

# Řízení kvality softwaru ve společnosti XY

Bc. Michal Františ

---

Diplomová práce  
2015



Univerzita Tomáše Bati ve Zlíně  
Fakulta managementu a ekonomiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta managementu a ekonomiky  
Ústav průmyslového inženýrství a informačních systémů  
akademický rok: 2014/2015

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Michal Františ**  
Osobní číslo: **M13667**  
Studijní program: **N6209 Systémové inženýrství a informatika**  
Studijní obor: **Průmyslové inženýrství**  
Forma studia: **kombinovaná**

Téma práce: **Řízení kvality softwaru ve společnosti XY**

Zásady pro vypracování:

### Úvod

Definujte cíle práce a použité metody zpracování práce.

#### I. Teoretická část

- Zpracujte přehled teoretických východisek zabývajících se problematikou zvoleného tématu diplomové práce.

#### II. Praktická část

- Analyzujte současný stav řízení kvality softwaru ve společnosti XY.
- Vytvořte projekt pro implementaci zlepšení ve vztahu k řízení kvality softwaru ve společnosti XY.
- Provedte zhodnocení projektu.

### Závěr

Rozsah diplomové práce: **cca 70 stran**  
Rozsah příloh:  
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

**BRÍŠ, Petr. Management kvality. Vyd. 2., uprav. Zlín: Univerzita Tomáše Bati ve Zlíně, 2010, 208 s. ISBN 978-80-7318-912-9.**

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

**PATTON, Ron. Testování softwaru. Vyd. 1. Praha: Computer Press, 2002, 313 s.**

**Programování. ISBN 80-722-6636-5.**

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

**SOMMERVILLE, Ian. Software engineering. 9th ed. Boston: Pearson, 2011, 773 p. ISBN 978-013-7053-469.**

Vedoucí diplomové práce: **doc. Ing. Petr Briš, CSc.**  
Ústav průmyslového inženýrství a informačních systémů  
Datum zadání diplomové práce: **16. února 2015**  
Termín odevzdání diplomové práce: **27. dubna 2015**

Ve Zlíně dne 16. února 2015

prof. Dr. Ing. Drahomíra Pavelková  
děkanka



prof. Ing. Felicita Chromjaková, PhD.  
ředitel ústavu

## **PROHLÁŠENÍ AUTORA DIPLOMOVÉ PRÁCE**

### **Prohlašuji, že**

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen na elektronickém nosiči v příruční knihovně Fakulty managementu a ekonomiky Univerzity Tomáše Bati ve Zlíně;
- byl jsem seznámen s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

### **Prohlašuji,**

1. že jsem na diplomové pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
2. že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....  
podpis diplomanta

## **ABSTRAKT**

Diplomová práce se zabývá procesem testování softwaru s ohledem na jeho řízení a zajištění kvality. Teoretická část se věnuje základním pojmům, které souvisí s tématem diplomové práce. Je zde pojednáno o kvalitě, procesním řízení a softwaru. Posléze se práce zabývá testováním softwaru v návaznosti na metodiky vývoje softwaru, zvyšováním kvality procesu testování prostřednictvím modelu TMMi a modelováním procesů.

Obsahem analytické části je procesní analýza současného stavu procesu testování softwaru ve společnosti XY. Identifikované podprocesy jsou posléze vymodelovány a zachyceny do Karet procesů. Zjištěné nedostatky jsou předmětem návrhu řešení na zlepšení řízení kvality procesu testování softwaru. Návrh řešení je poté implementován na současný stav procesu testování softwaru s cílem dosáhnout druhé úrovně zralosti procesů testování softwaru ohodnocenou oproti modelu TMMi.

Klíčová slova: řízení kvality, software, testování, proces, TMMi, modelování

## **ABSTRACT**

This master's thesis examines the process of software testing in terms of its management and guarantee of quality. The theoretical section is concerned with basic concepts related to the topic of the master's thesis. Therein are discussed the concepts of quality, process management, and software. Afterwards, the thesis covers software testing in relation with the methodology of software development, increasing the quality of the testing process through the TMMi model, and modeling processes.

The content of the analytical section is the process analysis of the current state of the software testing process in company XY. Identified sub-processes are then modeled and captured in the Card processes. Discovered shortcomings are the subject of a proposed solution for improving the management of the quality of the software testing process. The proposed solution is thereafter implemented into the current state of the software testing process, with the aim of reaching the second level of development in the software testing process evaluated in comparison with the TMMi model.

Keywords: control quality, software, testing, process, TMMi, modeling

Chtěl bych tímto způsobem poděkovat svému vedoucímu diplomové práce panu doc. Ing. Petru Brišovi, CSc. za odborné vedení, věcné připomínky a veškerou pomoc při psaní diplomové práce.

Dále bych rád poděkoval mé přítelkyni, která mě během celého studia podporovala a věřila v jeho zdárné dokončení.

# OBSAH

<b>ÚVOD</b> .....	<b>9</b>
<b>CÍLE A METODY ZPRACOVÁNÍ PRÁCE</b> .....	<b>10</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>13</b>
<b>1 ZÁKLADNÍ POJMY</b> .....	<b>14</b>
1.1 CO JE KVALITA .....	14
1.2 DEFINICE SOFTWARE.....	15
1.3 CHARAKTERISTIKA PROCESNÍHO ŘÍZENÍ.....	16
1.3.1 Definice pojmu proces .....	16
1.3.2 Vztah procesu k projektu a jeho charakteristické prvky .....	17
1.3.3 Základní atributy procesu.....	17
1.3.4 Kategorizace procesů .....	18
<b>2 TESTOVÁNÍ SOFTWARE</b> .....	<b>20</b>
2.1 DEFINICE TESTOVÁNÍ .....	20
2.2 TYPY TESTŮ .....	21
2.3 DEFEKT SOFTWARE .....	24
<b>3 ŘÍZENÍ A ZAJIŠTĚNÍ KVALITY SOFTWARE</b> .....	<b>26</b>
3.1 METODIKY VÝVOJE SOFTWARE VE VZTAHU K JEHO TESTOVÁNÍ.....	27
3.1.1 Vývoj softwaru prostřednictvím modelu vodopádu.....	28
3.2 METODY ZVYŠOVÁNÍ KVALITY SOFTWARE .....	29
3.3 ZVYŠOVÁNÍ KVALITY PROCESU TESTOVÁNÍ PROSTŘEDNICTVÍM MODELU TMMi .....	30
3.3.1 Struktura modelu .....	31
3.4 MODELOVÁNÍ SOFTWAREVÝCH PROCESŮ .....	34
3.4.1 Business Process Model and Notation .....	35
<b>II PRAKTICKÁ ČÁST</b> .....	<b>37</b>
<b>4 PŘEDSTAVENÍ SPOLEČNOSTI</b> .....	<b>38</b>
4.1 MISE FINANČNÍ SKUPINY .....	38
4.2 VIZE FINANČNÍ SKUPINY.....	38
4.3 HODNOTY FINANČNÍ SKUPINY .....	38
4.4 ORGANIZAČNÍ STRUKTURA SPOLEČNOSTI.....	39
4.5 PRODUKTY .....	40
4.6 PŘEDSTAVENÍ SOFTWARE POSKYTUJÍCÍ SOFTWAREVOU STRUKTURU PRO PRODUKT AUTOÚVĚR .....	41
<b>5 ANALÝZA SOUČASNÉHO STAVU ŘÍZENÍ KVALITY SOFTWARE PROSTŘEDNICTVÍM TESTOVÁNÍ SOFTWARE VE SPOLEČNOSTI</b> .....	<b>43</b>
5.1 IDENTIFIKACE A POPIS SOUČASNÉHO STAVU PROCESU TESTOVÁNÍ .....	45
5.1.1 Test Planning.....	45
5.1.2 Test Development .....	49
5.1.3 Test Implementation.....	52
5.1.4 Test Execution.....	55
<b>6 NÁVRH ŘEŠENÍ</b> .....	<b>61</b>

6.1	METRIKY URČENÉ K MĚŘENÍ PROCESU TESTOVÁNÍ SOFTWARE .....	63
6.2	ÚPRAVA STÁVAJÍCÍCH PODPROCESŮ .....	70
6.2.1	Test Planning .....	70
6.2.2	Test Development .....	71
6.2.3	Test Implementation.....	72
6.2.4	Test Execution.....	73
6.3	VYTVOŘENÍ CHYBĚJÍCÍCH PODPROCESŮ .....	74
6.3.1	Test Management .....	75
6.3.2	Test Evaluation.....	76
<b>7</b>	<b>NÁVRH ŘEŠENÍ PRO ZLEPŠENÍ ŽÍZENÍ KVALITY PROCESU TESTOVÁNÍ SOFTWARE VE SPOLEČNOSTI XY.....</b>	<b>80</b>
7.1	CÍLE PROJEKTU .....	80
7.2	ČASOVÝ HARMONOGRAM JEDNOTLIVÝCH ČINNOSTÍ PROJEKTU.....	80
7.3	ANALÝZA ZDROJŮ .....	81
7.4	NÁKLADOVÁ ANALÝZA PROJEKTU .....	82
7.5	ANALÝZA RIZIK PROJEKTU .....	84
7.5.1	Identifikace rizik .....	84
7.5.2	Ohodnocení rizik a jejich eliminace.....	85
7.6	VYHODNOCENÍ PROJEKTU .....	86
7.6.1	Vyhodnocení procesní oblasti PA1 – Test Policy and Strategy.....	88
7.6.2	Vyhodnocení procesní oblasti PA2 – Test Planning.....	89
7.6.3	Vyhodnocení procesní oblasti PA3 –Test Monitoring and Control.....	91
7.6.4	Vyhodnocení procesní oblasti PA4 – Test Design and Execution.....	92
7.6.5	Vyhodnocení procesní oblasti PA5 – Test Enviroment .....	94
7.6.6	Sumarizační report TMMi.....	95
	<b>ZÁVĚR .....</b>	<b>98</b>
	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>100</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>102</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>103</b>
	<b>SEZNAM TABULEK.....</b>	<b>104</b>
	<b>SEZNAM PŘÍLOH.....</b>	<b>106</b>



## ÚVOD

Proces testování softwaru je nedílnou součástí životního cyklu vývoje softwaru a úzce propojuje nám dobře známá a skloňovaná slova jako je kvalita, konkurenceschopnost, dynamika či vývoj. Software je všude kolem nás. S jeho selháváním se setkáváme každý den v obchodních centrech, kde na pokladnách platíme astronomické částky za kilo pomerančů, na bankovních účtech, kde nalézáme nesprávnou výši zůstatku či shledáváme hlášení, že program přestal pracovat. Velmi častá bývá také situace, kdy je naše celodenní práce nenávratně pryč a to jen proto, že jsme spoléhali na bezchybnost a kvalitu využívaného softwaru.

Pokud se ohlédneme do minulosti, defekty které nebyly objeveny v průběhu testování softwaru, způsobily nemalé problémy. Například v roce 1999 zapříčinily roztržštění přistávacího modulu NASA na Mars. O 8 let dříve neobjevený defekt způsobil časové zpoždění obranného raketového systému Patriot, díky čemuž přišla o život skoro třicítka amerických vojáků.

Kvalita softwaru a kvalita testovacího procesu je jednou z hlavních konkurenčních výhod, které mohou současné společnosti zabývající se vývojem softwaru, získat. V současném turbulentním prostředí tak vznikají nové standardy zabývající se dosažením potřebné úrovně kvality, nové nástroje sloužící k efektivnějšímu řízení vývoje softwaru potažmo procesu testování. Tlak na rychlost, nízké náklady a kvalitu tak dává vzniknout novým sofistikovaným nástrojům a přístupům, které ve vztahu k testování lze aplikovat pro technologické oblasti cloudových aplikací, mobilních platforem, automatizace aj.

Diplomová práce se zabývá právě procesem testování softwaru, který je v současné době stejně důležitý jako vývoj softwaru, i když je mnohdy širokou neodbornou veřejností vnímán jako nutná součást životního cyklu vývoje softwaru s cílem nalézání chyb. Tento chybný předpoklad počítá s tím, že testování softwaru slouží k pouhému nalezení chyb. Avšak účelem celého procesu testování je předcházení chybám, které jsou způsobeny defektem softwaru. Výsledkem takového přístupu je kvalitní softwarový produkt, který je podpořen kvalitou samotného procesu testování. Možnost deklarovat určitou úroveň řízení kvality neboli zralosti procesu testování softwaru navyšuje konkurenceschopnost na dynamicky se vyvíjejícím softwarovém trhu.

## CÍLE A METODY ZPRACOVÁNÍ PRÁCE

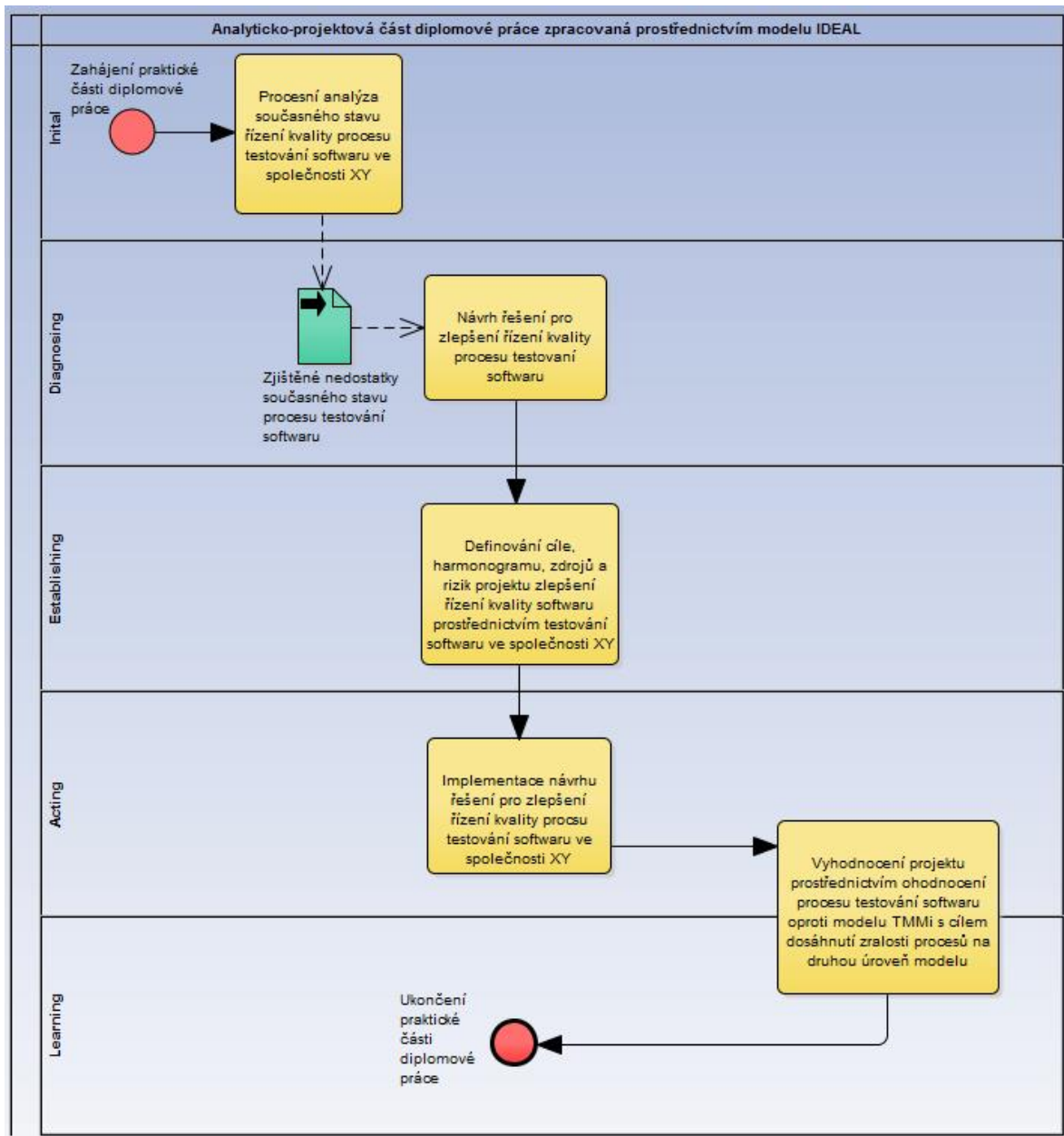
Cílem diplomové práce je dosažení druhé úrovně zralosti neboli kvality procesu testování softwaru ohodnocenou oproti modelu TMMi. Autorem zvolený cíl má za úkol podpořit společnost XY, která si klade za cíl zisk konkurenční výhody oproti ostatním členům finanční skupiny, čímž bude umožněno poskytovat softwarovou infrastrukturu svého finančního produktu Autoúvěr na nově vstupující americký trh. Možnost vyvíjet softwarovou infrastrukturu svého produktu Autoúvěr pro americký trh by přinesla transfer znalostí a know-how ze softwarových projektů z České republiky a Slovenska, čímž by došlo k podpoře hlavních cílů finanční skupiny.

Cílem společnosti tedy není pouhá redukce vynaložených nákladů na softwarové projekty inovující softwarovou infrastrukturu produktu Autoúvěr, zvýšení zisku prostřednictvím nárůstu počtu profinancovaných vozů či zkrácení doby testování softwaru. Ústředním cílem společnosti je dosažení konkurenční výhody umožňující proniknout na americký trh a podpořit transfer znalostí nabytých ze současných softwarových projektů realizovaných v České republice a na Slovensku s možnou budoucí vidinou uvedených efektů, tedy nárůst profinancované částky, redukce vynaložených nákladů na testování s podporou kvalitního procesu plánování a odhadech pracnosti.

Dosažení žádoucí úrovně zralosti procesu testování je cílem analyticko-projektové části diplomové práce. Stanovený cíl je podpořen návrhem řešení na zlepšení řízení kvality procesu testování softwaru. Vedlejším předpokládaným efektem návrhu na zlepšení řízení kvality procesu testování softwaru je dosažení nižšího počtu defektů a s tím spojených nákladů na neoprávněnou reklamaci vůči dodavateli softwaru a vyšší kvalitu vyvíjeného softwaru.

Analyticko-projektová část diplomové práce bude strukturována prostřednictvím aplikace modelu IDEAL vyvinutého institutem SEI (Software Engineering Institute). První Initial (Počáteční) a druhá Diagnosing (Diagnostická) fáze modelu vychází z procesní analýzy současného stavu. Tímto krokem je provedena komparace současného a požadovaného stavu, identifikace nedostatků a definice důvodů vedoucích k aplikování změn. Třetí fáze modelu IDEAL nesoucí název Establishing (Ustavovací) má za cíl vytvořit pracovní plán projektu a navázat na analytickou část projektu. Čtvrtá fáze Acting (Vykonávací) představuje implementaci návrhu řešení pro zlepšení řízení kvality procesu testování softwaru ve společnosti XY, který byl vytvořen v analytické části projektu. V páté fázi Learning

(Učení) probíhá vyhodnocení projektu pomocí ohodnocení procesu testování softwaru oproti modelu TMMi.



Obr. 1. Logická výstavba praktické části diplomové práce (vlastní zpracování)

V praktické části diplomové práce bude pro analytickou část projektu využita metoda analýzy rozčleňující proces testování softwaru jako celek na dílčí podprocesy. Uplatňující se technikou sběru dat bude procesní analýza.

Pro projektovou část diplomové práce bude využita metoda syntézy s induktivním přístupem, kdy budou užity poznatky získané v analytické části k návrhu řešení pro zlepšení

řízení kvality procesu testování softwaru ve společnosti XY. Aplikovanou analytickou technikou v této projektové části bude hodnocení úrovně zralosti procesů v modelu TMMi.

## **I. TEORETICKÁ ČÁST**

## 1 ZÁKLADNÍ POJMY

Cílem úvodní kapitoly je uvést čtenáře do souvislostí, které jsou typické pro zvolené téma diplomové práce. Budou představeny základní pojmy, které prostupují prací a pomáhají čtenáři vytvořit celistvý pohled na předloženou tematiku. Jako první pojem bude představena kvalita. Následovat bude definice softwaru a stručná charakteristika procesního řízení.

### 1.1 Co je kvalita

Pojem kvalita je využíván v současné době zcela běžně. Pro každého z nás ovšem může mít odlišný význam. Snahy sjednotit nejednotné definice a myšlenky daly vzniknout řadě mezinárodních norem, jež jsou kontinuálně revidovány a upravovány. Normou rozumíme společně dohodnutá doporučení stanovující základní požadavky na kvalitu, vhodnost použití k zamýšlenému účelu a případně i bezpečnost výrobku, procesu nebo služby. (Roudenský, 2013, s. 21-31) Normalizační proces na nadnárodní úrovni zabezpečují především tyto organizace:

- ISO – Mezinárodní organizace pro normalizaci,
- IEC – Mezinárodní elektrotechnická komise,
- IEEE – Institut pro elektrotechnické a elektronické inženýrství.

V České republice zajišťuje vydávání norem Úřad pro technickou normalizaci, metrologii a státní zkušebnictví.

Dle mezinárodní normy ISO 9000:2005, která popisuje základní principy systému managementu kvality, je kvalita (jakost) definována, jako stupeň splnění požadavků souborem inherentních charakteristik. Norma posléze vysvětluje pojmy, jež jsou v této definici použity. Za inherentní charakteristiky jsou vnímány takové charakteristiky, které jsou vnitřní, trvalé, neoddělitelné. Druhým důležitým pojmem jsou požadavky neboli potřeby či očekávání, které jsou stanoveny, obecně se předpokládají nebo jsou závazné. Takovéto požadavky definují zainteresované strany, mezi které řadíme např. zákazníky, vlastníky, zaměstnance, dodavatele, konkurenci či legislativu. Samotná norma vnímá pojem jakost za synonymum ke slovu kvalita a pro potřeby této práce bude uvedená jednotnost využívána.

Ve všech oblastech lidské činnosti roste tlak na kvalitu a ta se tak stává nástrojem konkurenceschopnosti produktu. Ovšem samotná kvalita v dnešní době nestačí. Musí být splněny

požadavky na další 3 atributy, které ve spojení tvoří tzv. Čtverec úspěšnosti. (Briš, 2010, s. 7-12) Doplňujícími atributy jsou čas, cena a kvantita.

Různorodost činností, které vykonávají výrobní i nevýrobní společnosti si postupem času vyžádaly řadu alternativních přístupů k managementu kvality. Tyto přístupy lze rozčlenit do následujících čtyř koncepcí:

- koncepce založena na vlastním přístupu,
- koncepce na bázi odvětvových standardů,
- koncepce postavená na mezinárodních normách ISO
- a koncepce managementu kvality na bázi TQM. (Nenadál, 2007, s. 25-48)

S pojmem kvalita se pojí management kvality a produktu. Dle mezinárodní normy ISO 9000:2005 je management kvality soubor koordinovaných činností pro usměrňování a řízení organizace s ohledem na kvalitu. Tato norma definuje pojem produkt, jako výsledek procesu. Lze tedy říci, že je produkt výsledkem souboru vzájemně souvisejících nebo vzájemně působících činností, které přeměňují vstupy na výstupy. A právě procesy jsou považovány za rozhodující faktor zabezpečení kvality produktů.

Produktem rozumíme výrobek či službu. Z povahy diplomové práce bude koncovým produktem software, kterému je věnována další podkapitola diplomové práce.

## 1.2 Definice softwaru

Při charakteristice pojmu software vycházíme z definice, která byla uveřejněna v roce 1991 institucí IEEE. Za software jsou považovány počítačové programy a jeho související dokumentace. Takové to produkty jsou vytvářeny nejen pro konkrétního zákazníka, ale také pro širokou veřejnost. (Galín, 2004, s. 14-35) Tato definice je posléze přejímána a rozšiřována Mezinárodní organizací pro normalizaci a vyskytuje se ve všech publikovaných normách, vztahujících se k softwaru. Software je rovněž označován za programové vybavení. Záměrně není programové vybavení vázané na počítače, protože se v dnešní době stírají hranice mezi počítačem, notebookem, tabletem, mobilem a dokonce i mezi tzv. informačnickými technologickými nositelnosti, mezi které řadíme chytré brýle, hodinky či sluchátka. Ve všech těchto zařízeních je jistý druh softwaru, který spolu s hardwarem tvoří informační technologie. (Vymětal, 2009)

Software lze členit na dvě základní kategorie. První kategorie tvoří systémový software, který zajišťuje samotný chod zařízení, zpravidla se jedná o operační systém. Druhou kate-

gorii tvoří aplikační software, se kterým se můžeme setkat v podobě kancelářských, grafických či vývojových programů.

### 1.3 Charakteristika procesního řízení

Významným faktorem úspěchu současných podnikatelských subjektů nacházejících se v informační společnosti je schopnost systematicky identifikovat a řídit procesy a jejich vzájemné působení. (Hromková, 2008, s. 47-50) Trvalé zajištění maximální výkonnosti je realizováno prostřednictvím neustálého zlepšování a optimalizace podnikových i mezipodnikových procesů. Cílem procesního přístupu k řízení podniku je odkrytí procesy, které jsou překryty funkční organizací, tyto procesy oprostít od všech činností, jež nepřidávají hodnotu, učinit je středem pozornosti a vytvářet infrastrukturu a podnikovou kulturu, která umožní hladké vykonávání a neustálé zlepšování stávajících procesů a podle potřeby tvorbu a neustálé zlepšování nových procesů. (Šmída, 2007, s. 29-38) Procesy představují způsob naplnění základního principu procesního řízení, tedy dosažení požadovaného produktu.

#### 1.3.1 Definice pojmu proces

Existuje mnoho různých vymezení pojmu proces, které byly publikovány řadou autorů. Níže uvedené definice se liší svou přesností či úplností, ale vždy mají určitou skupinu společných znaků:

- Proces je organizovaná skupina vzájemně souvisejících činností, které společně vytvářejí hodnotu pro zákazníka. (Hammer, 2002, s. 62)
- Proces je soubor provázaných činností, které vezmou vstup, transformují jej a vytvoří výstup. (Carr, 1995)
- Proces je ohraničená skupina vzájemně provázaných pracovních činností s předem definovanými vstupy a výstupy. Má jasně a přesně definovaný začátek a konec. (Nenadál, 2007, s. 13)
- Proces je definovaný sled činností, který je vykonáván za účelem přidání hodnoty. (Hromková, 2008, s. 48)
- Proces je organizovaná skupina vzájemně souvisejících činností a/nebo subprocesů, které procházejí jedním nebo více organizačními útvary či jednou (podnikový proces) nebo více spolupracujícími organizacemi (mezipodnikový proces), které spotřebovávají materiální, lidské, finanční a informační vstupy a jejich výstupem



je produkt, který má hodnotu pro externího nebo interního zákazníka. (Šmída, 2007, s. 29)

Poslední uvedená definice reflektuje autorem často opomíjený fakt, že kromě činností se může proces skládat i z podprocesů a dále pak to, že procesy procházejí napříč organizačními útvary nebo dokonce napříč spolupracujícími organizacemi.

### 1.3.2 Vztah procesu k projektu a jeho charakteristické prvky

Z publikovaných definicí vyplývá, že proces probíhá opakovaně a sekvenčně. Tento základní fakt tvoří rozdíl mezi procesem a projektem. Nicméně procesy a projekty spolu souvisí a mají společné rysy, mezi které řadíme cíl, vstupy, výstupy, definované činnosti, časový harmonogram atd. Vzájemný vztah je vyjádřen prostřednictvím změny procesu, která je uskutečněna pomocí projektu. V opakujícím procesu jsou identifikovány možnosti ke zlepšení, které je zapotřebí do procesu zavést. Menší změny může dle svých pravomocí a schopností zavést vlastník procesu. V případě, že se jedná o změny velkého charakteru, je zapotřebí běžící proces zastavit. Přerušování procesu musí být uskutečněno na co nejkratší dobu s co nejmenšími náklady. Z tohoto důvodu je zapotřebí před zastavením procesu stanovit odpovědnosti, činnosti a časový harmonogram. Pokud je vše připraveno, proces je zastaven s následnou realizací projektu. Výsledkem projektu je jednorázová implementace změn do procesu, spuštění zlepšeného procesu a opakované vykonávání stupně řízení zlepšeného procesu. (Šimonová, 2009. s. 49-82)

### 1.3.3 Základní atributy procesu

K základním charakteristikám procesu patří:

- Zákazník procesu je organizace, řídící se procesem nebo definováním pracovních rolí v organizaci (vlastník procesu), pro kterého jsou výsledky procesu určeny.
- Vlastník procesu je pracovní role v hierarchii organizace, která disponuje odpovědností a pravomocí k dosahování cílů procesu, monitorování jeho výkonnosti, plnění konkrétních ukazatelů, řešení problémů při realizaci procesu, systematické zlepšování a správu procesu. Působení odpovědností je ve vztahu k výsledku procesu, nikoliv k vykonávaným činnostem.
- Vstup procesu je generován z výstupů předcházejících procesů. Při spuštění procesu na jeho vstupu dochází ke spotřebě zdrojů pro přeměnu vstupů na výstupy. Proces je zpracováván a oproti vstupu je obohacen o přidanou hodnotu.

- Výstup procesu je charakterizován jako pracovní výsledek (výkon) činností z daného procesu ve formě produktu, shodný se vstupem do procesu následujícího.
- Rizika procesu jsou stav či událost při průběhu procesu, která má nežádoucí dopad na dosažení cíle a zabezpečení výsledku procesu.
- Regulátory řízení jsou závazná pravidla (zákony, vyhlášky, normy) s dočasnou nebo trvalou platností, které je pro provádění procesů nezbytné dodržovat.
- Činnosti procesu představují posloupnost pracovních úkonů prováděných v rámci organizační jednotky.
- Zdroje procesu jsou spotřebovány při přeměně vstupů do procesu na výstupy (materiál, technologie, finanční prostředky, lidské zdroje, informace a čas).
- Rozhraní procesu představuje vzájemnou vazbu mezi procesy, kdy výstup z předcházejícího procesu odpovídá vstupu do procesu následujícího. (Grasseová, 2008)
- Cíle nebo účel procesu obsahující odpověď na otázku, proč vlastně proces probíhá.
- Metriky neboli sledované parametry úspěšnosti procesu. (Gála, 2006, s. 169)

#### 1.3.4 Kategorizace procesů

Jeden z nejčastěji využívaných členění procesů vychází z Porterova modelu hodnotového řetězce a je používáno při zavádění norem ISO. Členění procesů jen následující:

- hlavní procesy,
- řídicí procesy
- a podpůrné procesy.

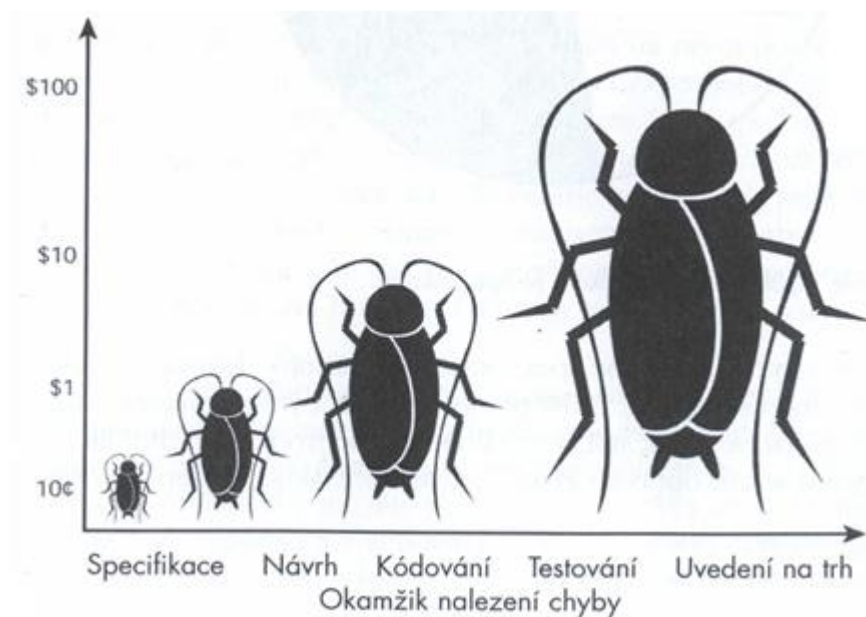
Hlavní procesy jsou hodnotovné procesy zajišťující splnění poslání společnosti, ve kterých přímo vzniká hodnota k uspokojení externího zákazníka, jsou tvořeny řetězcem přidané hodnoty, který představuje klíčovou oblast podnikání společnosti. (Hromková, 2008, s. 49) S ohledem na téma diplomové práce věnující se řízením kvality procesu testování softwaru lze za hlavní procesy označit přípravu testovacího prostředí a dat. Taktéž zde spadá nejdůležitější proces, kterým je vykonávání samotných testů.

Řídicí procesy jsou průřezové procesy pro zajištění říditelnosti a stabilizace společnosti, určují a zabezpečují rozvoj a řízení výkonu společnosti a vytváří podmínky pro fungování ostatních procesů tím, že zajišťují řízení a integritu firmy. (Hromková, 2008, s. 49) S ohledem na proces testování softwaru zde řadíme plánování, návrh testů či vyhodnocení testování.

Podpůrné procesy jsou procesy zajišťující produkt vnitřnímu zákazníkovi, nebo hlavnímu procesu, který je ale možné zajistit i externě bez ohrožení poslání společnosti. Tyto procesy, pokud jsou tedy vykonávány interně, jsou realizovány z důvodu ekonomické výhodnosti či minimalizace konkrétních rizik. Tyto procesy pouze zajišťují podmínky pro fungování ostatních procesů tím, že jim dodávají produkty, ale přitom nejsou součástí hlavních procesů. (Hromková, 2008, s. 49) Často se do této skupiny zařazují s ohledem na testování technologické zdroje a řízení či týmy specializující se na zajišťování kvality.

## 2 TESTOVÁNÍ SOFTWARE

Pro širokou veřejnost je proces testování vnímán mnohdy jako podružná činnost, která je vykonávána za účelem potvrzení funkčnosti softwaru. Opak je ovšem pravdou. Testování je nejen velmi významnou částí řízení kvality softwaru, ale taktéž vede ke snížení či k úplné eliminaci dodatečných nákladů. Pro testování platí pravidlo, že čím později jsou defekty v softwaru odhaleny, tím vyšší jsou náklady na jejich odstranění. Tento vztah je znázorněn níže na obrázku.



Obr. 2. Náklady na opravu chyb v časovém horizontu (Patton, 2002, s. 16)

### 2.1 Definice testování

Testování softwaru chápeme jako proces řízeného spouštění softwarového produktu s cílem zjistit, zda splňuje specifikované či implicitní potřeby uživatele. Za řízený proces testování je považován takový proces, který je prováděn vždy s určitým záměrem, předchází mu plánování ve formě návrhů testů a po něm následuje jejich vyhodnocení. Testování se zabývá zkoumáním softwarového produktu, čímž získává informace o jeho kvalitě, kterou chápeme jako stupeň naplnění požadavků a přání uživatelů. (Roudenský, 2013, s. 45-57)

Vedlejším produktem sběru a analýzy informací při procesu testování je nalezení defektů, o kterých bude pojednáno v samostatné podkapitole. Mnoho starších publikací uvádí, že

právě nalézání defektů je jediným cílem testování. Avšak vývoj vyspělosti testovacího procesu pokročil a testování je vnímáno jako nástroj sloužící k předcházení vzniku defektů.

Testování je proces skládající se z posloupnosti činností, u kterých je zapotřebí určit, kdy se mají provést. Rovněž se zde nachází definované vstupy a výstupy z dílčích činností. Proces testování má vydefinované odpovědnosti rolí, které se podílejí na činnostech.

## 2.2 Typy testů

Testy lze rozdělit dle několika hledisek:

- dle míry znalosti programového kódu,
- dle způsobu provádění testů,
- dle rozměru kvality, které testy ověřují
- a dle fáze vývoje softwaru.

Při testování lze pracovat s různým objemem informací potažmo znalostí, které jsou využívány. Na základě míry znalostí programového kódu rozdělujeme testování na následující přístupy:

- přístup černé skříňky,
- přístup bílé skříňky
- a přístup šedé skříňky.

Testy založené na přístupu černé skříňky odráží fakt, že neznáme strukturu programového kódu a ani jeho přesnou funkcionalitu. Při testování využíváme znalosti o vstupech a požadovaných výstupech. Neznáme, jak software pracuje. Opakem je přístup, založený na testování bílé skříňky, kdy víme, jak programový kód funguje a známe jeho strukturu. Kombinací obou výše uvedených vzniká přístup nazýván testování šedé skříňky. (Galín, 2004, s. 178-216)

Druhou známou kategorizací je členění testů dle způsobu provádění. Jedná se o manuální nebo automatizované testování. Manuální testování provádí tester, který posléze zaznamenává výsledky testování. Při manuálním testování využívá tester své myšlení, kreativitu a bázi znalostí. Oproti tomu automatizované testování provádí specializovaný software s cílem zvýšení efektivity prováděných testů. Automatizovaný test je tedy proveden opakovaně, rychle a bezchybně. Automatizované testování je velmi užitečnou doplňkovou aktivitou k manuálnímu testování, která si neklade za cíl manuální testování nahradit z výše uvedených důvodů. (Roudenský, 2013, s. 61-70)

Další důležité členění testů se odvíjí se zřetelem na hledisko rozměru kvality softwaru, které definovala společnost Hewlett-Packard prostřednictvím modelu kvality FURPS. Název modelu je složen z počátečních písmen anglických slov označující jednotlivé rozměry kvality softwaru:

- Functionality (funkčnost),
- Usability (použitelnost),
- Reliability (spolehlivost),
- Performance (výkonnost),
- Supportability (podporovatelnost).

Prvním rozměrem kvality softwaru je funkčnost. Do této kategorie spadají následující testy:

- testy funkčnosti ověřující, zda funkcionalita softwaru odpovídá specifikacím při jeho návrhu,
- testy zabezpečení zabývající se ověřením přístupových práv k funkčnosti či datům,
- testy objemu zjišťující stabilitu funkčnosti softwaru při vysokém objemu dat v databázích.

Druhým rozměrem je použitelnost. Zde se nachází testy zaměřené na uživatelské rozhraní, školící materiály či dokumentaci k softwaru.

Spolehlivost je třetím rozměrem kvality softwaru. K testům spolehlivosti náleží zejména:

- testy integrity zkoumající odolnost proti selhání po datové a funkční stránce,
- testy struktury softwaru zaměřené na otestování webového rozhraní,
- zátěžové testy ověřující stabilitu a funkčnost softwaru při nestandardních podmínkách jako je například snížení operační paměti aplikačního serveru.

Při čtvrtém rozměru kvality se testuje výkonnost softwaru prostřednictvím následujících testů:

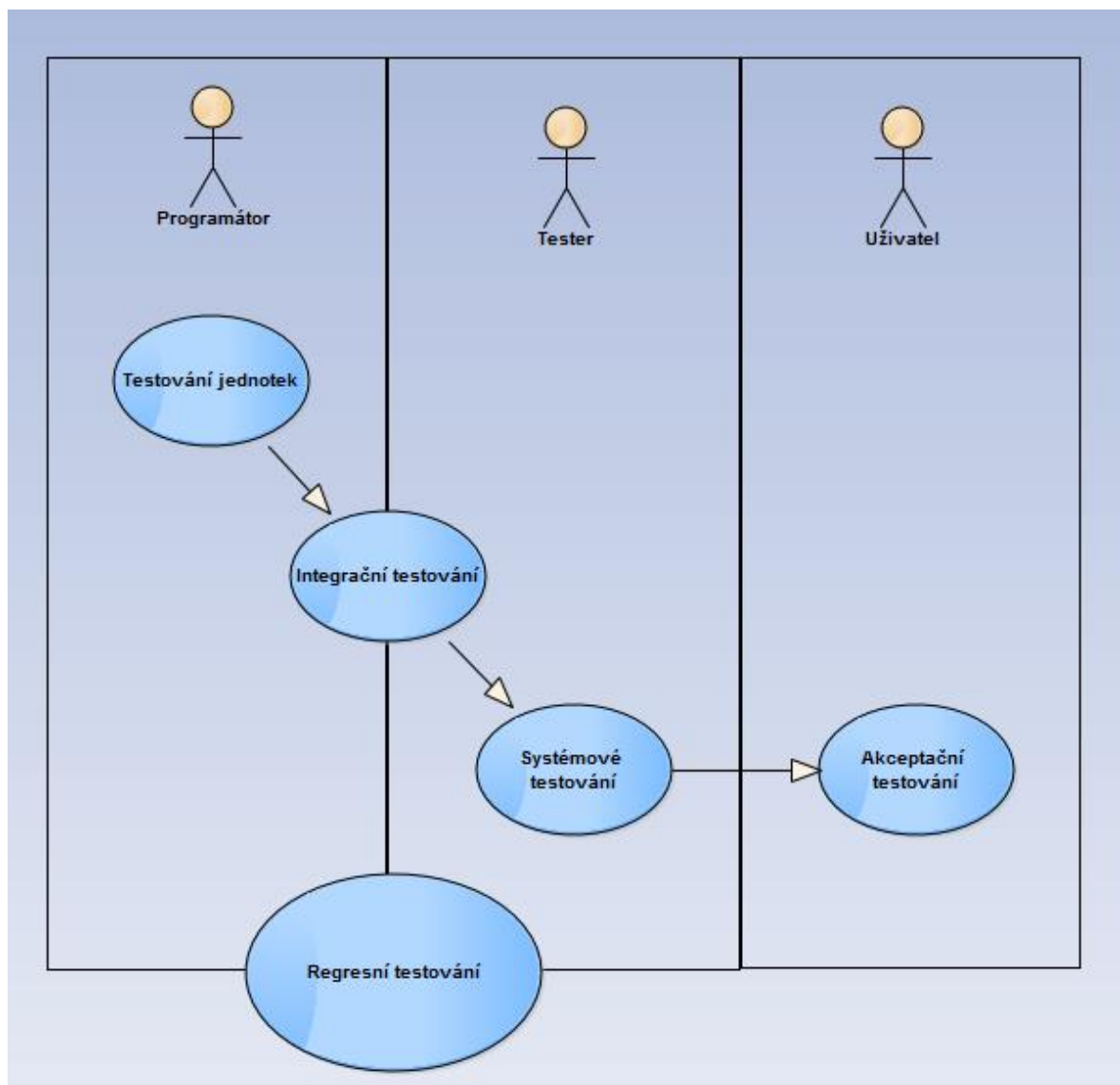
- měření odezev softwaru a nalézání výkonnostně úzkých míst,
- testování výkonu pod neobvyklou zátěží způsobenou vysokým počtem současně přistupujících a pracujících uživatelů.

Posledním rozměrem kvality softwaru v modelu FURPS je testování podporovatelnosti. Prostřednictvím níže uvedených testů ověřujeme stupeň podpory údržby softwaru:

- testy konfigurace ověřující, zda software bude plně funkční i na jiných hardwarových a softwarových konfiguracích, které jsou jiné než současné prostředí
- testy instalace softwaru na různé hardwarové a softwarové konfigurace. (Zelinka, 2013, s. 4-36)

Posledním důležitým členěním testů je dle jejich časového výskytu v procesu vývoje softwaru. Do této kategorie přísluší:

- Testování jednotek (Unit testing).
- Integrovaní testování (Integration testing).
- Systémové testování (System testing).
- Akceptační testování (Acceptance testing). (Roudenský, 2013, s. 89-96)



Obr. 3. Členění testů dle časového hlediska (vlastní zpracování)

Testování jednotek je prvním typem testování, které probíhá ze strany vývojového týmu zpravidla složeného z programátorů. Cílem jednotkových testů je otestovat jednotku při úplné izolaci od ostatních jednotek a prokázat, že se chová korektně. Za jednotky jsou považovány funkce, procedury, metody či třídy.

Po vytvoření a otestování dílčích jednotek programového kódu přichází na řadu integrační testování. Principem integračního testování je propojení izolovaných jednotek a otestování jejich vzájemné vazby. Postupně jsou integrovány všechny dílčí jednotky do jednoho celku, který musí být funkční a stabilní. Na integračních testech se již podílí kromě programátorů i specializovaní testeři.

Předposlední kategorií jsou systémové testy. Jedná se o nejdůležitější fázi testů, kdy se ověřuje, zda software splňuje deklarované požadavky zákazníka. Při systémovém testování je využívána velká část testů popsaných v modelu kvality FURPS. Systémové testování je taky pomyslnou hranicí mezi koncovým uživatelem a testerem, který má za úkol odchytnout a opravit defekty, které by objevil zákazník.

Akceptační testy jsou označovány pod zkratkou UAT. Jedná se o akronymum složené z počátečních písmen anglických slov User Acceptance Testing – neboli akceptační testování uživatele, respektive zákazníka. Cílem akceptačního testování je ověřit, zda předkládaný software splňuje akceptační kritéria zákazníka. Pokud ano, je software předán uživateli a jedna životní etapa softwaru se uzavírá.

Z obrázku číslo 3 je patrné, že v průběhu jednotkových, integračních a systémových testů probíhá tzv. regresní testování. Účelem regresního testování je ověřit, že změněná část softwaru neovlivní dosavadní funkčnost ostatních částí softwaru. K regresním testům patří i smoke a sanity testování. Smoke testování probíhá na základní množině funkcionalit, oproti tomu sanitní testování je zaměřeno na specifickou část.

### **2.3 Defekt softwaru**

Defektem v softwaru rozumíme vadu nacházející se v programovém kódu či datech, která je nejčastěji způsobena programátorem. Původ defektu je různý. Může se jednat o chybu v programovém kódu, chybný návrh softwaru, nepochopení specifikací zákazníka nebo se může dokonce jednat o plánovanou sabotáž. Zanesený defekt v softwaru je ve výchozím stavu neaktivní. Aktivace defektu způsobí odchýlení softwaru od jeho požadovaného stavu.



Tato situace je označována chybou. V případě, že bude chyba pro koncového uživatele viditelná, dochází k selhání softwaru. (Roudenský, 2013, s. 31-43)

Za situace vedoucí k chybám považujeme následující skutečnosti:

- software nedělá, co má dělat dle definované specifikace,
- software dělá, co nemá dělat dle definované specifikace,
- software dělá něco, co není specifikováno,
- software nedělá něco, co není specifikováno, ale asi by to dělat měl,
- software je nesrozumitelný, špatně se s ním pracuje, je pomalý apod.

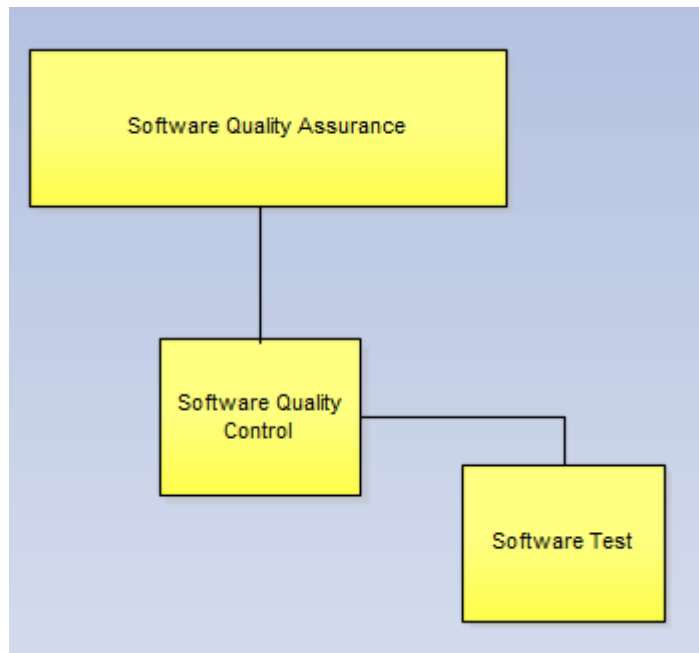
Pokud bychom chtěli vyjádřit poměr dopadu jednotlivých činností na vznik chyby, na prvním místě bude nedostatečné pochopení specifikace. Z 10 možných chyb připadá právě 6 chyb na specifikaci. Následuje návrh, který odráží skutečnosti definované ve specifikaci. Z 10 možných chyb připadají 3 chyby právě na část návrhu. (Patton, 2002, s. 267-278)

### 3 ŘÍZENÍ A ZAJIŠTĚNÍ KVALITY SOFTWARE

Řízení kvality softwaru (SQC) je zaměřeno na výstupy z jednotlivých procesů, u kterých ověřuje, zda odpovídají specifikacím a požadavkům. (Roudenský, 2013, s. 21-28) Za výstupy jsou považovány dílčí či výsledné produkty, které jsou například reprezentovány softwarovou dokumentací, programovým kódem, ale také defektem. K základním technikám využívaných při řízení kvality softwaru patří revize, inspekce, testování, simulace, verifikace, validace a audit. Ačkoliv je verifikace a validace jednou z výše uvedených technik, jsou taktéž součástí procesu testování.

V praxi jsou hojně pojmy verifikace a validace zaměňovány, což je dáno koncepcí a vzájemnou závislostí uvedených technik. Dle mezinárodní normy ISO/IEC 12207:2008 je verifikace proces ověření, zda dílčí produkt vývoje softwaru splňuje specifické požadavky a podmínky, které na něj byly kladeny. K jednotlivým aktivitám řadíme verifikaci požadavků, verifikaci návrhu, verifikaci zdrojového kódu, verifikaci integrace, verifikaci dokumentace. V téže mezinárodní normě nalezneme i definici validace. Jedná se o potvrzení, že vyvíjený produkt je správný a plní zamýšlené použití. Při vývoji softwaru se tak může stát, že software vyvíjíme správně a bude verifikován, ale nebude akceptován ze strany uživatele, tedy nebude validní.

Řízení kvality softwaru spadá pod kategorizaci nazývanou zajišťování kvality softwaru (SQA). Na rozdíl od řízení kvality softwaru, které je zaměřeno na výsledný produkt, se zajišťování kvality softwaru orientuje na kvalitu samotných procesů. A právě testování je jedním z mnoha procesů probíhajících při vývoji softwaru, který nám umožňuje dosáhnout kvalitního produktu. Dle definice zakotvené opět v mezinárodní normě ISO/IEC 12207:2008 je zajišťování kvality softwaru proces, který poskytuje dostatečnou záruku, že všechny procesy vztahující se k životnímu cyklu softwaru budou vyhovovat stanoveným kritériím. Na níže uvedeném obrázku vidíme, že proces testování je na nejnižší rozlišovací úrovni a spadá pod řízení kvality softwaru a současně je součástí procesu zajištění kvality softwaru.



Obr. 4. Vztah mezi řízením a zajištěním kvality softwaru (vlastní zpracování)

### 3.1 Metodiky vývoje softwaru ve vztahu k jeho testování

Vývoj softwaru, tak jako každou posloupnost kroků ve vymezeném prostoru a čase, lze označit pojmem projekt. Aby mohly být projekty, jež jsou si podobné, znovu opakovatelné a mohlo docházet k neustálým zlepšením, jsou k projektům vytvářeny jejich popisy. Na základě popisů mohou následně vznikat modely těchto projektů neboli metodiky. Více či méně formální popis činností, aktivit, vztahů mezi nimi a jejich časových posloupností se nazývá model životního cyklu projektu. (Doucek, 2004, s. 8-32)

Vývoj softwaru je rovněž realizován pomocí softwarových projektů a k obecnému životnímu cyklu řadíme specifikaci požadavků, analýzu, návrh, implementaci, testování, předání a údržbu softwaru.

S růstem moderních technologií, které svou hybností určují tempo vývoje ve všech oblastech lidské činnosti, úměrně roste potřeba invencí a inovací v modelech životního cyklu softwaru. Důsledkem je vznik řady modelů, resp. metodik zachycující vývoj softwaru s ohledem na zajištění kvality a konkurenceschopnosti.

Z pohledu průmyslového inženýrství je vývoj softwaru další dynamicky se rozvíjející oblastí, která se štěpí na dva základní směry. První směr vychází z metodiky rigorózní. Druhý směr je založen na agilní filozofii. Každá oblast má své jedinečné přístupy, které přispívají

ke zvýšení kvality softwaru. Tyto metody jsou představeny v následující podkapitole diplomové práce. Ve společnosti XY jsou aplikovány oba dva představené přístupy. Autor diplomové práce působí v týmu pracující rigorózní přístupem zastoupeným modelem vodopádu, který byl přizpůsoben v metodiku osvědčených postupů (best-practices). Z tohoto důvodu bude v rámci dalšího textu uvedena základní charakteristika modelu vodopádu.

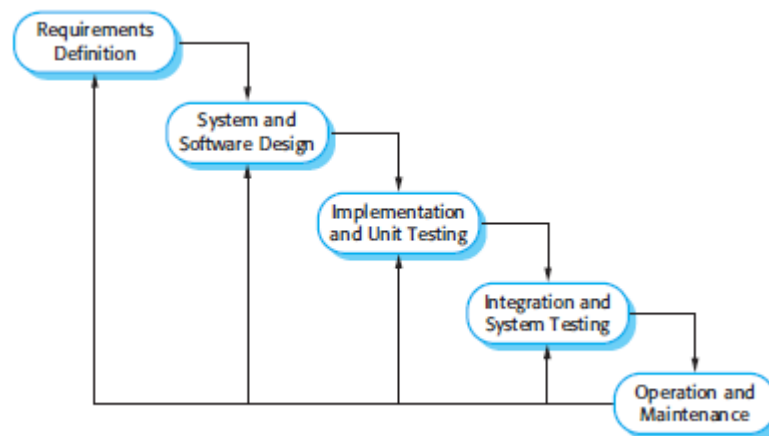
### 3.1.1 Vývoj softwaru prostřednictvím modelu vodopádu

Mezi první publikované modely vztahující se k procesu vývoje softwaru patří vodopádový model. Tento model je jednoduchý, elegantní a dává smysl u mnoha softwarových projektů. Životní cyklus softwaru vyjádřený prostřednictvím vodopádového modelu je charakteristický vzájemnou návazností jednotlivých fází. Výstup jedné fáze tedy představuje vstup pro fázi nadcházející. (Patton, 2002, s. 21-33)

Nejdůležitějším předpokladem vývoje softwaru pomocí tohoto modelu je úvodní fáze, jež je charakteristická definováním a specifikací požadavků. Na níže uvedeném obrázku se jedná o shromáždění nápadů a jejich důkladnou analýzu, která je předpokladem návrhu softwaru. Model rovněž očekává neměnnost specifikovaných požadavků, které byly shromážděny v první fázi. Dalším charakteristickým rysem je časový harmonogram, který by měl přispět k jednoduchému plánování dílčích fází. Je tedy nutné naplánovat všechny činnosti procesu dříve, než se na nich začne pracovat. Výstupem každé fáze modelu je zpravidla jeden nebo více artefaktů. Artefaktem je označován výstup z dílčích procesů, např. dokumenty. Prostřednictvím dokumentů se proces zviditelňuje a je pro účastníky procesu snazší sledovat postup vzhledem k plánovanému vývoji. (Sommerville, 2011, s. 37-61)

K hlavním fázím vodopádového modelu patří:

1. Analýza a definice požadavků
2. Návrh systému a softwaru
3. Implementace a jednotkové testování
4. Integrace a systémové testování
5. Provoz a údržba



Obr. 5. Vodopádový mode vývoje softwaru (Sommerville, 2011, s. 30)

### 3.2 Metody zvyšování kvality softwaru

Cílem této podkapitoly je představit metody zvyšující kvalitu softwaru prostřednictvím zlepšování procesů testování. Jedna z níže uvedených metod bude posléze aplikována v praktické části diplomové práce.

Zlepšování softwarových procesů představuje způsob, jak zvýšit kvalitu výsledného softwaru či snížit náklady nebo urychlení procesů vyskytujících se v celém jeho životním cyklu. Rozeznáváme dva základní přístupy, které se věnují zlepšováním procesů ve vztahu k softwaru:

1. Přístup zralosti procesů.
2. Agilní přístup. (Sommerville, 2011, s. 705-790)

Prvním jmenovaným přístupem je zralost procesů. Tento přístup je zaměřen na zlepšování správy procesů a projektů a zavádí do organizace optimální postupy. Přístup zralosti procesů je interpretován prostřednictvím vyvinutých modelů a metrik. Jedním z nejznámějších modelů využívaných ke stanovení a zvyšování procesní zralosti (maturity) organizace a vyspělosti (capability) je integrační model CMMI Institutu softwarového inženýrství pro zlepšování procesů. (Galín, 2004, s. 475-528).

Bohužel z pohledu praktické části diplomové práce věnující se primárně procesem testování je model CMMI nedostatečný. Model se zaměřuje na testování jen z části a nepokrývá činnosti testování v komplexním měřítku. Z tohoto důvodu bude využit v praktické části

TMMi, který reaguje na nedostatečné pokrytí procesů souvisejících s testováním softwaru v modelu CMMI, avšak z tohoto modelu vychází.

Agilní přístup je spojen s inkrementálním vývojem představující protipól rigorózního vývoje softwaru, který byl představen v předcházející podkapitole diplomové práce. Primárními charakteristickými znaky je iterativní vývoj, redukce režie v softwarovém procesu a rychlé reakční doby na proměnlivé požadavky zákazníků. (Sommerville, 2011, s. 651-681)

### **3.3 Zvyšování kvality procesu testování prostřednictvím modelu TMMi**

Za vznikem modelu TMMi stojí nezisková organizace TMMi Foundation. Cílem organizace bylo vytvoření rozsáhlého modelu zlepšování procesů testování, který bude dostupný široké veřejnosti. Současně stojí za vznikem modelu fakt, že v té době dobře známý a zavedený model CMMI nedostatečně pokrýval činnosti testování softwaru, za což byl také často kritizován. Struktura samotného modelu však vychází z modelu CMMI.

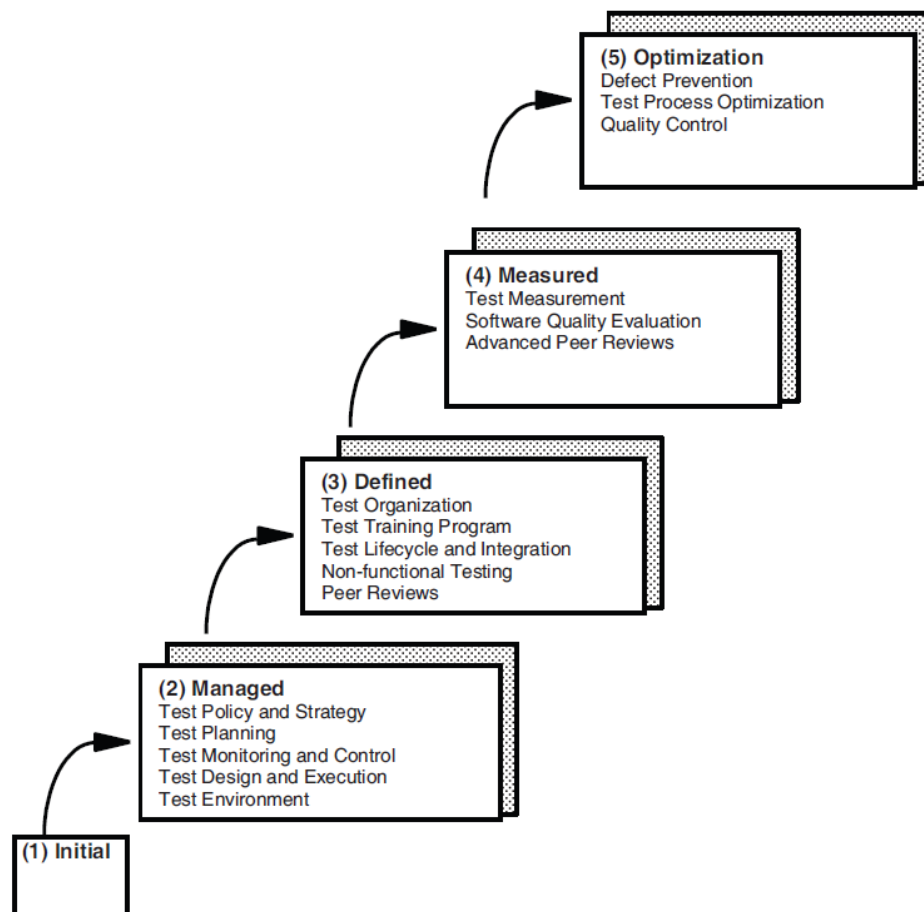
Hlavním dokumentem ukotvující model TMMi obsahující požadavky na procesy testování je Test Maturity Model integration (TMMi) v poslední vydané verzi nesoucí označení Release 1.0. Druhým důležitým dokumentem je TMMi Assessment Method Application Requirements (TAMAR) popisující požadavky na metody hodnocení procesů testování ve své poslední vydané verzi pod označením Release 1.0. Bohužel samotná metoda hodnocení nebyla společností publikována.

Dokument TAMAR podává dvojí pohled na možnosti hodnocení. Hodnocení může probíhat formálním a neformálním způsobem. Cílem formálního hodnocení je získání certifikace TMMi Foundation dokládající dosaženou úroveň zralosti. Získání certifikátu zaručuje také splnění normy ISO/IEC 15504-2. Aby bylo možné o tuto certifikaci požádat, musí mít společnost k dispozici akreditovanou metodu hodnocení nebo využít externí konzultační společnosti, které již akreditovanou metodu hodnocení vlastní. (TMMi Foundation, 2012)

V praktické části diplomové práce bude využita neformální metoda hodnocení navrhnuta Tomášem Doškem v diplomové práci Modely zlepšování procesů testování softwaru a zajištění kvality nevyžadující akreditovaného hodnotitele a akreditovanou metodiku hodnocení. (Došek, 2012, s. 42).

### 3.3.1 Struktura modelu

Model TMMi je založen na stupňovité reprezentaci hodnocení a zlepšování procesů. Model TMMi se tak zakládá na pěti úrovních. Cílem společnosti je při aplikaci modelu postupně dosáhnout vyšších úrovní zralosti. Hodnocení úrovně zralosti následně zodpovídá na otázky, jaká je úroveň procesů testování a jaké nedostatky musí být odstraněny, aby bylo dosaženo vyšší úrovně.



Obr. 6. Stupňovitá reprezentace modelu TMMi (TMMi Foundation, 2012)

První Initial (Počáteční) úroveň zralosti je výchozím stavem nedeklarující podmínky pro splnění, z čehož vyplývá, že tuto úroveň splní společnosti automaticky. Testování je chaotické, skládající se ze špatně definovatelné struktury procesů.

Druhá Managed (Řízená) úroveň již obsahuje procesní oblasti, které musí být ve společnosti zavedeny a jsou předmětem zlepšování. Jedná se o následující oblasti:

- politika a strategie testování,
- plánování testů,
- řízení a monitoring testování,

- návrh a provedení testů,
- testovací prostředí.

Politika testování spolu se strategií představuje základní společná pravidla a cíle pro testování. Součástí plánování testů je harmonogram testování ohraničující počátek a konec testů, analýza možných rizik, rozsah a prioritizace prováděných testů či odhad pracnosti. Na této úrovni jsou taktéž přítomny procesy řízení a monitorování. Nezbytným prvkem je identifikace potřeb s ohledem na testovací prostředí a řízení jejich dostupností. Testování probíhá už při vytváření programového kódu, ale odstraňování chyb nereflektuje první etapy životního cyklu softwaru, kterými jsou analýza, definice požadavků a návrh softwaru.

Třetí Defined (Definovaná) úroveň se skládá z následujících procesních oblastí:

- organizace testování,
- program školení,
- životní cyklus a integrace,
- ne-funkční testování,
- revize.

Cílem této úrovně je definování celopodnikových standardů, které určují základní pravidla pro dílčí projekty. Spolu se standardy jsou definovány základní role a pozice vztahující se k testování. Součástí třetí úrovně je i provádění ne-funkčního testování, za které lze považovat použitelnost, spolehlivost, výkonnost či podporovatelnost. Poslední důležitou oblastí jsou provádění revizí dílčích výstupů z jednotlivých fází životního cyklu softwaru (definice požadavků, návrh systému, programový kód, dokumentace aj.).

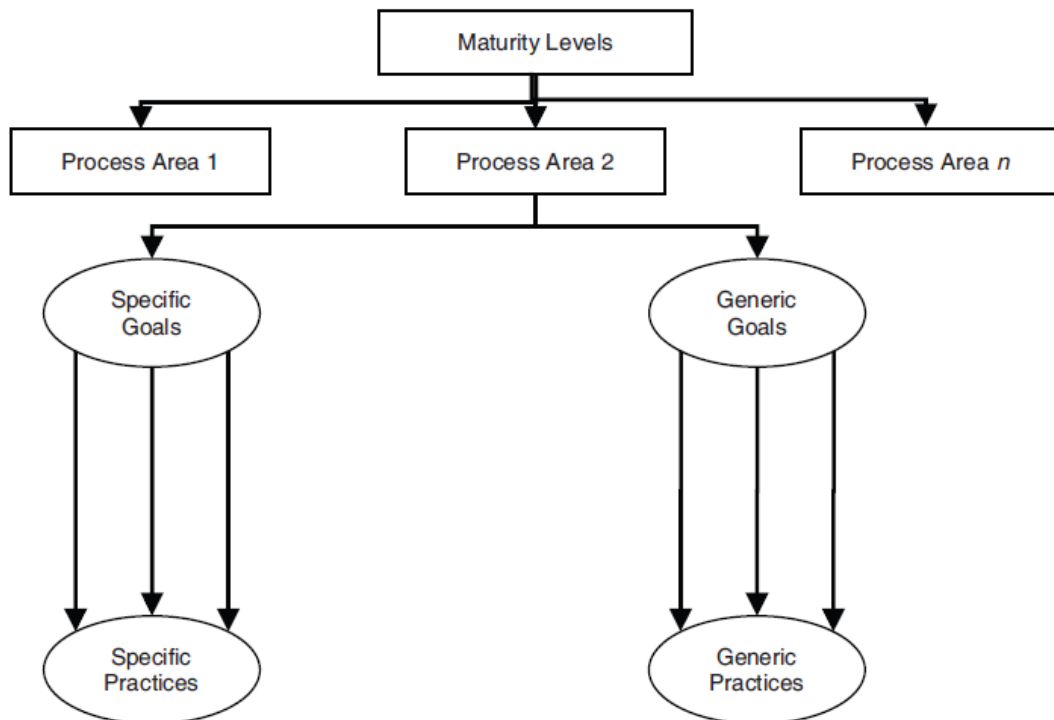
Čtvrtá Measured (Měřitelná) úroveň zavádí měření kvality softwarového produktu, produktivity testování a efektivity testování prostřednictvím tvrdých a měkkých metod. Jsou definovány základní metriky, postupy pro sběr dat a jejich vyhodnocení. K hodnoceným procesním oblastem tedy patří Měření, Hodnocení kvality produkty a Pokročilé revize.

Poslední pátá Optimization (Optimalizovaná) a zároveň nejvyšší úroveň se skládá z procesních oblastí prevence chyb, řízení kvality a optimalizace procesu testování. (TMMi Foundation, 2012)

Jak bylo uvedeno v úvodu této podkapitoly, model TMMi má stupňovitou strukturu. Dosažení požadované maturity level (úrovně zralosti) se odvíjí od dosažení úrovně ve všech



procesních oblastech, které jsou pro danou úroveň definovány. Na dílčí procesní oblasti jsou kladeny goals (cíle) a practices (praktiky).



Obr. 7. Vztah cílů a praktik vyskytujících se v modelu TMMi (TMMi Foundation, 2012)

Model TMMi se skládá z následujících Componets (komponent):

- Maturity Levels,
- Process Areas,
- Specific Goals – Required Components,
- Generic Goals – Required Components,
- Specific Practices – Expected Components,
- Generic Practices – Expected Components,
- Sub-practices – Informative Components,
- Supporting Informative Components – Informative Components,
- Example Work Products – Informative Components.

Komponenty modelu TMMi jsou rozděleny do třech základních skupin. Required Components (povinné komponenty) musí být splněny, aby bylo dosaženo požadované úrovně zralosti. Expected Components (očekávané komponenty) jsou využívány k dosažení požadovaných komponent. Pokud nejsou využity představené praktiky, musí být nahrazeny vhod-

nou alternativou. Informative Components (informativní komponenty) nejsou podmínkou a tedy nejsou ani vyžadovány k splnění deklarované úrovně zralosti.

Každá dílčí procesní oblast má deklarované své vlastní Specific Goals (specifické cíle), které jsou dosahovány pomocí Specific Practices (specifických praktik). Tyto specifické cíle a praktiky jsou pro procesní oblasti jedinečné. Mimo specifických cílů obsahují procesní oblasti i tzv. Generic Goals (generické cíle) a Generic Practices, které představují kvalitu procesní oblasti s ohledem na úroveň zralosti modelu, jež je dosahována prostřednictvím specifických praktik a cílů.

K dalším komponentám patří Sub-practices (sub-praktiky) sloužící k správné interpretaci specifických praktik, Example Work Products (příklady pracovních produktů) a Supporting Informative Components (pomocné informační komponenty), mezi které řadíme Notes (poznámky), Examples (příklady) a References (odkazy). (TMMi Foundation, 2012)

### 3.4 Modelování softwarových procesů

Začátky modelování softwarových procesů sahají k počátkům 80. let. 20. století. Modelování procesů je v současnosti jedním ze způsobů podpory řízení a zajišťování kvality softwaru. Základní náležitosti modelování procesů vycházejí z prvků modelu podnikového procesu, které byly představeny v kapitole 1.3.3 Základní charakteristiky procesu. Modelování procesů lze realizovat prostřednictvím různých metod, technik, nástrojů a standardů. K vybraným metodám a technikám modelování procesů patří:

- metodika ARIS profesora Scheera,
- metoda Business System Planning,
- metoda Lean Six Sigma,
- metodika DEMO profesora Dietze.

Mezi základní standardy určené pro modelování procesů náleží:

- Business Process Management Language,
- Business Process Model and Notation,
- standard IDEF,
- jazyk UML,
- standardy ISO. (Řepa, 2006, s. 37-55)

S ohledem na praktickou část diplomové práce, ve které autor práce využívá k modelování softwarových procesů Business Process Model and Notation, bude tento standard blíže představen.

### 3.4.1 Business Process Model and Notation

BPMN je moderní grafická notace pro modelování firemních procesů. Cílem notace je srozumitelnost popisu procesů pro člověka. Ve své aktuálně platné verzi 2.0 spadá pod softwarové konsorcium Object Management Group. BPMN umožňuje modelování spolupracujících B2B procesů či aktivit, ale také interních firemních procesů. Prostřednictvím grafické notace lze zachytit nejen vnitřní strukturu činností, ale také jejich workflow.

Firemním procesem rozumíme s ohledem na BPMN množinu činností vztahených k danému cíli organizace. Za pojmem workflow se skrývá označení návaznosti činností v rámci firemního procesu. Samotná činnost je jednotkou popisující průběh firemního procesu.

Notace BPMN se skládá z pěti základních skupin elementů:

1. Flow Object – základní objekty,
2. Data – množina dat,
3. Connecting Object – propojovací objekty,
4. Swimlanes – kontexty objektů,
5. Artifacts – artefakty. (Object Management Group, 2011)

Základní objekty jsou hlavními grafickými prvky notace BPMN a definují chování podnikových procesů:

- Event – události,
- Activities – činnosti,
- Gateways – brány.

Množina dat nám poskytuje informace o tom, jaké data činnost vyžaduje, aby mohla být provedena, nebo jaké informace jsou výstupem:

- Data Objects – datové objekty,
- Data Inputs – datové vstupy,
- Data Outputs – datové výstupy,
- Data Stores – datové sklady.

Propojovací objekty, jak už název napovídá, slouží k propojení základních objektů:

- Sequence Flow – sekvenční tok,
- Message Flow – tok zpráv,
- Associations – asociace,
- Data Associations – informační asociace.

Kontexty objektů se skládají z dvou typů elementů. Jedná se o Pools (bazény) a Lanes (dráhy).

Artefakty se využívají k dodatečné informaci vztahující se k procesu. Mezi nejčastěji využívané artefakty patří Text Annotation (poznámka) a Group (grafické zobrazení skupiny). Přehled grafické notace a workflow patterns (ustáleného modelování) BPMN je součástí přílohy P I – P VI diplomové práce.

## **II. PRAKTICKÁ ČÁST**

## 4 PŘEDSTAVENÍ SPOLEČNOSTI

Akciová společnost vznikla v České republice na konci 90. let 20. století a je členem finanční skupiny, která patří k předním evropským poskytovatelům spotřebitelského financování. Finanční skupina zaměstnává okolo 45 000 zaměstnanců a obsluhuje v celosvětovém měřítku 35 milionů zákazníků.

Hlavním předmětem podnikání akciové společnosti je poskytování spotřebitelského financování a úvěrů koncovým zákazníkům, jedná se například o nákup na splátky, hotovostní úvěry, revolvingové úvěry, kreditní karty a půjčky na auta.

### 4.1 Mise finanční skupiny

Poslání společnosti je následující:

„Chceme být předním poskytovatelem spotřebitelského financování na každém trhu, kde působíme – měřeno objemem obchodu, dlouhodobým růstem, pověstí společnosti i obchodními metodami.“ (Interní materiály, 2015)

### 4.2 Vize finanční skupiny

Budoucí stav společnosti je charakterizován následujícím tvrzením:

„Jsme plně zavázáni, abychom splnili potřeby a očekávání svých zákazníků a obchodních partnerů. Usilujeme o to být vždy dostupní, přátelští a efektivní. Snažíme se poskytovat svým zákazníkům bezprostřední přístup ke komplexní nabídce spotřebitelských finančních produktů, kterým je snadné porozumět a přináší skutečnou hodnotu.“ (Interní materiály, 2015)

### 4.3 Hodnoty finanční skupiny

Významné hodnoty společnosti jsou shrnuty níže:

„Naším cílem je zavést, udržovat a rozvíjet vysoký standard obchodních praktik ve styku s našimi klienty, obchodními partnery, zaměstnanci, investory a ostatními partnery. Znamená to zodpovědné a poctivé jednání, které je v souladu s platnými právními předpisy. Svým jednáním se na jedné straně hlásíme k mezinárodním standardům odpovědného obchodování, na straně druhé chceme respektovat tradice a kulturu obyvatel v zemích, kde působíme.“ (Interní materiály, 2015)

#### 4.4 Organizační struktura společnosti

Mezi orgány akciové společnosti patří valná hromada, představenstvo a dozorčí rada. Valná hromada je nejvyšším orgánem společnosti. Skládá se ze všech na ní přítomných akcionářů. Představenstvo je statutárním orgánem společnosti, který řídí její činnost. Představenstvo má 3 členy volené a odvolávané dozorčí radou. Dozorčí rada je kontrolním orgánem společnosti. Dohlíží na činnost představenstva a uskutečňování podnikatelské činnosti společnosti.

Struktura společnosti je členěna na tzv. organizační útvary, mezi které řadíme divize, sekce a oddělení. Divize je základní organizační útvar zajišťující zejména přípravu podkladů pro rozhodování představenstva a realizaci všech činností společnosti. Sekce je organizační útvar, jímž jednotlivé divize realizují svoji odbornou činnost v užších oblastech. Oddělení je organizační útvar, jímž jednotlivé divize a sekce realizují svoji odbornou činnost v úžeji vymezených oblastech. Společnost je členěna na následující divize:

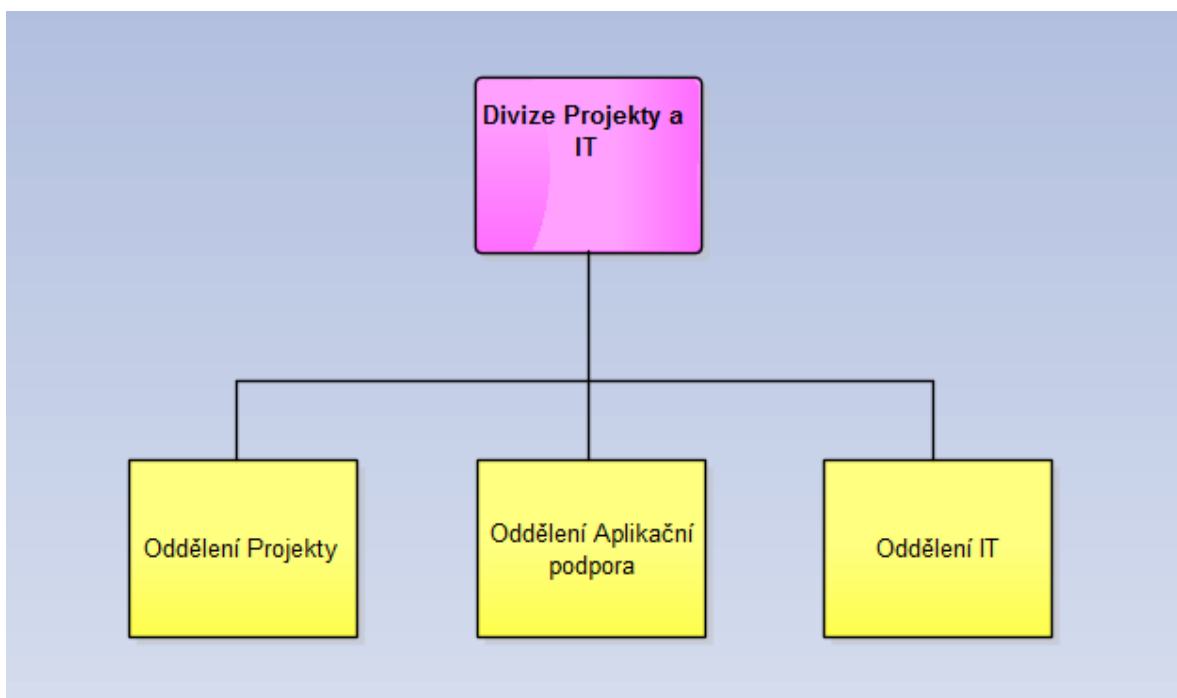
- CEO,
- Lidské zdroje,
- Právní a Compliance,
- Projekty a IT,
- Collections,
- Péče o klienta,
- Obchod,
- Auto Business a pojištění,
- Produkty a CRM,
- Marketing,
- Finance,
- Risk,
- Provoz a správa. (Interní materiály, 2015)

Z důvodu působení autora práce na Oddělení Aplikační podpory spadající pod divizi Projekty a IT, bude uvedena stručná charakteristika. Náplň činnosti divize je zejména:

- vývoj a realizace projektů,
- vývoj, správa a předávání projektové metodologie v rámci společnosti,
- příprava statutárního, provozního a manažerského reportingu,
- zpracování analýz a reportů týkajících se klientských databází,

- tvorba statických modelů (logistická regrese a shluková analýza),
- údržba datových zdrojů.

Divize je dále vnitřně členěna na Oddělení Projekty, Oddělení Aplikační podpora a Oddělení IT. Náplní činnosti Oddělení Aplikační podpory je koordinace vývoje aplikací společnosti, včetně definování požadavků na jejich vývoj směrem k dodavateli, definování změnových požadavků a jejich implementace a následná aplikační podpora. Dále se jedná o testování aplikací, odpovědnost za obsah verze softwaru, která byla uvolněna k řádnému nasazení a běžnému používání, tvorba metodik a dokumentace softwaru, správa, řešení a testování incidentů v aplikacích společnosti. (Interní materiály, 2015)



Obr. 8. Organizační struktura divize projektů a IT ve společnosti XY (vlastní zpracování)

## 4.5 Produkty

Akciová společnost disponuje širokou škálou finančních produktů, které nabízí svým koncovým zákazníkům. Stěžejní produkty jsou uvedeny níže a seřazeny dle podílového zastoupení na celkovém portfoliu:

- Karty.
- Hotovost.
- Nákup na splátky.
- Autoúvěry. (Interní materiály, 2015)



Za každým uvedeným produktem stojí rozsáhlý systém složený z procesní, organizační a informační struktury. Dílčím prvkem informační struktury je software a jeho řízení kvality, které je tématem této diplomové práce. Z důvodu zaměření této práce na software poskytující informační strukturu produktu Autoúvěr, bude tento produkt stručně uveden.

Společnost nabízí financování ojetých aut pro fyzické osoby, podnikatele a právnické osoby. Klienti mohou touto formou financovat až čtrnáct let stará vozidla. Výhodou je možnost přímé platby už od 0 %, variabilní délka splácení od šesti měsíců do osmi let a pevná výška měsíčních splátek podle dohodnutého splátkového kalendáře. Uzavření smlouvy spotřebitelského úvěru probíhá rychle a jednoduše přímo v místě, kde si klient automobil vybral. (Interní materiály, 2015)

#### **4.6 Představení softwaru poskytující softwarovou strukturu pro produkt Autoúvěr**

Název softwaru EkIS vznikl spojením počátečních písmen ze sousloví Ekonomický Informační Systém. Jedná se o software umožňující systém půjček na nákup ojetých i nových aut používaných v České republice a na Slovensku. Software je dodáván externí společností. Uvedený software umožňuje komunikaci se softwarem třetích stran, proto společnost XY spolupracuje i s dalšími softwarovými organizacemi.

Na samotný software lze nahlížet ze dvou základních úhlů:

- technologické složení,
- uživatelské členění.

Dle technologického složení dělíme software na části aplikace využívající operační systém Windows a na části aplikace využívající systém Linux.

Dle uživatelského členění softwaru rozlišujeme kategorie FRONT office a BACK office. FRONT office softwaru tvořící systém pro pořizování a schvalování žádostí o půjčku na nákup ojetých i nových aut. Pořizování žádostí probíhá ve webovém prostředí a tuto činnost provádí automobilový dealer. Schvalování žádostí provádějí uživatelé softwaru ve společnosti XY. Dílčí komponenty FRONT office musí být neustále provozuschopné a funkční, protože mají přímý vliv na chod celého obchodu.

BACK office softwaru tvoří systém pro kompletaci smluv a dalších procesů týkající se pohledávek do jejich doplacení. Jedná se zejména o evidenci a správu účetnictví, klientské služby, CRM, vymáhání pohledávek aj. (vlastní zpracování).

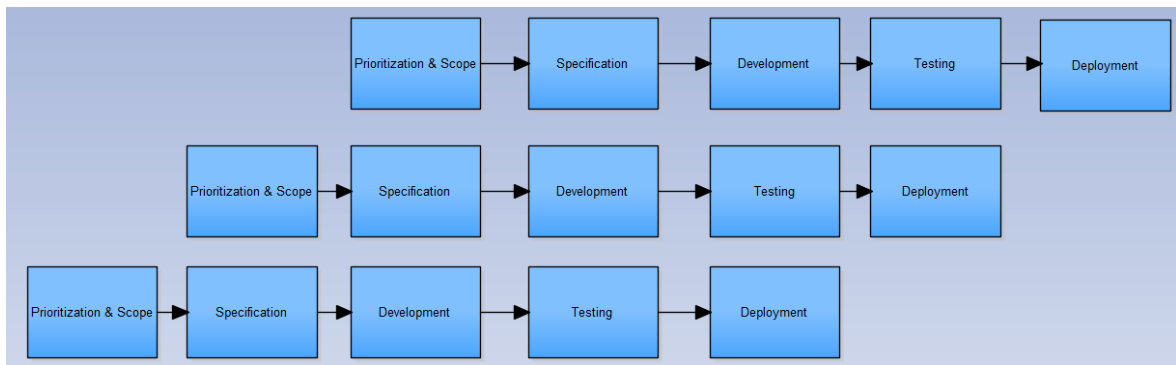
## 5 ANALÝZA SOUČASNÉHO STAVU ŘÍZENÍ KVALITY SOFTWARE PROSTŘEDNICTVÍM TESTOVÁNÍ SOFTWARE VE SPOLEČNOSTI

Proces testování softwaru ve společnosti XY je silně ovlivněn modelem jeho vývoje. Ten je postaven na principech vodopádu, který byl představen v teoretické části diplomové práce. Nejvýraznější charakteristikou vodopádového vývoje softwaru je fakt, že vstupem do následující činnosti je výstup z činnosti předchozí.

Společnost využívá k vývoji softwaru služeb externího dodavatele, který je tak úzce spjat s procesy probíhajícími ve společnosti. Životní cyklus vývoje softwaru ve společnosti XY se skládá z následujících částí:

1. Prioritization & Scope: definování požadavků a stanovení rámcového obsahu pro životní cyklus ze strany společnosti XY s následnou analýzou ze strany dodavatele softwaru.
2. Specification: specifikace požadavků na straně společnosti XY a návrh softwaru dle požadavků ze strany dodavatele softwaru.
3. Development: vývoj softwaru probíhá prostřednictvím externího dodavatele, který implementuje předchozí data, informace a znalosti vedoucí ke vzniku požadované podoby softwaru.
4. Testing: testování ze strany společnosti XY a dodavatele softwaru.
5. Deployment: nasazení softwaru na produkční prostředí neboli nasazení do ostrého provozu.

Životní cyklus vývoje softwaru je označován jako Release. Tento cyklus se opakuje z pravidla třikrát do roka. Četnost cyklů závisí na stanoveném rozpočtu společnosti XY, který je věnován na rozvoj kvality představeného softwaru v kapitole 4.6. Obsah rozvoje softwaru je vždy jedinečný, a proto lze považovat životní vývojový cyklus softwaru za softwarový projekt. Ten má vždy definované zdroje, rizika a časové milníky. Obsahová stránka projektu je dána specifickými požadavky koncových uživatelů, ale také legislativními změnami. Dílčí softwarové projekty jsou realizovány paralelně, čímž je docíleno překrytí životních fází životního cyklu vývoje softwaru, kdy je například software ze strany dodavatele vyvíjen a současně testován ve společnosti XY.



Obr. 9. Průnik realizovaných softwarových projektů během roku (vlastní zpracování)

Kromě dodavatele softwaru, který figuruje v různých intenzitách v každé části životního cyklu vyvíjeného softwaru, se můžeme setkat s dalšími subjekty vstupující do celého procesu. Jedná se o interní zaměstnance společnosti XY, tedy koncové uživatele softwaru. Dále to jsou zaměstnanci spolupracující společnosti AB zajišťující správu databázových prostředí, nad kterými běží vyvíjený software. Posledním subjektem jsou zaměstnanci působící v divizi Projekty a IT, oddělení aplikační podpory ve společnosti XY zajišťující řízení kvality softwaru v celém jeho životním cyklu. Jedná se o oddělení, ve kterém zároveň působí autor diplomové práce. Pro lepší orientaci jsou subjekty vyjmenovány v níže uvedeném seznamu:

- dodavatel softwaru, dále jen dodavatel,
- zaměstnanci oddělení aplikační podpory, dále jen zaměstnanci AP,
- oddělení správy databázových prostředí ve společnosti AB, dále jen zaměstnanec DP,
- interní zákazník softwaru ve společnosti XY, dále jen uživatel.

Současný stav testování není ve společnosti XY nikterak zdokumentován či zmapován. Z tohoto důvodu bude autorem diplomové práce využita procesní analýza současného stavu testování softwaru zahrnující návržení procesních skupin agregující současné činnosti vykonávané v procesu testování. Následně bude pro každou procesní skupinu vytvořen popis procesu a Karta procesu. Druhým krokem procesní analýzy bude vytvoření mapy procesů pomocí modelování současného stavu testování softwaru s využitím grafické notace BPMN 2.0 viz Příloha P I – P VI.

Karta procesu	
Číslo a název procesu:	
Cíl procesu:	
Vlastník procesu:	
Zákazník procesu:	
Vstupy procesu:	
Činnosti procesu	
Číslo činnosti	
1.	
2.	
3.	
4.	
Výstupy procesu:	
Metriky procesu:	

Obr. 10. Vzor Karty procesu (vlastní zpracování)

## 5.1 Identifikace a popis současného stavu procesu testování

Testování softwaru ve společnosti XY vychází z přístupu best-practices a jak bylo uvedeno, jsou tyto procesy poznamenány vývojem softwaru odrážející vodopádový přístup. Současný stav testování softwaru lze rozdělit do následujících čtyř procesních oblastí:

1. Test Planning (Plánování testů)
2. Test Development (Návrh testů)
3. Test Implementation (Implementace testů)
4. Test Execution (Provádění testů)

### 5.1.1 Test Planning

Proces testování ve společnosti XY začíná plánováním testů. Cílem této procesní oblasti je deklarace cílů testování s identifikací hlavních činností, úloh a zdrojů, které povedou

k vytyčenému cíli. Proces plánování testů probíhá již při plánování životního cyklu softwaru. Ve společnosti XY zodpovídá za plánování testování a celého softwarového projektu role spojující projektového a test manažera do jedné osoby. Výstupem tohoto procesu je plán testování.

Prvním zjištěným nedostatkem je fakt, že je plánování testů z důvodů nedostatečných kapacit jednorázovou aktivitou, která je vykonána na začátku softwarového projektu. Často se tak stává, že v průběhu procesu testování jsou zjištěny detailnější informace ovlivňující celý projekt, na které již plán testování nereaguje. Dopadem jsou posléze neaktuální informace v testovacím plánu, nízká flexibilita, dodatečné náklady a posouvání časových milníků.

Proces plánování testů je ve společnosti členěn na čtyři hlavní podprocesy:

- definice cílů a rozsahu testování,
- analýza rizik testování,
- definice přístupu k testování,
- ostatní úlohy.

Primárním cílem testování softwaru ve společnosti XY je prevence vzniku defektů. Tento cíl je fixní a je podpořen detekcí s následnou opravou defektu se sběrem informací o jeho vzniku. Nástrojem v průběhu procesu testování je řízení defektů vůči dodavateli softwaru. Deklarace komplexního cíle vychází ze strategie a politiky testování na celopodnikové úrovni. Další konkrétní cíle a záměry testování definuje pro softwarový projekt Projektový (Test) manažer, který ve spolupráci se zainteresovanými stranami definuje priority a hodnoty jednotlivých cílů. Dochází tak ke sladění cílů softwarového projektu s cíli testování softwaru. K zainteresovaným stranám patří vývojáři, programátoři, analytici dodávající software, vrcholový management, zaměstnanci AP a DP a koncoví uživatelé softwaru.

Rozsah testování vychází z projektového plánu, kde je stanoveno časové, rozpočtové a zdrojové omezení projektu. Na základě těchto vstupních informací sestavuje Projektový (Test) manažer seznam testovaných a netestovaných částí a vlastností softwaru spolu s odůvodněním proč není ona část systému či vlastnost předmětem testů.

Viditelným problémem této úlohy je nestanovení priorit k jednotlivým položkám na seznamu popisující rozsah testování. Priority by tak měly vycházet z potřeb zainteresovaných stran a z identifikovaných produktových a projektových rizik, které jsou výstupem podprocesu analýzy rizik testování.

V podprocesu analýzy rizik sestavuje Projektový (Test) manažer seznam produktových a projektových rizik. K nejčastěji stanoveným produktovým rizikům patří nedostatečná úroveň kvality formulována prostřednictvím modelu FURPS představený v teoretické části diplomové práce nebo validace, kdy je zjištěno, že software nefunguje tak, jak bylo zamýšleno uživatelem. Mnohem důležitější jsou pro softwarový projekt definované projektové rizika ovlivňující splnění cílů celého projektu. Jedná se na prvním místě o rizika dodavatelská, dále technická a organizační. Například nedodání softwaru inovovaného z důvodu legislativních změn představuje přímý dopad na chod společnosti zabývající se automobilovým businessem. Mezi technická rizika spadají problémy při specifikaci požadavků kladených na vyvíjený software, jejich rozsah, který nemůže být z důvodů omezení projektu splněn, nízká kvalita návrhu softwaru ze strany dodavatele, nízká kvalita testovacích dat, scénářů aj. K nejčastěji definovaným organizačním rizikům přísluší nedostatečné kapacity vymezené pro testování ze strany uživatelů softwaru ve fázi jeho akceptace.

Podproces definování přístupu k testování je silně ovlivněn modelem vývoje softwaru. Ten je v současné době postaven pouze na přístupu založeného na požadavcích. Podmínkou úspěchu je kvalitní specifikace požadavků v druhé fázi životního cyklu softwaru Specification. Po konzultaci s Projektovým (Test) manažerem bylo v této fázi zjištěno, že ačkoliv jsou rizika deklarována, nejsou nikterak do přístupu k testování promítnuta.

Součástí podprocesu definování přístupu k testování je specifikace úrovní testování. Zde jsou definovány dvě úrovně, které probíhají ve společnosti XY. Jedná se o systémové a akceptační testování. Testování jednotek při programování a integrační test při návrhu softwaru je prováděn na straně dodavatele softwaru. Projektový (Test) manažer mimo jiné stanovuje vstupní a výstupní kritéria pro každou testovanou úroveň. Vstupní kritéria určují, kdy může být zahájeno testování dané úrovně. Výstupní kritéria stanovují, kdy je testování pro danou úroveň ukončeno.

Po stanovení úrovní testování přichází na řadu deklarace typu testů, jež budou prováděny na daných úrovních. Zde vychází společnost opět z metody FURPS čímž je zajištěno testování pokrývající funkční i nefunkční části softwaru. Deklarována je taktéž jediná technika návrhů testů založená na přístupu černé skříňky. Technika bílé a šedé skříňky je využívána na straně dodavatele softwaru a proto není součástí procesů testování softwaru ve společnosti XY.

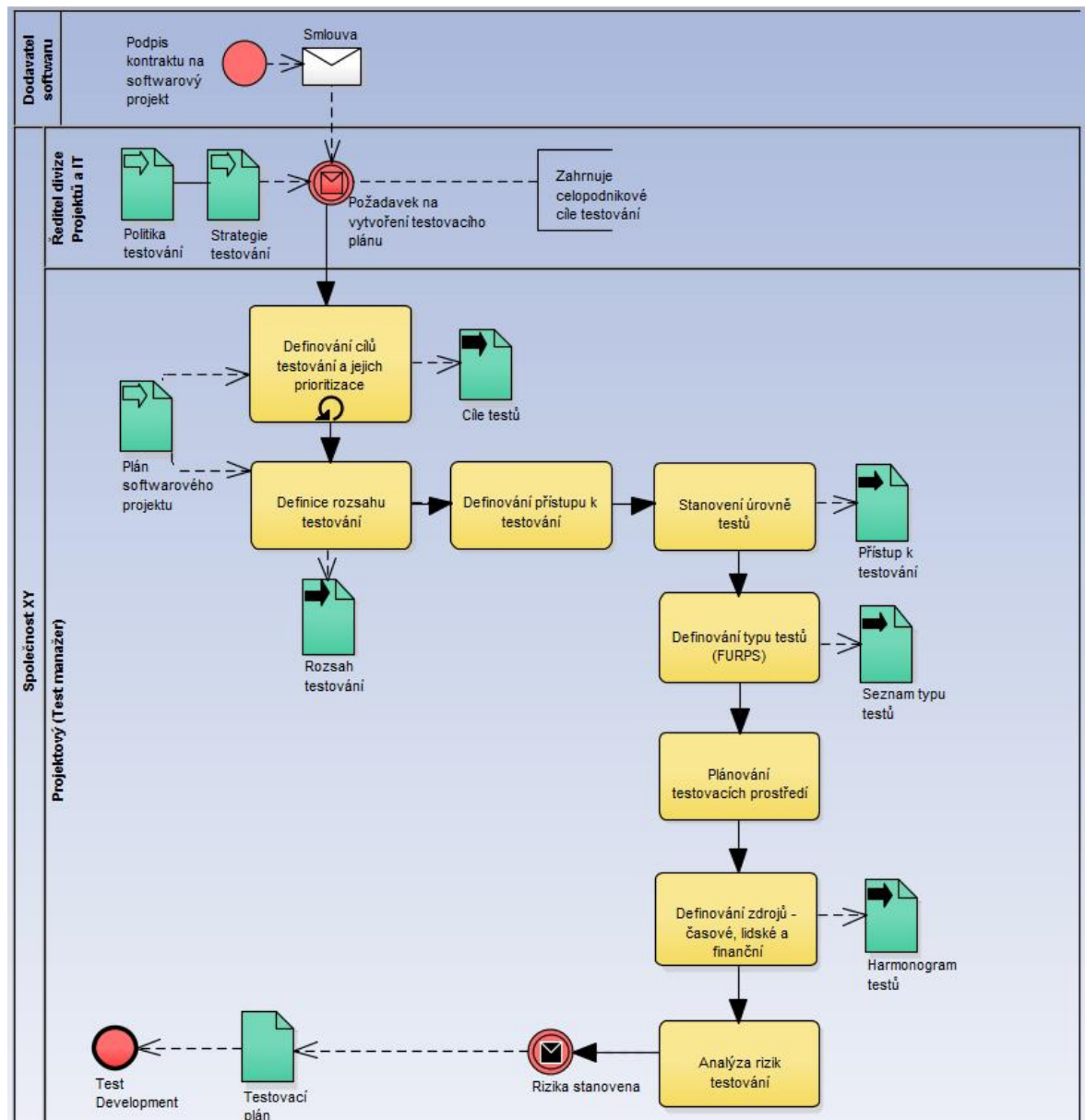
Do podprocesu ostatní úlohy jsou začleněny úlohy plánování potřebných testovacích prostředků, které má společnost k dispozici, sestavení harmonogramů odrážející časový odhad testování, stanovení rozpočtů testování a lidské zdroje.

V úvodu analýzy procesu plánování testování bylo zjištěno, že je proces plánování jednorázový a neodráží směr, kterým se testování ubírá. Mimo nedostatečné kapacity ze strany Projektového (Test) manažera tento fakt způsobuje nestanovení metrik testování. Pokud nejsou stanoveny základní měřicí veličiny, nelze předpokládat dostatečné řízení procesu testování, natož řízení kvality softwaru.

Karta procesu	
Číslo a název procesu:	P 1 - Test Planning (Plánování testů)
Cíl procesu:	Vytvoření plánu testování
Vlastník procesu:	Projektový (Test) manažer
Zákazník procesu:	Podproces Test Development (Návrh testů)
Vstupy procesu:	Softwarový projekt a jeho rozpočet, strategie a politika testování
Činnosti procesu	
Číslo činnosti	
1.	Definice cílů a rozsahu testování
2.	Analýza rizik testování
3.	Definice přístupu k testování
4.	Ostatní činnosti
Výstupy procesu:	Plán testování
Metriky procesu:	Žádné

Obr. 11. Karta procesu Plánování testů (vlastní zpracování)





Obr. 12. Proces Plánování testů (vlastní zpracování)

### 5.1.2 Test Development

Cílem procesu návrhu testů je vydefinování, jak bude software testován. Vstupem do návrhu testů je podproces analýzy vývojových studií, označovaných jako Release Study, který předkládá dodavatel softwaru. Obsahem vývojových studií jsou dílčí specifikace uživatelů, které byly definovány v druhé fázi životního cyklu vývoje softwaru pojmenovaném Specification. Pomocí vývojových studií jsou identifikovány podmínky k testování. Ty jsou posléze namapovány prostřednictvím procesu návrh testů na testovací případy (Test Case) a testovací scénáře (Test Scenario). Zodpovědnou rolí za proces návrhu testů je ve společ-

nosti XY Tester (Test Analytik). Uvedenou roli zastávají tři zaměstnanci AP. Při návrhu testů se zaměstnanci řídí plánem testování, který určuje přístup a samotný rozsah testů.

Hlavními činnostmi při návrhu testů je návrh testovacích případů a scénářů. Výstupem těchto úloh jsou artefakty neboli dokumentace nesoucí název Testovací případy a Testovací scénáře.

Název kapitoly v zadání	Předpoklady	Popis činnosti	Očekávaný výsledek	Testy	Odpovídá
<b>Kompletní název požadavku</b>					
číslo kapitoly v RS a název kapitoly				<u>Netestováno</u>	Michal Františ

Obr. 13. Vzor Testovacího případu (vlastní zpracování)

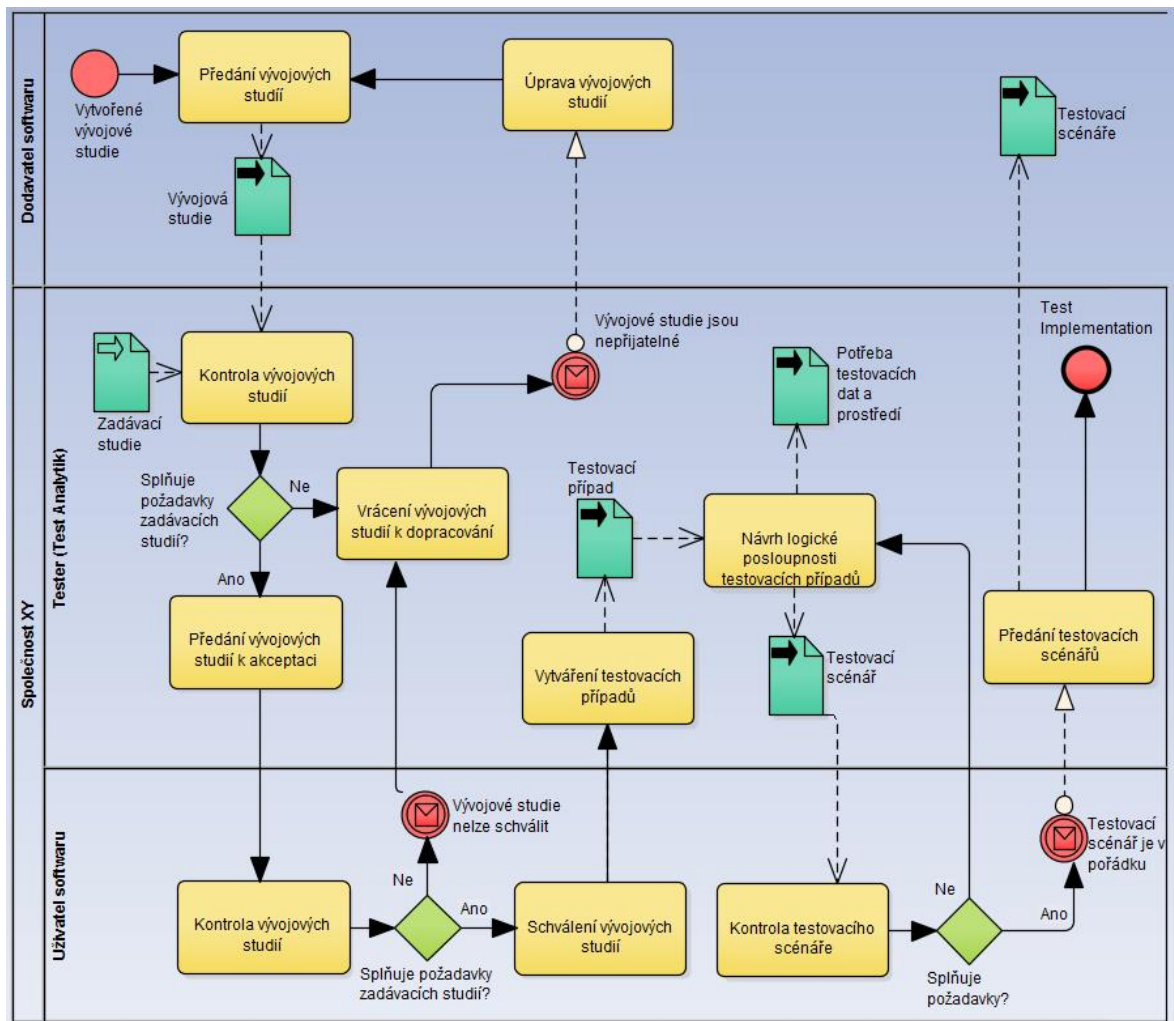
Aby bylo možné vytvořit testovací scénář, musí být navrhnutá logická posloupnost testovacích případů. Sada několika logicky navazujících testovacích případů tvoří posléze testovací scénář.

Během vytváření testovacích případů a scénářů jsou zjišťovány konkrétní podmínky na potřebu testovacích dat a testovacích prostředí. Zjištěné informace posléze vstupují do podprocesu Test Implementation (implementace testů). Vytvořené testovací scénáře jsou průběžně předkládány koncovým uživatelům softwaru ke schválení. V momentě dokončení všech testovacích scénářů pokrývajících rozsah vývojových studií, jsou zaslány dodavateli softwaru, který po skončení vývoje provede pomocí této dokumentace první systémové testování. Dochází tak k průběžné zpětné vazbě mezi koncovým uživatelem softwaru a jeho dodavatelem.

Po provedené analýze a konzultaci se zaměstnanci AP nebyly zjištěny mimo chybějících metrik daného procesu žádné jiné závažné nedostatky.

Karta procesu	
Číslo a název procesu:	P 2 - Test Development (Návrh testů)
Cíl procesu:	Vytvoření Testovacích případů a scénářů
Vlastník procesu:	Tester (Test Analytik)
Zákazník procesu:	Podproces Test Implementation (Implementace testů)
Vstupy procesu:	Plán testování, Vývojové studie
Činnosti procesu	
Číslo činnosti	
1.	Analýza vývojových studií
2.	Návrh testů
3.	Identifikace potřeb testovacích dat
4.	Identifikace potřeb testovacích prostředí
Výstupy procesu:	Testovací případ, Testovací scénář
Metriky procesu:	Žádné

Obr. 14. Karta procesu Návrh testů (vlastní zpracování)



Obr. 15. Proces Návrh testů (vlastní zpracování)

### 5.1.3 Test Implementation

Cílem implementace testů je připravení testovacích dat a testovacích prostředí, na kterých bude testování softwaru prováděno. Za samotný podproces přípravy databázových prostředí zodpovídá zaměstnanec DP v roli Databázového správce. Společnost využívá k testování šest testovacích prostředí. Tři testovací prostředí slouží k testování softwaru pro Českou republiku, další tři slouží k testování softwaru pro Slovensko. Řízení konfigurace a využitelnosti jednotlivých testovacích prostředí spadá pod roli Tester (Test Analytik) z řad zaměstnanců AP. Řízení databázových prostředí spočívá v předání identifikovaných potřeb na testovací prostředí v podprocesu Test Development. Průběh přípravy databázových prostředí sledují zaměstnanci AP pomocí specializovaného softwaru. O dokončení přípravy databázových prostředí jsou vždy pomocí tohoto softwaru informováni. Správu databázových prostředí provádí spolupracující společnost. Z toho důvodu je proces mimo společ-

nost XY a není dále identifikován. Jako viditelný problém byla analyzována chybějící kontrola po předání databázových prostředí zpět zaměstnancům AP, kdy není ověřena korektnost přípravy testovacích prostředí. Tato kontrola by mohla být provedena pomocí smoke testů ověřující základní funkcionalitu testovacích prostředí.

Činnost přípravy testovacích dat provádí zaměstnanec AP v roli Tester (Test Analytik). Ačkoliv se jedná mnohdy o obrovské softwarové projekty, není v současné době vytvořena automatizace přípravy testovacích dat ze strany společnosti XY. Testovací data jsou pro dílčí testovací případy a testovací scénáře vždy připravovány výhradně manuálně. Tato činnost spočívá v úpravě reálných dat, kde dochází k nahrazení citlivých či osobních údajů údaji fiktivními. Druhou možností je vytvoření úplně nových dat. Tato varianta je časově náročnější, přičemž není zachována původní struktura reálných dat.

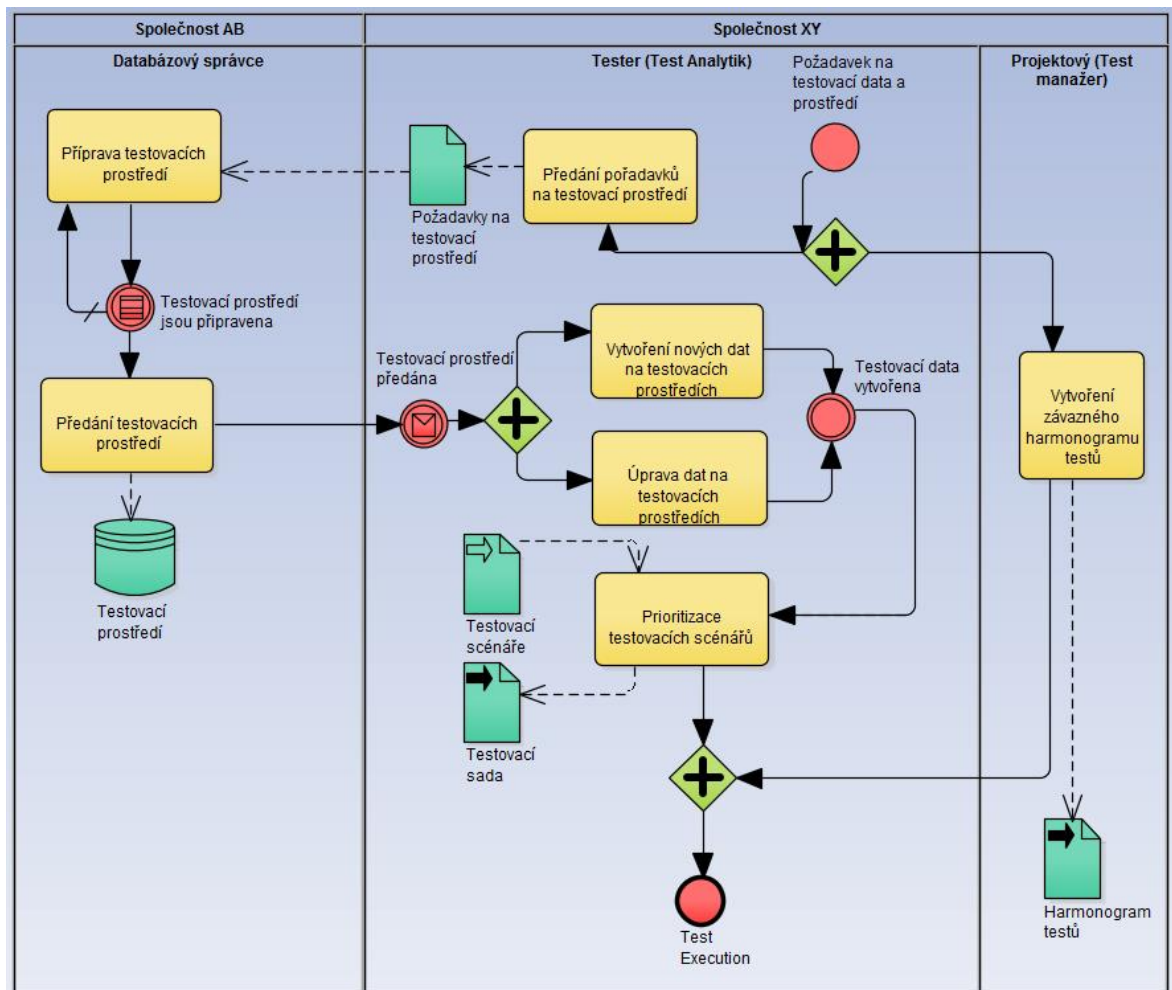
Implementace automatických testů by snížila vytíženost kapacit zaměstnanců AP, ale po konzultaci s Projektovým (Test) manažerem není z důvodu vysokých finančních nákladů tento krok možný. Činnosti přípravy testovacích dat a přípravy testovacích prostředí probíhají paralelně.

Poslední důležitou činností v rámci implementace testů je vytvoření závazného harmonogramu prováděných testů. Tuto činnost provádí Projektový (Test) manažer.

Výstupními produkty procesu implementace testů jsou Testovací data, Harmonogram testů a Testovací sada. Testovací sada je vytvářena na základě prioritizace a organizace testovacích případů a testovacích scénářů. Řízení testovacích sad spadá pod zaměstnance AP, který testovací případy a scénáře vytváří. Priority jsou přiřazovány na základě přístupu k testování. Tím je ve společnosti XY přístup založený na požadavcích a jejich pokrytí. V závěru analýzy implementace testů je opět zjištěno, že na proces nejsou navázány metriky, které by umožnili koncepční řízení procesu.

Karta procesu	
Číslo a název procesu:	P 3 - Test Implementation (Implementace testů)
Cíl procesu:	Příprava testovacích dat, prostředí, vytvoření testovacích sad, sestavení harmonogramu testování
Vlastník procesu:	Projektový (Test manažer), Tester (Test Analytik), Databázový správce
Zákazník procesu:	Podproces Test Execution (Provádění testů)
Vstupy procesu:	Požadavky na testovací data a prostředí, Testovací případy a scénáře
Činnosti procesu	
Číslo činnosti	
1.	Příprava testovacích prostředí
2.	Příprava testovacích dat
3.	Vytvoření harmonogramu prováděných testů
4.	Organizace a prioritizace testovacích případů a scénářů
Výstupy procesu:	Testovací data, Harmonogram testů, Testovací sada
Metriky procesu:	Žádné

Obr. 16. Karta procesu Implementace testů (vlastní zpracování)



Obr. 17. Proces Implementace testů (vlastní zpracování)

#### 5.1.4 Test Execution

Provádění testů je stěžejním podprocesem celého procesu testování. Cílem uvedeného podprocesu je vykonání manuálních testů dle vytvořených testovacích scénářů. Zodpovědnou rolí za provádění testů je ve společnosti Tester (Test Analytik). Provádění testů začíná ve chvíli, kdy jsou k dispozici připravené testovací prostředí s nasazenou verzí softwaru, která je obsahem softwarového projektu. Následuje ověření vstupních kritérií, které slouží k započítí testů. Činnost ověřující vstupní kritéria kladená na softwarový produkt je ve společnosti XY nazvána jako akceptační testování. Akceptační testování je založené na technice smoke testů ověřující obecnou funkcionální softwaru. Výstupním artefaktem je dokument typu checklist s názvem Test obecné funkcionality.

Testy obecné funkcionality	Destinace	Správná funkčnost	Poznámka (popis chyby)	Zodpovídá
<b>Sekce:</b>				
	PROD CZ	<u>Ne</u>		
	PROD CZ	<u>Ano</u>		
	PROD CZ	<u>N/A</u>		
	PROD CZ	<u>N/A</u>		
	PROD CZ	<u>N/A</u>		
	PROD CZ	<u>N/A</u>		
	PROD CZ	<u>N/A</u>		
	PROD SK	<u>N/A</u>		
<b>Sekce:</b>				
	PROD CZ	<u>N/A</u>		
	PROD CZ	<u>N/A</u>		
	PROD CZ	<u>N/A</u>		
	PROD CZ	<u>N/A</u>		
	PROD CZ	<u>N/A</u>		
	PROD CZ	<u>N/A</u>		

Obr. 18. Vzor artefaktu – Test obecné funkcionality (vlastní zpracování)

Provádění testů je vykonáváno dle plánu testování a harmonogramu testů. K hlavním činnostem realizovaných v procesu provádění testů náleží provádění testů a zaznamenávání výsledků z testování a analýza a reportování defektů. Průběžnou činností vztahující se k procesu provádění testů je regresní testování a tzv. retest.

Cílem činnosti provádění testů a zaznamenávání výsledků z testování je vykonání testů dle testovacích sad složených z logicky uspořádaných testovacích případů tvořící testovací scénáře seřazených dle stanovených priorit. Zaznamenání výsledků z testování probíhá přímo do vytvořených testovacích scénářů. Výsledek testovacího případu na testovacím scénáři má čtyři možné konečné stavy. Jedná se o stavy Netestováno, Otestováno, Chyba a N/A. Zkratka N/A vychází s anglických slov not available a jedná se o stav testování, kdy není možné provést testovací případ. Důvodem neprovedení je například nedostupnost testovacích dat či testovacích prostředí. Ačkoliv je prováděno zaznamenávání výsledků testování do dílčích testovacích scénářů, není vytvářen průběžný report o celkovém stavu testování. Úkolem test reportu by mělo být průběžné informování o celkovém počtu dokončených testovacích scénářů, jejich výsledků a počtu testů, které je zapotřebí vzhledem k harmonogramu testů dokončit.

Během exekuce testů dochází k nalezení defektů, jejich analýze a hlášení. Defekt je ve společnosti vydefinován jako stav, kdy existuje rozdíl mezi očekávaným výsledkem deklarovaným v testovacím scénáři a skutečným výsledkem testů. Před reportování defektu



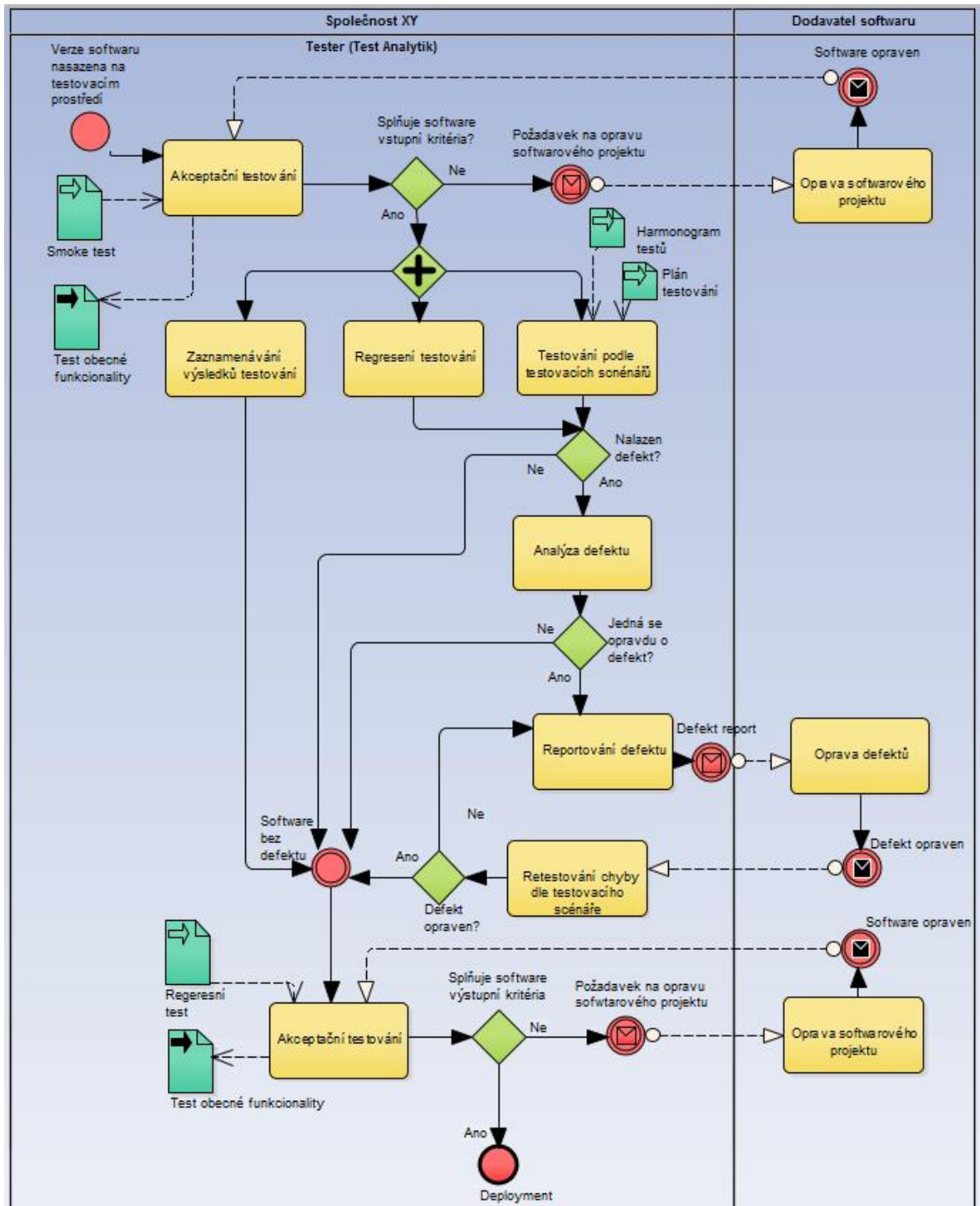
na dodavatele softwaru dochází k jeho analýze. Cílem analýzy je zjištění, zda se jedná o platný defekt report nebo pouze o chybně provedený test či požadované chování softwaru, které bylo dle požadavků uživatelů naprogramováno. V případě, že je od reportován neplatný defekt, dochází ze strany dodavatele softwaru k fakturaci neoprávněné reklamace navyšující náklady celého softwarového projektu.

V případě nahlášení platného defektu začíná životního cyklus defektu. Defekt report se zakládá pomocí specializovaného softwaru, který je ve správě dodavatele softwaru. Defekt report obsahuje popis nefunkční části softwaru, kroky vedoucí k defektu, prostředí, na kterém byl defekt zjištěn, závažnost, priorita, aj. Pomocí tohoto nástroje dochází i k monitorování stavu řešení defektu. Vlastníkem činnosti řízení životního cyklu defektu je dodavatel softwaru. Nalezené defekty jsou průběžně opravovány. Prostřednictvím Databázového správce jsou na žádost Testerů (Test Analytiků) nasazovány opravené verze softwaru na testovací prostředí, kde dochází k retestům softwaru.

Tester pomocí retestů prověří opravu defektu a pokud je vše v pořádku, uzavře Defekt report jako vyřešený. Pokud je retestem testovacího scénáře zjištěno neopravení defektu či zanesení nového defektu, je předán zpět dodavateli softwaru. Defekt je v řešení tak dlouho, dokud není potvrzeno ze strany Testera (Test Analytika) jeho vyřešení. Regresní testování je využíváno během akceptačního testování k ověření nezanesení defektů do částí softwaru, které nebyly předmětem softwarového projektu. Proces testování je řízen ze strany Testerů (Test Analytiků), ale není řízen pomocí měřitelných výstupů z důvodu chybějících metrik procesu a chybějícího test reportu. Po provedení všech naplánovaných testů dochází opět k akceptačnímu testování zahrnující testy obecných funkcionalit. V případě akceptace softwaru je proces provádění testů ukončen, čímž byl ukončen proces testování. Následuje poslední fáze životního cyklu softwaru označována jako Deployment, při kterém je otestovaný software nasazen během noční odstávky na produkční prostředí. Zde jsou již plnohodnotné reálné data a odstávka musí být vykonána mimo obchodní den (business time), kdy nejsou vykonávány obchodní případy a prováděny účetní operace.

Karta procesu	
Číslo a název procesu:	P 4 - Test Execution (Provádění testů)
Cíl procesu:	Cílem procesu je provedení testů a nalezení defektů
Vlastník procesu:	Tester (Test Analytik), dodavatel softwaru
Zákazník procesu:	Žádný
Vstupy procesu:	Testovací případy, scénáře a sady, připravené testovací data a prostředí s novou verzí softwaru
Činnosti procesu	
Číslo činnosti	
1.	Akceptační testování
2.	Provádění a zaznamenávání výsledků testů
3.	Analýza a reportování defektů dodavateli
4.	Retestování a regresní testování
Výstupy procesu:	Výsledky testů, Defekt report, Test obecné funkcionality.
Metriky procesu:	Žádné

Obr. 19. Karta procesu Provádění testů (vlastní zpracování)



Obr. 20. Proces Provádění testů (vlastní zpracování)

Kompletně zpracovaná analýza současného stavu byla následně předložena autorem diplomové práce Projektovému (Test) manažerovi, zaměstnancům AP a řediteli divize Projektů a IT. Po konzultaci dospěly všechny zúčastněné strany k závěru, že uvedené chybějící aspekty procesů jsou důsledkem chybějících procesů monitorování, řízení a vyhodnocení testování. Neexistující procesní oblasti jsou taktéž výsledkem chybějících metrik v ce-

lém procesu testování. Uvedené procesy byly z historického kontextu vývoje společnosti ignorovány a nebyla jim věnována žádná pozornost. Tento fakt byl způsoben dosavadním přístupem, kdy byl kladen důraz na agilní vývoj a testování obstarávající softwarovou infrastrukturu pro produkty a projekty zaměřené na spotřebitelské financování. Na agilním vývoji se podílí ve společnosti tým čítající 16 zaměstnanců.

## 6 NÁVRH ŘEŠENÍ

Cílem společnosti XY je možnost poskytovat softwarovou infrastrukturu svého finančního produktu Autoúvěr ve správě oddělení AP na nově i na americký trh. Aby byl tento krok možný, musí proces testování splňovat minimálně druhou úroveň zralosti (kvality) procesu testování softwaru ohodnocenou oproti modelu TMMi. Podmínka ohodnocení oproti modelu TMMi je správná, protože je model zralosti zaměřen na proces testování a současně poskytuje možnost využít neformální metody hodnocení s minimálními náklady.

Splněním této podmínky obdrží společnost XY konkurenční výhodu oproti ostatním členům finanční skupiny, čímž dojde k transferu znalostí a know-how ze softwarových projektů z České republiky a Slovenska na americký trh. Z výše uvedeného si společnost XY neklade za cílové spektrum tvrdé metriky, které jsou převoditelné na finanční prostředky. Cílem společnosti rovněž není redukce nákladů vynaložených na softwarové projekty, zvýšení zisku prostřednictvím nárůstu počtu profinancovaných vozů či zkrácení doby testování softwaru. Ústředním cílem je dosažení konkurenční výhody umožňující proniknout na americký trh s možnou budoucí vidinou uvedených efektů.

Dosažení požadované úrovně zralosti procesu testování je cílem analyticko-projektové části diplomové práce. Tento cíl je podpořen návrhem řešení na zlepšení řízení kvality procesu testování softwaru. Vedlejším předpokládaným efektem návrhu na zlepšení řízení kvality procesu testování softwaru je dosažení nižšího počtu defektů a s tím spojených nákladů na neoprávněnou reklamaci vůči dodavateli softwaru a vyšší kvalita samotného vyvíjeného softwaru.

Po vyhotovené procesní analýze současného stavu byly autorem diplomové práce zjištěny závažné nedostatky, které brání dosažení požadované zralosti procesu testování softwaru odrážející požadovanou úroveň řízení kvality testování softwaru. Níže uvedený přehled poukazuje na zjištěné nedostatky v procesu testování softwaru založeného na přístupu best-practices.

*Tab. 1. Zjištěné nedostatky v procesu testování softwaru ve společnosti XY (vlastní zpracování)*

Test Planning	Testovací plán nereflektuje změny, ke kterým v průběhu procesu testování softwaru dochází. Nedochází k prioritizaci rozsahu testování na základě
---------------	---

	<p>identifikovaných potřeb zainteresovaných stran na softwarovém projektu.</p> <p>Software je testován prostřednictvím přístupu založeného na požadavcích. Ačkoliv jsou rizika identifikována, nedochází k testování na základě přístupu založeného na rizicích.</p> <p>Nejsou definovány metriky pro řízení procesu testování.</p>
Test Development	Chybějící metriky procesu.
Test Implementation	<p>Nedochází ke kontrole databázových prostředí bezprostředně po jejich předání.</p> <p>Chybějící metriky procesu.</p>
Test Execution	<p>Nevytváří se Test report popisující průběžný stav testování softwaru.</p> <p>Chybějící metriky procesu.</p>

Mimo uvedené nedostatky jsou hlavním problémem chybějící podprocesy řízení a vyhodnocení procesu testování. Uvedené podprocesy jsou z pohledu řízení kvality softwaru klíčové. Není možné bez nich dosáhnout na druhou úroveň zralosti procesu testování softwaru v modelu TMMi.

Analyticko-projektová část diplomové práce se zabývá odstraněním zjištěných nedostatků v dílčích podprocesech procesu testování se současnou implementací chybějících podprocesů. Tento fakt je podpořen implementací chybějících metrik, bez kterých není možné účinně řídit jakýkoliv proces.

## 6.1 Metriky určené k měření procesu testování softwaru

Tab. 2. Produktivita tvorby testovacích scénářů (vlastní zpracování)

Název:	Produktivita tvorby testovacích případů (Test Case Productivity)	
Vstupní hodnoty:	Vzorec:	Příklad:
Celkový počet vytvořených testovacích řádků	$\frac{\text{(Celkový počet vytvořených testovacích řádků)}}{\text{(Počet hodin strávených psaním testovacích případů)}}$	Celkový počet vytvořených testovacích řádků = 50
Počet hodin strávených tvorbou testovacích případů		Počet hodin strávených tvorbou TC = 8
		Produktivita tvorby testovacích případů v počet kroků/h = $50/8 = 6,25$ kroků/h
Cíl:	Porovnávání trendu produktivity tvorby testovacích případů mezi dílčími vývojovými cykly	

Z důvodu omezeného času, který je vytyčen na psaní testovacích případů a scénářů bude sledována produktivita psaní testovacích případů. Každý testovací případ se skládá z testovacích kroků. Výsledná hodnota je uváděna v počtu vytvořených testovacích kroků za hodinu. Produktivita bude měřena na konci každého dne.

Tab. 3. Průběh provádění testů s výsledkem je v pořádku (vlastní zpracování)

Název:	Průběh provádění testů s výsledkem je v pořádku (Pass Trend)	
Vstupní hodnoty:	Vzorec:	Příklad:
Celkový počet testovacích případů ve stavu je proveden v pořádku	$\frac{\text{((Celkový počet testovacích případů ve stavu je proveden v pořádku))}}{\text{(Celkový počet provedených testovacích případů)}} * 100$	Celkový počet testovacích případů ve stavu je proveden v pořádku = 42
Celkový počet provedených testovacích případů		Celkový počet provedených testovacích případů = 87
		Průběh provádění testů s výsledkem je v pořádku v % = $(42/87) * 100 = 48,28\%$
Cíl:	Studium provádění testů s procentuálním zastoupením jednotlivých výsledků stavu testování na celkovém objemu provedených testů	

Tab. 4. Průběh provádění testů s výsledkem nelze provést (vlastní zpracování)

Název:	Průběh provádění testů s výsledkem nelze provést (Unable to Test Trend)	
Vstupní hodnoty:	Vzorec:	Příklad:
Celkový počet testovacích případů ve stavu nelze provést	$\frac{\text{((Celkový počet testovacích případů ve stavu nelze provést))}}{\text{(Celkový počet provedených testovacích případů)}} * 100$	Celkový počet testovacích případů ve stavu nelze provést = 15
Celkový počet provedených testovacích případů		Celkový počet provedených testovacích případů = 87
		Průběh provádění testů s výsledkem nelze provést v % = $(15/87) * 100 = 17,24\%$
Cíl:	Studium provádění testů s procentuálním zastoupením jednotlivých výsledků stavu testování na celkovém objemu provedených testů	

Tab. 5. Průběh provádění testů s výsledkem selhání (vlastní zpracování)

<b>Název:</b> Průběh provádění testů s výsledkem selhání (Fail Tend)		
<b>Vstupní hodnoty:</b>	<b>Vzorec:</b>	<b>Příklad:</b>
Celkový počet testovacích případů ve stavu dopadl selháním	$\left( \frac{\text{Celkový počet testovacích případů ve stavu dopadl selháním}}{\text{Celkový počet provedených testovacích případů}} \right) * 100$	Celkový počet testovacích případů ve stavu dopadl selháním = 30
Celkový počet provedených testovacích případů		Celkový počet provedených testovacích případů = 87
		Průběh provádění testů s výsledkem selhání v % = $(30/87) * 100 = 34,48\%$
<b>Cíl:</b>	Studium provádění testů s procentuálním zastoupením jednotlivých výsledků stavu testování na celkovém objemu provedených testů	

Sledování průběhu provádění testů v dílčích vývojových cyklech umožňuje klasifikovat testovací případy do kategorií dle jejich stavu. Testovací případy mohou být vykonány s následujícím výsledkem:

- testovací případ je proveden v pořádku,
- testovací případ nelze provést,
- testovací případ dopadl selháním softwaru.

Počet jednotlivých stavů testovacích případů bude posléze sledován k celkovému počtu provedených testovacích případů. Výsledné podílové hodnoty budou uváděny v procentuálním vyjádření. Průběh provádění testů bude reportován na konci každého dne během provádění testů. Informace budou současně sloužit ke sledování pokroku testování oproti harmonogramu testování.

Tab. 6. Míra zastoupení nahlášených defektů (vlastní zpracování)

<b>Název:</b> Míra zastoupení nahlášených defektů (Defect Acceptance)		
<b>Vstupní hodnoty:</b>	<b>Vzorec:</b>	<b>Příklad:</b>
Počet nahlášených defektů	$\left( \frac{\text{Počet nahlášených defektů}}{\text{Celkový počet nalezených defektů}} \right) * 100$	Počet nahlášených defektů = 13
Celkový počet nalezených defektů		Celkový počet nalezených defektů = 20
		Zastoupení nahlášených defektů v % = $(13/20) * 100 = 65\%$
<b>Cíl:</b>	Studium efektivnosti prováděných analýz defektů ze strany testerů	

V případě, že je objeven defekt, je následně tento defekt testerem před nahlášením analyzován. Pokud je defekt platný, je odeslán na dodavatele softwaru k opravení. Ukazatel sledující počet nahlášených defektů k celkovému počtu nalezených defektů testovacím



týmem je uváděn opět v procentech. Ukazatel se vztahuje k vývojovému cyklu a bude prezentován s denní odchylkou vždy na začátku pracovního dne.

Tab. 7. Míra zamítnutých defektů (vlastní zpracování)

<b>Název:</b> Míra zamítnutých defektů (Defect Rejection)		
<b>Vstupní hodnoty:</b>	<b>Vzorec:</b>	<b>Příklad:</b>
Počet zamítnutých defektů	(Počet zamítnutých defektů/Celkový počet nalezených defektů) *100	Počet zamítnutých defektů = 4
Celkový počet nalezených defektů		Celkový počet nalezených defektů = 20
		Míra zamítnutých defektů v % = (4/20)*100 = 20%
<b>Cíl:</b>	Studium chybovosti testerů při prováděných analýzách defektů navyšující náklady na testování	

Ukazatel pracuje s počtem zamítnutých defektů ze strany dodavatele softwaru, které byly nahlášený.

Spolu s ukazatelem Míry zastoupení nahlášených defektů bude výsledná hodnota prezentována na začátku pracovního dne. Jedná se o silný ukazatel. Čím nižší je výsledná hodnota, tím nižší budou dodatečné náklady na opravu defektu. Tento stav je nejčastěji způsoben nesprávnou interpretací požadavků kladených na softwarový projekt.

Tab. 8. Míra chybovosti oprav (vlastní zpracování)

<b>Název:</b> Míra chybovosti oprav (Bad Fix Defect)		
<b>Vstupní hodnoty:</b>	<b>Vzorec:</b>	<b>Příklad:</b>
Počet chybně opravených defektů	(Počet chybně opravených defektů/Celkový počet nahlášených defektů)*100	Počet chybně opravených defektů = 2
Celkový počet nahlášených defektů		Celkový počet nahlášených defektů = 13
		Míra chybovosti oprav v % = (2/13)*100 = 15,38%
<b>Cíl:</b>	Studium chybovosti dodavatele při opravách nahlášených defektů	

Při testování softwaru dochází občas ze strany dodavatele k neopravení nahlášeného defektu, nebo dokonce k zanesení defektu nového. Ukazatel je vyjádřen v procentech. Čím vyšší je ukazatel, tím horší je vykonaná práce ze strany dodavatele. Jedná se do budoucna o silný argumentační ukazatel, kdy bude docházet k dodatečným nákladům způsobeným pochybným ze strany společnosti při současné nekvalitně provedené opravě defektu ze strany dodavatele softwaru. V případě, že by ukazatel překročil 20%, bude společnost XY uvažovat o změně dodavatele softwaru.

Tab. 9. Odchylka odhadu pracnosti (vlastní zpracování)

Název:	Odchylka odhadu pracnosti (Effort Variation)	
Vstupní hodnoty:	Vzorec:	Příklad:
Současná pracnost	$\frac{(\text{Současná pracnost} - \text{Odhad pracnosti})}{(\text{Odhad pracnosti})} * 100$	Odhad pracnosti (MD) = 10
Odhad pracnosti		Současná pracnost (MD) = 12
		Odchylka odhadu pracnosti v % = $(12-10)/10*100 = 20\%$
Cíl:	Studium rozložení pracovní zátěže se současným snižováním hodnoty odchylky odhadu pracnosti	

Odhad pracnosti a současná pracnost je uváděna v jednotkách člověkodenní (MD). Výsledná hodnota bude uváděna v procentuálním vyjádření. V průběhu procesu testování budou uváděny následující Odchylky odhadu pracnosti:

- Odchylka odhadu pracnosti procesu Test Planning.
- Odchylka odhadu pracnosti procesu Test Development.
- Odchylka odhadu pracnosti procesu Test Implementation.
- Odchylka odhadu pracnosti procesu Test Execution.
- Odchylka odhadu pracnosti procesu Test Management.
- Odchylka odhadu pracnosti procesu Test Evaluation.

Odchylky odhadu pracnosti poukazují zároveň i na účinnost plánování s dopadem na cenu testování softwaru a časových harmonogramů testování.

Tab. 10. Odchylka časových termínů (vlastní zpracování)

Název:	Odchylka časových termínů (Schedule Variation)	
Vstupní hodnoty:	Vzorec:	Příklad:
Skutečné datum ukončení	$\frac{((\text{Skutečné datum ukončení} - \text{Plánované datum ukončení}) / (\text{Plánované datum ukončení} - \text{Plánované datum zahájení})) * 100$	Skutečné datum ukončení = 1.3.2015
Plánované datum ukončení		Plánované datum ukončení = 28.2.2015
Plánované datum zahájení		Plánované datum zahájení = 1.2.2015
		Odchylka časových termínů v % = $(1/28)*100 = 3,57\%$
Cíl:	Studium schopnosti dosažení predikovaných časových milníků s cílem snížení odchylky od stanovených časových termínů	

Metrika indikující schopnost dodržet stanovené termíny bude rovněž implementována pro všechny důležité termíny nacházejících se ve vytvořeném harmonogramu testování. Výsledná hodnota bude uváděna v procentuálním vyjádření. Čím více se bude ukazatel blížit 0, tím vyšší je schopnost dodržovat plánované termíny celého procesu testování.

Tab. 11. Míra vynaložených nákladů na testování (vlastní zpracování)

Název:	Míra vynaložených nákladů na testování (Cost of Testing)	
Vstupní hodnoty:	Vzorec:	Příklad:
Náklady na testování	$\left(\frac{\text{Náklady na testování}}{\text{Celkové náklady}}\right) * 100$	Náklady na testování = 250 000 Kč
Celkové náklady		Celkové náklady = 1 400 000 Kč
		Míra vynaložených nákladů na testování v % = $(250000/1400000) * 100 = 17,86\%$
Cíl:	Zjištění výše nákladů připadajících na proces testování	

Ukazatel vyjadřuje procentuální výši nákladů procesu testování na celkových nákladech softwarového projektu. Stav nákladů slouží k porovnání plánovaného finančního odhadu nákladů připadajících na proces testování se skutečným stavem.

Tab. 12. Hodnota nalezeného defektu (vlastní zpracování)

Název:	Hodnota nalezeného defektu (Cost of Defect)	
Vstupní hodnoty:	Vzorec:	Příklad:
Náklady na testování	$\frac{\text{Náklady na testování}}{\text{Počet nahlášených defektů}}$	Náklady na testování = 250 000 Kč
Počet nahlášených defektů		Počet nahlášených defektů = 13
		Hodnota nalezeného defektu v Kč = 19 230,77 Kč
Cíl:	Zjištění hodnoty nahlášeného defektu	

Tento ukazatel reflektuje hodnotu nalezeného defektu v průběhu testů. Ukazatel bude reportován směrem k projektovému manažerovi. Hodnota ukazuje, jak důležitý je proces testování. Cena defektu by vzrostla za předpokladů, že by nebyl defekt objeven a svým chováním by způsobil v systému nevyčíslitelné škody.

Tab. 13. Stav čerpání rozpočtu (vlastní zpracování)

Název:	Stav čerpání rozpočtu (Drawing Budget)	
Vstupní hodnoty:	Vzorec:	Příklad:
Současná výše nákladů na testování	Současná výše nákladů na testování/Rozpočet testování	Současná výše nákladů na testování = 250 000 Kč
Rozpočet testování		Rozpočet testování = 275 000 Kč
		Stav čerpání rozpočtu = 0,9
Cíl:	Posouzení aktuálního stavu čerpání přidělených finančních prostředků na proces testování softwaru	

Výslednou hodnotou ukazatele je bezrozměrný koeficient, který nám udává aktuální čerpání finančních prostředků stanovených pro proces testování. Koeficient menší než 1 říká, že jsme nevyčerpali přidělené finanční prostředky. Koeficient roven 1 udává stav, kdy jsou náklady na testování rovny rozpočtu na testování. Hodnota vyšší než 1 udává přečerpání finančních prostředků. V případě přečerpání finančních prostředků, podává Projektový (Test) manažer důvody řediteli Divize a IT, které způsobily tento stav.

Tab. 14. Úroveň kvality procesu testování (vlastní zpracování)

Název:	Úroveň kvality procesu testování (Quality of Testing)	
Vstupní hodnoty:	Vzorec:	Příklad:
Počet nahlášených chyb během testování	(Počet nahlášených chyb během testování / (Počet nahlášených chyb během testování + Počet chyb nahlášených během reálného provozu)) * 100	Počet nahlášených chyb během testování = 13
Počet chyb nahlášených během reálného provozu		Počet chyb nahlášených během reálného provozu = 2
		Úroveň kvality procesu testování v % = $(13/(13+2))*100 = 86,67\%$
Cíl:	Stanovení úrovně kvality procesu testování ve vztahu k nahlášeným defektům v průběhu testování a během provozu softwaru	

Počet nahlášených chyb během ostrého provozu snižuje procentuální vyjádření kvality procesu testování softwaru. Čím vyšší je počet nalezených chyb během reálného provozu, tím nižší je kvalita procesu testování, ale vyšší kvalita provozovaného softwaru. Úroveň kvality bude vyhodnocována jednou za dva týdny a předkládána všem zainteresovaným skupinám podílejících se na procesu testování softwaru.

Tab. 15. Míra pokrytí testů (vlastní zpracování)

<b>Název:</b> Míra pokrytí testů (Test Coverage)		
<b>Vstupní hodnoty:</b>	<b>Vzorec:</b>	<b>Příklad:</b>
Počet vytvořených testovacích scénářů	(Počet vytvořených testovacích scénářů/Celkový počet zadávacích studií)*100	Počet vytvořených testovacích scénářů = 20
Celkový počet zadávacích studií		Celkový počet zadávacích studií = 34
		Míra pokrytí testů v % = $(20/34)*100 = 58,82 \%$
<b>Cíl:</b>	Znalost průběhu pokrytí zadávacích studií tvorbou testovacích scénářů	

Výsledná hodnota je uváděna v procentech. Čím více se blíží výsledná hodnota 100%, tím vyšší je stupeň pokrytí testovacích scénářů reflektující testovací požadavky. Ukazatel bude předkládán v průběhu návrhu testů na konci pracovního dne.

Tab. 16. Výše opravenosti defektů (vlastní zpracování)

<b>Název:</b> Výše opravenosti defektů (Bug Fix Rate)		
<b>Vstupní hodnoty:</b>	<b>Vzorec:</b>	<b>Příklad:</b>
Počet opravených defektů	(Počet opravených defektů/Počet nahlášených defektů)*100	Počet opravených defektů = 8
Počet nahlášených defektů		Počet nahlášených defektů = 13
		Výše opravenosti defektů v % = $(8/13)*100 = 61,54\%$
<b>Cíl:</b>	Studium současného stavu procesu testování	

Ukazatel nám říká, kolik procent nahlášených defektů bylo opraveno. Poměr nahlášených a opravených defektů vypovídá o stavu softwarového projektu. Pokud se snižuje počet nahlášených chyb a zvyšuje se počet chyb opravených, je to náznak, že proces testování spěje ke svému konci.

Vytvořené metriky budou v rámci dalších podkapitoly přiřazeny k jednotlivým podprocesům v procesu testování softwaru. Tyto metriky budou uvedeny na Kartě procesu. Revize metrik bude probíhat vždy v podprocesu plánování testů. Tento krok zajistí stav, kdy metriky odpovídají aktuálním potřebám s ohledem na specifikace softwarového projektu. Může tak dojít k úplné eliminaci některých výše uvedených metrik, nebo naopak k vytvoření metrik nových, které budou odrážet současnou situaci.

## 6.2 Úprava stávajících podprocesů

Následující podkapitoly diplomové práce prezentují návrh úpravy stávajících podprocesů testování softwaru ve společnosti XY.

### 6.2.1 Test Planning

První úpravou v podprocesu plánování testů bude zjednodušení plánu testování se zachováním celopodnikových zásad (Test policy) a strategií (Test strategy). Plán testování zřetelně vydefinoje cíl testování, rámce testování, způsob testování (strategie), rizika, potřebné zdroje, artefakty, harmonogram testování, matice zodpovědností, metriky, identifikace vstupních, výstupních, pozastavovacích a obnovovacích kritérií. Plán testování bude v současné podobě zrevidován a při jeho tvorbě bude využita standardní forma popsaná v normě ISO/IEC/IEEE 29119-3 Software and systems engineering - Software testing – Test Documentation. Vytvořený plán testování bude předložen ke kontrole všem zainteresovaným skupinám s cílem potvrzení jeho obsahu. Rovněž bude docházet k průběžné aktualizaci plánu testování reflektující případné odchýlení od tohoto plánu testování.

Důležitým prvkem procesu plánování testů je odhadnutí pracnosti testování pomocí přístupu založeného na metrikách. Následně bude docházet k identifikaci zdrojů a sestavení harmonogramu testování.

Vydefinovaná rizika budou implementována do vstupních, pozastavujících, obnovovacích a výstupních kritérií testování čímž dojde k prolnutí principu testování založeného na požadavcích a rizicích. V případě, že bude mít riziko větší prioritu než požadavek, bude uplatněno testování zakládající se čistě na rizicích. Součástí úpravy obsahu procesu je i jeho organizační struktura. Projektový (Test) manažer bude zastávat roli Projektového manažera softwarových projektů pro vyvíjenou softwarovou infrastrukturu produktu Auto-úvěr. Stávající agendu spadající pod proces testování převezme zaměstnanec AP v roli Test manažera.

V podprocesu plánování testů bude rovněž docházet k revizím vytvořených metrik. Cílem je pokrytí současných potřeb v celém procesu testování.

Poslední úpravou je prioritizace rozsahu testování. Ten nebude prioritozován skrz zainteresované skupiny, ale bude reflektovat způsob testování, který je na požadavcích založen. Potřeby zainteresovaných stran budou identifikovány v první fázi životního cyklu softwa-

ru, což zaručí promítnutí těchto potřeb do rozsahu testování. Ten bude posléze testerem předkládán ke schválení zainteresovaným skupinám.

Karta procesu	
Číslo a název procesu:	P 1 - Test Planning (Plánování testů)
Cíl procesu:	Vytvoření plánu testování
Vlastník procesu:	Test manažer
Zákazník procesu:	Proces Test Development (Návrh testů)
Vstupy procesu:	Softwarový projekt a jeho rozpočet, strategie a politika testování
Činnosti procesu	
Číslo činnosti	
1.	Definice cílů a rozsahu testování
2.	Analýza rizik testování
3.	Definice přístupu k testování
4.	Ostatní činnosti
Výstupy procesu:	Plán testování
Metriky procesu:	Odchylka odhadu pracnosti, Odchylka časových termínů, Míra vynaložených nákladů na testování, Úroveň kvality procesu testování, Míra pokrytí testů

Obr. 21. Doplněná Karta procesu Plánování testů (vlastní zpracování)

### 6.2.2 Test Development

Podproces zabývající se návrhem testů netrpí závažným nedostatkem, který by vedl k úpravě podprocesu. Chybějící metriky jsou zaneseny do Karty procesu a následně sledovány, vyhodnocovány a interpretovány zainteresovaným skupinám.

Karta procesu	
Číslo a název procesu:	P 2 - Test Development (Návrh testů)
Cíl procesu:	Vytvoření testovacích případů a scénářů
Vlastník procesu:	Tester (Test Analytik), Test manažer
Zákazník procesu:	Podproces Test Implementation (Implementace testů)
Vstupy procesu:	Plán testování, Vývojové studie
Činnosti procesu	
Číslo činnosti	
1.	Analýza vývojových studií
2.	Návrh testů
3.	Identifikace potřeb testovacích dat
4.	Identifikace potřeb testovacích prostředí
Výstupy procesu:	Testovací případ, Testovací scénář
Metriky procesu:	Produktivita tvorby testovacích případů, Odchylka odhadu pracnosti, Odchylka časových termínů, Míra pokrytí testů

Obr. 22. Doplněná Karta procesu Návrh testů (vlastní zpracování)

### 6.2.3 Test Implementation

Podproces implementace testů bude nově obsahovat činnost zabývající se kontrolou databázových prostředí bezprostředně po jejich předání. Tato činnost bude vykonávána Teste-rem (Test Analytikem), který prostřednictvím smoke testů k tomu určených ověří korektnost databázových prostředí. Pomocnou vizuální kontrolou bude očíslování nasazené revize databázového prostředí, která se musí shodovat s číselným označením revize, jež bude uvedena v požadavcích na testovací prostředí. Logická stavba stávající podprocesu je dostačující. Chybějící metriky jsou uvedeny na Kartě procesu.



Karta procesu	
Číslo a název procesu:	P 3 - Test Implementation (Implementace testů)
Cíl procesu:	Přípravení testovacích dat, prostředí, vytvoření testovacích sad, sestavení harmonogramu testování
Vlastník procesu:	Test manažer, Tester (Test Analytik), Správce databází
Zákazník procesu:	Proces Test Execution (Provedení testů)
Vstupy procesu:	Požadavky na testovací data a prostředí, Testovací případy a scénáře
Činnosti procesu	
Číslo činnosti	
1.	Příprava testovacích prostředí a jejich kontrola
2.	Příprava testovacích dat
3.	Vytvoření harmonogramu prováděných testů
4.	Organizace a prioritizace testovacích případů a scénářů
Výstupy procesu:	Testovací data, Harmonogram testů, Testovací sada
Metriky procesu:	Odchylka odhadu pracnosti, Odchylka časových termínů, Míra vynaložených nákladů na testování

Obr. 23. Doplněná karta Procesu Implementace testů (vlastní zpracování)

#### 6.2.4 Test Execution

V průběhu podprocesu provádění testů bude docházet prostřednictvím nového podprocesu řízení testování k průběžné tvorbě Test reportů identifikující aktuální vývoj a stav podprocesu provádění testů. Při tvorbě Test reportu bude využita standardní forma popsána v normě ISO/IEC/IEEE 29119-3 Software and systems engineering - Software testing – Test Documentation. Za Test report bude zodpovědný Test manažer. Metriky budou uvedeny v Kartě procesu.

Karta procesu	
Číslo a název procesu:	P 4 - Test Execution (Provádění testů)
Cíl procesu:	Provedení testů a nalezení defektů
Vlastník procesu:	Tester (Test Analytik), dodavatel softwaru, Test manažer
Zákazník procesu:	Test Evaluation (Vyhodnocení testů)
Vstupy procesu:	Testovací případy, scénáře a sady, připravené testovací data a prostředí s novou verzí softwaru
Činnosti procesu	
Číslo činnosti	
1.	Akceptační testování
2.	Provádění a zaznamenávání výsledků testů
3.	Analýza a reportování defektů dodavateli
4.	Retestování a regresní testování
Výstupy procesu:	Výsledky testů, Defekt report, Test obecné funkcionality, Test report
Metriky procesu:	Průběh provádění testů s výsledkem je v pořádku, nelze provést, selhání, Míra zamítnutých defektů, Míra chybovosti oprav, Odchylka odhadu pracnosti, Odchylka časových termínů, Míra vynaložených nákladů na testování, Hodnota nalezeného defektu, Stav čerpání rozpočtu, Úroveň kvality procesu testování, Výše opravenosti defektů

Obr. 24. Doplněná Karta procesu Provádění testů (vlastní zpracování)

### 6.3 Vytvoření chybějících podprocesů

V průběhu procesní analýzy byly zjištěny chybějící podprocesy procesu testování. Vytvoření těchto podprocesů je náplní následujících podkapitol.

### 6.3.1 Test Management

Cílem nového podprocesu bude zajištění souladu mezi prováděnými aktivitami testování a plánem. Řízení testů se bude zakládat na průběžné analýze procesu testování a jeho výsledků, monitorování procesu testování, získávání vstupních hodnot pro metriky procesu testování. Případná náprava zjištěných problémů bude probíhat prostřednictvím dvou možných způsobů:

- úpravou testovacího plánu,
- sjednáním nápravných opatření.

Součástí procesu řízení testování bude tvorba tzv. Test reportů. Vstupní hodnoty budou získávány prostřednictvím vytvořených a namapovaných metrik procesu testování. Ty budou následně zpracovávány v průběhu celého procesu testování. Metriky, které budou sledovány v průběhu procesu testování, jsou uvedeny v předchozí podkapitole diplomové práce. Výsledky budou rovněž uvedeny do souvislostí a interpretovány zainteresovaným skupinám.

Monitorování v průběhu celého procesu testování přináší sebou i sledování životního cyklu defektu, které jsou zadávány v průběhu podprocesu Test Execution. Mimo životní cyklus defektu je monitorován stav testovacích prostředí. Komunikace musí být silnou stránkou Test manažera, který zajišťuje, zprostředkovává a předává informace všem zainteresovaným skupinám na testování.

Podproces řízení testování se tak zakládá na sledování postupu testování, čímž dochází ke sběru informací. Ty jsou zhodnoceny, což může vyvolat rozhodnutí zahrnující změnu priorit založených na výskytu rizika, změně testovacího harmonogramu z důvodu nedostupnosti testovacích prostředků či dat, pochybení ze strany dodavatele softwaru při opravě defektu aj. V průběhu testování dochází k reportování prostřednictvím výstupních artefaktů (výstupů z procesu) uvedených na Kartě procesu. Rolí zodpovědnou za podproces řízení testování je Test manažer.

Součástí řízení testování bude i aspekt motivace testerů, která má taktéž vliv na kvalitu vykonané práce.

Pro řízení kvality nebude vytvořen model procesů z důvodu komplexnosti a složitosti podprocesu zasahujícího do všech procesů vyskytujících se v procesu testování softwaru.

Karta procesu	
Číslo a název procesu:	P 5 - Test Management (Řízení testů)
Cíl procesu:	Cílem procesu je zajištění souladu mezi prováděnými aktivitami testování a testovacím plánem
Vlastník procesu:	Test manažer
Zákazník procesu:	Zainteresoované skupiny
Vstupy procesu:	Plán testování, Metriky testování, Výsledky testů, Defekt report
Činnosti procesu	
Číslo činnosti	
1.	Sledování postupu testování (monitorování)
2.	Průběžné měření testování
3.	Průběžné reportování o stavu testování
4.	Náprava zjištěných nedostatků
5.	Motivace členů testovacího týmu
Výstupy procesu:	Test report, Plán testování, Záznamy z jednání
Metriky procesu:	Míra vynaložených nákladů na testování, Hodnota nalezeného defektu, Úroveň kvality procesu testování

Obr. 25. Karta procesu Řízení testů (vlastní zpracování)

### 6.3.2 Test Evaluation

Podproces Test Execution neboli provádění testů byl posledním podprocesem ve stávající struktuře procesů testování. V případě akceptace softwaru byl proces provádění testů ukončen. Akceptace softwaru se zakládá na komparaci výstupních kritérií kladených na softwarový produkt v plánu testování a jeho aktuální stav, který je zjišťován při testech obecné funkcionality. Vytvořením nového podprocesu vyhodnocení testování bude docházet ke kontrole výstupních kritérií kladených na proces testování a k uzavření celého pro-

cesu testování. Tyto výstupní kritéria budou stanovena v plánu testování. Zodpovědnou roli za podproces vyhodnocení testování bude mít Test manažer. Výstupním artefaktem z podprocesu vyhodnocení testování bude Summary Test report.

K vyhodnocení testování ve vztahu k vyvíjenému softwaru dochází v samotném současném závěru podprocesu provádění testů, kdy probíhá akceptační testování ověřující kvalitu testovaného softwarového produktu. Tato činnost bude přenesena do podprocesu vyhodnocení testování. Této činnosti bude předcházet ověření kvality procesu testování. Vyhodnocení testování bude spočívat v komparaci vytvořených tvrdých a měkkých výstupních kritérií popsané v plánu testování. V případě nesplnění měkkých výstupních kritérií bude proces testování regulárně pokračovat koncovou akceptací softwaru. Pokud se však vyskytne atribut nesplnitelnosti u tvrdých výstupních kritérií, musí být sjednána náprava a proces testování nemůže pokračovat akceptační činností.

Přeskupení podprocesů má za cíl vyhodnotit testování s následnou akceptací, ověřující kvalitu softwarového produktu. Implementace této činnosti může vyvolat posunutí současných časových milníků harmonogramu z důvodu nesplnění tvrdých výstupních kritérií. V případě, že nebude možné sjednat nápravu tak, aby byla splněna v přiměřené době tvrdá kritéria, budou kontaktovány zainteresované skupiny s žádostí o vyjádření a schválení výjimky s vědomím, že při akceptačním testováním může mít tento krok dopad na kvalitu softwaru v podobě defektu, čímž dojde opět k posunutí časových milníků v harmonogramu testů a generování dodatečných nákladů.

Příkladem tvrdého výstupního kritéria je validace softwarového projektu. Při validaci nejsou zjištěny defekty, ale neshoda mezi tím, co zainteresované skupiny očekávali a tím, co je výstupem v procesu testování. Nápravou tohoto kritéria bude tzv. Změnový požadavek (Change Request), který bude předán dodavateli softwaru k zapracování. Změnový požadavek generuje dodatečné náklady nejen ze strany dodavatele, ale také ze strany společnosti XY, z důvodu posunutí časových termínů nasazení softwarového produktu do reálného provozu. Tento aspekt může posléze vyvolat ztrátu finančních prostředků z neprováděných obchodních činností. Po vyhodnocení testování a akceptačním testování bude docházet k uzavření testování. Za tuto činnost bude opět zodpovědný pracovník v roli Test manažera.

Ukončení testování začíná kontrolou úplnosti testování. Cílem této aktivity je ověření, že všechny práce vztahující se k procesu testování jsou dokončeny. Nejčastěji se bude jednat o kontrolu provedených naplánovaných testů, stavy oprav defektů a náprav, které byly

provedeny v průběhu celého procesu testování. Následuje aktivita archivace výstupních artefaktů z procesu testování. Cílem je uchování informací pro budoucí využití na nových softwarových projektech. Tyto archivované artefakty jsou umístěny na společné uložení, čímž dojde k předání informací všem zainteresovaným skupinám. Dostupnost informací bude řízena prostřednictvím přístupových práv, čímž bude zajištěno, že budou informace dostupné správným osobám.

Během celého procesu testování dochází k setkání mezi zainteresovanými skupinami, aby byly kontinuálně předávány informace týkající se aktuálního vývoje testů. Tyto setkání budou rozšířeny o tzv. retrospektivní meeting. Náplní tohoto setkání bude výměna zkušeností po ukončení testování mezi jednotlivými členy testovacího týmu, představení odvedené práce a vyhodnocení soutěží, které budou jedním z motivačních faktorů nacházejících se v procesu testování.

Karta procesu	
Číslo a název procesu:	P 6 - Test Evaluation (Vyhodnocení testů)
Cíl procesu:	Vyhodnocení testování a ověření kvality softwarového produktu s následným ukončením celého procesu testování
Vlastník procesu:	Test manažer
Zákazník procesu:	Zainteresované skupiny, testovací tým
Vstupy procesu:	Plán testování, vývojové studie, Test Case, Test Scenario, Metriky testování, Test report, Záznam z jednání, Testovací data, Testovací sada, Výsledky testů, Defekt report, Test obecné funkcionality
Činnosti procesu	
Číslo činnosti	
1.	Vyhodnocení testování vůči stanoveným kritériím
2.	Akceptační testování
3.	Kontrola úplnosti testování
4.	Archivace výstupních artefaktů
5.	Předání výstupních artefaktů
6.	Retrospektivní meeting
Výstupy procesu:	Summary Test report
Metriky procesu:	Odchylka odhadu pracnosti, Odchylka časových termínů, Úroveň kvality procesu testování

Obr. 26. Nová Karta procesu Vyhodnocení testů (vlastní zpracování)

## 7 NÁVRH ŘEŠENÍ PRO ZLEPŠENÍ ŽÍZENÍ KVALITY PROCESU TESTOVÁNÍ SOFTWARE VE SPOLEČNOSTI XY

Následující podkapitoly diplomové práce popisují hlavní prvky proveditelnosti projektu. Mezi tyto prvky patří specifikace cíle projektu, analýza jeho zdrojů s nákladovým ohodnocením a definice rizik. Dokument popisující uvedené prvky je ve společnosti označován jako Business Case.

### 7.1 Cíle projektu

Cílem projektu je dosažení druhé úrovně zralosti nazývanou Managed v procesu testování softwaru ohodnocenou oproti modelu TMMi. Projektový cíl má za úkol podpořit společnost XY, která si klade za cíl získat konkurenční výhody oproti ostatním členům finanční skupiny, čímž bude umožněno poskytovat softwarovou infrastrukturu jejího finančního produktu Autoúvěr na nově i na americký trh. Možnost vyvíjet softwarovou infrastrukturu svého produktu Autoúvěr pro americký trh by přinesla transfer znalostí a know-how ze softwarových projektů z České republiky a Slovenska, čímž by došlo k podpoře hlavních cílů finanční skupiny.

### 7.2 Časový harmonogram jednotlivých činností projektu

Fáze projektu	Říjen 2014 - Červen 2015								
	Říjen	Lистопад	Prosinec	Leden	Únor	Březen	Duben	Květen	Červen
<i>I. Přípravná fáze</i>									
Analýza zdrojů projektu									
Časový harmonogram projektu									
Nákladová analýza projektu									
Analýza rizik projektu									
<i>II. Návrh řešení pro zlepšení řízení kvality procesu testování softwaru</i>									
Definice cíle projektu									
Procesní analýza současného stavu									
Modelování současného stavu									
Vytvoření chybějících metrik v procesu testování									
Úprava současných podprocesů v procesu testování									
Vytvoření chybějících procesů									
<i>III. Implementace návrhu</i>									
Integrace změn do procesu testování									
Časová rezerva									

Obr. 27. Časový harmonogram projektu (vlastní zpracování)



### 7.3 Analýza zdrojů

Plánování zdrojů je činností, která předchází nákladové analýze projektu. Společnost XY uplatňuje při projektovém řízení rozčlenění zdrojů na tři kategorie:

- lidské zdroje,
- materiálové zdroje,
- finanční zdroje.

Vzhledem ke skutečnosti, že společnost XY zaměstnává 885 zaměstnanců a řadí se tak mezi velké podniky, všechny činnosti spojené s projektem budou realizovány z vnitřních zdrojů. Pro zpracování lidských zdrojů je využívána forma organizačního diagramu projektu s níže uvedenými rolemi.

Projektová role	Začlenění v org. struktuře	Role v org. struktuře
Schvalovatel projektu	Zaměstnanec divize Projektů a IT	Ředitel divize Projektů a IT
Zpracovatel a realizátor projektu, Hodnotitel projektu	Zaměstnanec oddělení Aplikační podpory	Test manažer
Konzultant projektu, Hodnotitel projektu	Zaměstnanec oddělení Aplikační podpory	Projektový manažer rigorózního týmu
Konzultant projektu	Zaměstnanec oddělení Projektů	Projektový manažer agilního týmu

Obr. 28. Organizační diagram projektu (vlastní zpracování)

V důsledku realizace projektu bude zapotřebí doplnit pracovní pozici Tester (Test Analytik), kterou zastává budoucí Test manažer. Tato pozice bude obsazena interním zaměstnancem společnosti XY v rámci probíhající restrukturalizace. Budoucí Projektový manažer rigorózního týmu zastával roli Projektové (Test) manažera. Zodpovědnost a pravomoc obdrží Test manažer. Tímto krokem bude moct Projektový manažer soustředit své kapacity na činnosti spojené se softwarovými projekty.

Finanční zdroje uvolněné pro realizaci projektu budou čerpány z vlastních prostředků společnosti. Společnost XY investuje ročně do projektů desítky miliony korun, a to ať už se jedná o projekty softwarové, obchodní či procesní, na všechny z nich jsou vždy předem alokovány finanční prostředky.

Společnost XY disponuje všemi materiálovými zdroji, které budou figurovat v průběhu realizace projektu. Jedná se o níže uvedené materiálové zdroje:

- technické vybavení pracovníků společnosti XY zahrnující pracovní notebooky a mobilní telefony
- softwarové vybavení Enterprise Architect sloužící pro modelování procesů
- softwarové vybavení kancelářským balíkem Microsoft Office
- vybavená zasedací místnost vyhrazená pro účely projektu (datapojektor, whiteboard, konferenční telefon, klimatizace, síťová a energetická infrastruktura)

#### **7.4 Nákladová analýza projektu**

Jak již bylo řečeno výše, společnost XY disponuje vlastními finančními, materiálovými a lidskými zdroji. Nákladovou položku překládaného projektu bude navýšení mezd ve formě odměn a benefitů zodpovědným osobám podílejících se na projektu. Mzdové náklady budou odhadnuty ve vztahu k odpovědnostem, pravomocem a kompetencím jednotlivých účastníků projektu, a to s ohledem na přípravu a realizaci projektu. Náklady za zaškolení nového zaměstnance nebudou připočítány, protože samotné školení bude realizováno interním zaměstnancem, který má tyto činnosti v popisu práce.

Součástí nákladových položek budou předpokládané nepřímé náklady a náklady na administrativní vedení projektu. Schvalovatel projektu zároveň trval na připočítání 10% rezervy z celkových plánovaných nákladů na nepředvídatelné výdaje. V případě kladného závěru projektu bude oněch nevyužitých 10% alokováno na teambuildingové aktivity rigorózního týmu.

Jednodenní školení na využití softwarového vybavení Enterprise Architect pro zpracovatele projektu poskytl v rámci rozvojového programu Projektový manažer agilního týmu.

Tab. 17. Mzdové náklady projektu (vlastní zpracování)

Nákladová položka	Kč
Odměna pro budoucího Test manažera	100 000
Odměna pro budoucího Projektového manažera rigorózního týmu	55 000
Odměna pro současného Projektového manažera agilního týmu	35 000
Náklady na zdravotní pojištění (9%)	17 100
Náklady na sociální pojištění (25%)	47 500
<b>Celkem</b>	<b>254 600</b>

Tab. 18. Ostatní náklady projektu (vlastní zpracování)

Nákladová položka	Kč
Administrativní náklady	9 000
Nepřímé náklady	12 500
<b>Celkem</b>	<b>21 500</b>

Tab. 19. Rezerva ve výši 10% z celkových plánovaných nákladů (vlastní zpracování)

Nákladová položka	Kč
Nepředvídatelné výdaje (10%)	27 610
<b>Celkem</b>	<b>27 610</b>

Tab. 20. Tab. 21. Celkové náklady na projekt (vlastní zpracování)

Nákladová položka	Kč
Mzdové náklady	254 600
Ostatní náklady	21 500
Nepředvídatelné výdaje	27 610
<b>Celkem</b>	<b>303 710</b>

Na základě výše uvedené nákladové analýzy projektu byly celkové náklady vyčísleny na 303.710,- Kč. Mzdové náklady činí skoro 84% z celkových nákladů na projekt.

## 7.5 Analýza rizik projektu

Předkládaný projekt může ohrozit řada rizik s odlišnou mírou dopadu a různorodou pravděpodobností jejich výskytu. Standardní postup společnosti XY při tvorbě analýze rizik je následující:

- identifikace rizik spojených s projektem,
- dílčí vyhodnocení rizik s ohledem na pravděpodobnost vzniku a míry dopadu rizika na překládaný projekt,
- stanovení možného postupu vedoucího k eliminaci potenciálních rizik.

### 7.5.1 Identifikace rizik

V této podkapitole jsou identifikována potenciální rizika, která se pojí s návrhem a realizací projektu:

1. První rizikovou oblastí je nedostačující aklimatizace realizátora projektu (autora diplomové práce) ve společnosti XY vedoucí k nepochopení současných firemních i nefiremních procesů, které se v průběhu procesu testování vyskytují. Uvedené riziko může mít za následek vysokou chybovost v modelování procesů.
2. Další rizikovou oblastí je odpor současných zaměstnanců pracujících v rigorózním týmu vůči navrženým a implementovaným procesním změnám pro současný proces testování postavený na metodě best-practices.
3. Třetí rizikovou skupinou je návrh a implementace projektu, která ve výsledku nepovede ke stanovenému cíli a společnost XY neobdrží konkurenční výhodu oproti ostatním členům finanční skupiny. Riziko se odvíjí od předchozích zkušeností realizátora projektu na obdobných projektech.
4. Za rizikovou oblast lze také považovat vyčerpání kapacit jednotlivých členů projektové týmu uvedeného v analýze lidských zdrojů. Mimo participace na projektu, musí všichni členové vykonávat svůj dennodenní popis práce v jednotlivých rolích.
5. S rizikem modelování procesů se pojí riziko správné kombinace metrik, která bude sloužit pro řízení procesu testování. Špatně zvolená metrika povede k tomu, že nebude možné vyhodnotit stávající stav či zřídit nápravu v dostatečném čase, s minimálními náklady a ve vysoké kvalitě.
6. Předposledním rizikem je nedodržení časových termínů v předpokládaném harmonogramu projektu s dopadem na náklady. Dodatečné náklady jsou generovány

v případě, že dojde k protáhnutí délky projektu, který je v současné době predikován na 9 kalendářních měsíců.

- Poslední skupinou rizik jsou tzv. rizika ostatní, která mohou mít dopad na celý projekt.

### 7.5.2 Ohodnocení rizik a jejich eliminace

Po identifikaci rizik přichází na řadu jejich ohodnocení s ohledem na pravděpodobnost výskytu rizika a míry dopadu na projekt. Tabulka vysvětlující způsob ohodnocení analýzy rizik je uvedena v příloze XY. K ohodnoceným rizikům byly následně doplněny eliminační faktory s cílem rizikům předcházet nebo alespoň částečně zmírnit jejich dopad.

Tab. 21. Identifikovaná rizika projektu (vlastní zpracování)

Identifikace rizika	Pravděpodobnost výskytu	Míra dopadu rizika na projekt	Celkové riziko	Eliminace rizika
Nedostatečné poznání firemních i nefiremních procesů	8	6	48	Pro rychlejší poznání firemních procesů je nutné zaškolení ze strany současného Projektového (Test) manažera
Odpor rigorózního týmu vůči změnám	5	5	25	Současný rigorózní tým musí mít pocit sounáležitosti s procesní úpravou a identifikací procesů testování. Nastolení pozitivního přístupu a poskytnutí neformálního pocitu účasti na projektu.
Nedostačující návrh a implementace projektu	7	10	70	Návrh a implementace musí být průběžně kontrolována s ohledem na ohodnocení projektu oproti druhé úrovni zralosti procesů testování v modelu TMMi

Vytížení kapacit účastníků projektu	5	6	35	Přislíbení odměn účastníkům projektů bude zčásti kompenzovat navýšené kapacitní vytížení, které se promítne do tzv. přesčasů
Špatně zvolené metriky procesů	8	7	42	Metriky budou průběžně vyhodnocovány, zda splňují svůj účel. Případně budou flexibilně nahrazeny či eliminovány.
Nedodržení časových termínů a generování dodatečných nákladů	6	6	36	Musí docházet k průběžné kontrole časových termínů a nákladů s případnou eliminací odchylek.
Ostatní rizika	3	5	15	Průběžná a pravidelná analýza spojená s kontrolou projektu

## 7.6 Vyhodnocení projektu

Vyhodnocení projektu diplomové práce proběhlo prostřednictvím ohodnocení procesu testování oproti modelu TMMi, který byl představen v podkapitole 3.3 Zvyšování kvality procesu testování prostřednictvím modelu TMMi.

Celkové hodnocení zralosti procesu testování proběhlo na pěti deklarováných procesních oblastech druhé úrovně modelu TMMi. Konečný výsledek byl prezentován prostřednictvím sumarizačního reportu nazvaného TMMi report.

Na vyhodnocení projektu se podílel autor diplomové práce ve spolupráci s projektovým manažerem společnosti XY řídicí agilní tým. Hodnotitelé procházeli každou komponentu zastoupenou specifickými či generickými praktikami. Každé z těchto komponent byl udělen po vyhodnocení stupeň splnění a výsledek byl zaznamenán do hodnotící karty procesní

oblasti. V případě, že se hodnotitelé rozcházel v hodnocení dané komponenty, byl uspořádán meeting s cílem dospět na hodnotící škále shody.

Formát šablony pro hodnotící kartu byl představen v diplomové práci Tomáše Doška s názvem *Modely zlepšování procesů testování softwaru a zajištění kvality*, odkud byla struktura hodnotící karty přebrána. (Došek, 2012) Do hodnotící karty byly umístěny jednotlivé komponenty TMMi pro druhou úroveň modelu v původním anglickém jazyce, aby nedošlo k záměně smyslu jednotlivých komponent.

Každá dílčí praktika mohla nabýt stupně splnění od 0 – 100% s ohodnocenou známkou F, L, P, N, NR nebo NA. Škála hodnocení byla využita dle dokumentu TAMAR věnující se hodnocením zralosti praktik modelu TMMi a je pro vyhodnocení dílčích komponent klíčové.

Tab. 22. Hodnotící škála pro model TMMi (TAMAR, 2012)

Hodnocení	Popis	Podmínky
F	Fully Achieved (Plně dosaženo)	85 - 100% Achieved
L	Largely Achieved (Z většiny dosaženo)	50 - 85% Achieved
P	Partially Achieved (Částečně dosaženo)	15 - 50% Achieved
N	Not Achieved (Nedosaženo)	0 - 15% Achieved
NR	Not Rated (Nehodnoceno)	Nemožnost vyhodnotit
NA	Not Applicable	Neaplikovatelné pro organizaci

Hodnocení známkou F předpokládá plné splnění hodnocené praktiky dle referenční dokumentace představenou v podkapitole 3.3 Zvyšování kvality procesu testování prostřednictvím modelu TMMi. Udělením hodnocení NR nevylučuje praktiku z hodnocení a je automaticky považována za nesplněnou. Oproti tomu hodnocení NA, neboli praktika pro společnost neaplikovatelnou je z hodnocení vyloučena. Pro takovou praktiku musí společnost uvést důvody vyloučení z hodnocení.

Hodnotitelé začali vyplněním stupně splnění na úrovni specifických a obecných praktik. Nejnižší hodnocení bylo následně přeneseno do pole pro specifický či obecný cíl. Nejnižší hodnocení na úrovni specifických a obecných cílů bylo následně přeneseno do pole pro hodnocení procesní oblasti. Pro splnění druhé úrovně zralosti procesů ohodnocenou oproti

modulu TMMi museli mít všechny procesní oblasti známku F (Fully Achieved). Vyhodnocení dílčích procesních oblastí je uvedeno v níže představených podkapitolách.

### 7.6.1 Vyhodnocení procesní oblasti PA1 – Test Policy and Strategy

Tab. 23. Tab. 24. Hodnocení 1. procesní oblasti – Politika a Strategie testování (vlastní zpracování)

Komponenty TMMi				Hodnocení		
PA1	Test Policy and Strategy			F		
	SG1	Establish a Test Policy			F	
		SP 1.1	Define test goals			F
		SP 1.2	Define test policy			F
		SP 1.3	Distribute the test policy to stakeholders			F
	SG2	Establish a Test Strategy			F	
		SP 2.1	Perform a generic product risk assessment			F
		SP 2.2	Define test strategy			F
		SP 2.3	Distribute the test strategy to stakeholders			F
	SG3	Establish Test Performance Indicators			F	
		SP 3.1	Define test performance indicators			F
		SP 3.2	Deploy test performance indicators			F
	GG2	Institutionalize a Managed Process			F	
		GP 2.1	Establish an organizational policy			F
		GP 2.2	Plan the process			F
		GP 2.3	Provide resources			F
		GP 2.4	Assign responsibilities			F



		GP 2.5	Train people			F
		GP 2.6	Manage configurations			F
		GP 2.7	Identify and involve relevant stakeholders			F
		GP 2.8	Monitor and control the process			F
		GP 2.9	Objectively Evaluate Adherence			F
		GP 2.10	Review status with higher lvl. management			F

### 7.6.2 Vyhodnocení procesní oblasti PA2 – Test Planning

Tab. 24. Tab. 25. Hodnocení 2. procesní oblasti – Plánování testů (vlastní zpracování)

Komponenty TMMi				Hodnocení		
PA2	Test Planning			F		
	SG1	Perform a Product Risk Assessment			F	
		SP 1.1	Define product risk categories and parameters			F
		SP 1.2	Identify product risks			F
		SP 1.3	Analyze product risks			F
	SG2	Establish a Test Approach			F	
		SP 2.1	Identify items and features to be tested			F
		SP 2.2	Define the test approach			F
		SP 2.3	Define entry criteria			F
		SP 2.4	Define exit criteria			F
		SP 2.5	Define suspension and resumption criteria			F
	SG3	Establish Test Estimates			F	

		SP 3.1	Establish a top-level work breakdown structure			F
		SP 3.2	Define test lifecycle			F
		SP 3.3	Determine estimates for test effort and cost			F
	SG4	Develop a Test Plan			F	
		SP 4.1	Establish the test schedule			F
		SP 4.2	Plan for test staffing			F
		SP 4.3	Plan stakeholder involvement			F
		SP 4.4	Identify test project risks			F
		SP 4.5	Establish the test plan			F
	SG5	Obtain Commitment to the Test Plan			F	
		SP 5.1	Review test plan			F
		SP 5.2	Reconcile work and resource levels			F
		SP 5.3	Obtain test plan commitments			F
	GG2	Institutionalize a Managed Process			F	
		GP 2.1	Establish an organizational policy			F
		GP 2.2	Plan the process			F
		GP 2.3	Provide resources			F
		GP 2.4	Assign responsibilities			F
		GP 2.5	Train people			F
		GP 2.6	Manage configurations			F
		GP 2.7	Identify and involve relevant stakeholders			F
		GP 2.8	Monitor and control the process			F

		GP 2.9	Objectively evaluate adherence			F
		GP 2.10	Review status with higher lvl. management			F

### 7.6.3 Vyhodnocení procesní oblasti PA3 – Test Monitoring and Control

Tab. 25. Hodnocení 3. procesní oblasti – Monitorování a řízení testů (vlastní zpracování)

Komponenty TMMi				Hodnocení		
PA3	Test Monitoring and Control			F		
	SG1	Monitor Test Progress against Plan			F	
		SP 1.1	Monitor test planning parameters			F
		SP 1.2	Monitor test environment resources provided and used			F
		SP 1.3	Monitor test commitments			F
		SP 1.4	Monitor test project risks			F
		SP 1.5	Monitor stakeholder involvement			F
		SP 1.6	Conduct test progress reviews			F
		SP 1.7	Conduct test progress milestone reviews			F
	SG2	Monitor Product Quality against Plan and Expectations			F	
		SP 2.1	Check against entry criteria			F
		SP 2.2	Monitor defects			F
		SP 2.3	Monitor product risks			F
		SP 2.4	Monitor exit criteria			F
		SP 2.5	Monitor suspension and resumption criteria			F
		SP 2.6	Conduct product quality reviews			F

		SP 2.7	Conduct product quality milestone reviews			F
	SG3	Manage Corrective Action to Closure			F	
		SP 3.1	Analyze issues			F
		SP 3.2	Take corrective action			F
		SP 3.3	Manage corrective action			F
	GG2	Institutionalize a Managed Process			F	
		GP 2.1	Establish an organizational policy			F
		GP 2.2	Plan the process			F
		GP 2.3	Provide resources			F
		GP 2.4	Assign responsibilities			F
		GP 2.5	Train people			F
		GP 2.6	Manage configurations			F
		GP 2.7	Identify and involve relevant stakeholders			F
		GP 2.8	Monitor and control the process			F
		GP 2.9	Objectively evaluate adherence			F
		GP 2.10	Review status with higher lvl. management			F

#### 7.6.4 Vyhodnocení procesní oblasti PA4 – Test Design and Execution

Tab. 26. Hodnocení 4. procesní oblasti – Návrh a provádění testů (vlastní zpracování)

Komponenty TMMi			Hodnocení		
PA4	Test Design and Execution		F		
	SG1	Perform Test Analysis and Design using Test Design Techniques		F	

		SP 1.1	Identify and prioritize test conditions			F
		SP 1.2	Identify and prioritize test cases			F
		SP 1.3	Identify necessary specific test data			F
		SP 1.4	Maintain horizontal traceability with requirements			F
	SG2	Perform Test Implementation			F	
		SP 2.1	Develop and prioritize test procedures			F
		SP 2.2	Create specific test data			F
		SP 2.3	Specify intake test procedure			F
		SP 2.4	Develop test execution schedule			F
	SG3	Perform Test Execution			F	
		SP 3.1	Perform intake test			F
		SP 3.2	Execute test cases			F
		SP 3.3	Report test incidents			F
		SP 3.4	Write test log			F
	SG4	Manage Test Incidents to Closure				F
		SP 4.1	Decide disposition of test incidents in configuration control board			F
		SP 4.2	Perform appropriate action to close the test incident			F
		SP 4.3	Track the status of test incidents			F
	GG2	Institutionalize a Managed Process			F	
		GP 2.1	Establish an organizational policy			F
		GP 2.2	Plan the process			F

		GP 2.3	Provide resources			F
		GP 2.4	Assign responsibilities			F
		GP 2.5	Train people			F
		GP 2.6	Manage configurations			F
		GP 2.7	Identify and involve relevant stakeholders			F
		GP 2.8	Monitor and control the process			F
		GP 2.9	Objectively evaluate adherence			F
		GP 2.10	Review status with higher lvl. management			F

### 7.6.5 Vyhodnocení procesní oblasti PA5 – Test Environment

Tab. 27. Hodnocení 5. procesní oblasti – Testovací prostředí (vlastní zpracování)

Komponenty TMMi				Hodnocení		
PA5	Test Environment			F		
	SG1	Develop Test Environment Requirements			F	
		SP 1.1	Elicit test environment needs			F
		SP 1.2	Develop the test environment requirements			F
		SP 1.3	Analyze the test environment requirements			F
	SG2	Perform Test Environment Implementation			F	
		SP 2.1	Implement the test environment			F
		SP 2.2	Create generic test data			F
		SP 2.3	Specify test environment intake test procedure			F
		SP 2.4	Perform test environment intake test			F

	SG3	Manage and Control Test Environments		F	
		SP 3.1	Perform systems management		F
		SP 3.2	Perform test data management		F
		SP 3.3	Coordinate the availability and usage of the test environments		F
		SP 3.4	Report and manage test environment incidents		F
	GG2	Institutionalize a Managed Process		F	
		GP 2.1	Establish an organizational policy		F
		GP 2.2	Plan the process		F
		GP 2.3	Provide resources		F
		GP 2.4	Assign responsibilities		F
		GP 2.5	Train people		F
		GP 2.6	Manage configurations		F
		GP 2.7	Identify and involve relevant stakeholders		F
		GP 2.8	Monitor and control the process		F
		GP 2.9	Objectively evaluate adherence		F
		GP 2.10	Review status with higher lvl. management		F

### 7.6.6 Sumarizační report TMMi

Níže uvedený sumarizační report TMMi shrnuje výsledky z dílčích specifických a generických cílů, kterých bylo dosaženo prostřednictvím ohodnocení stupně splnitelnosti na úrovni specifických a generických praktik každé procesní oblasti. Závěr hodnocení poukazuje na skutečnost, že společnost XY dosáhla na druhý stupeň zralosti procesů testování softwaru ohodnocenou prostřednictvím modelu TMMi. Hodnocení proběhlo neformálním způsobem. Neformální způsob hodnocení nevyžaduje akreditovaného hodnotitele a akreditovanou metodiku hodnocení, čímž byly ušetřeny finanční prostředky společnosti XY.

Tab. 28. Celkové hodnocení pro druhou úroveň zralosti procesů testování v modelu TMMi (vlastní zpracování)

Komponenty TMMi		Celkové hodnocení	
PA1		F	
	SG1	↑ ←	F
	SG2	←	F
	SG3	←	F
	GG2	←	F
PA2		F	
	SG1		F
	SG2		F
	SG3		F
	SG4		F
	SG5		F
	GG2		F
PA3		F	
	SG1		F
	SG2		F
	SG3		F
	GG2		F
PA4		F	
	SG1		F
	SG2		F
	SG3		F
	SG4		F
	GG2		F
PA5		F	
	SG1		F
	SG2		F
	SG3		F
	GG2		F

Cílem společnosti XY je možnost poskytovat softwarovou infrastrukturu svého finančního produktu Autoúvěr ve správě oddělení AP na nově i na americkém trhu. Aby byl tento krok možný, musela by společnost XY dosáhnout konkurenční výhody prostřednictvím zralosti (kvality) procesu testování softwaru ohodnocenou oproti druhé úrovni modelu TMMi. Tento cíl byl splněn a proces testování dosahuje po implementaci návrhu řešení pro

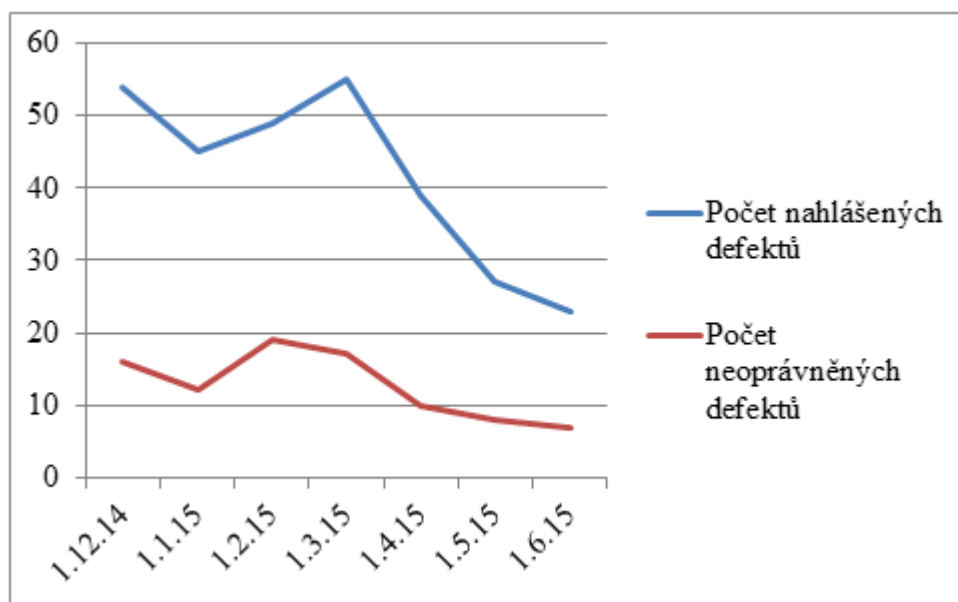


zlepšení řízení kvality procesu testování druhé úrovně zralosti procesu testování softwaru ohodnocenou oproti modelu TMMi.

Podružným cílem autora diplomové práce bylo v rámci návrhu řešení snížit počet hlášených defektů, které jsou zadávány na dodavatele softwaru. S počtem hlášených defektů se pojí i počet neoprávněných reklamací neboli počet neoprávněných hlášení defektů. Z níže uvedených tabulek je vidět klesající tendence v čase, kdy byl návrh implementován. Hodnoty uvedené v měsíci červnu 2015 jsou predikovány na základě 2 předchozích měsíců.

Tab. 29. Přehled hlášených defektů (vlastní zpracování)

Počet nahlášených defektů		Neoprávněný defekt	
Měsíc	Defekt report	Měsíc	Neoprávněný defekt
12.14	54	12.14	16
1.15	45	1.15	12
2.15	49	2.15	19
3.15	55	3.15	17
4.15	39	4.15	10
5.15	27	5.15	8
6.15	23	6.15	7



Obr. 29. Průběh hlášených defektů v čase (vlastní zpracování)

## ZÁVĚR

Diplomová práce byla zaměřena na proces testování softwaru s cílem navrhnout a implementovat návrh na zlepšení řízení kvality procesu testování softwaru ve společnosti XY tak, aby proces testování softwaru dosáhl na druhou úroveň zralosti neboli kvality ohodnocenou oproti modelu TMMi.

V úvodu se teoretická část zabývala vydefinováním základních pojmů, které se vážou k tématu diplomové práce. Byla představena kvalita, pojem software a základní charakteristické znaky procesního řízení. Teoretická část se dále věnovala samotnému procesu testování softwaru a jeho základním atributům. Po představení procesu testování softwaru byla zmíněna základní typologie testů, kterou lze kategorizovat dle různých hledisek. Posléze se práce zaměřila na vymezení pojmu defekt, jenž je nedílnou součástí celého procesu testování softwaru.

Po uvedení do tématu se práce zaměřila na samotné řízení kvality softwaru, přičemž bylo poukázáno na synergii mezi testováním, řízením a zajišťováním kvality procesu testování. Následně byla uvedena rigorózní metodika vývoje softwaru uplatňující se ve společnosti XY, jež ovlivňuje celý proces testování softwaru. Dále se teoretická část zaměřila na metody zvyšující kvalitu procesu testování prostřednictvím dvou známých přístupů. Jednalo se o agilní přístup ve vztahu k testování softwaru a modelů kvality založených na zralosti procesů. Závěr teoretické části se zabýval modelováním procesů s představením nástroje BPMN.

Úvod analytické části projektu se zabýval představením společnosti s její organizační hierarchií, ve které byl projekt realizován. Následovalo uvedení do produktového portfolia se zaměřením na poskytování produktu Autoúvěr s jeho softwarovou infrastrukturou, která je předmětem inovačních softwarových projektů a tedy i procesu testování. Stěžejní částí analytické části projektu byla procesní analýza tehdejšího stavu procesu testování softwaru produktu Autoúvěr. Ta se zakládala na slovním zdokumentování a vytvoření procesních karet s následným modelováním podprocesů vyskytujících se v procesu testování. Závěrem byly shrnuty základní nedostatky vycházející z procesní analýzy procesu testování.

Projektová část diplomové práce byla zaměřena na návrh řešení vedoucí ke zlepšení řízení kvality procesu testování softwaru s následnou realizací návrhu do procesní struktury. Autor diplomové práce při svém návrhu vycházel ze zkušeností a znalostí, které načerpal při účasti na jiných softwarových projektech v roli testera a test analytika. V průběhu analytické

ky projektových prací bylo neustále bráno na zřetel, že je zapotřebí dosáhnout druhé úrovně zralosti procesů testování ohodnocenou oproti modelu TMMi. Pro účely projektu byl rovněž vytvořen plán projektu zahrnující harmonogram prací, analýzu zdrojů, nákladovou analýzu a analýzu rizik. V závěru celého projektu došlo k vyhodnocení stanoveného cíle s využitím představeného modelu TMMi v teoretické části diplomové práce.

Výsledkem projektu byl proces testování softwaru, který dosáhl na druhou úroveň zralosti procesů ohodnocenou oproti modelu TMMi. Tímto krokem naplnil projekt stanovený cíl a byl prohlášen za úspěšný. Zisk konkurenční výhody oproti ostatním členům finanční skupiny umožňuje vstoupit se softwarovou infrastrukturou produktu Autoúvěr na americký trh, kde začala finanční skupina vyvíjet své aktivity. Perspektivou projektu je již zmíněný transfer znalostí a know-how získaných z řad velkých projektů v České republice a na Slovensku.

Námětem pro pokračování s ohledem na řízení kvality procesu testování softwaru je získání nejen vyšších stupňů zralosti procesů testování v modelu TMMi, ale také získání akreditované metody hodnocení či využití externích konzultačních společností, které již akreditovanou metodu hodnocení vlastní. Společnost XY přislíbila, že bude uvedenou možnost zvažovat a zahrne projekt do svého rozvojového programu.

**SEZNAM POUŽITÉ LITERATURY**

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

ČSN EN ISO 9000:2005. *Systémový managementu kvality - Základní principy a slovník*. Praha: Český normalizační institut, 2006.

BRÍŠ, Petr. *Management kvality*. Vyd. 2., uprav. Zlín: Univerzita Tomáše Bati ve Zlíně, 2010, 208 s. ISBN 978-80-7318-912-9.

NENADÁL, Jaroslav, Darja NOSKIEVIČOVÁ, Růžena PETŘÍKOVÁ, Jiří PLURA a Josef TOŠENOVSKÝ. *Moderní systémy řízení jakosti: quality management*. 2. dopl. vyd. Praha: Management Press, 2007, 282 s. ISBN 80-726-1071-6.

GALIN, Daniel a Anna HAVLÍČKOVÁ. *Software quality assurance: průvodce testováním*. 1. vyd. New York: Pearson Education Limited, 2004, xxvi, 590 p. ISBN 02-017-0945-7.

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

HROMKOVÁ, Ludmila a Zuzana TUČKOVÁ. *Reengineering podnikových procesů*. 1. vyd. Zlín: Univerzita Tomáše Bati ve Zlíně, 2008, 139 s. ISBN 978-80-7318-759-0.

ŠMÍDA, Filip. *Zavádění a rozvoj procesního řízení ve firmě*. 1. vyd. Praha: Grada, 2007, 293 s. ISBN 978-80-247-1679-4.

HAMMER, Michael. *Agenda 21: co musí každý podnik udělat pro úspěch v 21. století*. Vyd. 1. Praha: Management Press, 2002, 258 s. ISBN 80-726-1074-0.

CARR, David K a Henry J JOHANSSON. *Best practices in reengineering: what works and what doesn't in the reengineering process*. Vyd. 1. New York: McGraw-Hill, c1995, xii, 235 p. ISBN 00-701-1224-X.

ŠIMONOVÁ, Stanislava. *Modelování procesů a dat pro zvyšování kvality*. Pardubice: Univerzita Pardubice, 2009. ISBN 978-80-7395-205-1.

GRASSEOVÁ, Monika, Radek DUBEC a Roman HORÁK. *Procesní řízení ve veřejném sektoru: teoretická východiska a praktické příklady*. Vyd. 1. Brno: Computer Press, 2008, v, 266 s. ISBN 978-80-251-1987-7.

GÁLA, Libor. *Podniková informatika: počítačové aplikace v podnikové a mezipodnikové praxi, technologie informačních systémů, řízení a rozvoj podnikové informatiky*. 1. vyd. Praha: Grada, 2006, 482 s. ISBN 80-247-1278-4.

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

ZELINKA, Bořek. UNICORN SYSTEMS A.S. *Testování softwaru*. 2013. Dostupné z: [http://d3s.mff.cuni.cz/teaching/commercial\\_workshops/previous/1213/zelinka-zajisteni\\_kvality\\_softwarovych\\_produkту.pdf](http://d3s.mff.cuni.cz/teaching/commercial_workshops/previous/1213/zelinka-zajisteni_kvality_softwarovych_produkту.pdf)

ISO/IEC 12207:2008. *Systems and software engineering - Software life cycle processes*. Switzerland: ISO, 2008.

DOUCEK, Petr. *Řízení projektů informačních systémů*. 1. vyd. Praha: Professional publishing, 2004. 168 s. ISBN: 80-86419-71-1.

SOMMERVILLE, Ian. *Software engineering*. 9th ed. Boston: Pearson, c2011, xv, 773 p. ISBN 978-013-7053-469.

TMMI FOUNDATION. *TMMi Assessment Method Application Requirements: TAMAR*. 2014, 35 s. Dostupné z: <http://www.tmmi.org/pdf/TMMi.TAMAR.pdf>

TMMI FOUNDATION. *Test Maturity Model integration: TMMi*. 2012, 219 s. Dostupné z: <http://www.tmmi.org/pdf/TMMi.Framework.pdf>

ISO/IEC 15504-2:2003. *Information technology - Process assessment - Part 2: Performing an assessment*. Switzerland: ISO, 2003.

DOŠEK, Tomáš. *Modely zlepšování procesů testování softwaru a zajištění kvality*. Praha, 2012. Dostupné z: [http://www.vse.cz/vskp/show\\_file.php?soubor\\_id=1082632](http://www.vse.cz/vskp/show_file.php?soubor_id=1082632). Diplomová práce. Vysoká škola ekonomická v Praze. Vedoucí práce Mgr. Anna Borovcová.

ŘEPA, Václav. *Podnikové procesy: procesní řízení a modelování*. 1. vyd. Praha: Grada, 2006, 265 s. ISBN 80-247-1281-4.

OBJECT MANAGEMENT GROUP. *Business Process Model And Notation (BPMN) Version 2.0*. 2011, 508 s. Dostupné z: <http://www.omg.org/spec/BPMN/2.0/PDF>

Interní materiály společnosti XY, 2015

ISO/IEC/IEEE 29119-3. *Software and systems engineering - Software testing: Part 3: Test documentation*. USA: IEEE, 2013, 138 s. Dostupné z: <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6588538>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

BPMN	Business Process Model and Notation.
CEO	Chief Executive Officer.
CMMI	Capability Maturity Model Integration.
EkIS	Ekonomický Informační Systém.
IEC	International Electrotechnical Commission.
IEEE	Institute of Electrical and Electronics Engineers.
ISO	International Organization for Standardization.
NASA	National Aeronautics and Space Administration.
MD	Man-Day.
SEI	Software Engineering Institute.
SQA	Software Quality Assurance.
SQC	Software Quality Control.
TMMi	Test Maturity Model integration.
TQM	Total Quality Management.
UAT	User Acceptance Testing.

**SEZNAM OBRÁZKŮ**

<i>Obr. 1. Logická výstavba praktické části diplomové práce (vlastní zpracování)</i> .....	11
<i>Obr. 2. Náklady na opravu chyb v časovém horizontu (Patton, 2002, s. 16)</i> .....	20
<i>Obr. 3. Členění testů dle časového hlediska (vlastní zpracování)</i> .....	23
<i>Obr. 4. Vztah mezi řízením a zajištěním kvality softwaru (vlastní zpracování)</i> .....	27
<i>Obr. 5. Vodopádový mode vývoje softwaru (Sommerville, 2011, s. 30)</i> .....	29
<i>Obr. 6. Stupňovitá reprezentace modelu TMMi (TMMi Foundation, 2012)</i> .....	31
<i>Obr. 7. Vztah cílů a praktik vyskytujících se v modelu TMMi (TMMi Foundation, 2012)</i> .....	33
<i>Obr. 8. Organizační struktura divize projektů a IT ve společnosti XY (vlastní zpracování)</i> .....	40
<i>Obr. 9. Průnik realizovaných softwarových projektů během roku (vlastní zpracování)</i> ....	44
<i>Obr. 10. Vzor Karty procesu (vlastní zpracování)</i> .....	45
<i>Obr. 11. Karta procesu Plánování testů (vlastní zpracování)</i> .....	48
<i>Obr. 12. Proces Plánování testů (vlastní zpracování)</i> .....	49
<i>Obr. 13. Vzor Testovacího případu (vlastní zpracování)</i> .....	50
<i>Obr. 14. Karta procesu Návrh testů (vlastní zpracování)</i> .....	51
<i>Obr. 15. Proces Návrh testů (vlastní zpracování)</i> .....	52
<i>Obr. 16. Karta procesu Implementace testů (vlastní zpracování)</i> .....	54
<i>Obr. 17. Proces Implementace testů (vlastní zpracování)</i> .....	55
<i>Obr. 18. Vzor artefaktu – Test obecné funkcionality (vlastní zpracování)</i> .....	56
<i>Obr. 19. Karta procesu Provádění testů (vlastní zpracování)</i> .....	58
<i>Obr. 20. Proces Provádění testů (vlastní zpracování)</i> .....	59
<i>Obr. 21. Doplněná Karta procesu Plánování testů (vlastní zpracování)</i> .....	71
<i>Obr. 22. Doplněná Karta procesu Návrh testů (vlastní zpracování)</i> .....	72
<i>Obr. 23. Doplněná karta Procesu Implementace testů (vlastní zpracování)</i> .....	73
<i>Obr. 24. Doplněná Karta procesu Provádění testů (vlastní zpracování)</i> .....	74
<i>Obr. 25. Karta procesu Řízení testů (vlastní zpracování)</i> .....	76
<i>Obr. 26. Nová Karta procesu Vyhodnocení testů (vlastní zpracování)</i> .....	79
<i>Obr. 27. Časový harmonogram projektu (vlastní zpracování)</i> .....	80
<i>Obr. 28. Organizační diagram projektu (vlastní zpracování)</i> .....	81
<i>Obr. 29. Průběh hlášených defektů v čase (vlastní zpracování)</i> .....	97

**SEZNAM TABULEK**

<i>Tab. 1. Zjištěné nedostatky v procesu testování softwaru ve společnosti XY (vlastní zpracování)</i> .....	61
<i>Tab. 2. Produktivita tvorby testovacích scénářů (vlastní zpracování)</i> .....	63
<i>Tab. 3. Průběh provádění testů s výsledkem je v pořádku (vlastní zpracování)</i> .....	63
<i>Tab. 4. Průběh provádění testů s výsledkem nelze provést (vlastní zpracování)</i> .....	63
<i>Tab. 5. Průběh provádění testů s výsledkem selhání (vlastní zpracování)</i> .....	64
<i>Tab. 6. Míra zastoupení nahlášených defektů (vlastní zpracování)</i> .....	64
<i>Tab. 7. Míra zamítnutých defektů (vlastní zpracování)</i> .....	65
<i>Tab. 8. Míra chybovosti oprav (vlastní zpracování)</i> .....	65
<i>Tab. 9. Odchylka odhadu pracnosti (vlastní zpracování)</i> .....	66
<i>Tab. 10. Odchylka časových termínů (vlastní zpracování)</i> .....	66
<i>Tab. 11. Míra vynaložených nákladů na testování (vlastní zpracování)</i> .....	67
<i>Tab. 12. Hodnota nalezeného defektu (vlastní zpracování)</i> .....	67
<i>Tab. 13. Stav čerpání rozpočtu (vlastní zpracování)</i> .....	68
<i>Tab. 14. Úroveň kvality procesu testování (vlastní zpracování)</i> .....	68
<i>Tab. 15. Míra pokrytí testů (vlastní zpracování)</i> .....	69
<i>Tab. 16. Výše opravenosti defektů (vlastní zpracování)</i> .....	69
<i>Tab. 17. Mzdové náklady projektu (vlastní zpracování)</i> .....	83
<i>Tab. 18. Ostatní náklady projektu (vlastní zpracování)</i> .....	83
<i>Tab. 19. Rezerva ve výši 10% z celkových plánovaných nákladů (vlastní zpracování)</i> .....	83
<i>Tab. 20. Tab. 21. Celkové náklady na projekt (vlastní zpracování)</i> .....	83
<i>Tab. 21. Identifikovaná rizika projektu (vlastní zpracování)</i> .....	85
<i>Tab. 22. Hodnotící škála pro model TMMi (TAMAR, 2012)</i> .....	87
<i>Tab. 23. Tab. 24. Hodnocení 1. procesní oblasti – Politika a Strategie testování (vlastní zpracování)</i> .....	88
<i>Tab. 24. Tab. 25. Hodnocení 2. procesní oblasti – Plánování testů (vlastní zpracování)</i> .....	89
<i>Tab. 25. Hodnocení 3. procesní oblasti – Monitorování a řízení testů (vlastní zpracování)</i> .....	91
<i>Tab. 26. Hodnocení 4. procesní oblasti – Návrh a provádění testů (vlastní zpracování)</i> .....	92
<i>Tab. 27. Hodnocení 5. procesní oblasti – Testovací prostředí (vlastní zpracování)</i> .....	94



---

<i>Tab. 28. Celkové hodnocení pro druhou úroveň zralosti procesů testování v modelu TMMi (vlastní zpracování) .....</i>	<i>96</i>
<i>Tab. 29. Přehled hlášených defektů (vlastní zpracování) .....</i>	<i>97</i>

**SEZNAM PŘÍLOH**

P I BPMN 2.0 – notace 1

P II BPMN 2.0 – notace 2

P III BPMN 2.0 – notace 3

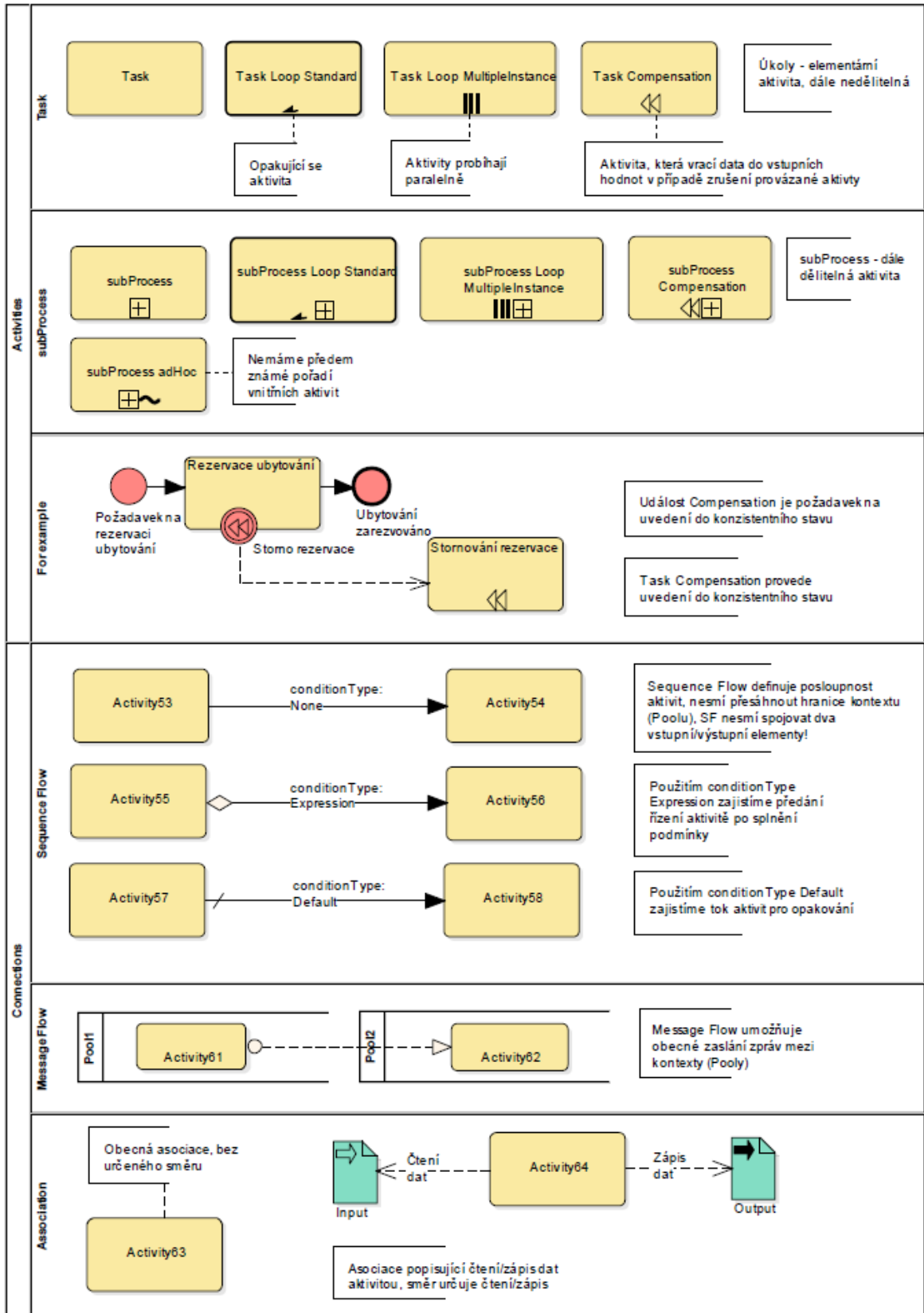
P IV BPMN 2.0 – workflow patterns 1

P V BPMN 2.0 – workflow patterns 2

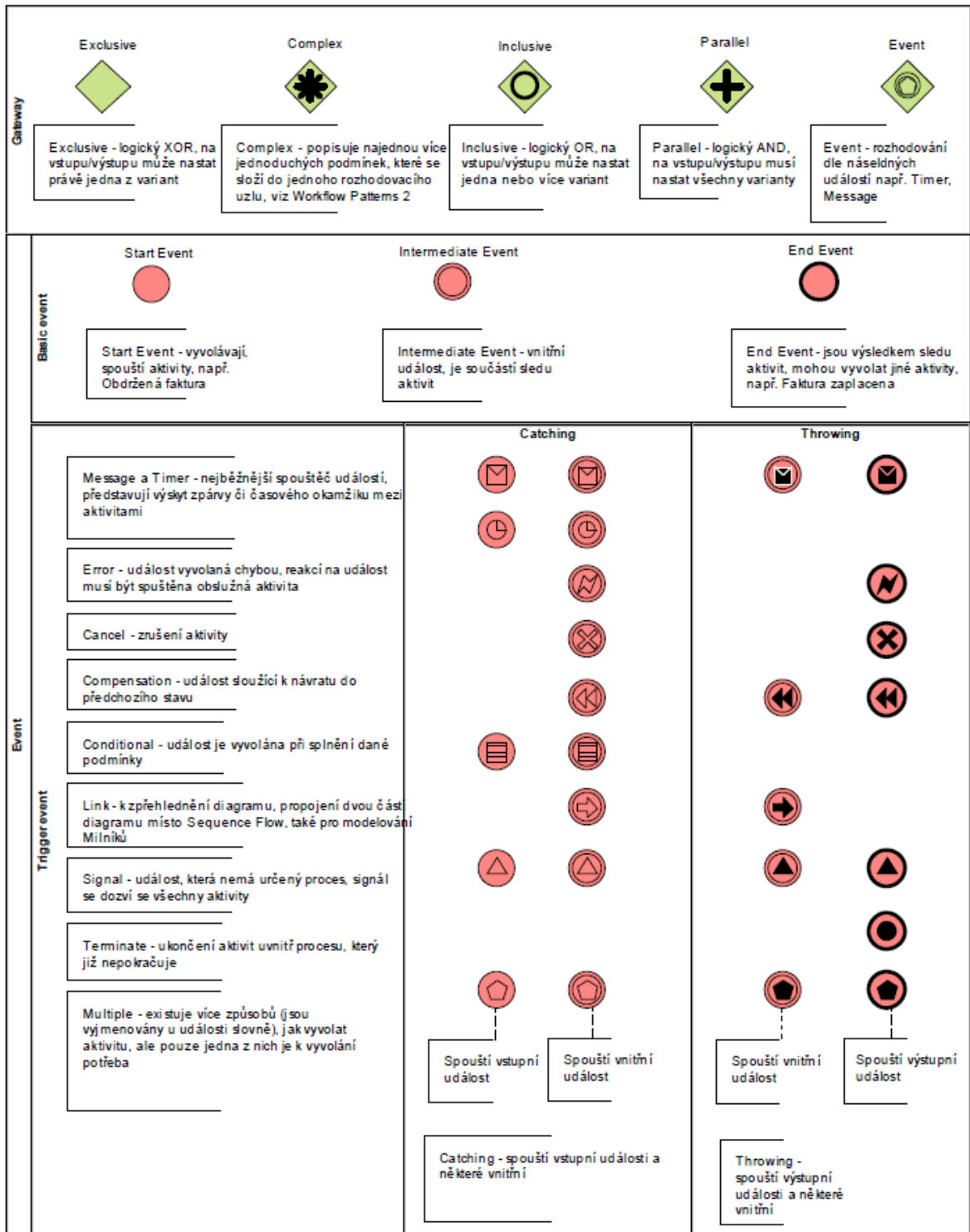
P VI BPMN 2.0 – workflow patterns 3

P VII Riziková analýza

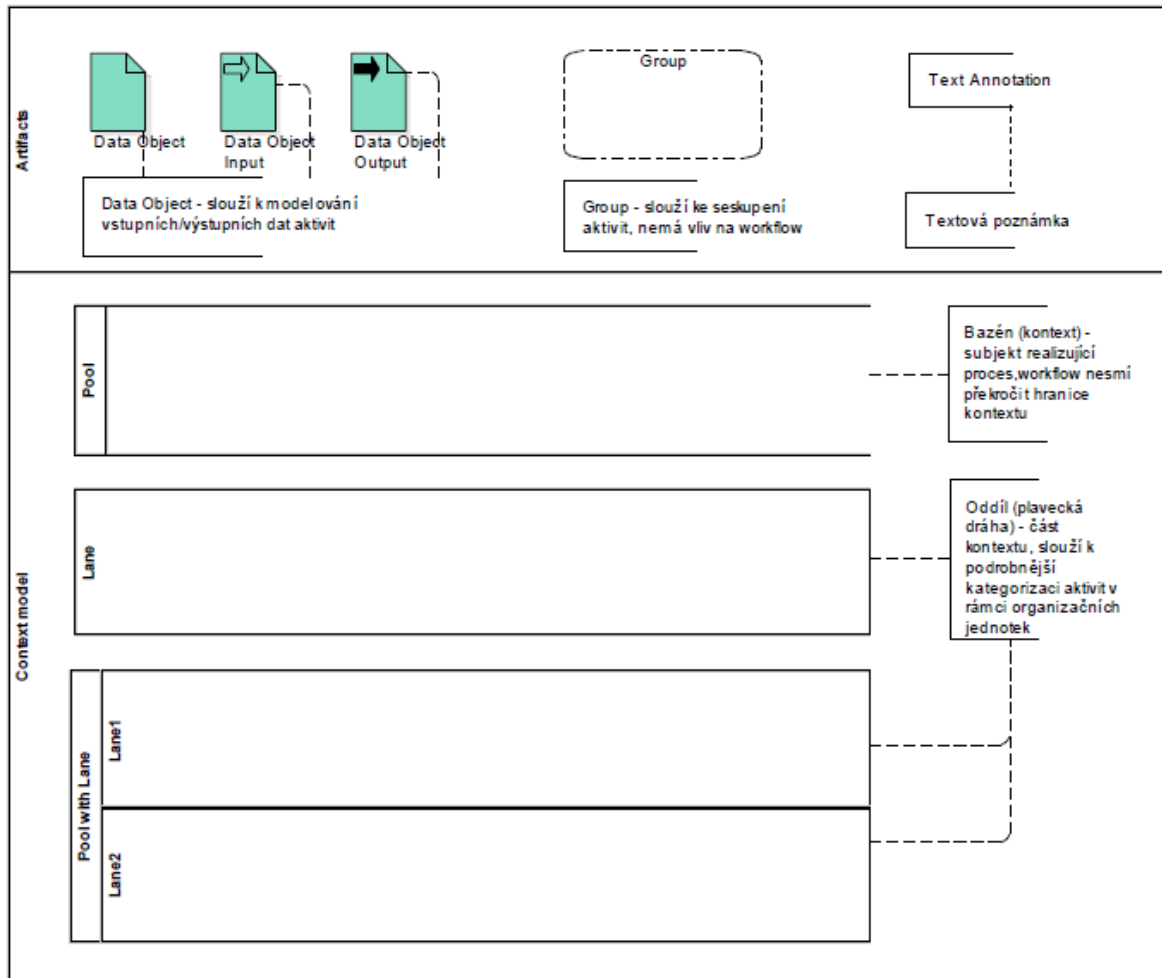
# PŘÍLOHA P I: BPMN 2.0 – NOTACE 1



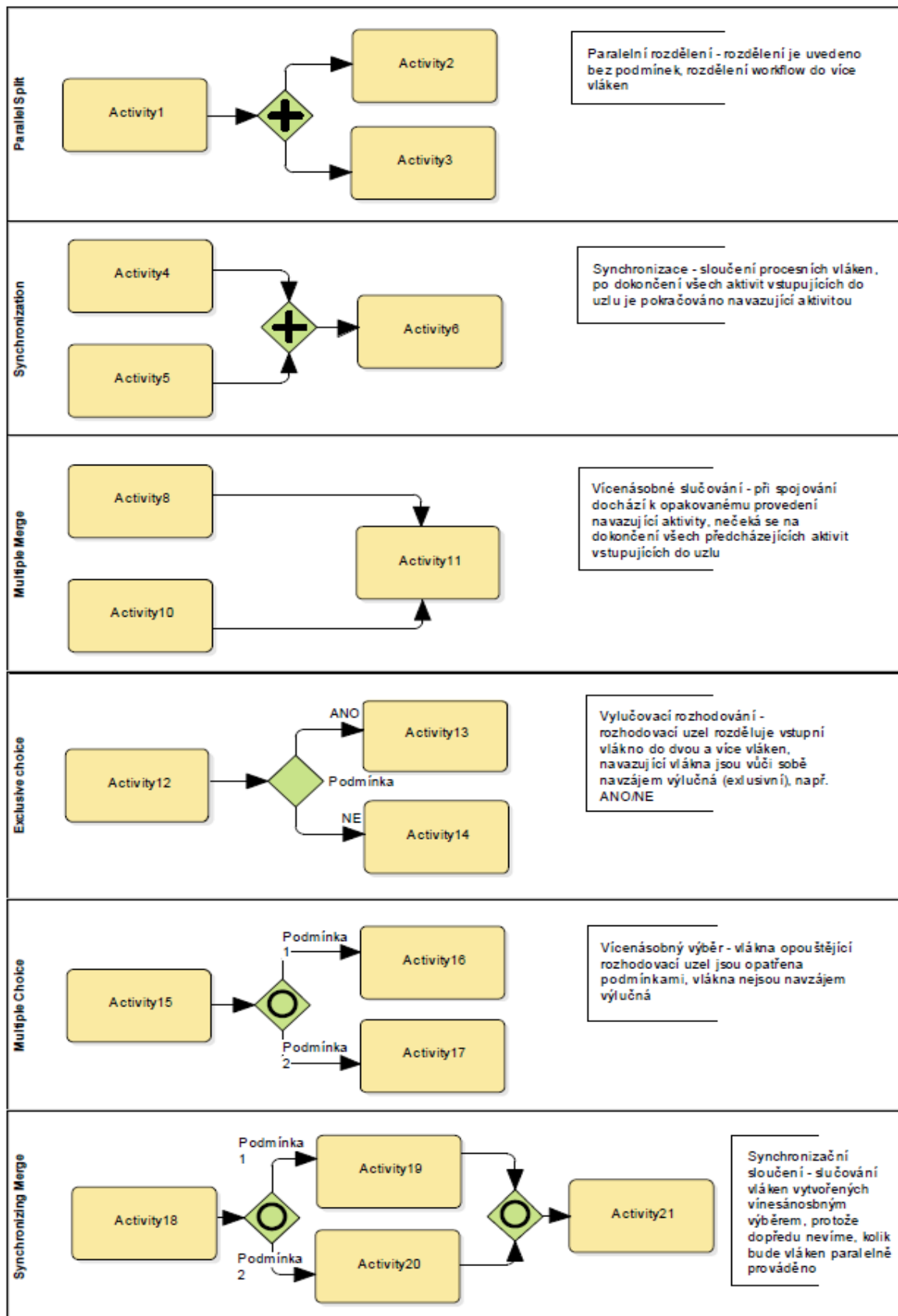
## PŘÍLOHA P II: BPMN 2.0 – NOTACE 2



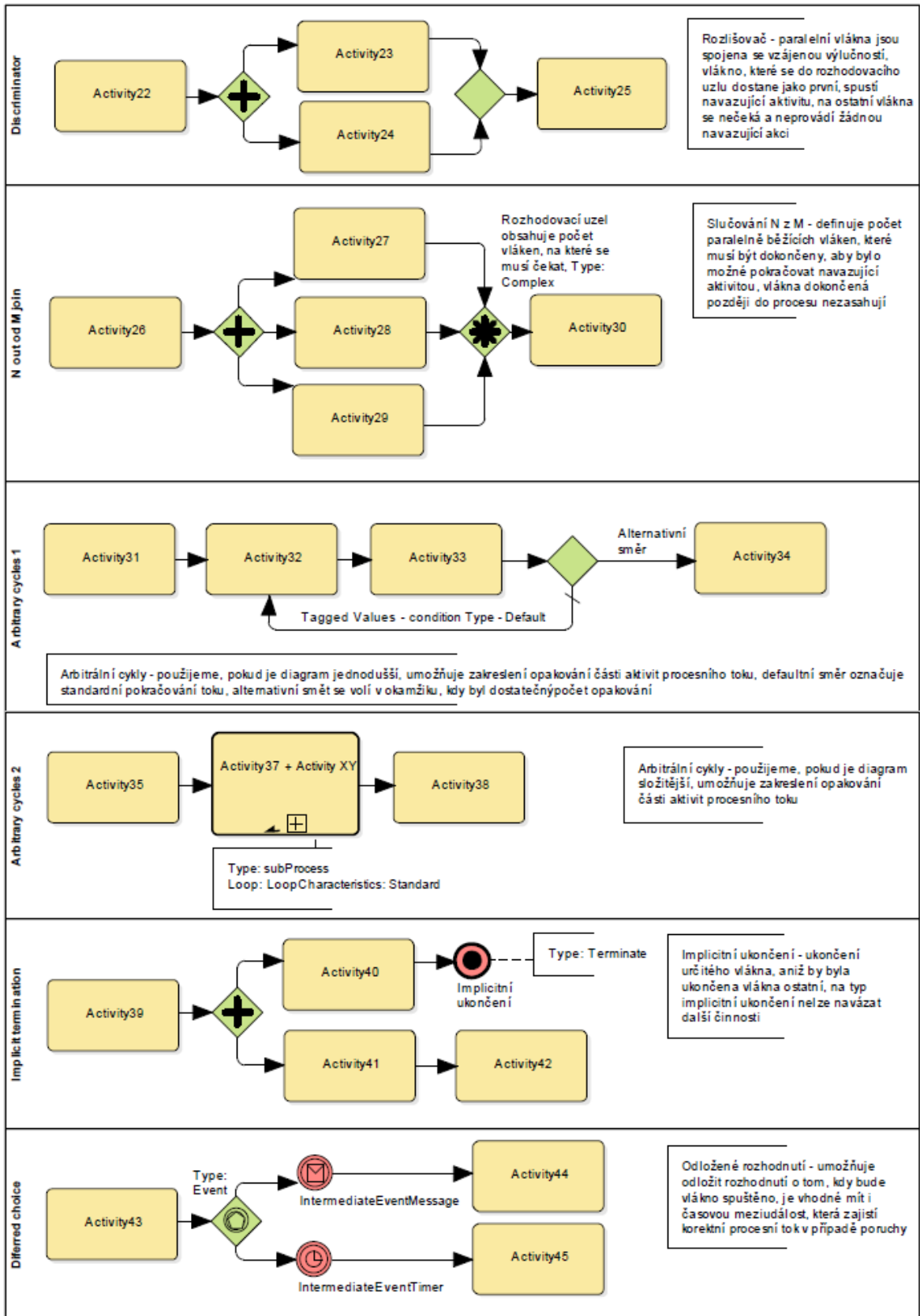
## PŘÍLOHA P III: BPMN 2.0 – NOTACE 3



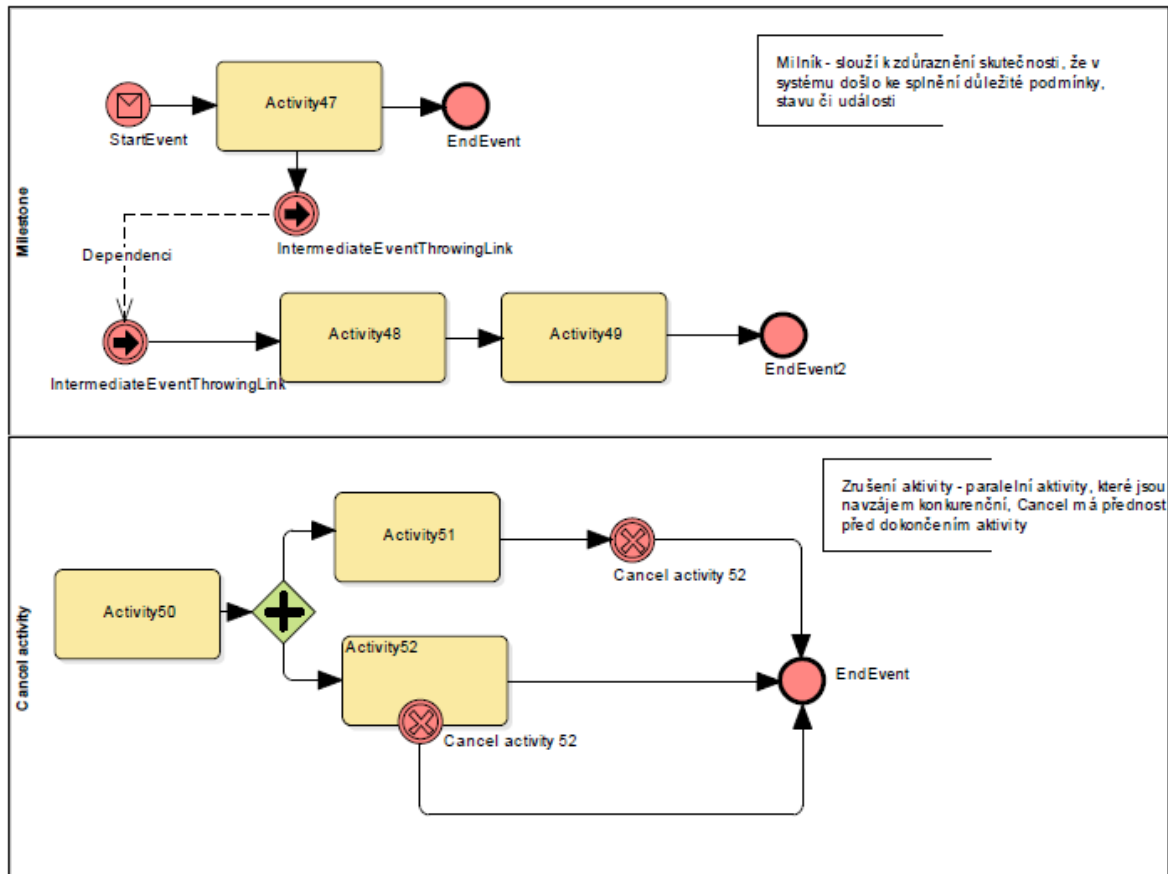
## PŘÍLOHA P IV: BPMN 2.0 – WORKFLOW PATTERNS 1



## PŘÍLOHA P V: BPMN 2.0 – WORKFLOW PATTERNS 2



## PŘÍLOHA P VI: BPMN 2.0 – WORKFLOW PATTERNS 3





## PŘÍLOHA P VII: RIZIKOVÁ ANALÝZA

Pravděpodobnost vzniku	Bodová hodnota
Nejnižší	1 - 2
Nízké	3 - 4
Pravděpodobné	5 - 6
Významnější	7 - 8
Nejvýznamnější	9 - 10

*Pravděpodobnost vzniku rizika (vlastní zpracování)*

Míra dopadu rizika	Bodová hodnota
Nejnižší riziko dopadu na projekt	1 - 2
Nízké riziko dopadu na projekt	3 - 4
Pravděpodobné riziko dopadu na projekt	5 - 6
Významnější riziko dopadu na projekt	7 - 8
Nejvýznamnější riziko dopadu na projekt	9 - 10

*Míra dopadu rizika (vlastní zpracování)*

Celkové riziko	Bodová hodnota
Nejnižší riziko	0 - 20
Nízké riziko	21 - 40
Pravděpodobné riziko	41 - 60
Významnější riziko	61 - 80
Nejvýznamnější riziko	81 - 100

*Celkové riziko (vlastní zpracování)*

Výpočet celkového rizika vychází ze součinu pravděpodobnosti vzniku rizika a míry dopadu rizika.