

Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS	
Semestr	LS 2017/2018
Autoři – jméno, příjmení , xname	Alexey Kim, kima03 Anastasia Frolova, xfroa01
Téma	Porovnání disciplíny Test v RUP, MMSP a Metodice testování dle mezinárodních praktik a standardů
Datum odevzdání	12.05.2018

Abstrakt

Obsahem této práce je porovnávací rešerše různých disciplín testování v metodikách RUP, MMSP a Metodice testování dle mezinárodních praktik a standardů. Po stručném seznámení s metodikami je provedeno porovnání procesu testování ve zvolených metodikách a vyhodnocení přínosů. Po přečtení této práce čtenář by si měl udělat celkovou představu o vybraných metodikách a pochopit jaky je mezi nimi rozdíl ve procesu testování.

Klíčová slova

RUP, MMSP, ISTQB, Testování, Porovnání, Tester

Obsah

	2
1 Úvod	3
2 Definice metodik	3
2.1 Metodika RUP	3
2.1.1 Základní charakteristika	3
2.1.2 Testování v metodice	5
2.1.2.1 Role	5
2.1.2.1.1 Test Manager	5
2.1.2.1.2 Test Analytik	6
2.1.2.1.3 Test Designer	6
2.1.2.1.4 Tester	6
2.1.2.2 Techniky	6
2.2 Metodika MMSP	7
2.2.1 Základní charakteristika	7

2.2.2 Role	8
2.2.2.1 Analytik	8
2.2.2.2 Tester	8
2.2.3 Testování v metodice	9
2.2.3.1 Plánování testů	9
2.2.3.2 Příprava testů	9
2.2.3.3 Provedení testů	10
2.3 Metodika mezinárodních praktik a standardů (MP)	10
2.3.1 Základní charakteristika	10
2.3.2 Role	11
2.3.2.1 Test manažer	11
2.3.2.2 Test analytik	12
2.3.2.3 Tester	12
2.3.3 Techniky	12
2.3.3.1 Testování rozdělením na třídy ekvivalence	12
2.3.3.2 Testování založené na rizicích	13
2.3.3.3 Průzkumné testování	13
2.4 Porovnání metodik dle uvedených kritérií	13
2.4.1 Porovnání dle rolí	13
2.4.1.1 Tester	14
2.4.1.2 Test Analytik	14
2.4.1.3 Test Manažer	14
2.4.1.4 Test Designer	14
2.4.2 Porovnání dle technik testování	14
2.4.2.1 RUP	15
2.4.2.2 MMSP	15
2.4.2.3 MP	15
3 Závěr	15
4 Literatura	16

1 Úvod

Informační systémy jsou dnes neoddělitelnou částí každého úspěšného podniku. Jejich implementace je většinou podporovaná moderními softwarovými nástroji. Může to být jednoduchá mobilní aplikace nebo rozsáhlý bankovní portál. S rostoucím počtem softwarových řešení paralelně narůstá počet negativních uživatelských zkušeností s aplikacemi, které nefungují jak se očekává. Nedostatečná kvalita dodaného softwaru může vést k závažným důsledkům jako ztráta peněz, zpomalení práce, poškození reputace podniku a v některých případech i smrti.

Člověk není stroj, proto existuje jistá míra chybovosti ve výsledcích jeho práce. Chyba vývojáře softwaru způsobuje chyby v softwaru, tzv “bugy”. Existuje řada dalších faktorů, které ovlivňuje výstupy softwarových produktů: práce analytika a kvalita dodaných požadavků, metodiky vývoje softwaru, vnější fyzické a ekonomické faktory. Bohužel nelze zachytit a ošetřit všechny faktory, proto čím víc opatření bude zapracováno, tím stabilnější a kvalitnější bude dodaný software. Hlavní bránou, která posuzuje kvalitu navařeného řešení, je proces testování.

V rámci této seminární práce se podíváme jak funguje testování ve zvolených metodikách RUP, MMSP a testování dle mezinárodních standardů. Hlavním cílem této práce je seznámení čtenáře se zvolenými metodikami a porovnání procesu testování v rámci každé metodiky. V první části práce bude definována každá vybraná metodika s podrobnějším popisem fáze testování. Ve druhé části práce bude provedena srovnávací analýza testování a závěrečné vyhodnocení.

2 Definice metodik

2.1 Metodika RUP

2.1.1 Zakladní charakteristika

Metodika RUP (Rational Unified Process) je iterativní metodikou pro vývoj softwaru. Vznik metodiky sahá do roku 1995, sloučením metodiky Objectory Process a Rational Approach. Důvodem sloučení byla fúze dvou firem Objectory AB s Rational Software Corporation, kde již z názvu je jasné, že společnost Objectory AB používala metodiku Objectory Process a Rational Software Corporation metodiku Rational Approach. V roce 2003 byla metodika odkoupena společností IBM.

Tato metodika zdůrazňuje užití nejlepších praktik při vývoji softwaru, mezi které patří:

1. Iterativní vývoj
2. Řízení požadavků
3. Vizuální modelování
4. Komponentová architektura
5. Kontinuální ověřování kvality
6. Řízení změn

Iterativní vývoj neboli rozdělení projektu na iterace umožňuje již v rámci jedné iteraci odhalit rizika v počátcích projektu. Tohle je výhodné v tom pohledu, že náklady na odstranění odhalených rizik jsou relativně nízké, než by byly v dalších fázích vývoje projektu. Neustálé řízení požadavků vede k dřívějšímu zaznamenání měnících se požadavků na změnu systému. Užití vizuálního modelování napomáhá při utváření představ o vyvíjeném systému. Metodika využívá Unified Modeling Language (UML). Pro implementaci softwaru doporučuje metodika zvolit komponentovou architekturu, která je založená na užití znovupoužitelných komponentů. To zvyšuje produktivitu a umožňuje doplňovat architekturu v případě vývoje dalších komponentů. Metodika také doporučuje sledovat kvalitu po celém životním cyklu projektu, tedy provádět testování po každé iteraci, nikoli na konci projektu. Průběžné ověřování kvality umožňuje včasější odstranění vzniklých chyb. Při vývoje je nezbytné sledovat veškeré změny prováděné v systému. Metodika zároveň popisuje způsob řešení změnového řízení, jehož cílem je úspěšný iterativní vývoj.

Grafická interpretace metodiky RUP je zachycena na níže uvedeném obrázku, viz Obrázek 1.

Obrázek 1: Schéma projektu dle metodiky RUP. Zdoj: kitscm.vse.cz

Na svislé ose jsou vyneseny disciplíny, které znázorňují rozdělení všech činností a aktivit do logických celků. Jsou to obchodní model, požadavky, analýza a návrh, implementace, testování a předání. Kromě těchto šesti disciplín jsou zde vyneseny další tři disciplíny podporující základní, jsou to konfigurační a změnový management, projektový management a správa prostředí. Ke každé disciplíně graf znázorňuje, jaký důraz je na jakou disciplínu v průběhu životního cyklu projektu kladen.

Na vodorovné ose je vynesen čas, tedy plynutí času od zahájení projektu do jeho konce. Také jsou zde uvedené fáze a iterace metodiky RUP. Přerušované čáry znázorňují milníky jednotlivých fází.

Životní cyklus projektu je podle metodiky RUP rozdělen do 4 fází:

1. Zahájení (Inception)
2. Příprava (Elaboration)
3. Konstrukce (Construction)
4. Předávání (Transition)

Fáze zahájení projektu definuje účel, rozsah projektu a jeho obchodní případ, který obsahuje kriteria úspěchu, určení rizik, odhad potřebných zdrojů a také definování jednotlivých milníků projektu.

Dále následuje fáze příprava, hlavním cílem které je tvorba projektového plánu, analýza problematických oblastí a eliminace rizik projektu. Na základě sebrané informace se dále rozhoduje jestli tento projekt bude předán do konstrukce.

Dále následuje fáze konstrukce nebo realizační fáze, ve které probíhá tvorba zdrojových kódů. Účelem této fázi je vytvoření všech komponentů a vlastností aplikace, jejich následná integrace do produktu a důkladné otestování všech jeho součástí.

V poslední fázi předávání probíhá předání softwarového produktu uživatelům. Jakmile se produkt dostane ke koncovému uživateli mohou se objevit připomínky, což může následně vest k vývoji nových verzí.

Jak je patrné z výše uvedeného obrázku, jednotlivé fáze jsou prováděny v iteracích. Pod iteraci se představuje podмноžiny jednotlivých fází, každá s kterých je dokončena milníkem. Po dokončení každé fáze následuje její vyhodnocení a jen pokud byla splněna všechna kritéria, je možné zahájit další fázi projektu.

2.1.2 Testování v metodice

Testování v metodice RUP je samostatnou disciplínou. Z obr. 1 je patrné, že je na testování kladen důraz během celého modelu. Hlavní filosofií metodiky je nalézt co nejvíce problémů již v průběhu vývoje. Nejvíc však těsně ve fázi Konstrukce před předáním produktu zákazníkovi. Oblasti testování se metodika RUP věnuje velmi rozsáhle. Definuje jednotlivé role, které se mají testování věnovat. Poskytuje vzory procesů, návodů, šablon a dalších součástí disciplíny testování.

2.1.2.1 Role

S testovací disciplínou jsou spojené čtyři testové role - Test Manager, Test Analytik, Test Designer a Tester. Pro každou role metodika vymezuje činnosti, které vykonává a artefakty, se kterými pracuje. Testeři mají vlastní testovací tým v rámci projektového týmu, který je řízen manažerem testování. Níže je uveden podrobnější popis odpovědností pro každou roli.

2.1.2.1.1 Test Manager

Jedná se o řídicí roli. Představují vedoucí pozice v disciplíně testování. Je zodpovědný za úspěch celého testovacího procesu. Jeho úkolem je řídit proces testování, řešit problémy a záležitosti, které zdržují celý proces testování. Má také na starost plánování a řízení zdrojů.

2.1.2.1.2 Test Analytik

Jedná se o prováděcí roli. Test analytik má na starostí identifikace cílových součástí testů a definice potřebných testů a s nimi spojených dat. Je také zodpovědný za sledování jejich průběhu a analýzu výstupu proběhlých testů.

2.1.2.1.3 Test Designer

Jedná se o prováděcí roli. Test Designer je zodpovědný za definici přístupů k testování a jejich úspěšnou implementaci. Určuje techniky, nástroje a metodické pokyny k realizaci potřebných testů.

2.1.2.1.4 Tester

Jedná se o prováděcí roli. Tester je zodpovědný za hlavní činnosti v rámci disciplíny testování. Má na starostí implementaci jednotlivých testů, jejich připravení a provádění, monitoringu, zaznamenávání výstupů a následné vyhodnocení výsledků testů.

Za zajištění kvality produktu zodpovídají všichni členové projektového týmu společně. Testovací tým má za úkol pouze ověřování kvality softwaru, díky kterému může nasměrovat zbytek týmu k dosažení softwarové kvality. Nenese ovšem přímou odpovědnost za zajištění kvality. [1]

2.1.2.2 Techniky

Celý proces testování je rozdělen do několika kroků, z nichž některé probíhají paralelně a opakují se v každé iteraci. [2]

Na začátku každé iterace probíhá určení cílů testování pro danou iteraci. Po nastavení cílů následuje ověření, zda aktuální sestava je stabilní. Pokud je stabilní, je podrobena testování a ohodnocení, v opačném případě sestava je vrácena zpět do vývoje. Paralelně je sestavena zpráva o průběhu a výsledcích testování a následně předaná řídicímu výboru projektu. Dále následuje proces, který je v metodice označován jako Test Assets. Mezi činnostmi probíhající v procesu patří údržba a zkvalitnění testovací dokumentace, testovacích dat, vytvořených testů a dalších náležitostí testování.

Celý ten proces od ověření stability sestavy až po zkvalitnění náležitostí testování je cyklický, neboli v rámci jedné iterace může proběhnout i vícekrát. Paralelně běží proces, který kontroluje, zda zvolený přístup k testování je vhodný, jestli jsou testy kvalitní a jsou relevantní ve výsledku. Metodika doporučuje použít přístup TFD, hlavní myšlenkou kterého je vytvoření testu před psaním kódu. Jinými slovy testy zde představují nástroj pro návrh systému a ověřují jeho funkcionalitu. Při návrhu testů, je potřeba důkladně zamýšlet nad tím, kdy je ten přesný čas na jejich návrh a postupovat v souladu s cíli iterace a fáze projektu.

Metodika RUP pokrývá různé úrovně testování, mezi které lze zařadit jednotkové, integrační, systémové a akceptační testy. Jednotkové testy se zabývají testováním jedné zvolené komponenty a tím pádem neřeší chyby jiných komponentů. Tyto testy se provádějí podle potřeby, obvykle několikrát během iterace. Pro testování více komponentů dohromady slouží

integrační testy. Dle metodiky tyto testy jsou rozděleny na testy podsystémů, které by měly proběhnout aspoň jednou za iteraci, a testy systému jako celku, běh kterých již záleží na tvorbě sestavy. Po skončení integračních testů přichází čas systémových testů, které se zaměřují na aplikaci jako celek v podobě, v jaké by ji měl používat zákazník. Systémové testy by měly proběhnout několikrát během iterace. Po úspěšném průběhu integračních a systémových testů následují akceptační testy. Výsledky těchto testů musí splňovat stanovené kritéria pro přechod do další fáze předávání hotového řešení zákazníkovi. Dle metodiky, jednotkové a integrační testy by měli provádět vývojáři. Testovací tým by se měl zaměřit hlavně na systémové a akceptační testy.

Metodika RUP zmiňuje dva druhy testů. Jedním je smoke test, který slouží k ověření stability a funkcionality aktuální sestavy systému. Další druh testu je tak zvaný beta test, který se provádí ve fázi zavedení, kdy je produkt nasazen do zkušebního provozu. V metodice jsou popsány tyto testy spolu s doporučením jak je provádět a jaké nástroje je vhodné k tomu použít.

Mezi základní artefakty, které metodika vymezuje patří plán testů (Test Plan), testovací skript (Test Script), test protokol (Test Log), testovací případ (Test Case), Test Results, Test Strategy, Test Evaluation Summary, aj. Metodika RUP při popisu artefaktů uvádí účel, povinnost artefaktu a informace, které by měl obsahovat.

V metodice RUP je také uveden obecný koncept popisující výhody a nevýhody automatizovaného testování, avšak metodika doporučuje využívat především manuální testování. Nabízí taky základní postupy k automatizovaným aktivitám, které lze provádět s pomocí nástrojů společností IBM. Ovšem tyto nástroje jsou komerční, což je velkou nevýhodou pro mnoha firmy.

2.2 Metodika MMSP

2.2.1 Základní charakteristika

Metodika MMSP je určená pro malé softwarové projekty. MMSP je iterativní metodika, která je založena na principu RUP a OpenUP. Iterativní přístup může být závažným omezením uplatnění metodiky pokud průběh projektu není iterativní. Metodika však zahrnuje částí dalších agilních metodik, "best practices" apod. Metodika MMSP byla vytvářena jako plug-in k metodice OpenUP a může být dále upravována prostřednictvím nástroje Eclipse Process Framework Composer. [4]

Základní pojmy metodiky:

Název	Popis	Příklady
Pracovní produkt	To, co je v rámci procesu vývoje vytvářeno, modifikováno či používáno.	List pracovních položek, Seznam rizk
Úloha	Popis, jak pracovat, aby bylo dosaženo stanovených cílů či byly vytvořeny určité pracovní produkty. Úlohy jsou vykonávány rolami a jsou obvykle definovány jako série dílčích kroků.	Návrh architektury, Plánování testů
Role	Charakteristika člena vývojového týmu, který provádí v rámci procesu vývoje jednotlivé úlohy a vytváří požadované pracovní produkty. Role definují schopnosti a zodpovědnosti, ne konkrétní osoby.	Analytik, Tester
Proces	Popis struktury práce a workflow činností, který spojuje dohromady úlohy, role a pracovní produkty.	Životní cyklus
Návod	Šablony, příklady, rady a další pomůcky, které je možné v rámci procesu vývoje využít.	Plán projektu - šablona

Tabulka 1: Základní pojmy metodiky. Zdroj: [5]

Projekty vhodné pro nasazení metodiky MMSP by měly splňovat následující kritéria:

1. Délka projektu by neměla přesahovat hranici 6 měsíců
2. Počet členů týmu je maximálně 10 lidí
3. Náročnost vyvíjeného systému není vysoká

2.2.2 Role

Podle MMSP na projektu by měli být obsazené následující role: Projektový manažer, Analytik, Tester, Vývojář, Architekt a Zainteresovaná strana. Z pohledu metodiky v procesu testování jsou dvě klíčové role: tester a analytik. [4]

2.2.2.1 Analytik

Analytik je osoba která se stará o sběr požadavků, jejich zapracování a určení priorit. Role je svazuje vývojový tým a koncové uživatele softwaru. Analytik především zodpovídá za: detailní vymezení nefunkčních požadavků, identifikace a zaznamenání požadavků, modelování případů užití, návrh technické vize, plánování konfiguračního řízení. Pro složitější projekty je dobré mít samostatného analytika. Jelikož MMSP se zabývá menšími projekty, role analytika může současně zastupovat zkušený tester. [4]

2.2.2.2 Tester

Tester je představitel QA oddělení. Pověřená osoba má na starosti především 3 činnosti: plánování testů, provádění testů, příprava testů. MMSP doporučuje mít na projektu několik

osob zodpovědných za testování. O plánování testů a přípravu testovacích dat by se měl starat zkušenější tester. Metodika doporučuje kombinovat roli testera s roli analytika.

2.2.3 Testování v metodice

Fáze testování v MMSP zahrnuje 3 základní činnosti: plánování testů, provádění testů, příprava testů. Model testování je znázorněn na následujícím obrázku:

Obrázek 1: Proces testování v metodice MMSP. Zdroj: [4]

Testování probíhá pomocí manuálních a automatických testů. Vzhledem k tomu že podle MMSP jsou řízené menší projekty, preferují se manuální testy z důvodu úspory času na psaní automatických skriptů. Výjimkou jsou předem připravené testovací skripty z předchozích projektů nebo explicitní požadavek na zátěžové a výkonové testy. Jednotkové testy v rámci této metodiky připravuje vývojový tým. Všechny testovací činnosti jsou iterativní a probíhají v každé iteraci od začátku. [4]

2.2.3.1 Plánování testů

V rámci této aktivity se připravuje testovací strategie a způsoby testování, definice nástrojů pro podporu testování, rozsah testů. Následně se určuje harmonogram testů a definice výstupů pro každou iteraci testů. Plánování testů v MMSP probíhá dle FURPS+. [4]

2.2.3.2 Příprava testů

Příprava testů je nejrozsáhlejší etapa v rámci testování v MMSP. Na základě testovacího plánu, strukturovaných požadavků a případů užití se provádí hlavní kory v rámci přípravy testů:

1. **Mapování testovacích nápadů** - připravuje se kompletní seznam toho, co by mělo být otestováno na základě požadavků na software.
2. **Tvorba testovacích případů** - tvorba klasických testovacích scénářů na základě kterých tester bude vykonávat testování.
3. **Tvorba testovacích sad** - testovací scénáře se dávají dohromady do testovacích sad. Každá sada představuje testování nějakého většího modulu softwaru.
4. **Definice testovacích dat** - definují se testovací data jako různé validní a nevalidní vstupní hodnoty pro jednotlivé pole.

2.2.3.3 Provedení testů

V momentě kdy všechny požadavky jsou zpracované a testovací scénáře vytvořené, nastává fáze provedení nebo taky exekuce testů. Testování probíhá dle následujících kroků:

- 1. Výběr testovacích sad** - testovací sada se vybírá dle stanovené priority. Pořadí se určuje ve fázi plánování testů.
- 2. Realizace testů** - tester vykonává testy podle navržených testovacích scénářů. Testování taky může být průzkumné. Průzkumnému testování se občas dává větší přednost z důvodu nevysoké komplexity řešení a časového omezení. V neposlední řadě se provádějí regresní testy. U regresních testů je výhodnější používat automatické testování.
- 3. Vyhodnocení výsledků** - v případě testování dle testovacích scénářů tester porovnává skutečný výstup testu s očekávaným. Pokud budou nalezeny nějaké chyby, tester by měl založit ticket na opravu.

2.3 Metodika mezinárodních praktik a standardů (MP)

2.3.1 Základní charakteristika

Táto metodika se primárně zabývá testováním softwaru. Kompletně popisuje proces testování od přípravy testů až po řízení a spuštění. Metodika je založena na základě materiálů pro certifikační zkoušku ISTQB (International Software Qualification Board). Samotné materiály ISTQB nejsou popisem metodiky, ale souhrnem "best practices" v testování softwaru. Specifikace většiny pracovních produktů je definována na základě standardu IEEE 829. Některé úlohy a produkty jsou definované na základě metodiky RUP. Popis elementů metodiky je částečně popsán v následující tabulce:

Název	Popis	Příklady
Hlavní aktivita	Seskupení úloh, které se vztahují k určité oblasti zájmu a jsou vykonávány ve stejném časovém období.	Plánování testů, implementace testů
Úloha	Nejmenší jednotky práce, které má smysl definovat jako samostatný proces. Jsou vykonávány rolemi a jedná se o „popis jak pracovat, aby bylo dosaženo stanovených cílů či byly vytvořeny určité pracovní produkty“ [MMSP, 2011].	Návrh testovacích případů, příprava testovacích dat, analýza a reportování defektů
Pracovní produkt	Výstup z procesu testování, který je v průběhu testování vytvářen, modifikován a používán [IBM, 2005].	Plán testování, defekt report
Technika	Postupy a přístupy k tomu, jak dosáhnout určitého cíle, jako je snížení míry rizika, redukce množství testovacích případů nebo	Rozdělení tříd ekvivalencí, průzkumné testování

	zvýšení účinnosti a efektivity testování.	
Role	Abstraktní vyjádření souboru zodpovědností za jednotlivé úlohy a pracovní produkty, které mohou být naplněny jednou nebo více osobami [IBM, 2005].	Test manažer
Koncept	Poskytují informace, na které se odkazují ostatní prvky metodiky a které je vhodné vyčlenit zvlášť.	Klasifikace defektů
Šablona	Pomůcka pro opakované vytváření pracovních produktů. Definuje náležitosti, které má pracovní produkt mít a při vytváření vlastního pracovního produktu stačí tyto náležitosti doplnit bez potřeby řešit jejich formu.	Šablona pro testovací případ

Tabulka 2: Prvky metodiky. Zdroj: [autor; IBM, 2005; MMSP, 2011]

Podle modelu testování ISTQB (V-model) fáze testování se začíná po implementaci řešení v následující posloupnosti:

1. Jednotkové testování
2. Testování komponentů
3. Systémové integrační testování
4. Systémové testování
5. Akceptační testování

Metodika mezinárodních praktik primárně pokrývá testování na úrovni systémových testů. Podle potřeby lze metodiku přizpůsobit na jiné úrovni V-modelu a nadbytečné prvky metodiky vynechat. Testování podle této metodiky vyžaduje dostatečně kvalifikovaný tým testerů. Metodika není vhodná pro malé týmy a společnosti ve kterých testování většinou řeší programátor, protože objem definovaných testovacích prvků je poměrně velký. [1]

2.3.2 Role

Role je abstraktním vyjádřením souboru zodpovědností za jednotlivé úlohy a pracovní produkty, které mohou být naplněny jednou nebo více osobami [IBM, 2005]. Každá role vyžaduje určité schopnosti a dovednosti, které jsou vyžadované pro naplnění práce. V materiálech od ISTQB není kompletně popsány role a jejich funkce. Podle ISTQB všechny testovací činnosti spadají jenom do dvou rolí test lídr a tester. Vzhledem k opravdu velkému počtu činností je třeba zavést další podpůrnou roli - test analytik. [1]

2.3.2.1 Test manažer

Test manažer má na starost vedení týmu testování v průběhu celého procesu testování. Je zodpovědný za kvalitu výsledného produktu, plánování a řízení zdrojů na projektu a řešení problémů, které stojí testovacímu týmu na cestě ke splnění cílů [IBM, 2006]. Do činností test manažera především patří: analýza rizik testování, definice cílů a rozsahu testování, řízení testování, reportování o stavu testování.

Tuto roli lze přiřadit několika osobám, avšak z praxe se ukazuje že na jednom projektu pracuje nejvýše jeden test manažer. Projektový manažer může taky mít podpůrnou roli test manažera jestli má dostatečné zkušenosti s testováním. [3]

2.3.2.2 Test analytik

Test analytik je osoba, která poskytuje veškeré potřebné zdroje pro testování. Mezi hlavní činnosti patří: analýza podkladů pro testování, návrh testovacích scénářů, Příprava testovacích dat, vytváření testovacích skriptů.

Test analytik má za úkol nejen vytvářet podklady pro testování, ale taky pomáhat Projektovému manažeru. [3]

2.3.2.3 Tester

Tester je osoba, která zodpovídá za vykonání testů a zaznamenání výsledků. Vzhledem k objemu práce, role testera je většinou disjunktní a pracovník s roli testera nemá další role. Do činností testera spadají: analýza a reportování defektů, ověření připravenosti infrastruktury pro testování, provádění testů a logování výsledků, retesty a regresní testování. [3]

2.3.3 Techniky

Podle ISTQB existuje řada testovacích technik. Metodika mezinárodních praktik avšak přebrala jenom několik hlavních technik.

2.3.3.1 Testování rozdělením na třídy ekvivalence

Technika černé skříňky, kdy testovací případy jsou navrženy tak, aby pokryly jednotlivé třídy ekvivalence a z každé třídy ekvivalence otestovaly alespoň jednoho zástupce. [ISTQB, 2012, s. 20]

Základní axiom testování je v tom, že není možné otestovat úplně vše. Čím víc věcí chceme otestovat, tím víc testovacích scénářů je třeba připravit. Snížit počet testovacích scénářů a zachovat pokrytí testování umožňuje technika rozdělení na třídy ekvivalence. Příkladem je testování věrnostního programu letecké společnosti, která na základě nasbíraných mil poskytuje zákazníkovi speciální členství:

Třída ekvivalence	míle < 0	$0 \leq \text{míle} < 10000$	$10000 \leq \text{míle} < 25000$	$25000 \leq \text{míle}$
Očekávaný výsledek	chyba	Standardní členství	Stříbrné členství	Zlaté členství
Hodnota pro testování	-10	4500	15000	30000
Testování mezních hodnot	0, -1	0,1, 9999, 10000	10000, 10001, 24999, 25000	25000, 25001

Tabulka 3: Rozdělení na třídy ekvivalence. Zdroj: Autor

2.3.3.2 Testování založené na rizicích

Přístup k testování, jehož účelem je snížit úroveň produktových rizik a informovat zainteresované strany projektu o stavu těchto rizik v průběhu projektu. Začíná v již v počátečních fázích projektu a zahrnuje identifikaci produktových rizik a využívání úrovní rizik pro řízení procesu testování. [ISTQB, 2012c, s. 37]

Tato technika má především preventivní charakter a slouží k včasnému odhalení potenciálních problémů a následný návrh a provádění testů, které pomůžou snížit pravděpodobnost určitých rizik. Testování založené na rizicích zahrnuje čtyři hlavní aktivity: identifikace rizik, hodnocení rizik, snížení úrovně rizik, řízení rizik v životním cyklu. [3]

2.3.3.3 Průzkumné testování

Technika průzkumného testování (Exploratory testing) je nejmíň formální technika ze všech zmíněných. Testování je založeno na vlastních zkušenostech testera a není řízeno podle striktně definovaných testovacích korek. Průzkumné testování je vhodné dělat zkušenějším testerům které mají dostatečné doménové znalosti projektu a produktu. Existuje nevýhoda průzkumného testování, která se spočívá v reprodukovatelnosti testovacích korek, které vedou k chybě. Tester musí vzpomenout a analytický určit jaké kroky vyvolali chybu a následně založit defekt. [3]

2.4 Porovnání metodik dle uvedených kritérií

2.4.1 Porovnání dle rolí

Z předchozích kapitol je patrné, že každá z popsaných metodik určuje specializované testovací role. V následující tabulce jsou zobrazeny hlavní role procesů testování v jednotlivých metodikách.

	RUP	MMSP	Metodika dle mez. praktik a standardů (MP)
Role	Test Manažer Test Analytik Test Designer Tester	Analytik Tester	Tester Test Analytik Test Manažer

Tabulka 4: Porovnání dle rolí. Zdroj: Autor

Je třeba zmínit, že pro každou metodiku je společné zapojení do procesu testování i dalších rolí, které se podílí na vývoje softwaru. Jinými slovy testování není zodpovědností pouze testovacích rolí, ale probíhá také v rámci vývojářského týmu. Hlavní myšlenkou je společensky přístup k kvalitě produktu.

2.4.1.1 Tester

Role testera je obsažená ve všech metodikách. Metodika MMSP doporučuje tuto roli kombinovat s rolí analytika, avšak podle metodiky mezinárodních praktik role testera je atomická. Pracovní povinnosti testera jsou ve všech metodikách skoro totožné. Dle MMSP a RUP tester by neměl řešit jednotkové testování. V ISTQB se popisují techniky jednotkového testování kterými by se měl zabývat tester. Výstup práce testera je ve všech metodikách se moc neliší.

2.4.1.2 Test Analytik

Role je definovaná v metodikách RUP a MP. Rozdíl je v metodice MMSP kde činnosti test analytika vykonává tester ve fázi "Plánování testů". V MP se zdůrazňuje, že test analytik by měl pomáhat projektovému manažerovi.

2.4.1.3 Test Manažer

Test manažer je řídicí role která především obsazena v metodikách RUP a MP. Jejich hlavní činnost je stejná. Podle RUP a MP test manažer má na starosti vedení týmu testování v průběhu celého procesu testování. V MMSP není definována role test manažera, ale podobné činnosti může vykonávat osoba v roli testera a zároveň analytika. Dle RUP role test manažer navíc má na starosti řízení testerských zdrojů.

2.4.1.4 Test Designer

Role s názvem Test Designer je uvedena pouze v metodice RUP, ovšem u ostatních metodik vyskytují role, které jsou zodpovědné za podobné činnosti. Test designer v metodice RUP je zodpovědný za definici činností potřebných k testování. Podobné činnosti má na starosti role test analytika v metodice MP, která je zodpovědná za vytváření podkladů pro testování. Na těchto činnostech se v metodice MMSP podílí tester.

2.4.2 Porovnání dle technik testování

	RUP	MMSP	Metodika dle mez. praktik a standardů (MP)
Techniky			
	<ul style="list-style-type: none">- TFD- Smoke test- Beta test	<ul style="list-style-type: none">- Black-box- Průzkumné testování	<ul style="list-style-type: none">- Rozdělení na třídy ekvivalence- Testování založené na rizicích- Průzkumné testování- Black-box

			- White-box
Úrovně testování			
Jednotkové	Vývojář (dle potřeby Tester)	Vývojář, výjimečně	Tester
Komponentní	Vývojář (dle potřeby Tester)	Vývojář, výjimečně	Tester
Integrační	Vývojář (dle potřeby Tester)	Tester	Tester
Systémové	Vývojář, Tester	Tester	Tester
Akceptační	Tester	Tester	Tester

Tabulka 5: Porovnání dle technik testování. Zdroj: Autor

2.4.2.1 RUP

Metodika RUP je velmi rozsáhlá a plně pokrývá disciplínu testování, která má vazbu i na ostatní disciplíny metodiky. Je zaměřená na všechny úrovně testování a doporučuje využít TFD přístup. Metodika také doporučuje automatizované testování u jednotkových testů a nabízí řadu technik pro jeho provedení.

2.4.2.2 MMSP

Na rozdíl od RUP a MP, v metodice MMSP především kladen důraz na rychlost testování z důvodu šetření času na malém projektu. Proto především tester se zabývá Black-box a průzkumným testováním. V MMSP se dává přednost manuálním testům než automatickým ale automatické testy se výjimečně provádějí. Metodika MMSP je ulehčenou verzí metodiky RUP a na rozdíl od metodiky RUP nepokrývá oblast testování zcela. Kromě toho metodika je v českém jazyce, což je výhodou pro české týmy.

2.4.2.3 MP

Metodika mezinárodních praktik na rozdíl od RUP a MMSP poskytuje velký rozsah testovacích technik pro důkladné otestování softwaru. MMSP a RUP jsou definované jako metodiky pro vývoj softwaru, ale MP je soustředěna výhradně na testování.

3 Závěr

Semestrální práce se věnovala porovnání disciplíny testování ve vybraných metodikách. Hlavním cílem práce bylo seznámení čtenáře se zvolenými metodikami a s procesem testování, který je nedílnou součástí každé metodiky, a jeho následné porovnání dle vybraných kritérií. Nejdříve byla sepsána základní informace o každé metodice. Dále se práce věnovala popisu disciplíny testování, jejímu významu, cílům a návaznostem na jiné procesy. Zde byla také identifikována podstata testovacího procesu, popsány odpovědnosti a aktivity jednotlivých rolí a vymezeny základní techniky testování. Na základě uvedených

informace dále bylo provedeno porovnání oblastí testování ve vybraných metodikách dle zvolených kritérií.

Po provedení rešerše testovacích procesů ve vybraných metodikách lze doporučit využití metodiky MMSP pro opravdu malé a nenáročné projekty, kde rozpočet projektu a počet lidských zdrojů striktně omezen. Na druhou stranu u rozsáhlých a náročných projektů má smysl se zabývat testováním a řízením projektu pomocí metodiky RUP. Obě metodiky můžou být rozšířeny o další komponenty které poskytuje metodika MP. Samozřejmě výběr metodiky testování hodně závisí na časovém rámci, rozpočtu, zdrojích, komplexitě řešení a dalších faktorech. Avšak je třeba aby výsledný produkt odpovídal požadované kvalitě a zároveň proces testování moc nekomplikoval celý proces dodání produktu.

4 Literatura

[1] KRÁLOVÁ, Iveta. *Metodika testování podle mezinárodních praktik a standardů*. Praha, 2013. DIPLOMOVÁ PRÁCE. Vysoká škola ekonomická v Praze. Vedoucí práce Doc. Ing. Alena Buchalcevová, Ph.D.

[2] KUČERA, Jan. *Hodnocení metodik vývoje informačních systémů z pohledu testování*. Praha, 2008. Bakalářská práce. Vysoká škola ekonomická v Praze. Vedoucí práce Doc. Ing. Alena Buchalcevová, Ph.D.

[3] International Software Testing Qualification Board: *Certified Tester Foundation Level Syllabus*. Version 2011, 31. 3. 2011, 78 s. Dostupné z: <http://www.istqb.org/downloads/viewcategory/16.html>

[4] REJNKOVÁ, Petra. *Lokalizace a přizpůsobení metodiky OpenUP*. Praha, 2011. Diplomová práce. Vysoká škola ekonomická v Praze. Vedoucí práce Doc. Ing. Alena Buchalcevová, Ph.D.

[5] REJNKOVÁ, Petra. MMSP. *MMSP: Jak s metodikou pracovat* [online]. Praha: Petra Rejnková, 2011 [cit. 2018-05-08]. Dostupné z: http://spicenter.vse.cz/wp-content/uploads/2018/02/publikovana_MMSP/index.htm