časová náročnost přípravy některých specifických dat (47 %)

U všech ostatních problémů byla zjištěná schopnost řešení problému nad 50 %.

V rámci zjišťování možností řešení problémů respondenti uvedli u každého problému několik možných způsobů řešení, která jsou popsána podrobně v kapitole 5.4 *Možnosti řešení identifikovaných problémů*. Obecným závěrem skrze všechny identifikované problémy je, že jakmile nastane jakýkoliv z výše uvedených problémů, pak řešení daného problému bývá většinou náročné na lidské, časové i finanční zdroje, což respondenti zmínili u většiny řešení identifikovaných problémů jako návazný problém k danému řešení. Možnost předcházení vzniku problému byla respondenty zmíněna ve třech případech a to u špatné specifikace množiny dat, časové náročnosti přípravy některých specifických dat a (ne)konzistence dat napříč různými aplikacemi nebo prostředími. U ostatních problémů byla navržena různá řešení. Největší návazné problémy byly identifikovány u řešení anonymizace dat, kde se jedná především o bezpečnostní otázku a vysokou časovou náročnost samotné anonymizace, která nemusí být právě z důvodu náročnosti na zdroje efektivní.

Nakonec zmíním dvě zásady, které je dobré dodržovat:

1. Provést kvalitní úvodní analýzu požadavků zákazníka a cílů projektu a také zmapovat dobře technicky situaci u zákazníka, a to ještě předtím, než se finálně stanoví rozsah projektu a jeho cena. Přizvat testera k tvorbě požadavků a podmínek projektu. Praktické je i u designu řešení již mít jasno v tom, jak se funkčnost bude testovat a jak se budou simulovat „krizové stavy“. Je nutné vyčlenit dostatečný prostor pro testy a v případě nejasností zahrnout dostatečnou rezervu. Toto pomáhá předcházet problémům, či pomáhá řešit problémy - s protichůdnými požadavky na data, problémy s anonymizací dat a špatnou specifikací množiny dat.
2. Zajistit přípravu a údržbu samostatných testovacích prostředí (nejlepší je oddělit je od vývojových prostředí a také od testovacího prostředí klienta), kde budou dlouhodobě konzistentní data a bude jasně dohodnutý mechanismus jejich vzniku, údržby a aktualizace i zálohování. Toto pomáhá zmenšit riziko častého ničení dat.

Na základě kvalitativního průzkumu i dotazníkového šetření se mi podařilo zodpovědět na dotazníkové otázky, formulovat závěry a splnit tak cíle dotazníkového šetření. Díky těmto výsledkům a závěrům se mi nejen podařilo splnit dílčí cíle práce „identifikovat jednotlivé oblasti problémů s testovacími daty“, „realizovat a vyhodnotit průzkum ohledně problémů s testovacími daty“, „vyhodnotit, se kterými problémy se pracovníci setkávají nejčastěji“ a „zjistit možnosti jejich řešení, které se uplatňují v praxi v České republice“, ale zároveň se mi podařilo provést první průzkum na toto téma v České republice, vyhodnotit nejčastější problémy s testovacími daty a shrnout různé způsoby řešení identifikovaných problémů v České republice. Výsledky a závěry tohoto dotazníkového šetření rozšiřují obraz současného stavu testování v České republice tak, jak bylo popsáno v kapitole 2 *Charakteristika současného stavu problémové oblasti* a přináší nám tak ucelenější pohled na celou problematiku testování a to včetně testovacích dat.

6 ZÁVĚR

V této diplomové práci jsem se zabývala testováním, testovacími daty, nejčastějšími problémy s testovacími daty a možnostmi jejich řešení.

Nejprve jsem v kapitole 2 *Charakteristika současného stavu problémové oblasti* na základě nedávného průzkumu popsala stav testování v České republice v porovnání se zahraničím. V kapitole 2.1 *Rešerše prací* jsem provedla rešerši českých i zahraničních prací s podobným tématem a dále prací i elektronických zdrojů o testování a testovacích datech. Pak jsem v kapitole 3 *Testování a testovací data* a jejích podkapitolách definovala pojmy testování, testovací data, test data management a zaměřila se na kategorizaci testování podle typů, úrovně a způsobu testování a na kategorizaci testovacích dat. Také jsem zde představila rozdíly mezi manuálním a automatizovaným testováním.

V kapitole 4 *Průzkum nejčastějších problémů s testovacími daty a možností jejich řešení* jsem si nejprve stanovila cíle prováděného průzkumu. Dále jsem zde představila použité metody sběru dat- kvalitativní výzkum formou diskuze a dotazníkové šetření. Poté jsem v kapitole 4.3 *Dotazník a výběr otázek* představila samotný dotazník a popsala výběr jednotlivých otázek v dotazníku a jejich návaznost na stanovené cíle této diplomové práce.

Kapitola 5 *Vyhodnocení průzkumu* je praktickou částí této diplomové práce. Zde jsem popsala metodu analýzy jednotlivých odpovědí na dotazníkové otázky. Analýzu odpovědí na dotazníkové otázky jsem si rozdělila do tří částí. Nejprve jsem charakterizovala vzorek respondentů, kteří zodpověděli dotazník, a to dle délky praxe v oboru testování a dle počtu projektů, kterých se v rámci testování účastnili. Dále jsem se věnovala nejčastějším problémům s testovacími daty a to jak z pohledu četnosti výskytu a závažnosti těchto problémů, tak také jestli dané problémy mají nějaká řešení. V poslední fázi jsem zpracovávala slovní odpovědi respondentů na otevřené otázky týkající se právě způsobů řešení identifikovaných problémů s testovacími daty a případných návazných problémů, které dané řešení může způsobovat. V závěru páté kapitoly jsem shrnula výsledky dotazníkového šetření a vyhodnotila splnění cílů průzkumu, které jsem si stanovila.

Jako hlavní cíl této diplomové práce jsem si stanovila definovat, identifikovat a kategorizovat nejčastější problémy s testovacími daty, se kterými se dnes a denně testeři, test analytici a ostatní pracovníci v oblasti testování setkávají a zjistit možnosti jejich řešení, které se uplatňují v praxi v České republice. Pro dosažení tohoto cíle práce jsem si dále definovala tyto dílčí cíle:

- definovat a kategorizovat testování jako samostatnou oblast
- definovat a kategorizovat samotná testovací data
- identifikovat jednotlivé oblasti problémů s testovacími daty
- realizovat a vyhodnotit průzkum ohledně problémů s testovacími daty
- vyhodnotit, se kterými problémy se pracovníci setkávají nejčastěji

Testování a testovací data jsem ve třetí kapitole definovala a kategorizovala pomocí rešerší českých i zahraničních tištěných i elektronických zdrojů. Rešerši i se zdroji jsem popsala v kapitole *2.1 Rešerše prací*.

Jednotlivé oblasti problémů s testovacími daty jsem identifikovala pomocí kvalitativního průzkumu na 23. setkání [pro]Test!u v Praze, jehož téma bylo Test data management. Tyto identifikované oblasti problémů s testovacími daty jsem dále použila při tvorbě dotazníku pro splnění dílčího cíle realizace a vyhodnocení průzkumu problémů s testovacími daty. Tento dotazník (viz Příloha 1) jsem tvořila v souladu s cíli této diplomové práce. Návratnost dotazníku byla přibližně 50 %.

Nakonec jsem data získaná dotazníkovým šetřením zpracovala a vyhodnotila. Vyhodnocení průzkumu je podrobně popsáno v kapitole *5 Vyhodnocení průzkumu*. V rámci průzkumu jsem si také stanovila cíle průzkumu, které jsem provedeným průzkumem splnila a závěry popsala v kapitole *5.5 Závěry vyplývající z průzkumu*. Zde jsem popsala výsledky šetření, se kterými problémy s testovacími daty se testeři, test analytici, test manažeři a další pracovníci v oblasti testování setkávají nejčastěji. Také jsem zpracovala a vyhodnotila možnosti řešení nebo předcházení identifikovaných problémů s testovacími daty a případné další návazné problémy, které tato řešení přinášejí.

Realizováním všech dílčích cílů jsem splnila i hlavní cíl této diplomové práce, který jsem si stanovila v úvodu této diplomové práce.

Přínosem této diplomové práce je ucelený pohled na testování, testovací data a jejich význam v celé oblasti testování a na překážky, se kterými se pracovníci v oblasti testování setkávají. Dále tato práce přináší shrnutí možností řešení identifikovaných problémů s testovacími daty a různé způsoby, jak těmto problémům předcházet případně je řešit. Vzhledem k tomu, že jsem při řádné rešerši neobjevila žádnou práci zabývající se testovacími daty z pohledu problémů a překážek, tato práce přináší komplexní pohled na danou problematiku.

7 SEZNAM LITERATURY

ABOWD, John M. a Julia LANE. New approaches to confidentiality protection: synthetic data, remote access and research data centers. In: *Privacy in statistical databases* [online]. Berlin: Springer, 2004, s. 282-289 [cit. 2015-10-25]. ISBN 978-3-540-25955-8. Dostupné z http://link.springer.com/chapter/10.1007/978-3-540-25955-8_22

AMMANN, Paul a Jeff OFFUTT. *Introduction to software testing*. New York: Cambridge University Press, 2008, xxii, 322 s. ISBN 0521880386.

BACH, James. Testing and checking refined. In: *James Bach's blog: creator of rapid software testing* [online]. 26. 3. 2013 [cit. 2015-10-25]. Dostupné z: <http://www.satisfice.com/blog/archives/856>

BOROVCOVÁ, Anna. Kvalita a testování v České republice. *Systémová integrace*. 2011, roč. 18, č. 1, s. 93-108. ISSN 1210-9479.

ERIKSSON, Ulf. How to streamline your test data management. In: *EuroSTAR software testing conference* [online]. 12. 02. 2014 [cit. 2015-10-20]. Dostupné z: <http://conference.eurostarsoftwaretesting.com/2014/how-to-streamline-your-test-data-management/>

GABRIEL, Antonín. Testovací data. In: *[pro]Test!: test data management* [online]. 25. 11. 2014 [cit. 2015-12-03]. Dostupné z: <https://drive.google.com/folderview?id=0Bzr2hyQZJdwkUkhrQmFQYWWhRRIU>

HAMURO, Yoshitomo, Yuto KATOH a Keiji YADA. MUSASHI: flexible and efficient data preprocessing tool for KDD based on XML. *Proceedings of the first international workshop on data cleaning and preprocessing*. 2002, č. 12, s. 38-49.

HEWLETT-PACKARD. *How to address top problems in test data management: data reuse, sub-setting and masking* [online]. 2010 [cit. 2015-10-20]. Dostupné z: http://www.hp.com/hpinfo/newsroom/press_kits/2010/HPSoftwareUniverseBarcelona2010/How_to_address_top_problems_WP.pdf

HLAVA, Tomáš. Fáze a úrovně provádění testů. In: *Testování softwaru: portál zabývající se problematikou ověřování kvality software. Manuální i automatizované testování* [online]. 21. 08. 2011 [cit. 2015-11-10]. Dostupné z: <http://testovanisoftwaru.cz/metodika-testovani/druhy-typy-a-kategorie-testu/faze-testu/>

ISO/IEC/IEEE 29119 *Software testing: The international standard for software testing* [online]. 2013, 10/2014 [cit. 2015-12-12]. Dostupné z: <http://www.softwaretestingstandard.org/>

ISTQB. Certifikovaný tester: učební osnovy pro základní stupeň. In: *CaSTB: Czech and Slovak testing board* [online]. 15. 6. 2013 [cit. 2015-12-01]. Dostupné z: <http://castb.org/wp-content/uploads/2013/11/ISTQB-CTFL-Syllabus-v2011-CZ-Beta1.pdf>

KANER, Cem, Jack FALK a Hung Quoc NGUYEN. *Testing computer software*. 2nd ed. New York: Wiley, 1999, xv, 480 s. ISBN 0-471-35846-0.

KHURANA, Purnima a Purnima BINDAL. Test data management. *International journal of computer trends and technology* [online]. 2014, vol. 15, no. 4, s. 162-167 [cit. 2015-12-02]. DOI: 10.14445/22312803/IJCTT-V15P135. ISSN 22312803. Dostupné z: <http://ijcttjournal.org/archives/ijctt-v15p135>

LANGROVÁ, Kamila. *Analýza projektových rolí v banko-pojišťovacím sektoru ČR*. Praha, 2013. Bakalářská práce. Vysoká škola ekonomická v Praze, Fakulta Informatiky a statistiky. Vedoucí práce: Ing. Tomáš Bruckner, Ph.D.

LANGROVÁ, Kamila. Zápis z 23. setkání [pro]Test!. 2014. Viz Příloha 2.

PATTON, Ron. *Testování softwaru*. 1. vyd. Praha: Computer Press, 2002, xiv, 313 s. ISBN 80-7226-636-5.

POTHARST, R. a Michiel von WEZEL. Generating synthetic data with monotonicity constraints: Econometric institute report EI 2005-06. In: *EconPapers* [online]. Erasmus University Rotterdam, Econometric Institute, 11. 03. 2005 [cit. 2015-11-17]. Dostupné z: <http://econpapers.repec.org/paper/emseureir/1916.htm>

Přehled testů řízených daty. In: *IBM knowledge center* [online]. IBM, [2015] [cit. 2015-12-02]. Dostupné z: http://www-01.ibm.com/support/knowledgecenter/SSBLQQ_8.5.0/com.ibm.rational.test.ft.doc/topics/dpaboutdatadrivingscripts.html?lang=cs

RAHMAN, Kalilur a Matthias RASKING. Test data management – security compliance whilst performing production data extraction/data anonymisation using tools. In: *EuroSTAR software testing conference* [online]. 26. 09. 2013 [cit. 2015-10-20]. Dostupné z: <http://conference.eurostarsoftwaretesting.com/2013/test-data-management-security-compliance-whilst-performing-production-data-extraction-data-anonymisation-using-tools/>

ROSS, Sheldon M. *Simulation*. 4th ed. Boston: Elsevier Academic Press, c2006, xiii, 298 s. ISBN 978-0125980630.

ROUDENSKÝ, Petr a Anna HAVLÍČKOVÁ. *Řízení kvality softwaru: průvodce testováním*. 1. vyd. Brno: Computer Press, 2013, 208 s. ISBN 978-80-251-3816-8.

ŘEZANKOVÁ, Hana. *Analýza dat z dotazníkových šetření*. 3., aktualiz. vyd. Praha: Professional Publishing, 2011, 223 s. ISBN 978-80-7431-062-1.

Software and testing training: what is test data and what are the types of test data. In: *Youtube* [online]. 27. 11. 2013 [cit. 2015-11-17]. Dostupné z: <https://www.youtube.com/watch?v=XNZ2Pm2TAUs>

STRAUSS, Anselm L. a Juliet M. CORBIN. *Základy kvalitativního výzkumu: postupy a techniky metody zakotvené teorie*. 1. vyd. Boskovice: Albert, 1999, 196 s. ISBN 80-85834-60-x.

ŠIMŮNEK, Milan. Příprava umělých dat pro výuku a testování pomocí evolučního algoritmu. *Systémová integrace* [online]. 2013, roč. 20, č. 2, s. 62–72 [cit. 2015-11-17]. ISSN 1804-2716. Dostupné z: http://www.cssi.cz/cssi/system/files/all/SI_2013_2_06_Simunek.pdf

ŠTRBÁK, Martin. Příprava testovacích dat I. In: *Blog o testování* [online]. 07. 03. 2011 [cit. 2015-11-17]. Dostupné z: <http://cz-testing.blogspot.com/2011/03/priprava-testovacich-dat-i.html>

ŠTRBÁK, Martin. Příprava testovacích dat II. - data z produkce. In: *Blog o testování* [online]. 31. 3. 2011 [cit. 2015-11-17]. Dostupné z: <http://cz-testing.blogspot.com/2011/03/priprava-testovacich-dat-ii-data-z.html>

Testing the new system. In: *IGCSE ICT* [online]. 1999 [cit. 2015-11-17]. Dostupné z: <http://www.igcseict.info/theory/8/test/>

Test data. In: *Collins English dictionary* [online]. c2015 [cit. 2015-12-02]. Dostupné z: <http://www.collinsdictionary.com/dictionary/english/test-data>

Test data management. In: *Techopedia* [online]. c2010-2015 [cit. 2015-12-02]. Dostupné z: <https://www.techopedia.com/definition/30193/test-data-management>

What is test data. In: *Software testing mentor* [online]. c2015 [cit. 2015-11-02]. Dostupné z: <http://www.softwaretestingmentor.com/test-deliverables/what-is-test-data/>

What is test data & guidelines to design your test data? In: *Software testing class* [online]. c2015 [cit. 2015-12-02]. Dostupné z: <http://www.softwaretestingclass.com/what-is-test-data-and-guidelines-to-design-your-test-data/>

ŽABKAR, Jure a kol. Learning qualitative models from numerical data. In: *Proceedings of the 23rd international joint conference on artificial Intelligence*, August 3-9, 2013, Beijing, China. AAAI Press, [2013], s. 1604-1619.

8 SEZNAM OBRÁZKŮ

Obrázek 1: Trojúhelník kvality. Bod A představuje drahý, velmi kvalitní systém s dlouhou dobou dodání. V bodě B se sice každému atributu věnujeme stejně, avšak ve výsledku všechny zbytečně ztrácí. [Zdroj: Roudenský a Havlíčková, 2013, s. 129]	14
Obrázek 2: Vztah mezi fází, kdy je defekt nalezen a náklady na jeho odstranění [Zdroj: Roudenský a Havlíčková, 2013, s. 47]	14
Obrázek 3: Pevně naprogramovaný testovací skript s literálovými odkazy [Zdroj: Přehled testů řízených daty]	25
Obrázek 4: Testovací skript řízený daty s datovým fondem [Zdroj: Přehled testů řízených daty]	25
Obrázek 5: Vliv testovacích dat na trojúhelník kvality [Zdroj: Gabriel, 2014]	31
Obrázek 6: Vliv testovacích dat na trojúhelník kvality v tabulce [Zdroj: Gabriel, 2014]	31
Obrázek 7: 3 typy testovacích dat [Zdroj: Testing the new system, 1999].....	33
Obrázek 8: Graf rozdělení respondentů dle délky praxe v oblasti testování [Zdroj: autorka]	44
Obrázek 9: Graf rozdělení respondentů dle délky praxe v oblasti testování (v %) [Zdroj: autorka]	44
Obrázek 10: Graf rozdělení respondentů dle počtu absolvovaných projektů v testování [Zdroj: autorka]	45
Obrázek 11: Graf rozdělení respondentů dle počtu absolvovaných projektů v testování (v %) [Zdroj: autorka]	45
Obrázek 12: Graf četnosti výskytu problémů s testovacími daty [Zdroj: autorka].....	47
Obrázek 13: Graf četnosti výskytu problémů s testovacími daty (v %) [Zdroj: autorka].....	47

Obrázek 14: Poměr schopnosti řešit problémy s testovacími daty s jejich výskytem [Zdroj:
autorka] 49

9 SEZNAM TABULEK

Tabulka 1: Oblasti problémů s testovacími daty definovány na [pro]Test! [Zdroj: autorka] ..	39
Tabulka 2: Oblast problémů s testovacími daty ze zahraničních zdrojů [Zdroj: Khurana a Bindal, 2014].....	40
Tabulka 3: Doba praxe respondentů v oblasti testování [Zdroj: autorka].....	46
Tabulka 4: Počet absolvovaných projektů respondenty [Zdroj: autorka]	46
Tabulka 5: Přehled četností výskytu jednotlivých problémů s testovacími daty [Zdroj: autorka]	48
Tabulka 6: Seznam problémů s testovacími daty dle četnosti výskytu [Zdroj: autorka]	48
Tabulka 7: Poměr výskytu problémů a schopnosti řešení problémů s testovacími daty (v %) [Zdroj: autorka]	49
Tabulka 8: Seznam problémů s testovacími daty dle schopnosti vyřešení problému [Zdroj: autorka]	50

10 PŘÍLOHY

Příloha 1 – Dotazník

Příloha 2 – Zápis z 23. setkání [pro]Test!

NEJČASTĚJŠÍ PROBLÉMY S TESTOVACÍMI DATY A MOŽNOSTI JEJICH ŘEŠENÍ

DOTAZNÍK

- ☐ *U takto označených odpovědí lze zvolit jen jednu možnost*
- ☐ *U takto označených odpovědí lze vybrat více možností*

1. Jak dlouho pracujete v oblasti testování?

- ☐ Méně než 2 roky
- ☐ 2-5 let
- ☐ Více než 5 let

2. Na kolika různých projektech jste se věnoval(a) / vedl(a) testování?

- ☐ 1 projekt
- ☐ 2-5 projektů
- ☐ Více než 5 projektů

3. Se kterými z těchto problémů s testovacími daty jste se v praxi setkal(a)?

Označte prosím všechny, se kterými jste se setkal(a).

- ☐ Protichůdné požadavky na data
- ☐ Problematika anonymizace dat
- ☐ (Ne)konzistence dat napříč různými aplikacemi nebo prostředími
- ☐ Nemožnost tvorby historických dat (například starších typů produktů)
- ☐ Špatná specifikace množiny dat
- ☐ Časté ničení dat (malým používáním, sdílením dat mezi testery, atp.)
- ☐ Časová náročnost přípravy některých specifických dat (dlouhodobá data nebo data zpětně)

4. Nalezl(a) jste způsob, jak předejít nebo vyřešit některý z výše uvedených problémů?

- | | |
|----------|---|
| ANO - NE | Protichůdné požadavky na data |
| ANO - NE | Problematika anonymizace dat |
| ANO - NE | (Ne)konzistence dat napříč různými aplikacemi nebo prostředími |
| ANO - NE | Nemožnost tvorby historických dat (například starších typů produktů) |
| ANO - NE | Špatná specifikace množiny dat |
| ANO - NE | Časté ničení dat (malým používáním, sdílením dat mezi testery, atp.) |
| ANO - NE | Časová náročnost přípravy některých specifických dat (dlouhodobá data nebo data zpětně) |

5. Popište prosím způsob předcházení či řešení problémů.

(U všech, u kterých jste ve 4. otázce zaškrtnl(a) ANO)

6. Způsobilo řešení vámi popsané v 5. otázce nějaké jiné problémy nebo překážky? Jaké? Prosím popište.

Zápis z 23. setkání [pro]Test!

Téma: Test Data Management

Datum: 25. 11. 2014, 17:00

Místo: 18. patro, Raiffeisenbank a.s., Hvězdova 1716/2b, 140 78 Praha 4

Počet účastníků: 28

Vyhotovil: Bc. Kamila Langrová

A. Program setkání

	Prezentující	Téma
1	Bc. Kamila Langrová	Prezentace s žádostí o pomoc s Diplomovou prací
2	Antonín Gabriel	Testovací data - Požadavky na testovací data z hlediska zákazníka - Způsoby zajištění testovacích dat - Náklady a prosaditelnost jednotlivých způsobů zajištění testovacích dat
3	Luboš Fryc	Test Data Management - Fundamentals (Methods, Strategy, Roles - DBA / Test Engineers) - Concepts (Summary, Key Concepts, Processes) - Best Practice (Good Test Data, Requirements, Building Test Data) - Increased Business Value - Security & Compliance

B. Průběh diskuze k první části programu

Byla položena otázka „S jakými problémy s testovacími daty se v praxi setkáváte?“. Jako první problémy byly vyhodnoceny zaprvé nekonzistence dat napříč aplikacemi nebo prostředím – anonymizovaná vs. reálná data na stejném testovacím prostředí, případně při manuální tvorbě dat se ne vždy všechno propíše do všech aplikací na testovacím prostředí a zadruhé špatná specifikace množiny dat v zadání projektu. Poté byl diskutován problém protichůdných požadavků na data (například Data warehouse jako online zdroj), který jsme se také rozhodli do seznamu zařadit. Poté se otevřela velká diskuze o tvorbě, časové náročnosti tvorby a udržování testovacích dat. Z toho byly nakonec vykrystalizovány tři větší oblasti a to zaprvé nemožnost tvorby historických dat (například starších typů produktů), zadruhé časté ničení dat (malým používáním, sdílením dat mezi testery, atp.) a zatřetí časová náročnost přípravy některých specifických dat (dlouhodobá data nebo data zpětně). Závěrem diskuze se ještě zmínila problematika anonymizace dat, která byla vyhodnocena jako stále aktuální a časově náročná. Diskuze byla tedy uzavřena sedmi oblastmi problémů s testovacími daty, se kterými se přítomní účastníci setkávají. Tyto oblasti uvádím v tabulce v části C.

C. Diskutované oblasti

Oblast problémů s testovacími daty
Protichůdné požadavky na data
Problematika anonymizace dat
(Ne)konzistence dat napříč různými aplikacemi nebo prostředími
Nemožnost tvorby historických dat (například starších typů produktů)
Špatná specifikace množiny dat
Časté ničení dat (malým používáním, sdílením dat mezi testery, atp.)
Časová náročnost přípravy některých specifických dat (dlouhodobá data nebo data zpětně)