

Univerzita Pardubice
Fakulta ekonomicko-správní
Ústav systémového inženýrství a informatiky

Proces správy a řízení požadavků v rámci vývoje systémů

Bc. Terézia Špicová

Diplomová práce
2017

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Terézia Špicová**
Osobní číslo: **E15673**
Studijní program: **N6209 Systémové inženýrství a informatika**
Studijní obor: **Regionální a informační management**
Název tématu: **Proces správy a řízení požadavků v rámci vývoje systémů**
Zadávající katedra: **Ústav systémového inženýrství a informatiky**

Z á s a d y p r o v y p r a c o v á n í :

Cíl práce: navrhnout proces řízení požadavků s využitím dostupných standardů, praktik a metodik v prostředí vybrané firmy.

Osnova:

- Vymezení základních pojmů.
- Problematika procesu řízení požadavků.
- Definování postupů v rámci tvorby uživatelských požadavků, systémových specifikací a jejich rozpad na nižší úrovně.
- Popis potenciálních rolí v procesu, potřebné vstupy, zdroje informací a výstupy.
- Popis možných způsobů či technik pro zajištění sledování (traceability) mezi požadavky.

Rozsah grafických prací:


Rozsah pracovní zprávy: **cca 50 stran**

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. WIEGERS, K. Požadavky na software. 448 S. Brno: Computer Press, a.s., 2008. ISBN 978-80-251-1877-1
2. SOMMERVILLE, I. Softwarové inženýrství. 680 S. Brno: Computer Press, a.s., 2013. ISBN 978-80-251-3826-7
3. BRUCKNER, T. Tvorba informačních systémů: principy, metodiky, architektury. 360 s. Praha: Grada, 2012. ISBN 978-80-247-4153-6
4. WIEGERS, K., BEATTY, J. Software Requirements. 672 s. Microsoft Press, 2013. ISBN 978-0-7356-7966-5

Vedoucí diplomové práce:

Ing. Hana Kopáčková, Ph.D. 

Ústav systémového inženýrství a informatiky

Datum zadání diplomové práce: **4. září 2016**

Termín odevzdání diplomové práce: **28. dubna 2017**


doc. Ing. Romana Provažníková, Ph.D.

děkanka

L.S.


doc. Ing. Pavel Petr, Ph.D.

vedoucí ústavu

V Pardubicích dne 4. září 2016

PROHLÁŠENÍ

Prohlašuji, že jsem tuto práci vypracovala samostatně. Veškeré literární prameny a informace, které jsem v práci využila, jsou uvedeny v seznamu použité literatury.

Byla jsem seznámena s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako Školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 28. 4. 2017

Bc. Terézia Špicová

PODĚKOVÁNÍ:

Tímto bych ráda poděkovala své vedoucí práce, paní Ing. Haně Kopáčkové, Ph.D., za její odbornou pomoc a cenné rady, které mi pomohly při zpracování diplomové práce. Stejně tak děkuji firmě, která mi poskytla možnost tuto práci zpracovávat, za jejich věnovaný čas, ochotu a připomínky při konzultacích.

Velké díky také patří mé rodině a blízkým, kteří mě během celého studia velice podporovali.

ANOTACE

Diplomová práce je zaměřena na pochopení problematiky týkající se řízení požadavků v rámci vývoje systémů v jednotlivých fázích projektu. V první části jsou vysvětleny pojmy týkající se projektového managementu a systémového inženýrství. Následuje uvedení do problematiky řízení, vývoje a správy požadavků. Poté je zde zmapován proces řízení požadavků ve fázích projektu. V závěrečné fázi je vytvořen návrh modelu pro sledovatelnost požadavků pomocí Entity Relationship Diagram (ERD).

KLÍČOVÁ SLOVA

Řízení požadavků, správa požadavků, fáze projektu, proces řízení požadavků, sledovatelnost

TITLE

Process of Requirement Engineering and Management within the system development

ANNOTATION

Thesis is focused on understanding the issues related to the Requirements Engineering and Management in the system development in various phases of the project. The first part explains the concepts related to Project Management and Systems Engineering. Another part is introduction to the problems of Requirements Engineering, Requirements Development and Requirements Management. Then there is mapped process of Requirements Engineering in the phases of project. In the final of the thesis is formed model of traceability requirement using the Entity Relationship Diagram (ERD).

KEYWORDS

Requirement Engineering, Requirement Management, phases of a project, Process of Requirements Engineering, Traceability

OBSAH

ÚVOD.....	9
1. ÚVOD DO PROJEKTOVÉHO MANAGEMENTU.....	10
1.1. INICIALIZACE.....	12
1.2. ZAHÁJENÍ.....	13
1.3. PLÁNOVÁNÍ.....	14
1.4. REALIZACE	15
1.5. MONITOROVÁNÍ A KONTROLA	15
1.6. UKONČENÍ.....	16
2. ÚVOD DO SYSTÉMOVÉHO INŽENÝRSTVÍ	17
2.1. SYSTÉMOVÉ INŽENÝRSTVÍ	17
2.2. POŽADAVEK	17
2.2.1. Typy požadavků	18
2.2.2. Vlastnosti požadavků.....	20
2.3. ŘÍZENÍ POŽADAVKŮ.....	21
2.4. VÝVOJ POŽADAVKŮ	22
2.5. SPRÁVA POŽADAVKŮ	28
2.5.1. Proces správy změn požadavků.....	29
2.5.2. Sledovatelnost požadavků	32
2.5.3. Nástroje pro správu požadavků	33
2.5.4. Dokumentace požadavků.....	34
2.6. ROLE PŘI ŘÍZENÍ POŽADAVKŮ V RÁMCI VÝVOJE SYSTÉMU	35
3. PROCES ŘÍZENÍ POŽADAVKŮ VE VYBRANÉ FIRMĚ	37
3.1. PŘEDSTAVENÍ FIRMY	37
3.1.1. Identifikace současného stavu	37
3.1.2. Návrh cílového stavu	38
3.2. ROLE PŘI ŘÍZENÍ A SPRÁVĚ POŽADAVKŮ VE VYBRANÉ FIRMĚ	39
3.3. PROCES ŘÍZENÍ POŽADAVKŮ VE FÁZÍCH PROJEKTU.....	40
3.3.1. Subproces Sběr požadavků.....	44
3.3.2. Subproces Tvorba specifikace požadavků.....	44
3.3.3. Subproces Plánování verifikace požadavků	45
3.3.4. Subproces Identifikace a analýza změn.....	46
3.3.5. Subproces Implementace změny	46
3.3.6. Subproces Verifikace a Validace.....	47
3.4. SLEDOVATELNOST POŽADAVKŮ VE VYBRANÉ FIRMĚ.....	47
3.4.1. Matice sledovatelnosti	47
3.5. NÁVRH SLEDOVATELNOSTI POŽADAVKŮ	51
3.5.1. Návrh modelu databáze	51
3.5.2. Příklady využití databáze pro správu požadavků	53
3.6. ZÁVĚREČNÉ SHRNTUÍ.....	57
ZÁVĚR.....	58
POUŽITÁ LITERATURA	59
SEZNAM PŘÍLOH.....	63

SEZNAM TABULEK

Tabulka 1: Nefunkční požadavky příklad	19
Tabulka 2: Výhody a nevýhody V-modelu	27
Tabulka 3: Základní rozdíl mezi vývojem a správou požadavků	29
Tabulka 4: Uživatelé dokumentu požadavků	35
Tabulka 5: Subprocesy řízení požadavků ve fázích projektu	41
Tabulka 6: Karta procesu pro interní projekt	43
Tabulka 7: Příklad matice sledovatelnosti mezi požadavky	48
Tabulka 8: Příklad matice sledovatelnosti mezi požadavky a testy	50
Tabulka 9: Příklad matice sledovatelnosti mezi požadavky a návrhem	50
Tabulka 10: Příklad 1	54
Tabulka 11: Příklad 2	54
Tabulka 12: Příklad 3	55
Tabulka 13: Příklad 4	55
Tabulka 14: Příklad 5	56
Tabulka 15: Příklad 6	56

SEZNAM OBRÁZKŮ

Obrázek 1: Systémové inženýrství a projektový management	10
Obrázek 2: Fáze projektu	11
Obrázek 3: Životní cyklus projektu	11
Obrázek 4: Trojimperativ projektu	12
Obrázek 5: SMART	13
Obrázek 6: Příklad myšlenkové mapy	14
Obrázek 7: Vztahy mezi požadavky	20
Obrázek 8: Řízení požadavků	22
Obrázek 9: Vývoj požadavků dle Ian Sommerville	23
Obrázek 10: Proces analýzy požadavků	25
Obrázek 11: Specifikace požadavků	26
Obrázek 12: V-model	28
Obrázek 13: Proces správy změn požadavků	30
Obrázek 14: Proces RTC	31
Obrázek 15: Způsoby reprezentace sledovatelnosti požadavků	33
Obrázek 16: Postup vypracování práce	37
Obrázek 17: Spolupráce systémového analytika s dalšími členy	40
Obrázek 18: Přehled subprocesů dle druhu projektu	42
Obrázek 19: Přehled sledovatelnosti mezi požadavky a jejich testováním	48
Obrázek 20: Návrh ERD pro sledovatelnost požadavků	52
Obrázek 21: Přehled řešení problému	57

SEZNAM ZKRATEK

BPMN	Business Process Model and Notation
ERD	Entity Relationship Diagram
HW	Hardware
ID	Identifikátor
MS	Microsoft Office
SQL	Structured Query Language
SW	Software
SRS	System/Software Requirements Specification

ÚVOD

Nejdůležitějšími úkoly projektu, a zároveň také nejsložitějšími, je jejich správná realizace, včasné předání, dodržení rozpočtu a dosažení definovaných cílů. Splněním těchto podmínek se očekává spokojenost při předání zákazníkovi či konečnému uživateli. Tomu, aby byl projekt úspěšný, předchází správná formulace požadavků, které je zapotřebí naplnit. Často nastává situace, kdy jsou v projektu nevhodně určeny a nastaveny procesy, kde realizace ověřování požadavků a jejich správa při změnách je nevhodná nebo dokonce žádná. Rozdílnost a nepochopení při mezinárodních projektech může způsobit jazyková bariéra, standardy, různé styly řízení atd., což může vést k nesplnění zákaznických požadavků.

Proces řízení požadavků může napomoci k tomu, aby se předešlo k nesplnění cíle projektu a byly uspokojeny zákaznickovy požadavky. Jedná se o systematický přístup, kde jsou jednotlivé požadavky kontrolovány zúčastněnými stranami. Využití tohoto přístupu je přínosné zejména pro větší a komplexně zaměřené (mezinárodní) projekty. V průběhu může docházet ke změnám požadavků, jak ze strany zadavatele, či vyplynuly při vývoji. Díky správě požadavků, se tento problém stává snadněji řešitelným, a nemusí být projekt ohrožen zvýšenými náklady, sníženou kvalitou, časovým prodloužením předání konečného projektu atd. Výhoda plynoucí z řízení požadavků je ve zlepšení komunikace mezi zákazníkem a dodavatelem, a zároveň napomáhá v komunikaci a práci celému projektovému týmu.

Do správy spadá i problematika sledovatelnosti (traceability) požadavků. Díky ní lze sledovat a kontrolovat jednotlivé požadavky, které zároveň usnadňují práci plynoucí ze změn.

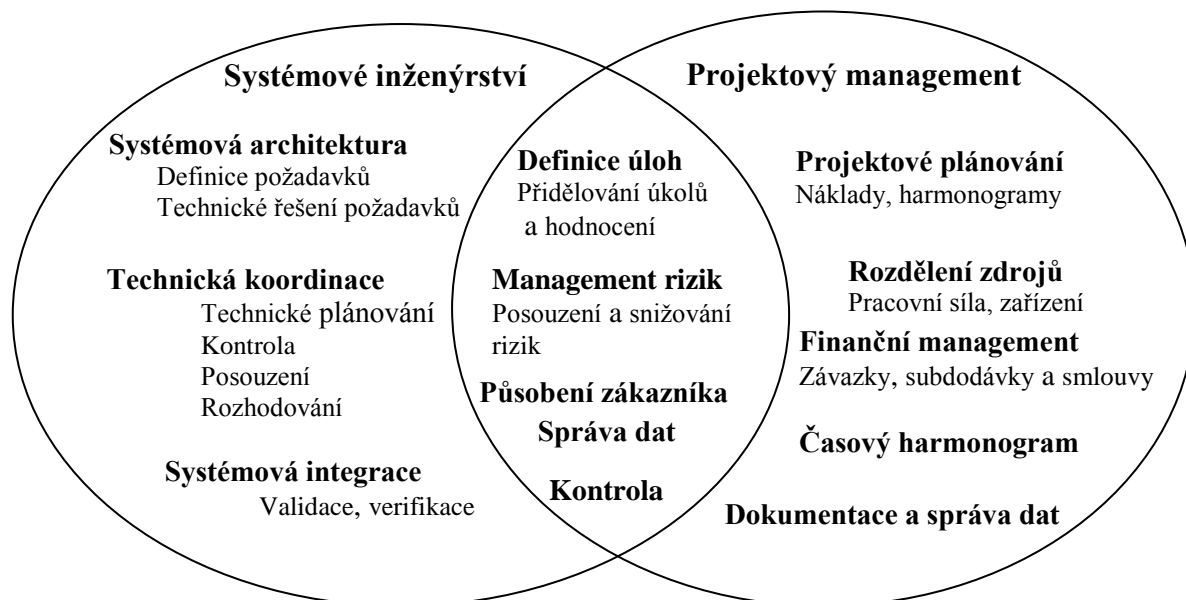
V úvodu práce budou vysvětleny základní pojmy z oblasti projektového managementu a systémového inženýrství, aby si čtenář dokázal zařadit souvislosti s následnou problematikou řízení a správy požadavků v rámci vývoje systému v jednotlivých fázích projektu. Dále bude zmapován proces řízení požadavků a v závěru bude vytvořen návrh modelu pro sledovatelnost požadavků.

Cílem práce je navrhnout proces řízení požadavků s využitím dostupných standardů, praktik a metodik v prostředí vybrané firmy.

1. ÚVOD DO PROJEKTOVÉHO MANAGEMENTU

Při vytváření projektu dochází k několika projektovým fázím, které jsou popsány v následujících kapitolách. V jednotlivých fázích by se měly uplatňovat a zároveň dodržovat určité metody a techniky, na kterých se zúčastněné strany předem dohodly. Tyto metody zároveň navazují na inženýrství požadavků, někdy také nazývané řízení (Requirements Engineering). Ve fázích projektu je zapotřebí dodržovat předem stanovené požadavky, které si zúčastněné strany předem určily. Jejich propojení a dodržování je velice důležité, jelikož zejména díky nim, dochází k úspěšně dokončenému projektu a naplnění cíle. Zároveň se v projektu uplatňují znalosti inženýrských oborů a jiných vědeckých disciplín. Jedním z nejdůležitějších aspektů systémového inženýrství je rozvoj nových technologií, což je spojuje s něčím jedinečným, jednou opakovatelným, jako je projekt.

Při vytváření projektu se využívají přístupy systémového inženýrství, zejména v rámci vývoje komplexních systémů obsahující hardware, software anebo kombinace obojího. Tyto oblasti jsou úzce propojené a navazují na sebe. Vztah mezi systémovým inženýrstvím a projektovým managementem zobrazuje obrázek 1. [19] Projektový management zde vystupuje spíše z hlediska kontroly nad projektem. Stará se především o dohlížení, plánování, definování cíle a nákladů pro spokojenost všech zúčastněných stran. Naproti tomu systémové inženýrství poskytuje zejména technickou podporu. [22]



Obrázek 1: Systémové inženýrství a projektový management

Zdroj: upraveno podle [19][22]

Projektové fáze jsou logicky uspořádané činnosti z pohledu řízení projektu. Je to časový úsek, který je od ostatních jasně oddělen a vymezen. Dle [9] se fáze dělí na koncepční, plánovací, realizační a fázi předání. Podobně je tak uvedeno i dle [4] na zahájení, plánování, realizaci, monitorování, kontrolu a ukončování.

V této práci budou popsány fáze dle členění na obrázku 2. Jednotlivé fáze v projektu jsou definovány dle potřeb dodavatele a zákazníka projektu, na kterých se shodli. Některé fáze mohou být sloučeny do jedné či vynechány. Výsledkem jedné z částí projektových fází, před zahájením realizace, může být například dokument či vytvořený prototyp. Díky těmto postupným výstupům je jasná kontrola toho, co je již v dané fázi splněno a zda jsou dodrženy například zákaznickovy požadavky. [7]

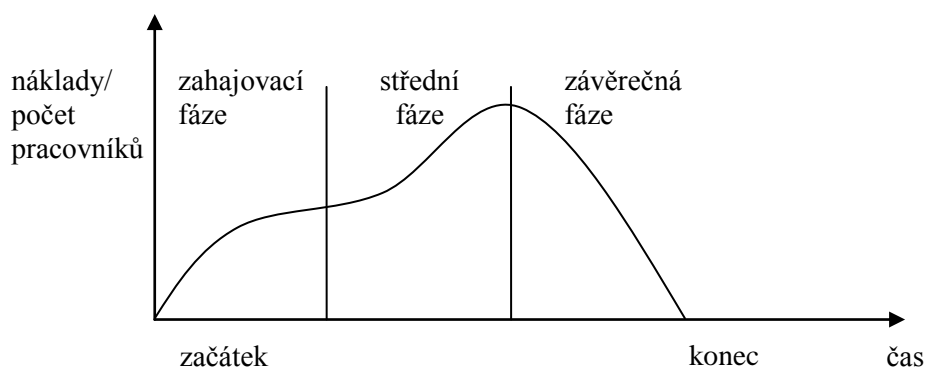


Obrázek 2: Fáze projektu

Zdroj: vlastní zpracování

Pro každou fázi projektu je vhodné určit: vstupy, procesy, klíčové činnosti, výstupy a možná rizika. [9]

Pro znázornění projektových fází se používá například životní cyklus projektu, který ukazuje obrázek 3. Na osách jsou čas a náklady nebo počet pracovníků. Křivka znázorňuje průběh projektu. Jednotlivé úseky patří do zahajovací fáze, střední a závěrečné fáze. Jak již bylo zmíněno, jejich rozložení je subjektivní. Každý projekt jich může mít různý počet. Stejně, tak každý projekt může mít různý průběh křivky. Například na obrázku 3 je na začátku počet zapojených pracovníků nízký, s čímž také souvisí nižší náklady. Ve střední fázi poté prudce rostou a ke konci projektu opět klesají. [27][7]



Obrázek 3: Životní cyklus projektu

Zdroj: upraveno podle [27]

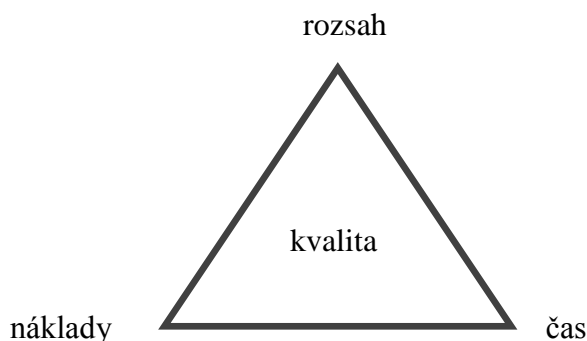
1.1. Inicializace

V této fázi se převážně identifikují potřeby a cíle, vybírají se členové týmu, hodnotí se rizika a připravují se požadavky na jednotlivé zdroje. Mohou se vytvářet návrhy na řešení a z nich je poté vybrána nejvhodnější varianta (projekt). [9] Pro tento účel se sestavuje logický rámec projektu, kde je sepsána logická návaznost, cíl, záměr, výstupy, aktivity, prostředky, zdroje ověření, objektivně ověřitelné ukazatele, předpoklady a rizika, předběžné podmínky atd. [6]

Hlavním účelem a zároveň také výstupem inicializační fáze je dle [34] Zakládací listina projektu. V souvislosti s řízením požadavků může být výsledkem dokument Specifikace požadavků.

V inicializační fázi, z pohledu řízení požadavků, probíhá především příprava a formulace se zákazníkem. Jedná se o dohodu, jakým způsobem bude probíhat komunikace v rámci vývoje systémů, sladují se procesy, probíhá sběr požadavků atd.

V inicializaci je důležité se zaměřit na prvky projektu, jako jsou čas, náklady, rozsah a vzájemné vztahy mezi nimi. Tyto vazby jsou znázorněné v projektovém trojúhelníku (nazývaný i **trojimperativ projektu**) na obrázku 4. Na vrcholech trojúhelníku jsou umístěny čas, náklady a rozsah projektu. Vazby jsou znázorněny hranami. Je důležité najít takové místo mezi těmito třemi vrcholy, aby došlo k rovnováze a dospělo se k požadované kvalitě. Někdy je to obtížné, jelikož zákazník preferuje například co nejnižší náklady, což se může projevit na zhoršené kvalitě či prodloužením vymezeného času na projekt. Proto je důležité hned na počátku vymezit jasný cíl mezi jednotlivými účastníky projektu. [27]



Obrázek 4: Trojimperativ projektu

Zdroj: upraveno podle [18]

V této souvislosti je důležité zmínit, že při nedodržení trojimperativu dochází k nesplnění stanoveného cíle. Častými příčinami neúspěšnosti projektů jsou [25][6]:

- špatná příprava plánu projektu, práce na poslední chvíli,
- nedostatečná dokumentace a sledování,
- neschopnost definovat požadavky,
- komunikační problémy,
- špatné vedení a nezkušenost projektových manažerů,
- nepřesné odhady nákladů, času,
- nepředpokládané vlivy na životní prostředí,
- nereálné cíle atd.

S touto problematikou souvisí i téma této práce a to, proč je správa a celkové řízení požadavků pro projekt důležité. Dle studie AMR Research z roku 2006, kde se zjišťovala nespokojenost zákazníka s projektem, je nejčastější příčinou nespokojenost zákazníka s produktem a pozdní uvedení produktu na trh. Obě tyto příčiny souvisí se správou požadavků. Naopak dle studie Aberdeen Group z roku 2008, by projekt dopadl úspěšně, pokud by se zlepšila komunikace a spolupráce mezi jednotlivými disciplínami a viditelnost stavu požadavků. [8]

Dále je nutné v inicializaci jasně stanovit cíl a náležitě ho popsat. Pokud je stanoven cílový stav, což je velice náročná záležitost, může se začít plánovat projekt. Jednou z technik pro definování cíle je **SMART**. Jednotlivá písmena znázorňují vlastnosti pro správné stanovení cíle a jsou znázorněny na obrázku 5. [7] Součástí jsou i studie proveditelnosti, příležitostí, ekonomické analýzy. [9][7]

S	M	A	R	T
<ul style="list-style-type: none"> • specifický • cíle mají být specifické a konkrétní 	<ul style="list-style-type: none"> • měřitelný • musí mít měřitelné parametry, podle kterých se pozná, zda bylo cíle dosaženo 	<ul style="list-style-type: none"> • akceptovaný • cíle mají být přidělené subjektu s odpovědností 	<ul style="list-style-type: none"> • realistický • cíle mají být dosažitelné 	<ul style="list-style-type: none"> • termínovaný • cíle mají mít jasně stanovený začátek a konec

Obrázek 5: SMART

Zdroj: upraveno podle [7][33]

1.2. Zahájení

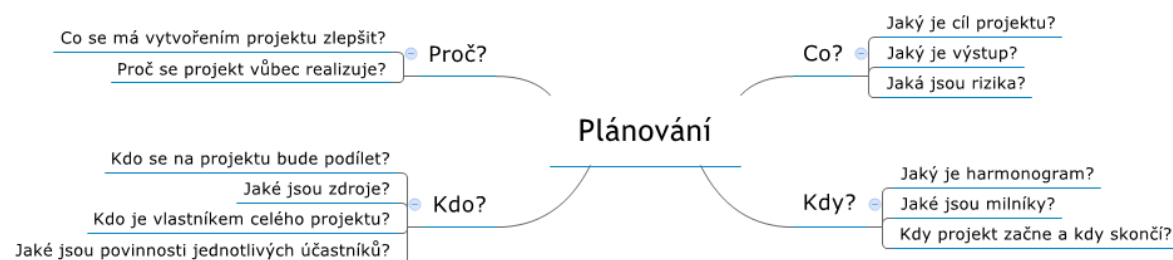
Před rozhodnutím, zda projekt bude zahájen či ne, musí proběhnout důsledné posouzení a vyhodnocení projektu. V této chvíli je možné využít detailní studii proveditelnosti nebo

vytvoření logického rámce. Bez této fáze by k realizaci nemělo vůbec dojít. Vlastník projektu dá většinou souhlas k tomu, aby se mohlo přistoupit k další etapě, kterou je plánování. Pokud všechno předzpracování proběhlo úspěšně, ještě stále nelze s jistotou usoudit, zda zahájení projektu nastane. Součástí této fáze je Zakládací listina projektu¹, což může být zároveň i výstupem anebo se používá Plán řízení projektu. Jsou v ní sepsány všechny meze rozpočtu, harmonogram. Je určen projektový manažer, projektový tým a požadované výsledky. Z tohoto dokumentu vycházejí veškeré další kroky přípravy a realizace. Každá organizace si svůj dokument upravuje dle vlastních potřeb. Je ale doporučeno, aby v něm nechyběly alespoň části týkající se názvu projektu, cíle, hlavní milníky. [7] Zahájení je v některých literaturách přiřazována do fáze inicializační nebo plánování. [34]

1.3. Plánování

Plánování je důležitým krokem před začátkem realizace projektu. Je to příprava budoucích kroků a postupů pro dosažení stanoveného cíle. Je zde vyjádřena detailní struktura činností, které se mohou modelovat například pomocí síťového grafu nebo Ganttova diagramu. [9] Vychází ze Zakládací listiny projektu. [34] Plán musí být kvalitní a realistický. Realistický v tomto smyslu znamená přizpůsobení se možnostem, které firma má.[7][18]

Při plánování projektu je potřebné odpovědět si na všechny potřebné otázky, a to především „co“ a „jak“ se bude dělat. Pro takovéto případy může sloužit **myšlenková mapa** zobrazená na obrázku 6, která propojuje myšlenky, vztahy a souvislosti. [18]



Obrázek 6: Příklad myšlenkové mapy

Zdroj: vlastní zpracování

Součástí plánování je celkové zjištění všech potřebných vstupů a dostupných zdrojů pro vyhotovení projektu. Je potřeba si ujasnit celé řízení a koordinaci projektu, rozsah, čas, rozpočet, sledování a definování kvality, řízení lidských zdrojů a komunikaci. Důležité je si

¹ Někdy je tento dokument nazýván jako Identifikační listina projektu, Zadání projektu či Definiční dokument.

stanovit, jaká mohou nastat rizika a jak budou v rámci projektu řízena. Nedílnou součástí je nákup a obstarávání zboží, služeb a průběh komunikace se zainteresovanými stranami. [7]

Ve fázi plánování, v souvislosti se řízením požadavků, jde především o předání oficiální dokumentace manažerovi projektu (nebo realizátorovi). Následuje přibližný návrh systému a vytvoření směrné verze Specifikace požadavků.

1.4. Realizace

Realizace projektu nastává ve chvíli, kdy je schválen Plán řízení projektu nebo také nazývaný Směrný plán. Dle metodiky PMBOK projektový plán slouží „...*pro zdokumentování předpokladů a rozhodnutí, usnadnění komunikace mezi zúčastněnými stranami, zdokumentování schváleného rozsahu, ceny, harmonogramu....*“. Dle metodiky PRINCE2 je to „...*prohlášení o tom, jak a kdy má být dosaženo cílů projektu, definuje hlavní produkty, milníky, činnosti a zdroje potřebné pro realizaci projektu...*“. [21] Zahájit realizaci je vhodné kick-off meetingem². Což je setkání důležitých zainteresovaných stran, kde se představí členové projektového týmu, vysvětlí se, jak se bude projekt realizovat, jeho přínosy, jaký je harmonogram, následuje přehled o rozpočtu, matice zodpovědnosti, jak budou sledovány změny a stavy atd. [7] V průběhu realizace je potřebné kontrolovat odchylky od plánu a na základě nich přijímat korekční opatření. [9] V realizačních fázích se vytváří reporting. **Reporting** je podávání zpráv o průběhu projektu. Pro efektivní podávání zpráv je nutné stanovit, kdo, kdy a jakým způsobem bude zprávy zpracovávat, komu budou předávány, jaký bude jejich obsah a forma. Poté se vyhodnocuje a porovnává skutečnost s vyhotoveným plánem. [7]

Z hlediska řízení požadavků zde již projektový tým, s pomocí Specifikace požadavků, vytváří detailní návrh systému a vytváří rozpad Uživatelských požadavků na dílčí subsystémy, komponenty či sestavy. Připravují se návrhy na testování. Zároveň se Specifikace požadavků využívá při případných změnách.

1.5. Monitorování a kontrola

Zde se uplatňuje dohled nad jednotlivými fázemi projektu. Zaměřuje se na soulad mezi všemi realizačními složkami porovnávanými s projektovým plánem. Monitorování a kontrola probíhá většinou ve všech fázích průběžně. [34] V souvislosti se řízením požadavků se provádí verifikace.

² Kick-off meeting je doporučován také již ve fázi inicializace.

1.6. Ukončení

Ve fázi ukončení je výstup projektu testován a ověřován. Zároveň se zde uplatňuje jako i v jiných fázích systémový přístup. [9] Po předání a akceptování všech výstupů podle Plánu řízení projektu (Směrného plánu) bez dalších připomínek vlastníka projektu a zákazníka, může být projekt ukončen. Pro ukončení se vypracovává Závěrečná zpráva. V této zprávě jsou sepsány zkušenosti z realizace, případně zde mohou být uvedeny i rady pro další projekty. [7]

Výsledkem projektu je produkt nebo služba, která je uvedena do provozu. Zároveň je nutné projekt ukončit i časově. V této fázi se doporučuje vytvořit i poučení, kde jsou sepsány chyby, které by se v dalších projektech již neměly opakovat. [7]

Fáze ukončení v rámci řízení požadavků, je spojena s verifikací a validací požadavků a následnou akceptací zákazníkem.

2. ÚVOD DO SYSTÉMOVÉHO INŽENÝRSTVÍ

V této kapitole budou vysvětleny základní pojmy, jako je systémové inženýrství, softwarové inženýrství, co je to požadavek a jaké jsou jeho vlastnosti. Dále zde bude uveden pojem řízení, správa a vývoj požadavků. Úvod do systémového inženýrství je v této práci uveden zejména z toho důvodu, že jeho dílčí součástí je inženýrství požadavků.

2.1. Systémové inženýrství

Je mnoho způsobů pro výklad pojmu systémové inženýrství. Skládá se ze slov systém a inženýrství. **Systém** v tomto smyslu znamená vzájemné propojení prvků a komponent, pro dosažení stanoveného cíle. **Inženýrství** je propojení poznatků z různých vědeckých disciplín, ať se již aplikují technické, systémové, procesní znalosti, které splňují určitá kritéria. [19][17] Zároveň je systémové inženýrství nedílnou součástí projektového řízení v případě vývoje komplexních systémů.

Systémové inženýrství (System Engineering) zahrnuje vývoj systémů, které mohou nebo nemusí zahrnovat software. Systémoví inženýři se zaměřují na transformaci potřeb zákazníků, jejich očekávání a omezení do produktového řešení v celém životním cyklu výrobku. [17] Je to tedy mezioborový proces, který vyvíjí, verifikuje a integruje sadu systémových řešení, které vyhovují potřebám, respektive požadavkům zákazníka. [36]

Softwarové inženýrství (Software Engineering) zahrnuje vývoj softwarových produktů. Zaměřuje se na uplatňování různých systematických přístupů k vývoji, provozu a údržbě softwaru. [17][20]

Hardwarové inženýrství (Hardwar Engineering) je již aplikace a transformace požadavků, které představují potřeby zúčastněných stran a jejich očekávání za použití dostupných technik a technologií pro implementaci hmatatelného produktu. Hardwarové inženýrství tedy převádí požadavky do hmotných a vyrobitelných produktů. [17]

2.2. Požadavek

Slovníček softwarové terminologie (IEEE Standard Glossary of Software Engineering Terminology) z roku 1990 definuje, požadavek je [39][20]:

1. Podmínka či funkce, kterou potřebuje uživatel pro řešení problému nebo je nezbytná pro dosažení daného cíle.

2. Podmínka či funkce, která musí být splněna pro splnění smlouvy, standardu, specifikace nebo jiného dokumentu v rámci systému či jeho části.
3. Dokumentovaná forma některého z předchozích bodů.

Ačkoliv je tato definice brána pro softwarové systémy, je vhodná i pro vyjádření ne-softwarových specifikací (požadavků). [20]

Další definice vyjadřují, že požadavek je popis toho, co vše je nutné implementovat. Je to popis chování a vlastností systému, který je žádaný. Obecně, pojem požadavek není přesně ustanoven, proto je potřeba si při vytváření nového projektu stanovit, co pod tímto pojmem zúčastněné strany očekávají. [39] Ideálním případem je, pokud jsou definované požadavky nezávislé na jejich konstrukci. Ukazují jen „co“ by měl systém dělat, a ne „jak“ by to mělo být konstruováno.[2]

2.2.1. Typy požadavků

V literatuře je možné se setkat s různými typy požadavků, některé z nich budou v této kapitole uvedeny.

Dle [42] lze dělit typy požadavků na softwarové a hardwarové. Softwarové požadavky dělí na funkční a nefunkční. Hardwarové dělí dle požadavků na výkon, rozhraní, životní prostředí a potřebnou odbornost. Následně rozděluje požadavky na obchodní, uživatelské, environmentální, produktové.

Dále dle [2] lze například dělit požadavky na ty, které se vztahují k celkovým cílům organizace (souvisejí s obchodními cíly), požadavky na doménu (vztahují se k problémové oblasti). Poté dělí požadavky na základní (zadané zúčastněnými stranami) a odvozené (odvozené ze základních). Následně klasifikuje požadavky na obchodní a technické, produktové a procesní (co podnik potřebuje, oproti tomu s jakým systémem budou lidé pracovat). A v poslední řadě jsou to požadavky dle rolí (požadavky uživatelské, IT, systémové a bezpečnostní).

Dle [39] lze dělit požadavky na podnikatelské, funkční a uživatelské. **Podnikatelské požadavky** vyjadřují, proč je systém potřeba, co se od něho očekává či co je cílem organizace, které má daný systém pomoci. Tyto požadavky přicházejí například od hlavních investorů či od nabývajících zákazníků. Příkladem podnikatelského požadavku může být například: *Snížení nákladů na vývoj požadavku REQ_1 o 25%*. V **uživatelských požadavcích** jsou uvedeny cíle a úkoly, které musí při své práci vykonávat. Pro zápis uživatelských požadavků se používají například případy užití, scénáře a tabulky s reakcemi na různé události. Příklad uživatelského požadavku: *Systém změří a uloží číselnou hodnotu teploty*

zařízení XY v rámci požadavku REQ_2 ve °C. **Funkční požadavky** již popisují funkcionality, které musí systém zvládat, aby byly uživatelské a tím i podnikatelské požadavky uspokojeny. Jedná se o požadavky na chování systému. Příklad funkčního požadavku: *Systém automaticky nastaví stav zařízení XY připravený, po vykonání události A.* **Systémové požadavky** jsou již celkové požadavky kladené na výsledek. Do nich jsou zařazeny i podsystémy, software, hardware nebo lidé. [39]

Dle [29] lze dělit **požadavky** také na **funkční, nefunkční a doménové**. Nefunkční požadavky jsou takové, které popisují omezení služeb a funkcí. Může se do nich zařadit omezení časové (jako je doba odezvy), výkon, zabezpečení nebo dostupnost. Rozdíly mezi funkčními a nefunkčními požadavky nejsou tak jednoznačné. Nefunkční požadavky mohou při detailním rozložení na subpožadavky obsahovat i funkční požadavky. Jakmile je to možné doporučuje se požadavky zapisovat kvantitativně, aby se daly objektivně testovat, tj. použít metriky. Příklad jak ohodnotit danou vlastnost zobrazuje tabulka 1. K jednotlivým metrikám se doporučuje napsat i konkrétní hodnotu dosahovaného výsledku.

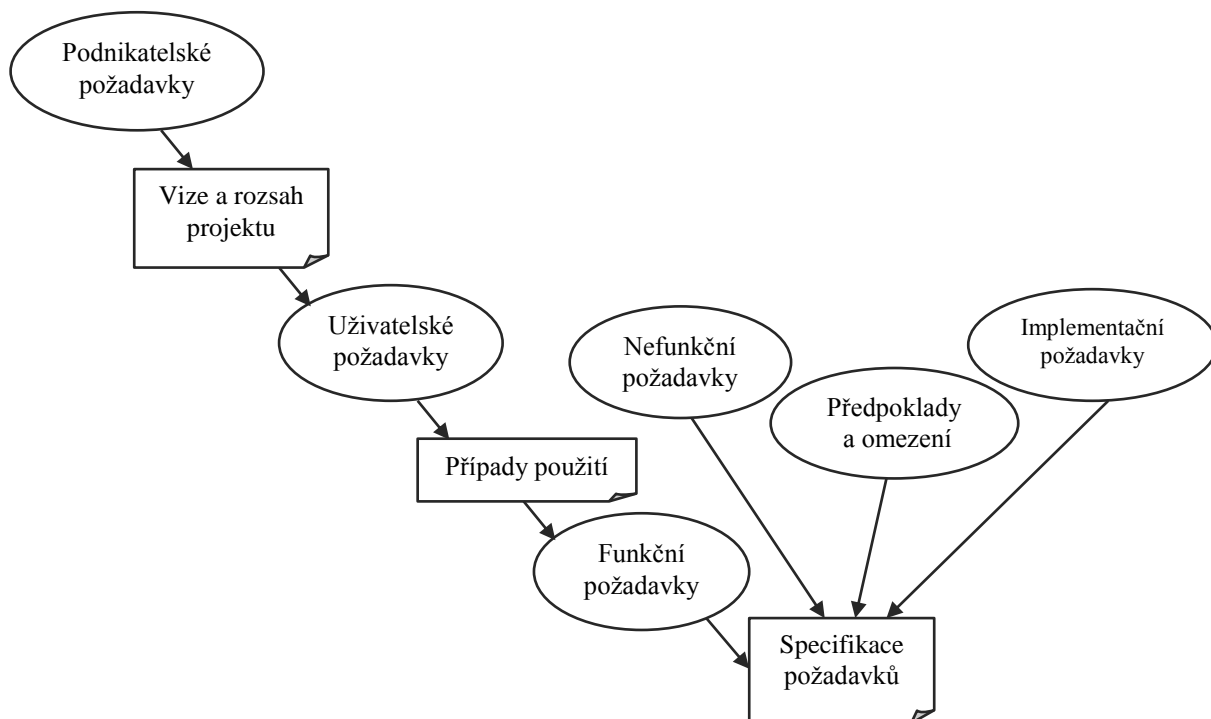
Tabulka 1: Nefunkční požadavky příklad

Vlastnost	Metrika
Rychlost systému	Zpracování transakce za sekundu. Čas aktualizace je do 5 sekund.
Velikost systému	Megabajty.
Spolehlivost	Frekvence výskytu chyb, pravděpodobnost nedostupnosti, průměrná doba odezvy.

Zdroj:[29]

Vztahy mezi druhy požadavků

Vazby mezi jednotlivými druhy požadavků zobrazuje obrázek 7. Zároveň je z obrázku viditelné, jakým způsobem na sebe požadavky navazují a z jakých dokumentů vycházejí. Podnikatelské požadavky vycházejí z podnikatelských cílů a strategií, jsou zaznamenány ve vizích a rozsahu projektu. Obsahem jsou především cíle, důvody a také metriky. Následují uživatelské, funkční, nefunkční, implementační požadavky a předpoklady a omezení. To vše je zaznamenáno do dokumentu Specifikace požadavků. [38]



Obrázek 7: Vztahy mezi požadavky

Zdroj: upraveno podle [38]

2.2.2. Vlastnosti požadavků

Každý požadavek musí vykazovat určité vlastnosti, jako jsou úplnost, správnost, proveditelnost, jednoznačnost, priorita a ověřitelnost. [39] Dle [2] slouží tyto vlastnosti převážně k vyjádření kvality.

Problematikou kvality softwaru se zabývá také standard ISO 9126. Dle této normy je možné rozdělit kvalitu do šesti kategorií: funkčnost, spolehlivost, použitelnost, efektivnost, udržitelnost a přenositelnost. Následně těchto 6 hlavních kategorií je dále rozpracováno normou ISO 9126-1. [32]

Úplný požadavek je takový, který obsahuje všechny informace, které potřebují vývojáři pro návrh a implementaci. Jsou to všechny funkcionality, které jsou žádoucí a musí být konkrétně popsány. [39]

Správný požadavek je popsán přesně. Správnost požadavku se může ověřit uživatelem, který bude výsledný produkt využívat. [39]

Proveditelný požadavek je myšlen ve smyslu realizovatelnosti dle známých možností a omezení systému v jeho prostředí. Dobrým pomocníkem pro zjištění, zda je požadavek proveditelný, je jeho prototypování. Vývojář zároveň může odborně posoudit, jakým způsobem by se požadavek dal vytvořit technicky. [39]

Nepostradatelný požadavek je takový, který si zákazník vyžádal, či jiný zdroj, který má právo klást požadavky. A musí být v systému obsažen. [39]

Jednoznačný požadavek je pouze ten, když každý účastník projektu chápe pod jedním požadavkem zcela to samé a dochází pouze k jedinému logickému výkladu. Je důležité popsat požadavky jednoduchým, stručným a srozumitelným jazykem. Je dobré zavést slovníček všech odborných výrazů.[39]

Priorita vysvětluje, jak moc je daný požadavek důležitý. [39]

Ověřitelný požadavek je takový, který je možný v průběhu či na konci nějakým způsobem otestovat, vyzkoušet nebo si ho prohlédnout. Pokud by nebylo možné požadavek ověřit, jednalo by se o neúplné, neproveditelné, nejednotné a nejednoznačné požadavky. [39]

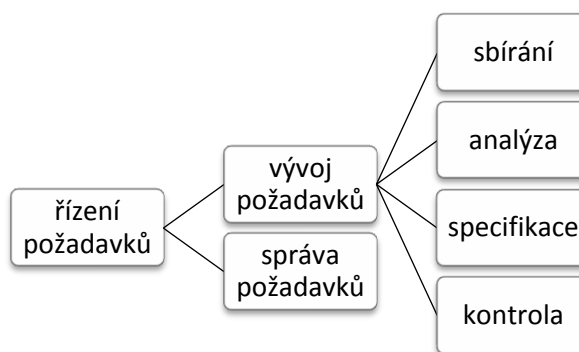
Podle [11] lze považovat požadavek za správný, pokud splňuje kritéria SMART, tj. specifický (Specific), měřitelný (Measurable), dosažitelný (Attainable), realistický (Realistic) a časově ohraničený nebo také sledovatelný (Time-bound, Traceable).

2.3. Řízení požadavků

Součástí této kapitoly je vysvětlení, co je to Requirements Engineering, Requirements Management, Requirements Development. Mezi těmito pojmy dochází často k záměně a rozporům, co si pod daným názvem představit. Přibližně od konce roku 1990 bylo trendem rozdělovat pojem Requirements Management od Requirements Development. Některé organizace braly pojmy Development Requirements a Management Requirements, jako Requirements Engineering. Některé organizace s tím nesouhlasily a považovaly pojmy Requirements Management a Requirements Engineering za totožné. [12] Snahou této kapitoly je objasnit a přiblížit tyto pojmy.

Requirements Engineering se často překládá jako řízení požadavků nebo požadavkové inženýrství. V různých literaturách je tento pojem často zaměňován s pojmem Requirements Management, což se dá přeložit jako správa požadavků či Requirements Development, což je vývoj požadavků. Řízení požadavků se zabývá činnostmi, které přesně specifikují potřeby zákazníků, které vyžadují od systému. Zároveň tyto požadavky musí být přesné a jednoznačné, aby se daly využít při následném vývoji systémů. Dalo by se říci, že řízení požadavků je proces vytváření specifikací požadavků, který analyzuje daný problém, dokumentuje výsledek pozorování v různých výstupních formátech a kontroluje správnost pochopení získaných informací. [20][12]

Tato práce vychází z výkladu problematiky řízení požadavků dle [39]. Souvislost řízení požadavků s vývojem a správou požadavků názorně zobrazuje obrázek 8, díky kterému je možné si lépe představit, kam jednotlivé pojmy zařadit. [39] Problematika řízení požadavků je také upravena dle standardu ISO/IEC/IEEE 29148 – Systems and software engineering – Life cycle proces - Requirements Engineering. Kde jsou zmíněné nejdůležitější zásady, které lze využít při implementaci řízení požadavků.



Obrázek 8: Řízení požadavků

Zdroj: upraveno podle [39]

2.4. Vývoj požadavků

Vývoj³ požadavků je dle [12] brán jako součást či dokonce jako totéž co, řízení požadavků. Smyslem vývoje požadavků je analyzovat zákazníka, produkt, komponenty produktu v souvislosti se všemi požadavky. Dle [39] do vývoje požadavků spadá sbírání, analýza, specifikace a kontrola. Jedná se tedy o počátek, kdy je potřeba identifikovat třídy uživatelů, kteří systém budou používat. Dále pochopení uživatelských úkolů a podnikatelských cílů apod. Výstupem vývoje požadavků je dokument, který popisuje rozsah a vize projektu. Obsahuje dokumentaci případů užití, specifikaci požadavků, datové modely včetně slovníku. Následuje celková kontrola, která poté schvaluje vše výše uvedené. Tímto vznikne tzv. Směrná verze požadavků⁴. Tímto odsouhlasením končí jejich vývoj a následuje jejich správa.

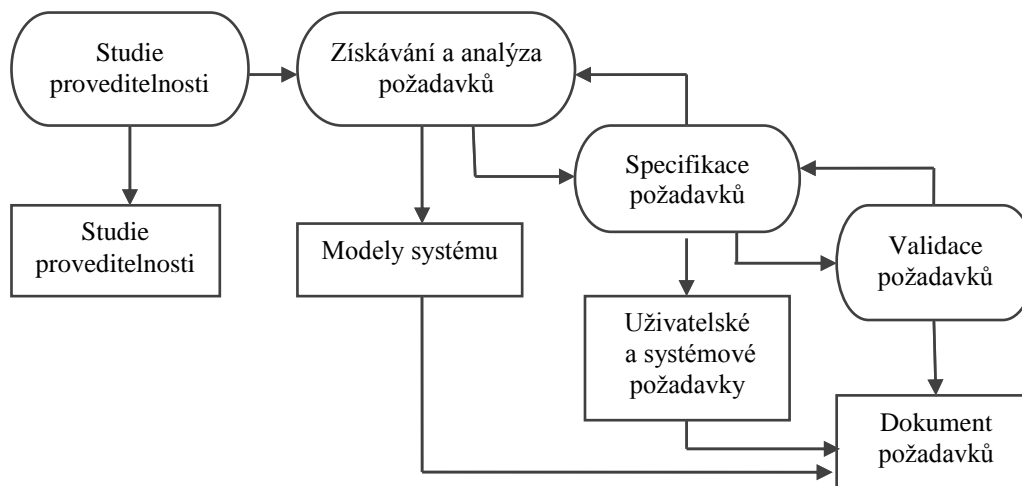
Činnosti v rámci vývoje požadavků jsou v různých odborných literaturách děleny odlišným způsobem, jelikož vznikají v závislosti na příslušné oblasti využití, či zapojení lidí v organizaci atd. Například dle [20], se vývoj požadavků skládá ze získávání znalostí, reprezentace a validace. Naproti tomu se v [2] uvádí aktivity, jako je elicitace (získávání/sběr)

³ Někdy překládané též jako Formulování požadavků.

⁴ Pod směrnou verzí požadavků si lze představit dokument, ve kterém jsou sepsané všechny požadavky a cíle, na kterých se dohodly osoby s přidělenou kompetencí (např. zákazník s vedoucím projektu a zároveň se vše konzultovalo s ostatními členy týmu). Zároveň je tento dokument označován též jako Specifikace požadavků.

interpretace a strukturování (analýza a dokumentace), validace a verifikace. Podle [28][29] jsou do této oblasti zařazovány aktivity, získávání požadavků a analýza, specifikace, validace.

V této práci budou zmíněny činnosti, které jsou navrženy dle [28] a jsou znázorněny na obrázku 9.



Obrázek 9: Vývoj požadavků dle Ian Sommerville

Zdroj: upraveno podle [28]

Všechny aktivity procesu vývoje požadavků nemusí postupovat sekvenčně, mohou se v průběhu procesu vzájemně prolínat. [29]

a) Studie proveditelnosti

Vstupem do studie proveditelnosti je sada předběžných obchodních požadavků. Jedná se o stručný popis systému a jak má systém podporovat obchodní procesy. Výsledkem studie proveditelnosti by měla být zpráva (zhodnocení), zda se systém bude vytvářet a bude vstupem do analýzy požadavků. [29][28]

Studie proveditelnosti je účelná a klade si pár základních otázek, kterými například jsou [28][29]:

- Jak má systém přispívat celkovým cílům organizace? A jaké dopady to bude mít na organizaci, pokud systém nebude zaveden?

Na druhou stranu z pohledu dodavatele systému se studie proveditelnosti může ptát, jaké náklady, čas a odborníci budou potřeba na vytvoření takového projektu.

- Jak může být systém realizován pomocí současných technologií při daných nákladech, časovém harmonogramu a daných omezeních?
- Jaké technologie jsou zapotřebí pro zapojení nového systému?

- Může být systém integrován s dalšími systémy, které jsou již na svém místě?

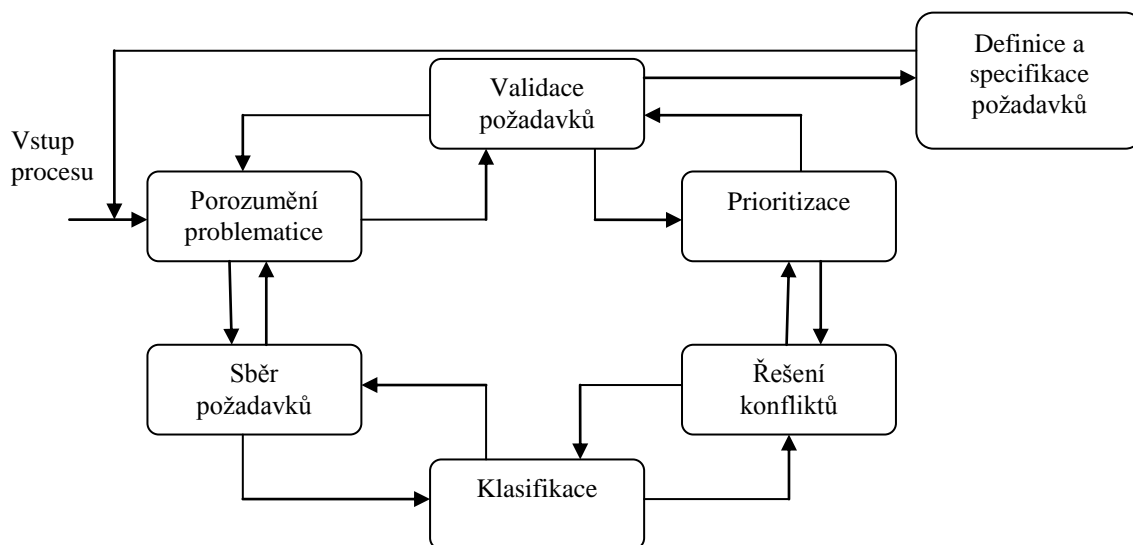
b) Získávání a analýza požadavků

V této fázi převážně analytici dodavatele projektu spolupracují se zákazníky či koncovými uživateli a chtějí se dozvědět vše o tom, co má výsledný systém poskytovat (např. výkonnost systému, hardwarové omezení atd.). Jde především o identifikaci nejasných, protichůdných požadavků a tyto nesrovnalosti zároveň vyřešit. Následně je nutné požadavky dokumentovat v různých formách, jako např. do podoby běžného dokumentového textu či využití modelů (např. Use Case). [28]. Tato část může zahrnovat i tvorbu modelů, prototypů, které by měly napomoci specifikovanému systému lépe porozumět. [29]

Procesní činnosti při analýze požadavků [28][30]:

- Sběr požadavků a jejich porozumění: komunikace a spolupráce se zúčastněnými stranami a zjišťování jejich požadavků. Je to proces shromažďování informací o navrhovaném systému. Zdroje těchto informací a specifikací se mohou získávat z podobných systémů a dokumentací.
- Klasifikace požadavků a jejich organizace: jedná se o správné rozdělení požadavků do skupin, například [30]:
 - Měnitelné požadavky: mění se v důsledku prostředí systému.
 - Náhle vzniklé požadavky: jsou ty, které se vyvodily při vývoji systému.
 - Vyplývající požadavky: vyplývají ze zavedení nového systému.
 - Požadavky na kompatibilitu: závislé na jiných systémech nebo organizačních procesech.
- Stanovení priorit a vyjednávání: upřednostňování požadavků a řešení nesrovnalostí mezi nimi.
- Dokumentace požadavků: jednotlivé požadavky jsou neustále dokumentovány a jsou vstupem do dalších fází procesu.

Obrázek 10 zobrazuje obecný proces analýzy požadavků dle [30].



Obrázek 10: Proces analýzy požadavků

Zdroj: upraveno podle [30]

V této části získávání požadavků se uplatňují například: rozhovory, dokumentování, brainstorming, workshopy, prototypování, dotazníky, analýza dokumentů, analýzy návratnosti investic (ROI⁵), analýza zákazníka, průzkumy atd. a jejich kombinace. Pokud je dobře vytvořená analýza, pokračuje se fází specifikace požadavků. [10][35][6]

V souvislosti s analýzou požadavků mohou nastat například následující problémy [30]:

- Zúčastněné strany nevědí, co opravdu od systému očekávají a v čem jim má napomoci.
- Zúčastněné strany mohou mít různé požadavky.
- Mohou je ovlivňovat organizační či politické faktory.
- Změny v průběhu vývoje systému, které mohou být zapříčiněné i novými osobami zúčastněných stran.
- Nedorozumění v používání odborných názvů mezi zúčastněnými osobami v projektu.

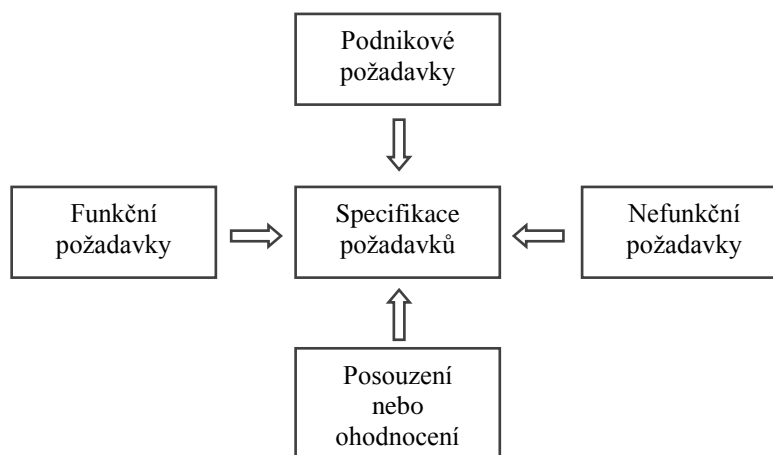
c) Specifikace požadavků

Specifikace požadavků je klíčovým prvkem ve vývoji celého projektu. Slouží pro správné definování a pochopení potřeb zákazníků nebo uživatelů systému. Zabývá se vlastnostmi systému, omezeními apod. Specifikace požadavků je důležitá nejen z pohledu vyjádření požadavků zákazníka, ale je nedílnou součástí při komunikaci mezi všemi zúčastněnými stranami, tak i uvnitř projektového týmu vyvíjející daný systém. Specifikace jsou součástí

⁵ Návratnost investic je jednoduchý ukazatel, který říká, kolik peněžních jednotek zisku nám přinese každá investovaná jednotka (nákladů). $ROI = (\text{výnos} - \text{investice}) / \text{investice}$

i smluvních ujednání a zároveň díky přesně dané specifikaci je možné posoudit konečný výrobek. [20][35]

Tradiční pohled na specifikaci požadavků systému je dle jeho funkčních, nefunkčních, dále podnikových nebo také uživatelských požadavků a zároveň musí být vše posuzováno a hodnoceno, viz obrázek 11. [20]



Obrázek 11: Specifikace požadavků

Zdroj: upraveno podle [20]

Výstupem ze specifikace požadavků je dokument. Pokud se jedná o softwarové požadavky je jím SRS dokument (Software Requirement Specification). Obecně lze říci, že i pro nesoftwarové požadavky je nutné dokument se specifikací požadavků sestavit (System Requirement Specification).

Výhodou dobře navrženého SRS dokumentu je možné dosáhnout [35]:

- Zákazník či koncový uživatel dostane zpětnou vazbu o tom, že poskytovatel systému dobře rozumí problému, který je potřeba vyřešit a jaký systém je potřeba vytvořit. S tím také souvisí jednoznačnost v používaných termínech. Soubor může obsahovat grafy, diagramy, tabulky, slovníky pro větší srozumitelnost.
- Zároveň SRS dokument pomáhá ověřit správnost dosažených výsledků požadavků při testování.

d) Validace požadavků

Validace systému v souvislosti s touto prací, znamená sestavení „správného“ systému. Ujistit se, že dělá jen to, co má dělat, co od něj bylo vyžadováno. Validace určuje správnost a úplnost výrobku a zároveň zajišťuje, aby systém splňoval požadavky zadané zákazníkem. Pomocí validace požadavků je možné kontrolovat, aby soubor požadavků byl správný, úplný,

konzistentní a reálný. V průběhu tohoto procesu se zjišťují chyby v dokumentu požadavků. Následně se řeší a opravují. Ověřuje se, že systém požadavky splňuje, a je možné je reálně sestavit či v případě opaku projekt včas zastavit. [26][29]

Se špatně nebo neúplně zadanými požadavky vznikají nemalé problémy a po finanční stránce mohou být náklady až stonásobně vyšší než ty předpokládané na začátku projektu. Opravy již implementovaných požadavků stojí nemalé úsilí. Stačilo by, pokud by se na danou chybu přišlo již během analýzy a specifikace. Časová náročnost odstranění chyby již při zpracovávání požadavků je minimální. Čím déle se na chybu přijde, tím více to stojí času a nákladů, aby se nesrovnalosti odstranily.[39][31]

V-model

Při vývoji systémů se doporučuje testovat vytvořené subsystémy či modely co nejdříve a k tomuto lze využít například tento model. V-model znamená verifikace a validace. Vychází z vodopádového modelu. Je ve tvaru písmene V a je zde vytvořena cesta, pro vyjádření realizace jednotlivých fází tvorby systému. Z jedné fáze do druhé je možné přejít až v té chvíli, pokud je dokončena ta předchozí. Aplikace je možná dle typu projektu, který se řeší a model si částečně přizpůsobit dle potřeby. [19][39][16]

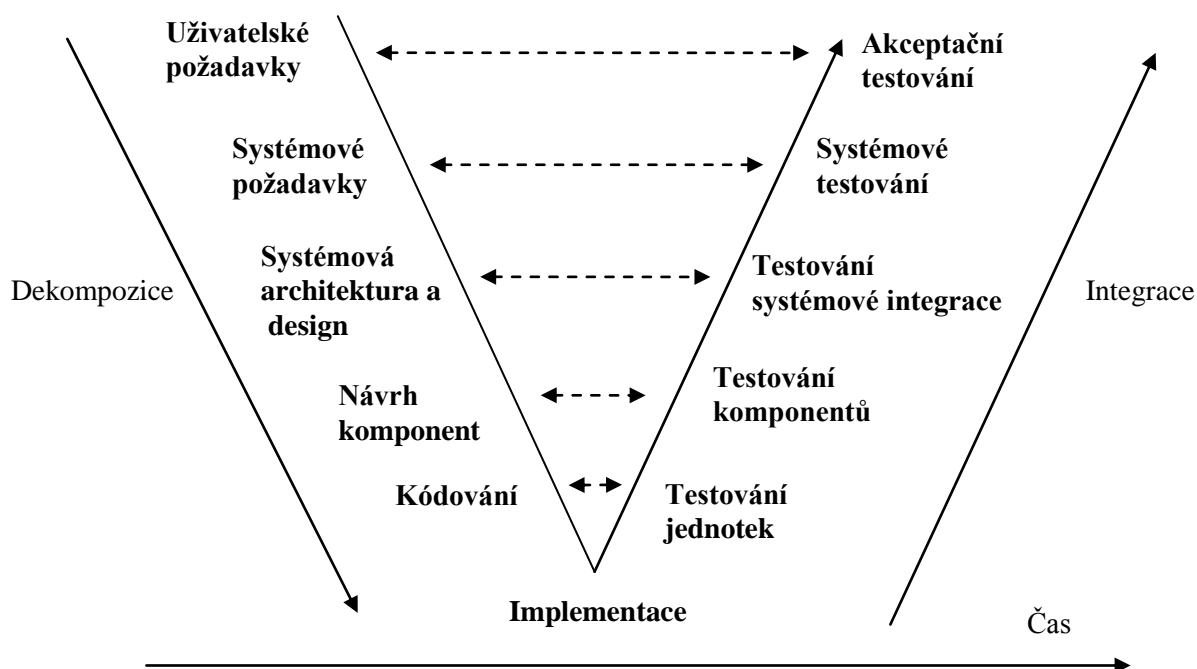
Tabulka 2 znázorňuje výhody a nevýhody V-modelu.

Tabulka 2: Výhody a nevýhody V-modelu

Výhody V-modelu	Nevýhody V-modelu
<ul style="list-style-type: none"> + Jednoduché, snadné použití. + Testovací aktivity plánování, návrhu nastane ještě před samotným kódováním či implementací modelu. To ušetří mnoho času. + Proaktivní sledování chyb. Tzn. že cílem je zachytit chyby systému již v rané fázi. Vyhýbá se tedy chybám, které by putovaly z jedné fáze do druhé a odhalila by se až např. při implementaci. + Vhodné spíše pro menší projekty, kde jsou požadavky snadno srozumitelné. 	<ul style="list-style-type: none"> - Málo flexibilní. - Pokud nastane v průběhu projektu změna, je potřebné změnit i všechny předchozí kroky v jednotlivých fázích (tzn. například zásah do zkušebních dokumentů, dokumentů požadavků). - Jednoduchá interpretace vede k odlehčenému pohledu na V-model. Snadno se tak stává, že pokud nedojde k potřebnému rozšíření v praxi, mohou nastat velké potíže s úspěšným uplatněním.

Zdroj: upraveno podle [16]

Na obrázku 12 je zobrazen V-model. Uživatelskými a systémovými požadavky začíná model životního cyklu. Zároveň paralelně probíhá testování. Následuje fáze systémové architektury a designu, která je často nazývána high-level design (HLD). Zde jsou zohledněna jednotlivá řešení systému. Zároveň zde probíhá i testování integrace, které je uskutečněno za účelem testování jednotlivých částí systémových komponentů, a ověřuje se, jak jsou spolu schopny spolupracovat. Následuje návrh komponent, neboli low-level design (LLD). Zde je definována každá součást systému. V této části se navrhuje například diagram tříd. Zároveň jsou v této fázi uskutečňovány testy komponent. Následuje kódování (pokud je součástí projektu i vývoj SW), kde vývojáři přetváří návrh modulu do kódu, to může probíhat už i v samotné implementaci, záleží na vytvářeném systému. Fáze implementace probíhá, jakmile je kódování kompletní.[19][16][39]



Obrázek 12: V-model

Zdroj: upraveno podle [19][39][16]

2.5. Správa požadavků

Správa požadavků je soubor postupů, které podporují rozvoj při plánování sledovatelnosti, analýze dopadů, řízení rizik a tak dále. Může to být rozhraní mezi vývojem požadavků a jinými obory, jako je například projektový management. Účelem správy požadavků je „spravovat“ požadavky v projektu a jejich produktové komponenty a identifikovat rozpory nebo i rizika mezi požadavky a projektovým plánem. [12] Dle [39] do této oblasti spadají například činnosti, jako je definice směrné podoby požadavků, posuzování a navrhování změn

a vyhodnocení jejich následků ještě před schválením, vyjednávání nových závazků podle očekávaného dopadu změn požadavků, sledování stavu a změn v průběhu projektu.

Rozdíl mezi vývojem a správou požadavků dle [39][22] je znázorněn následující tabulkou 3:

Tabulka 3: Základní rozdíl mezi vývojem a správou požadavků

Vývoj požadavků	Správa požadavků
<p>Během vývoje se požadavky:</p> <ul style="list-style-type: none"> - získávají, - analyzují, - specifikují, - kontrolují. <p>Výstupem z vývoje požadavků je:</p> <ul style="list-style-type: none"> - dokument, který popisuje rozsah a vize projektu, - dokumentace se specifikací požadavků, - mohou to být i datové modely a slovníky. 	<p>Během správy se v požadavcích:</p> <ul style="list-style-type: none"> - řeší změny, - sledují jednotlivé části až k jejich implementaci, - řídí se změny, navrhují se nápravná opatření, sleduje se stav a stabilita požadavku, aktualizuje se dokumentace, - provádí se verzování dokumentace a vytváří se definice o způsobu označení verzí, - provádí se správa testů, - kontrola požadavků, - udržují se odkazy na další požadavky.

Zdroj: upraveno podle [39][22]

Správa požadavků nejčastěji řeší splnění obchodních cílů a požadavků zákazníka, rychlejší dodání a lepší kvalitu systému, lepší kontrolovatelnost nákladů a efektivnější audit. [8]

2.5.1. Proces správy změn požadavků

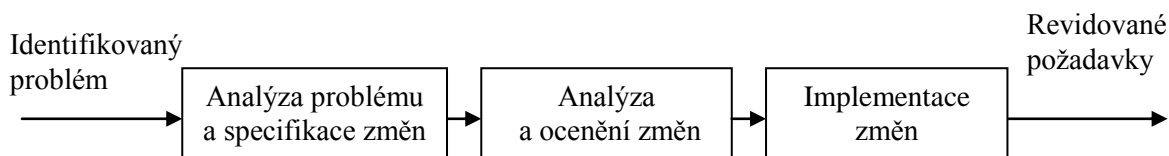
Dle [28][29] se požadavky na systém neustále mění. Děje se tomu tak i z důvodu složitějších problémů, které se musí vyřešit vzniklým systémem a nelze je na začátku úplně definovat. Dále se v průběhu realizace projektu, povědomí a znalosti dané problematiky zúčastněných osob mění a doplňují. To vede k tomu, že se požadavky mohou rozdělit na trvalé a přechodné. Trvalé jsou považovány za klíčové a nepředpokládá se jejich změna. Naopak přechodné souvisejí s pracovními činnostmi a jejich pravděpodobnost změny je vyšší. Je důležité vzniklým změnám porozumět a dostat je pod kontrolu, udržovat vazby mezi závislými požadavky a vyhodnocovat změny. Například, zdali se musí upravit části návrhu, zdrojové kódy, testovací scénáře apod. Zároveň je potřeba odhadnout, kolik práce bude stát nejen změna, ale i odhadování důsledků.

Postupy pro správu proměnlivých požadavků je doporučeno plánovat již v procesu získávání (sběru) požadavků. [29]

Zásadní význam je v rozhodnutí, zda výhody nových požadavků převažují nad náklady jejich implementace. Důležitým krokem je i dokumentace a zpracování požadavku na změny, aby se daly evidovat kontrolovatelným způsobem. [29] S tím souvisí také sledovatelnost požadavků, aby bylo možné zajistit vztahy a provázanost mezi nimi.

Proces správy změn obnáší tři fáze [29], které jsou znárodněné na obrázku 13:

1. Analyzování problému a v čem by spočívala změna: Identifikuje se problém a navrhuje se nové řešení, které se pošle žadateli o změnu.
2. Analyzování a oceňování změny: V této fázi se provádí vyhodnocování dopadů, které by změna přinesla. Využívají se k tomu informace např. z matice sledovatelnosti. Následně se analyzují náklady, které by souvisely s celkovou úpravou systému či dokumentů a následné implementace. Poté se rozhoduje, zda má smysl změnu provést.
3. Implementace změn: Pokud dojde k rozhodnutí o změně, je vhodné uspořádat dokument požadavků tak, aby se dal měnit bez rozsáhlého přepisování.



Obrázek 13: Proces správy změn požadavků

Zdroj: upraveno podle [29]

Dle [39] je navíc dobré vytvořit komisi, která bude mít na starosti řízení změn (Change Control Board). Dále se doporučuje stanovit proces, díky kterému se poté budou všechny změny řídit. Vhodné je verzování požadavkové dokumentace⁶. Jakmile se požadavková dokumentace podepíše, musí již každá změna projít procesem řízení změn. Doporučuje se používat běžný verzovací software. Zároveň je dobré vést historii změn. Což navazuje na verzování požadavkové dokumentace. Jde o zápis dat, kdy došlo ke změnám a o jaké změny se jednalo. Je doporučeno sledování stavu každého požadavku (např. navržený, schválený, implementovaný, otestovaný atd.). Stačí například vytvořit soubor (např. v MS Excel/Access), kde ke každému požadavku bude přiřazen stav, aby dodavatel, ale i zákazník měl přehled, o počtu požadavků v daných stavech. [39]

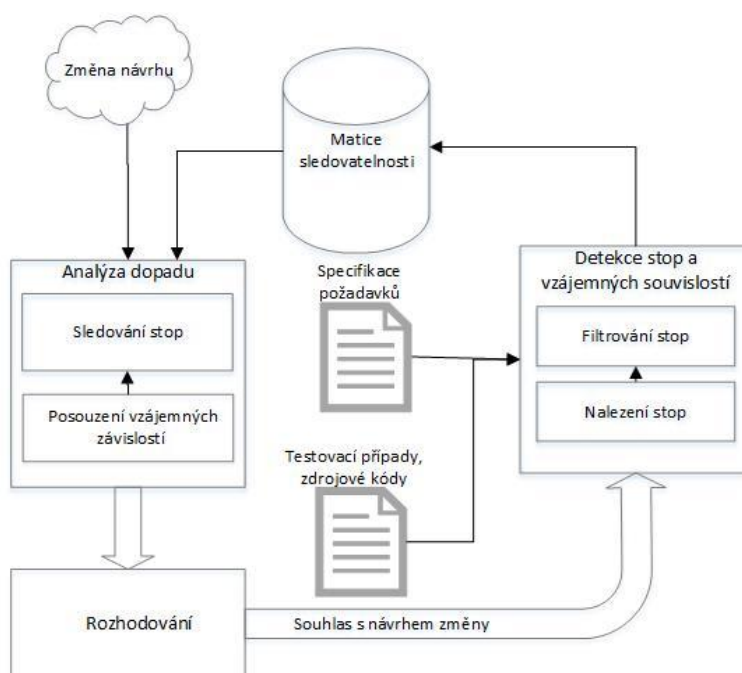
⁶ Jedná se o dokumentaci, kterou se dodavatel zavázal splnit, a na které se se zákazníkem domluvili. Například Specifikace požadavků, Uživatelské požadavky, Zadávací listina projektu atd.

Analýza dopadu změn

Analýza dopadu změn (Change Impact Analysis) je dle [39] důležitou součástí řízení požadavků. Jejím hlavním cílem je zjistit, jaké mohou nastat důsledky, pokud dojde ke změně, a jaké části (požadavky) by se případně musely upravit, nahradit, odstranit apod.

Jelikož častokrát nastávají v průběhu projektu změny, ať již ze strany zákazníků, zákonů, technologií, uživatelů, vývojářů atd., mohou způsobit mnohé problémy a těmto změnám by mohla napomoci sledovatelnost požadavků (viz kapitola 2.5.2) při jejich řešení. [3]

Analýzou dopadu se mají identifikovat potenciální důsledky, včetně vedlejších účinků, ještě před tím, než budou změny zavedeny. Studie [41] navrhuje též proces RTC Change Impact Analysis, který lze využít při změně na požadavek, který nastane například i v pozdním stádiu vývoje systému. Obrázek 14 zobrazuje proces, který začíná identifikací změny na požadavek. Ta je analyzována v analýze dopadu podle matice sledovatelnosti a detekcí stop. Fáze detekce stop je určena pro hledání souvislostí s dalšími požadavky. Čím více souvisejících požadavků se objeví, tím budou větší náklady na změnu. Nakonec se vypočítává dopad změny, který pomůže rozhodovateli pro zavedení návrhu změny či zamítnutí. Pokud bude změna přijata, aktualizuje se matice sledovatelnosti. [41]



Obrázek 14: Proces RTC

Zdroj: upraveno podle [41]

2.5.2. Sledovatelnost požadavků

Sledovatelnost požadavků (Requirements Traceability) slouží k zajištění správnosti všech kroků procesu vývoje systému. Aby byly v souladu s předešlými potřebami. Sledovatelnost požadavků je zařazována jako sub-disciplína správy požadavků v rámci řízení požadavků při vývoji systému, a zároveň tedy spadá i do oblasti systémového inženýrství. Sledovatelnost požadavků pomáhá zajistit, že projekt přináší to, co zainteresované strany očekávají. Při správném zavedení sledovatelnosti do praxe, to může mít pozitivní vliv na zvýšení kvality a spolehlivosti, minimalizaci nákladů a ostatních souvisejících negativních dopadů vzniklých s chybami v požadavcích. [5] Také dle [15] sledovatelnost požadavků pomáhá při řízení změn, rizik, času nákladů a komunikace.

Při ověřování správnosti a úplnosti projektové práce, se za pomoci matice sledovatelnosti ověřuje, zda daný výsledek projektu obsahuje požadované vlastnosti, a to v rámci celého životního cyklu projektu. [5] Sledování požadavků by mělo probíhat od plánování, analyzování, kódování i testování a následné implementace (viz. například všechny části V-modelu).

Ke sledování požadavků se nejčastěji využívají automatické nástroje, ale požadavky na sledovatelnost se mohou vytvářet i pomocí tabulek či relačních databází. Správný systém pro sledovatelnost požadavků by měl zajišťovat [5]:

- jedinečný identifikátor (ID) pro každý požadavek (pro lepší dohledatelnost a spojitost s dalšími požadavky),
- sledovat požadavky v rámci celého životního cyklu projektu,
- zachycení odpovědnosti za požadavek.
- verifikaci požadavků,
- historii změn požadavků.

Sledovatelnost požadavků se používá převážně pro větší projekty a provádí se buď interně v rámci firmy a/nebo ve vztahu se zákazníkem (dle V-modelu je například součástí akceptačních testů).

Způsoby vyjádření sledovatelnosti požadavků dle [40] jsou následující:

- a) Matice sledovatelnosti (Traceability Matrix)

Matice sledovatelnosti je jedna z nejčastěji používaných forem reprezentace. Jedná se o dvojrozměrnou mřížku, která představuje odkazy například mezi dvěma požadavky či návrhy atd. Propojení mezi nimi zobrazuje zvolený znak (viz obrázek 15 (a) je znakem černý

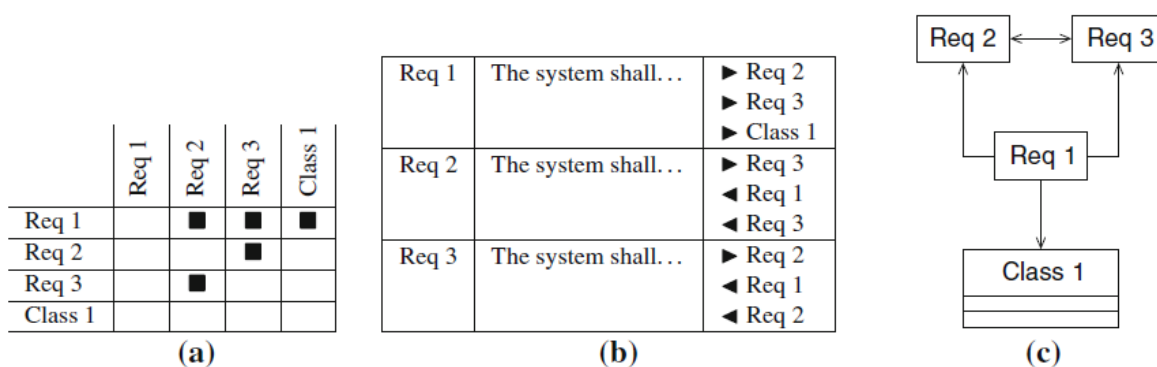
čtverec). Matici lze doplnit i o odkazy a další informace. Zároveň se používá i znázornění pomocí různých barev a symbolů. Matice sledovatelnosti avšak může být hůře přehledná a využitelná pro rozsáhlé požadavky. [40]

b) Křížové odkazy (Cross-references)

Sledovatelnost může být zobrazena i pomocí křížových odkazů. Jednotlivé požadavky mají ve svých vlastnostech umístěny hypertextové odkazy, které slouží pro navigování (viz obrázek 15 (b)). To umožňuje zobrazit přichodzí spojení (např. od jiného požadavku). Na rozdíl od matice sledovatelnosti je možné propojit více spojení s dalšími požadavky. Na druhou stranu, pokud je v jednom požadavku více odkazů, mohl by nastat problém s rozhodnutím, který odkaz je více důležitý. [40]

c) Grafická vizualizace (Graph-based visualization)

V grafické vizualizaci jsou požadavky zobrazeny jako uzly a sledovatelnost mezi nimi je vyjádřena pomocí hran. Stejně jako u matice sledovatelnosti lze identifikovat požadavky pomocí jedinečného identifikátoru (viz obrázek 15 (c)). Tímto způsobem lze vyjádřit rozsáhlejší informace (atributy) o požadavku. V tomto případě mohou být přehledněji zobrazeny i n-ární odkazy. U matice sledovatelnosti a křížových odkazů by se jejich znázornění stalo méně čitelným a srozumitelným. [40]



Obrázek 15: Způsoby reprezentace sledovatelnosti požadavků

Zdroj: [40]

2.5.3. Nástroje pro správu požadavků

Pouze pro přehled o dostupnosti komerčních nástrojů pro správu požadavků jsou zmíněny například tyto nástroje [39]:

- a) Databázové: IBM Rational Doors
- b) Dokumentové: IBM Rational Requisite Pro, RMTrak

Databázové programy ukládají všechny požadavky, atributy i dotazy do databáze. Používají buď relační, nebo objektovou databázi. V některých programech lze připojit k požadavkům i externí soubor s dodatečnými informacemi. V dokumentových programech je základem textový soubor. Pokud se firma rozhodne některý z těchto podpůrných programů zakoupit, je zapotřebí si promyslet, jaké další náklady s tím souvisí. Nejenže počáteční vklad není zrovna nízký, ale náleží k tomu i platba za roční licenci, dále aktualizace a údržba, správa, nároky na hardware, školení uživatelů apod. [39]

Často využívaným nástrojem je **IBM® Rational® DOORS®**. Je to aplikace pro správu požadavků a pro optimalizaci komunikace nad požadavky. Dále slouží pro spolupráci, verifikace v rámci organizace, ale i v dodavatelském řetězci. Umožňuje zachytit, sledovat, analyzovat, spravovat a řídit změny v souvislosti s požadavky. Poskytuje zejména správu požadavků pro lepší týmovou spolupráci, sledování požadavků skrze návrh požadavku, testovací plány s dalšími požadavky, a řízení změn nad požadavky. Cena za licenci pro jednoho uživatele činí 5,300\$ na 12 měsíců. [14] Naproti tomu podobný nástroj, **Atlassian JIRA** uvádí cenu za licenci pro 1-10 uživatelů 10\$. [1]

2.5.4. Dokumentace požadavků

Jako návod pro správnou dokumentaci požadavků lze využít známý standard IEEE/ANSI 830-1998. Jedná se o obecný rámec, který může být přizpůsoben a upraven dle konkrétních potřeb organizace. Tento standard může udávat strukturu, pro psaní těchto dokumentů [31]:

1. Úvod
 - a. Účel dokumentu požadavků
 - b. Rozsah výrobku
 - c. Definice, akronyma a zkratky
 - d. Reference
 - e. Přehled zbývajících částí dokumentu
2. Obecný popis
 - a. Funkce produktů
 - b. Charakteristika uživatele
 - c. Hlavní omezení
 - d. Předpoklady a závislosti
3. Specifikace požadavků

V části Specifikace požadavků jsou zahrnuty informace týkající se funkčních a nefunkčních požadavků a rozhraní. Tato část se považuje za nejpodstatnější. Jedná se o variabilní část dokumentu, proto se těžko definuje, jak by tato sekce měla být strukturována. Obecně se zde nachází popis funkčnosti, výkonnosti systému, požadavky na externí rozhraní, databáze, různá omezení, požadavky na kvalitu a další vlastnosti systému.

4. Dodatky

Tabulka 4 zobrazuje jednotlivé uživatele dokumentu požadavků.

Tabulka 4: Uživatelé dokumentu požadavků

Uživatelé dokumentu požadavků	
Zákazníci	Specifikují požadavky, zároveň jsou odpovědní je číst a ověřovat, aby docházelo k uspokojení jejich potřeb. Zákazníci zároveň mohou určovat změny požadavků.
Manažeři/analytici	Používají tento dokument pro plánování procesu vývoje systému.
Vývojáři/řešitelé/architekti	Používají požadavky pro návrh architektury celého systému, či jeho subsystémů a komponent.
Testeři	Používají požadavky na rozvoj a testování celého systému.
Osoby odpovědné za údržbu a kontrolu systému	Používají tento dokument pro pochopení celého systému a vztahů mezi jeho částmi.

Zdroj: upraveno podle [28]

2.6. Role při řízení požadavků v rámci vývoje systému

Obecným pojmem je **stakeholder**, neboli **zúčastněné strany**. Což je skupina nebo i jednotlivec, který je ovlivněn výsledkem projektu přímo či nepřímo, nebo může mít vliv na vytváření či výstup projektu. Mezi zúčastněné strany lze zařadit zákazníky, koncové uživatele, vývojáře, testery, odborníky na kvalitu, dodavatele, právníky, manažery atd. [17]

Pro vysvětlení je možné si představit koncového uživatele, který má zájem o daný požadavek. Zprostředkovatelem může být zákazník, který převezme požadavky od koncového uživatele a zadá je dodavateli. Dodavatel a jeho projektový tým požadavek zpracuje a dodá. V některých případech může být zákazník a uživatel jedna a tatáž osoba. Dodavatelé požadavků a vývojový tým jsou například vývojáři, testeři, vedoucí projektu, podpůrný personál, ale také právníci nebo marketingoví pracovníci. Jsou to zkrátka všichni, kteří se nějakým způsobem podílí na definování požadavků, vývoji výsledného produktu, a kterým má daný výsledek při práci pomoci. [39][17][28] Zúčastněné strany mají vliv v rámci definování požadavků, pomáhají je definovat, vyjasňovat, formulovat.

Koncový uživatel je to osoba, či organizace, která bude výstup projektu užívat. Zároveň to může být i skupina zaměstnanců podniku, kteří jsou do projektu zapojeni. [23]

Zákazník má zájem na realizaci projektu, je jeho investorem nebo zadavatelem. Může se zároveň jednat i o budoucího uživatele systému projektu. [33] Zákazníkem je jednotlivec, skupina, anebo také organizace, kterým výsledný produkt přináší užitek. Jsou jimi všichni účastníci, kteří si o něj požádali, zaplatili za něj, specifikovali ho a následně ho používají nebo dostávají z něj nějaký výstup. Zákazníci jsou podmnožinou zúčastněných stran tzv. stakeholders. [39][17]

Vedoucí projektu nebo také projektový manažer komunikuje se zákazníkem či koncovým uživatelem. Obecně vedoucí projektu musí [23]:

- Identifikovat, kdo je zákazník.
- Identifikovat relativní význam a důležitost. Zaměřením se na důležité zákazníky.
- Určuje, které informace jsou pro zákazníky důležité. Každý zákazník má jiné informační potřeby. Zákazník, který bude například zároveň i plátcem projektu má zájem o to, kolik za daný projekt bude povinný zaplatit. Uživatele systému bude například více zajímat konečná kvalita systému, zda pokrývá jeho potřeby.
- Určuje, jak nejlépe načasovat, zpracovat a jaký způsob předání informací bude optimální. Někteří zákazníci potřebují pravidelné reporty. Tyto reporty mohou být například určeny po dokončení každé fáze projektu.

Zároveň projektový manažer zodpovídá za splnění cílů projektu (splnění trojimperativu), musí mít dostatečnou autoritu k řízení lidí, koordinaci úkolů, rozhodovat i v neočekávaných situacích tak, aby se neodchýlil od cíle. [4]

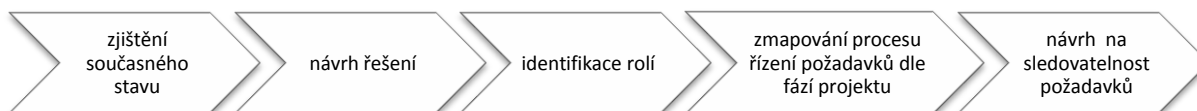
Systémový řešitel a systémový analytik nesou zodpovědnost za výstupy v rámci přijatých uživatelských požadavků. Plní požadavky na sběr, analýzu a technickou řešitelnost požadavků. Připravují matici sledovatelnosti. Projektový manažer provádí kontrolu nad jejich výstupy práce. [13]

Verifikační manažer často tvoří, udržuje, kontroluje, řeší a zlepšuje vytvářené systémy. Spolupracuje se systémovým analytikem a testerem.

Verifikátor (tester) tvoří, udržuje, kontroluje a zlepšuje vytvářené testy pro požadavky. Spolupracuje s vývojářem a systémovým analytikem.

3. PROCES ŘÍZENÍ POŽADAVKŮ VE VYBRANÉ FIRMĚ

Cílem této kapitoly je navrhnout a nově popsat proces řízení požadavků a jednotlivé subprocesy v již existující firmě. Bude obsahovat stručné představení firmy, modelování procesu, identifikace potřebných rolí a konkrétní činnosti, které jsou nutné vykonávat, s jakými vstupy a výstupy. Zároveň se vytvoří návrh modelu pro sledovatelnost požadavků. Postup zpracování práce je znázorněn na obrázku 16.



Obrázek 16: Postup vypracování práce

Zdroj: vlastní zpracování

3.1. Představení firmy

Pro diplomovou práci byla vybrána existující firma, dále označována jako firma XY. Jedná se o technologickou společnost, která vyvíjí, vyrábí a také modernizuje elektronické a záznamové systémy. Její produkty jsou dodávány nejen pro Českou republiku, ale i do několika dalších desítek zemí světa.

3.1.1. Identifikace současného stavu

Prvotním krokem muselo dojít ke zjištění, jaký je současný stav firmy XY v rámci řízení požadavků v projektu, pro který byl využit Hlas zákazníka z metodického rámce Six Sigma. To probíhalo zejména formou osobních rozhovorů se systémovým analytikem. Společně byl identifikován problém, kterému je i tato práce určena.

V první řadě z rozhovorů vyplynulo, že největším problémem je řízení a správa požadavků zejména v dlouhotrvajících projektech. Využívají ji pouze omezeně či intuitivně. Zároveň příslušné činnosti uvnitř procesu nejsou kompletně zmapovány. Tím vzniká problém, že jednotlivé role nemají jasně definované odpovědnosti a činnosti, za které zodpovídají a nemají přehled o tom, jaké procesy a jakým způsobem by se měly v jednotlivých fázích projektu vykonávat. Z toho plyne, že i dohled nad činnostmi je značně nepřehledný a dochází k horší schopnosti kontrolovat výstupy fází projektu. Veškeré výhody plynoucí z řízení požadavků tak mohou být potlačeny.

Dále bylo zjištěno, že vzhledem k určitým podmínkám ve firmě XY, kde nevyužívají žádný komerční nástroj ke správě požadavků, je nutné najít přijatelný způsob, jak požadavky sledovat. Současné využívání tabulek v MS Excel pro firmu XY je nevyhovující. Požadavky jsou zpracovávány do různých druhů tabulek, dokumentů a v rámci složitých a velkých projektů je velice náročné spravovat jejich změny. S touto souvislostí může lehce nastat situace, že se změnou požadavku se zapomene přepsat změna požadavků na ní závislých v jiných tabulkách či dokumentech.

3.1.2. Návrh cílového stavu

Identifikací současného stavu je tedy patrné, že se musí nastavit proces řízení požadavků a navrhnout způsob sledovatelnosti. Byl navrhnout následující postup:

- 1) V první řadě formálně nadefinovat/zmapovat proces pro řízení požadavků při vývoji systému ve fázích projektu.
 - Zmapováním procesu se sníží riziko plynoucí z opomenutí některé činnosti, která měla být vykonána. Zároveň bude patrné, které role jsou za činnost odpovědné. Zpřehlední se tím práce, zejména v rámci realizace složitých a dlouhotrvajících projektů, jejichž úkolem je vyvinout komplexní elektronické systémy.
 - Výhodou zmapování procesu se sníží riziko plynoucí z nesplnění požadavků, a tím se předejde k nadbytečným nákladům, které by mohly vzniknout.
 - Získá se větší přehlednost a podpoří se rychlejší adaptovatelnost nových pracovníků, které by firma přijala.
 - Zmapováním procesu řízení požadavků může být v budoucnosti firma XY lépe připravena na procesní audit.
- 2) Poté navrhnout model systému pro sledovatelnost požadavků.
 - Nejdříve bude nutné zjistit, jaké úrovně sledovatelnosti mezi požadavky je potřebné vykonávat.
 - Následně se definují entity, atributy a bude vytvořen návrh modelu. Jeho funkčnost se ověří pomocí MS Access a vytvoří se základní dotazy, které by mohly podpořit sledovatelnost požadavků.
 - Tímto způsobem by firma mohla využít příležitosti ke snížení případných nákladů, které vyplývají z nákupů licencí komerčních nástrojů pro správu požadavků.

Následující kapitoly popisují postup, jakým způsobem bude dosaženo navrhovaného řízení požadavků.

3.2. Role při řízení a správě požadavků ve vybrané firmě

Prvním krokem před začátkem mapování jednotlivých subprocesů, je identifikovat všechny role, které se do projektu v rámci řízení požadavků zapojují. Patří sem koncový uživatel, zákazník, vedoucí projektu, systémový řešitel, systémový analytik, verifikační manažer, tester a další osoby, které jsou zapojeny a budou označovány obecně jako projektový tým. V následujících odstavcích jsou role popsány.

Koncový uživatel je v tomto případě zejména uživatele, který bude s výsledným systémem pracovat. Při jeho vývoji specifikuje požadavky, přispívá připomínkami, případně se zúčastňuje testování společně se zákazníkem.

Příkladem koncového uživatele může být například armáda, která používá vyvinutý systém, například radar.

Zákazník spolupracuje při specifikování požadavků na vyvíjený systém společně s koncovým uživatelem. Bývá často zadavatelem projektu a předává dokumentaci uživatelských požadavků. Následně se zúčastňuje průběžných kontrol a také akceptačních testů v závěru projektu.

Příkladem zákazníka může být Ministerstvo obrany. Zadá daný projekt firmě XY, může ho zároveň také specifikovat a jednotlivé požadavky upravit či upřesnit. Výsledný systém ale nebude používat přímo, je to zprostředkovatel/zadavatel projektu.

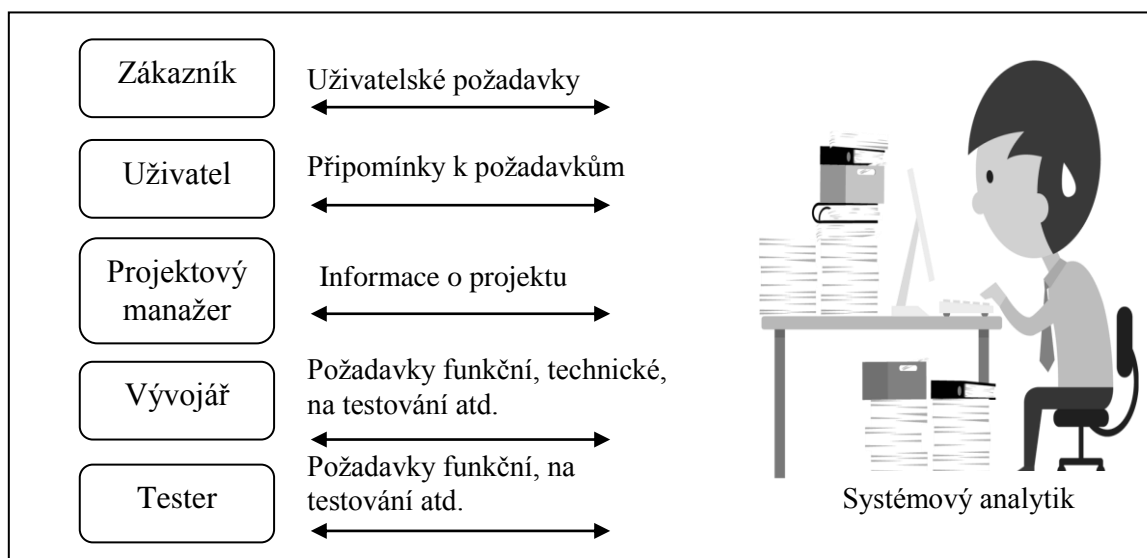
Projektový manažer analyzuje a čte příslušné dokumenty, které specifikují požadavky. Jeho úloha ale nespočívá ve vymýšlení technické stránky. Vytváří časový harmonogram, plánuje ho ze všech potřebných stran (lidé, materiál, čas atd.). S ohledem na procesy řízení požadavků zodpovídá za to, že bude systém vyvíjen v projektu dle požadavků zákazníka, a ty budou splněny na základě jejich ověření (verifikace).

Systémový řešitel zejména analyzuje a pomáhá systémovému analytikovi s rozpadem požadavků na nižší úrovně, pro přehlednější a logický vývoj systému. Všechny požadavky řeší ze systémového hlediska.

Systémový analytik převádí požadavky získané z dokumentů od zákazníka nebo uživatele např. do MS Excel. Následně, pokud je to možné, požadavky dekomponuje na dílčí subsystémy či komponenty. Rozsah dekompozice je závislý na velikosti projektu. Úlohou systémového analytika je i spolupráce s testery. Řeší, jakým způsobem se budou požadavky testovat, spravovat změny atd.

Dle [39] analytik požadavků, neboli v tomto případě obecně systémový analytik, představuje komunikační most mezi zákazníky a projektovým týmem, který znázorňuje

obrázek 17. Nejsou zde uvedeny všechny případy a požadavky, které se řeší, jedná se o ukázkou.



Obrázek 17: Spolupráce systémového analytika s dalšími členy

Zdroj: vlastní zpracování

Verifikační manažer navrhuje HW/SW řešení na základě zadaných požadavků (např. pro jednotlivé subsystémy či komponenty/sestavy). Zároveň pomáhá při tvorbě metodik testů a plánů pro jednotlivé požadavky.

Tester dle dokumentu specifikace požadavků vytváří metodiky testů, a to jak pro celý systém, tak i pro jednotlivé subsystémy či komponenty/sestavy. *Například vymýšlí, který test bude optimální pro ověření napájení systému z elektrické sítě.* Analyzuje již hotové, vytvořené požadavky. Navrhuje a definuje testy, verifikační metody.

3.3. Proces řízení požadavků ve fázích projektu

Nejdříve bylo nutné zjistit, jakým způsobem se člení fáze projektu ve firmě XY. Jsou jimi fáze inicializační, plánovací, realizační a fáze předávání výsledného systému neboli akceptace. Poté, na základě identifikovaných rolí, mohlo dojít k samotnému mapování procesu. To prošlo několika fázemi. Nejdříve, díky teoretickým poznatkům, byly vytvořeny návrhy subprocesů, které by se mohly ve fázích projektu vykonávat. Tyto návrhy byly předloženy během konzultací systémovému analytikovi. Dle jeho připomínek následně probíhaly úpravy, aby se daný proces mohl aplikovat v jejich firmě. Navržené subprocesy jsou zobrazeny v tabulce 5. Na základě konzultací bylo navrženo fáze inicializace a plánování seskupit. Důvodem je shodný subproces, který se v nich využívá. V inicializační fázi probíhá sběr požadavků. V rámci plánování již dochází ke konečné dohodě na uživatelských požadavcích. Sběr požadavků je ukončen a probíhá zde podpis směrné verze požadavků.

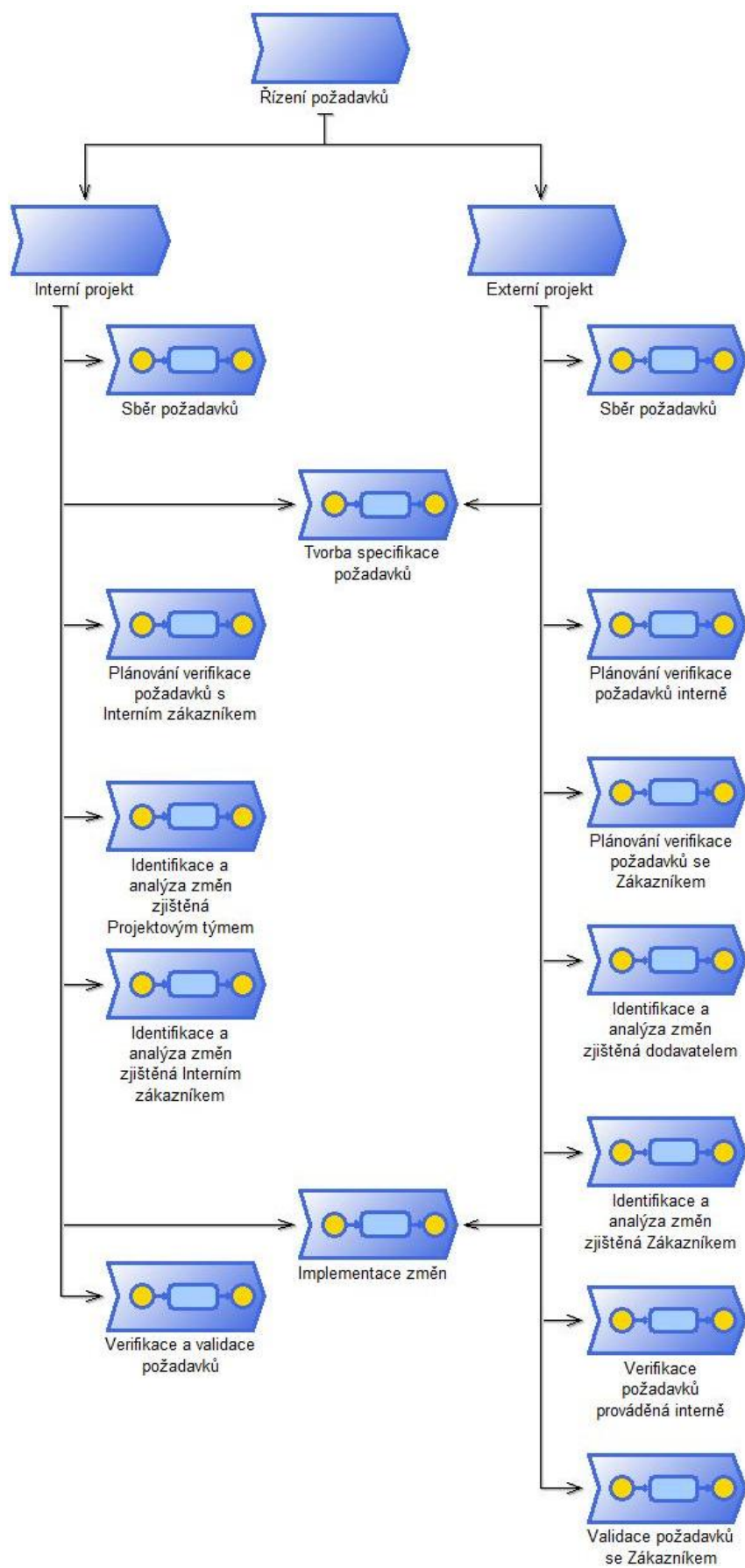
Pokud se jedná o externí projekt, subproces *Plánování verifikace požadavků – interně* předchází *Plánování verifikace se Zákazníkem*. Naopak v případě interního projektu je *Verifikace a validace požadavků – interně* přesunuta do fáze akceptace.

Tabulka 5: Subprocesy řízení požadavků ve fázích projektu

Fáze projektu	Subprocesy řízení požadavků
Inicializace a plánování	<ul style="list-style-type: none"> • Sběr požadavků
Realizace	<ul style="list-style-type: none"> • Tvorba specifikace požadavků • Zajištění sledovatelnosti • Identifikace a analýza změn zjištěná Projektovým týmem • Identifikace a analýza změn zjištěná Zákazníkem • Implementace změn • Plánování verifikace požadavků – interně
Akceptace	<ul style="list-style-type: none"> • Verifikace a validace požadavků – interně • Plánování verifikace se Zákazníkem • Validace požadavků se Zákazníkem

Zdroj: vlastní zpracování

Poté došlo ke zjištění, že by bylo přehlednější, pokud by se řízení požadavků rozdělilo dle druhu projektu. Proto byly jednotlivé subprocesy přepracovány dle následující přehledu znázorněného na obrázku 18. Z něj je patrné, jaké subprocesy se vykonávají, pokud se jedná o interní nebo externí projekt, a které subprocesy jsou společné pro oba druhy. Rozdíl mezi nimi je zejména v zapojení zákazníka do projektu. Pokud se jedná o interní projekt, zákazník zde vystupuje samozřejmě také, ale jedná se o interního zaměstnance (ve vypracovaných subprocesech bude nazýván jako Interní zákazník), který se může i na projektu podílet. Naopak v externích projektech je dodán výsledný systém Zákazníkovi, kterým může být například Ministerstvo obrany a jeho role spočívá zejména v definování Uživatelských požadavků, kontroly a schvalování Specifikace požadavků a účasti při akceptačních testech systému. Zákazník zároveň zjišťuje, jaké další připomínky má koncový uživatel a ty poté formuluje do požadavků na výsledků systém.



Obrázek 18: Přehled subprocesů dle druhu projektu

Zdroj: vlastní zpracování

Na základě tohoto rozdělení byla vypracovaná karta procesů, zobrazená v tabulce 6, která se týká řízení požadavků z hlediska interního a externího projektu.

Tabulka 6: Karta procesu pro interní projekt

Název procesu	Řízení požadavků v projektu
Účel procesu	Řízení a správa požadavků při vývoji systému v rámci projektu.
Vlastník procesu	Systémový analytik
Role v procesu	Projektový manažer, Verifikační manažer, Tester, Systémový analytik, Systémový řešitel, Zákazník, Uživatel
Zákazník	Pokud se jedná o interní projekt, zákazníkem je interní zaměstnanec, nazývaný ve vypracovaných subprocesech jako Interní zákazník. Naopak u externího projektu je to Zákazník.
Vstupy	Dokumentace uživatelských požadavků, Zadávací dokumentace projektu, Šablona specifikace požadavků
Výstupy	Matice sledovatelnosti, Testovací plány, Testovací metodiky, Specifikace požadavků
Regulátory	<ul style="list-style-type: none"> • IEEE 830-1998 – Recommended Practice for Software Requirements Specifications • ISO/IEC/IEEE 29148:2011 – Systems and Software Engineering – Life cycle processes – Requirements Engineering • ISO/IEC 15288 nebo ISO/IEC 12207 Systems and Software Engineering – System life cycle processes • IEEE 1233-1999 – Guide for Developing System Requirements Specification
Metrika	<ul style="list-style-type: none"> • Doba trvání sběru požadavků – v každém projektu jiná. • Počet analyzovaných požadavků v projektu – dle toho se odvíjí čas a náklady na výsledný projekt. • Počet navržených změn. • Počet implementovaných změn. • Podíl realizovaných změn na celkovém počtu identifikovaných změn v %. • Podíl neúspěšně provedených změn na celkovém počtu identifikovaných změn v %. • Celkový objem nákladů na realizaci uživatelských požadavků. • Počet odhalených chyb pomocí testů (pro porovnání testů). • Rychlost objevení chyby (=množství chyb/doba testování).

Zdroj: vlastní zpracování

Následující kapitoly již popisují navržené subprocesy, které byly vytvářeny na základě standardu BPMN. Použitá notace je zobrazena v Příloze A. V navržených subprocesech jsou uvedeny činnosti, které se vykonávají jak v interním, tak externím projektu a zároveň jsou začleněny do fáze projektu, ve které se nacházejí.

3.3.1. Subproces Sběr požadavků

Subproces *Sběr požadavků* byl vytvořen pro fázi inicializace a plánování projektu. Tento subproces je zaměřen na získávání požadavků ze vstupní Zadávací dokumentace projektu a Uživatelských požadavků. Pro lepší přehlednost byl rozdělen pro interní a externí projekty firmy XY. Výstupním dokumentem je Specifikace požadavků.

Pokud se jedná o interní projekt, vystupuje zde Interní zákazník. Jedná se o interního zaměstnance, kterému je výsledek projektu určen. Interní zákazník zde provádí kontrolu nad vytvořenou Specifikací požadavků. Následně ji schvaluje nebo předává k úpravě. Pokud by systém nešlo za daných podmínek a požadavků vytvořit, sběr ukončuje. Dále zde vystupují: Projektový manažer, Systémový analytik, a obecně Projektový tým. Projektový manažer dostává prvotní impulz k zahájení projektu. Následují činnosti, které vykonává Systémový analytik. Sbírá požadavky v rámci firmy, zpracovává je, upravuje, odesílá ke kontrole Internímu zákazníkovi. Projektový tým zde zasílá podklady pro sběr a úpravu požadavků (viz Subproces 1 v Příloze B).

Pokud se jedná o externí projekt, vystupuje zde navíc Uživatel. Zahájení probíhá opět u Projektového manažera, který získává požadavky od Zákazníka. Zákazník poté identifikuje a formuluje své či Uživatelovy požadavky, které následně přeposílá Projektovému manažerovi. Následuje rozpracování Specifikace a odesílá se ke kontrole. Pokud je Specifikace v pořádku, Zákazník ji schválí. Pokud ne, dochází k přepracování či ukončení sběru požadavků (viz Subproces 2 v Příloze B).

V této fázi ještě nejsou požadavky detailně rozpracovány. Jedná se ale o směrnou verzi požadavků, které mají být v projektu vytvořeny. Následně jsou vytvořené dokumenty dále používány v subprocesu Tvorby specifikace požadavků, které jsou již rozpracovány na detailní úrovni.

3.3.2. Subproces Tvorba specifikace požadavků

Pokud je v rámci *Sběru požadavků* specifikace schválena, dochází k rozpracování požadavků na nižší úrovni (subsystémy/komponenty/sestavy). Tento subproces je vykonáván v interním i externím projektu a je začleněn do realizační fáze. Součástí jsou činnosti, které

analyzují a vytvářejí dokument Specifikace požadavků i na nižších úrovních (sestavy/komponenty). Případně, pokud je to potřebné, analyzují se dílčí subsystémy pro subdodavatele. Součástí je zajišťování sledovatelnosti požadavku a Tvorba matice sledovatelnosti. Systémový řešitel zde zejména tvoří specifikaci, kde určuje, kterého subsystému/sestavy/komponenty se dílčí požadavky týkají. Verifikační manažer naproti tomu vyplňuje atributy požadavků, jako je verifikační metoda atd. Výsledkem subprocesu je vytvořená Specifikace požadavků pro nižší úrovně a Matice sledovatelnosti (viz Subproces 3 v Příloze B).

Součástí subprocesu je **zajištění sledovatelnosti**. Zde probíhají činnosti, které mají za úkol vytvořit vazby mezi požadavky, testy a návrhem (designem). Výsledným dokumentem je Matice sledovatelnosti.

3.3.3. Subproces Plánování verifikace požadavků

Subproces Plánování verifikace požadavků je začleněn do fáze realizace a akceptace. Pokud se jedná o externí projekt, subproces *Plánování verifikace požadavků – interně* předchází *Plánování verifikace požadavků se Zákazníkem*.

Pro interní projekt byl navržen subproces *Plánování verifikace požadavků s Interním zákazníkem* (viz Subproces 4 v Příloze B). Tester zde udává pokyn k zahájení plánování verifikace. Následně se do procesu zapojuje Systémový řešitel a Verifikační manažer. Tester vytváří plány a metodiky pro testování. Verifikační manažer je schvaluje a odesílá ke schválení Internímu zákazníkovi.

Pro externí projekt byly vytvořeny subprocesy *Plánování verifikace interně* a *Plánování verifikace se Zákazníkem*. *Plánování verifikace interně* (viz Subproces 5 v Příloze B) je navržen z toho důvodu, že verifikace musí být navržena a schválena ještě před tím, než bude plánována se Zákazníkem. Je obdobná jako u interního projektu. Poté se vykonává *Plánování verifikace se Zákazníkem* (viz Subproces 6 v Příloze B), který schvaluje nejdříve testovací plány, a pokud jsou v pořádku, přechází se k plánování testovacích metodik.

Jelikož se jedná o fáze realizace a akceptace, je žádoucí, aby zde došlo k domluvě na obou stranách. Projekt by se již neměl zrušit, a proto jsou zde vždy vypracovávány nápravná opatření. Výstupem jsou Metodiky testování a Testovací plány. Cílem tohoto subprocesu je vytvořit plán verifikace.

3.3.4. Subproces Identifikace a analýza změn

Tento subproces je více rozdělen na činnosti, které probíhají v rámci interního a externího projektu a probíhá v rámci realizace projektu.

Pokud se jedná o interní projekt, je zde vykonáván subproces *Identifikace a analýza změn zjištěná Interním zákazníkem a Identifikace a analýza změn zjištěná Projektovým týmem*. V subprocesu *Identifikace a analýza změn zjištěná Interním zákazníkem* (viz Subproces 8 v Příloze B), zde vystupuje Interní zákazník jako ten, který požaduje určitou změnu. Následně ji předává Projektovému manažerovi. Systémový analytik identifikuje související požadavky, poté Systémový řešitel a Verifikační manažer analyzují dopady na technické řešení či testování. Pokud je návrh změny přijat či zamítnut jsou oznámení o návrhu na změnu odesílány zpět k Internímu zákazníkovi. Pokud je změna zjištěna Projektovým týmem, je tento postup znázorněn v subprocesu *Identifikace a analýza změn zjištěná Projektovým týmem* (viz Subproces 7 v Příloze B). Dochází zde k identifikaci změn, analyzují se dopady a rizika, a pokud jsou významná, odesílají se ke kontrole a schválení Internímu zákazníkovi. Pokud ne, je návrh na změnu přijat.

U externího projektu jsou navrženy subprocesy identifikace změn obdobné. Rozdělují se na ty, které navrhne Projektový tým (dodavatel), (viz Subproces 9 v Příloze B), a které navrhne Zákazník (viz Subproces 10 v Příloze B). Pokud dojde k návrhu na změnu v rámci Projektového týmu, schvaluje změnu Zákazník, pokud se jedná o změnu navrženou Zákazníkem, schvaluje jí Projektový manažer.

3.3.5. Subproces Implementace změny

Jedná se o subproces, v němž probíhají činnosti, které zajišťují aktualizace, nová řešení v případě přijetí změn. Výstupem jsou nové upravené Testovací plány, Matice sledovatelnosti, Specifikace požadavků a Metodiky testů. Dochází zde k úpravě Specifikace požadavků, jako jsou její texty, atributy, verifikační metodiky. Aktualizace sledovatelnosti probíhá u požadavků na nižších úrovních. Systémový řešitel vytváří nová technická řešení a aktualizuje sledovatelnost mezi požadavky a návrhem (designem). Verifikační manažer upravuje dotčené testovací případy, aktualizuje sledovatelnost v rámci testů požadavků a testovací plány a metodiky. Testovací metodiky a plány se skládají z testovacích případů. Systémový analytik má na starosti aktualizaci sledovatelnosti mezi požadavky a vytváří novou verzi Specifikace požadavků (viz Subproces 11 v Příloze B). Probíhá ve fázi realizace pro interní i externí projekt.

3.3.6. Subproces Verifikace a Validace

Posledními navrženými subprocesy jsou verifikace a validace. Jsou zařazeny do fáze realizace a akceptace projektu. Cílem verifikace je ověření, že výsledek projektu je v souladu se specifikacemi navrženého systému. Validace je kontrola, zda vyvíjený systém splňuje to, co od něj zákazník očekává (provádějí se akceptační testy).

U interního projektu dochází ke kontrole výsledků testů Interním zákazníkem a Testerem. Pokud nejsou v souladu, nastávají nápravná opatření. Poté dochází k ukončení verifikace a validace (viz Subproces 12 v Příloze B). Subproces je zařazen do fáze akceptace.

Pokud se jedná o externí projekt, nejdříve je vykonána *Verifikace požadavků prováděná interně* (viz Subproces 13 v Příloze B) ve fázi realizace. Zde musí dojít k odhalení případných nesrovnalostí ještě před tím, než se zúčastní testování Zákazník. Poté je vykonáván subproces *Validace požadavků se Zákazníkem* (viz Subproces 14 v Příloze B) ve fázi akceptace. Zákazník se účastní testování. Pokud jsou výsledky v souladu, je validace ukončena a výsledek projektu předán Zákazníkovi. Pokud ne, jsou zajištěna nápravná opatření a určuje se další postup.

3.4. Sledovatelnost požadavků ve vybrané firmě

Zmapováním subprocesů v rámci řízení požadavků se mohlo přejít k další části. Dle současného stavu byl objeven problém ve sledovatelnosti požadavků.

Na základě konzultací bylo zjištěno, jaké matice sledovatelnosti firma využívá. Bylo dohodnuto, že tento způsob je zapotřebí zakomponovat i do návrhu databáze. V následujících podkapitolách jsou popsány druhy matic sledovatelnosti, které firma používá a jaký je jejich důvod. Součástí je vždy stručný příklad ve formě tabulky, jak daná matice sledovatelnosti může být sestavena. Nejsou zde ale uvedeny všechny atributy, které je zapotřebí evidovat. Ty budou zmíněny až v kapitole 3.5.1.

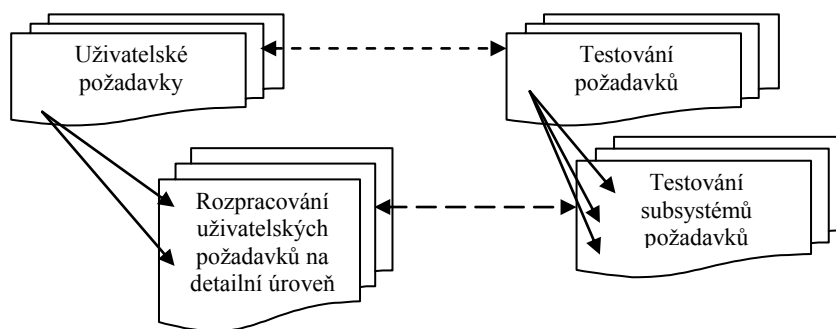
3.4.1. Matice sledovatelnosti

Matice sledovatelnosti (Traceability matrix) se skládají z různých atributů vytvořených pro lepší sledovatelnost, jak již bylo popsáno. Ve firmě XY se využívají následující druhy matic sledovatelnosti:

- a) sledovatelnost mezi požadavky: uživatelskými na celý systém a požadavky na nižších úrovních (např. subsystémů),
- b) sledovatelnost mezi požadavky a testy,

c) sledovatelnost mezi požadavky a návrhem (designem).

Obrázek 19 zobrazuje vztahy mezi požadavky a jejich testováním. Každý uživatelský požadavek systému je dále rozpracováván pro dílčí subsystémy, přičemž může být spojen i s více dílčími požadavky jednotlivých subsystémů atd. Následně každý požadavek by měl být součástí alespoň jednoho testu pro ověření jeho správnosti.



Obrázek 19: Přehled sledovatelnosti mezi požadavky a jejich

Zdroj:vlastní zpracování

a) Sledovatelnost mezi požadavky

Pomocí sledovatelnosti mezi požadavky je možné zajistit a sledovat zejména propojenost mezi požadavky na jednotlivých úrovních (viz. obrázek 19). Konkrétně mezi:

- jednotlivými uživatelskými požadavky a rozpracovanými požadavky pro jednotlivé subsystémy/komponenty/sestavy.

Na základě propojenosti požadavků na různých úrovních je možné sledovat a analyzovat změny požadavků. Následující tabulka 7 zobrazuje příklad matice sledovatelnosti uživatelských požadavků a požadavků na daný subsystém.

Tabulka 7: Příklad matice sledovatelnosti mezi požadavky

ID uživ. požadavku	Text uživ. požadavku	Subsystém	ID požadavku na subsystém	Text požadavku na subsystém
USER_REQ_1	System musí být schopen pracovat při teplotách od 0°C do 40°C.	KOMUNIKACE	SUBSYST-KOM_12	Komunikační systémy umístěné mimo systém musí být schopny pracovat při teplotách od 0°C do 40°C.
USER_REQ_1	System musí být schopen pracovat při teplotách od 0°C do 40°C.	KLIMATIZACE	SUBSYST-KLIM_36	Klimatizace musí zajistit volitelnou teplotu uvnitř systému mezi 18° až 27°C .
USER_REQ_2	System musí komunikovat s prvkem ABX umístěné v síti X.	KOMUNIKACE	SUBSYST-KOM_5	Komunikační subsystém musí komunikovat s prvkem ABX umístěné v síti X.

Zdroj: vlastní zpracování

Je rovněž doporučeno vhodným způsobem zajistit propojení nejvyšší úrovně požadavků (uživatelských) přes úroveň subsystémů až po úroveň sestav/komponent. Tento přístup umožní sledovat změny přes všechny úrovně požadavků. Například se uživatel rozhodne změnit znění požadavku USER_REQ_1 (např. rozmezí teplot, ve kterých musí systém pracovat). Systémový analytik pomocí sledovatelnosti určí všechny dotčené požadavky na nižších úrovních. Výsledkem může být, že změna jednoho uživatelského požadavku ovlivní další požadavek (požadavky), sestavu (y) či komponentu (y), což umožní ohodnotit dopad změny z hlediska času, nákladů, kvality, a to v rámci vývoje dané sestavy či komponenty, tak i celého (sub)systému.

b) Sledovatelnost mezi požadavky a testy

Sledovatelnost mezi jednotlivými požadavky a testy slouží ke sledování, zda jsou požadavky součástí nějakého testovacího případu či scénáře a je tedy zajištěna jejich verifikace. V průběhu vývoje systému vznikají postupně metodiky testů definující krok za krokem, jak budou požadavky prakticky ověřeny. Nejdříve uvnitř firmy, a v konečné fázi se zákazníkem (akceptační testy). Zajistit sledovatelnost mezi požadavky a testy je potřebné mezi:

- jednotlivými uživatelskými požadavky a akceptačními testy (test case),
- jednotlivými požadavky subsystémů/sestav/komponent a vývojovými testy (interními test case).

Pro zajištění sledovatelnosti je potřebné, aby ve firmě XY vznikl takový mechanismus, který umožní přiřadit každý požadavek k jednomu či více testům ověřující jeho splnění a dále sledovat výsledek daného testu (např. úspěšné/neúspěšné/ úspěšné s podmínkami/odloženo). To umožní kontrolovat, zda jsou všechny požadavky pokryty min. jedním testem, včetně jejich stavu a dalších atributů.

V opačném případě by hrozilo, že požadavek nebude ověřen a mohlo tak dojít v konečném důsledku k neověření interně rozpadlých požadavků či nesplnění některých uživatelských požadavků. Zároveň je potřebné zobrazit požadavky, které nejsou součástí (ať už nechtěně, nebo záměrně) žádného testu.

Výhodou zajištění sledovatelnosti je možnost zamezení vzniku reklamací na systém. Jelikož díky testování je zároveň ověřeno, že systém požadavky splňuje a vše je odsouhlaseno a potvrzeno příslušnými osobami (např. interní kontrolou jakosti a státním zástupcem pro ověřování jakosti) na základě schválených testovacích metodik (mezi dodavatelem projektu

a zákazníkem), podle kterých testy proběhly. Příklad takovéto matice sledovatelnosti je uveden v následující tabulce 8.

Tabulka 8: Příklad matice sledovatelnosti mezi požadavky a testy

ID uživ. požadavku	Text požadavku	ID testu	Název testu	Stav testu	Poznámka
USER_REQ_1	Systém musí být schopen pracovat při teplotě od 0°C do 40°C	TEST_AKCEPT_1	Test teplot	Nenaplánován	
USER_REQ_2	Systém musí být schopen pracovat nepřetržitě 8 hodin ze záložního napájení.	TEST_AKCEPT_2	Test elektrické infrastruktury	Neúspěšné	
USER_REQ_3	Systém musí být schopen diagnostikovat otevření dveří.				Nutno požadavek zařadit do testu

Zdroj: vlastní zpracování

Další výhodou je, že jednotlivé testovací případy/scénáře jsou lépe a přehledněji předem plánovány, strukturovány, což umožňuje zlepšit komunikaci jednak uvnitř projektového týmu (např. v rámci interní verifikace požadavků), tak i směrem k zákazníkovi (akceptační testy).

c) Sledovatelnost mezi požadavky a návrhem (designem)

Sledovatelnost mezi jednotlivými požadavky a návrhem (designem) slouží ke sledování, zda a jakým způsobem budou požadavky splněny. V rámci každé specifikace požadavků je potřeba sledovat atribut, kde bude uveden odkaz na místo s popisem návrhu na jeho splnění. Příklad je zobrazen v tabulce 9.

Tabulka 9: Příklad matice sledovatelnosti mezi požadavky a návrhem

ID uživ. požadavku	Text požadavku	Traceabilita design	Poznámka
SUBSYST_1	Systém musí být schopen pracovat při teplotě od 0°C do 40°C	Návrh systému XYZ v 1.0.docx (kapitola č. 3.2 „Teploty“)	
SUBSYST_2	Systém musí být schopen pracovat nepřetržitě 8 hodin ze záložního napájení.		Nutno vymyslet způsob záložního napájení systému a přidat referenci na popis.

Zdroj: vlastní zpracování

Díky tomuto přístupu je možné určit, zda je plán způsobu splnění požadavků určen. Tím se předejde opomenutí, že některý z požadavků nemá přiřazen odkaz na jeho návrh. Rovněž umožní prokázat zákazníkovi plánované plnění požadavků a případně včas zamezit nežádoucímu způsobu implementace ze strany dodavatele.

3.5. Návrh sledovatelnosti požadavků

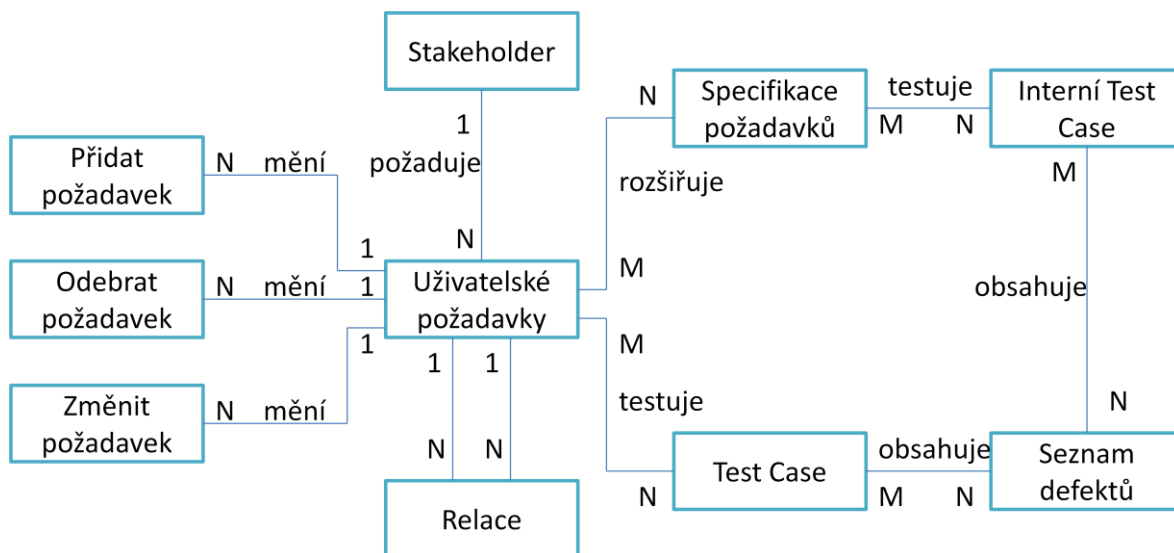
V předchozí etapě bylo zjištěno, jaká sledovatelnost požadavků je potřebná. Na základě konzultací a dohody s firmou XY bylo rozhodnuto vytvořit návrh struktury systému, pomocí kterého by bylo možné požadavky sledovat. Došlo se k závěru, že bude vytvořen návrh pomocí ERD. Ten se následně po úpravách může implementovat do relační databáze, například MS Access, kde se nad jednotlivými relacemi (tabulkami) budou moci provádět operace. Tím by se mohlo zajišťovat propojení mezi požadavky i mezi požadavky a testy na různých úrovních (uživatelské požadavky a akceptační testy, specifikace požadavků (subsystémy – komponenty/sestavy) a interní testy).

Požadavky na návrh systému:

- tvorba návrhu a následné ověření funkčnosti implementací do MS Access,
- tvorba návrhu formou relační databáze,
- možnost přidání, odebrání a změny uživatelských požadavků,
- zajištění sledování požadavků a jejich verzí,
- propojení uživatelských požadavků se specifikací požadavků (se subsystémy/komponentami/sestavami),
- propojení uživatelských požadavků s akceptačními testy,
- propojení specifikací požadavků s interními testy,
- sledovatelnost vztahů mezi uživatelskými požadavky.

3.5.1. Návrh modelu databáze

Následující obrázek 20 zobrazuje vytvoření návrhu modelu pomocí ERD dle metodiky datového modelování. Jsou zde vytvořeny entity, které jsou pro sledovatelnost žádoucí. Následná funkčnost systému bude ověřena pomocí tvorby příkladů dotazů v MS Access.



Obrázek 20: Návrh ERD pro sledovatelnost požadavků

Zdroj: vlastní zpracování

Popis navržených entit a atributů:

Entita **Přidání požadavku** [ID_P (text, "USER_REQ_"0000000;;_), ID_V (číslo), Název (text), Datum (datum), typ (text, uživatelský), Status (text, "přijatý" Or "odložen" Or "analyzovaný" Or "odmítnutý" Or "změněn"), Priorita (text, "neutrální" Or "kritická" Or "velmi kritická", Poznámka (text), Design (text), ID_S (číslo)]

Entita **Změna požadavku** [ID_Z (číslo), ID_P (text, "USER_REQ_"0000000;;_), ID_V (číslo), Název (text), Datum (datum), Typ (text, uživatelský), Status (text, "přijatý" Or "odložen" Or "analyzovaný" Or "odmítnutý" Or "změněn"), Priorita (text, "neutrální" Or "kritická" Or "velmi kritická", Poznámka (text), Design (text), ID_S (číslo)]

Entita **Odstranění požadavku** [ID_Delete (číslo), ID_P (text, "USER_REQ_"0000000;;_), ID_V (číslo), Název (text), Datum (datum), typ (text, uživatelský), Status (text, "přijatý" Or "odložen" Or "analyzovaný" Or "odmítnutý" Or "změněn"), Priorita (text, "neutrální" Or "kritická" Or "velmi kritická", Poznámka (text), Design (text), ID_S (číslo)]

Entita **Uživatelské požadavky** [ID_P (text, "USER_REQ_"0000000;;_), ID_V (číslo), Název (text), Datum (datum), typ (text, uživatelský), Status (text, "přijatý" Or "odložen" Or "analyzovaný" Or "odmítnutý" Or "změněn"), Priorita (text, "neutrální" Or "kritická" Or "velmi kritická", Poznámka (text), Design (text), ID_Z (číslo), ID_Delete (číslo)]

- Přidání požadavku, Změna požadavku, Odstranění požadavku, slouží především jako správa požadavků. Jednotlivé akce může navrhnout Stakeholder a po přijetí

přidání/změny/odstranění je požadavek zapsaný do Uživatelských požadavků, a to až v tom okamžiku, když atribut Status je nastaven na „přijatý“.

Entita **Stakeholder** [ID_S (číslo), Jméno (text), Příjmění (text), Role (text)]

Entita **Relace** [ID_P (text, "USER_REQ_"0000000;;_), ID_V (číslo), ID_P2 (text, "USER_REQ_"0000000;;_), ID_V2 (číslo)]

- Entita Relace slouží k vyjádření vzájemných závislostí mezi Uživatelskými požadavky.

Entita **Test Case** [ID_TC (text, "TEST_AKCEPT_"0000000;;_), ID_V (číslo), Popis (text), Status ("úspěšné" Or "neúspěšné" Or "úspěšné s podmínkami" Or "odloženo"), Lokalita (text), Kritéria úspěšnosti (text), Kritéria neúspěšnosti (text), Poznámka (text), Provedl (text), Datum (datum)]

- Uživatelské požadavky mají své Test Case. Jedná se o akceptační testy, které slouží převážně k prověření splnění požadavku, v závěru předávání projektu zákazníkovi.

Entita **Specifikace požadavků** [ID_SP (text, "SUBSYST_"0000000;;_), ID_V (číslo), Typ (text, "komponenta" Or "subsystém" Or "sestava"), Popis (text)]

- Specifikace Požadavků je entita, ve které je atribut Typ, který může nabývat hodnot „subsystém“, „komponenta“ či „sestava“. Tímto je Uživatelský požadavek v rámci firmy XY dále dekomponován na nižší subsystémy či komponenty někdy též sestavy.

Entita **Interní Test Case** [ID_ITC (text, "TEST_INTER_"0000000;;_), Popis (text), Status (text, "úspěšné" Or "neúspěšné" Or "úspěšné s podmínkami" Or "odloženo"), Poznámka (text), Provedl (text), Lokalita (text), Kritéria úspěšnosti (text), Kritéria neúspěšnosti (text), Datum (datum)]

- Každý subsystém/komponenta/sestava má své Interní Test Case, díky kterým lze sledovat splnění požadavku či případné odhalení nepřiznání testu(ů) k požadavku.

Entita **Seznam defektů** [ID_D (číslo), Popis (text)]

- Zde jsou zaznamenány neshody testů.

3.5.2. Příklady využití databáze pro správu požadavků

V této části bude vytvořeno na ukázkou jen několik dotazů, které by mohly pomoci při práci s požadavky. V každém příkladu je uveden vždy problém, který se bude dotazem řešit, následně je zde ukázán zpracovaný dotaz pomocí SQL a zobrazený výsledek.

Příklad 1: V rámci řízení a správy požadavků je potřebné sledovat subsystémy či komponenty uživatelského požadavku (viz tabulka 10). Výsledek je možné využít pro následné vyhotovování potřebných testů.

Tabulka 10: Příklad 1

Příklad dotazu 1	Vyhledání všech komponent a subsystémů pro uživatelský požadavek USER_REQ_0000001.												
<pre>SELECT PožadavkySpecifikace.ID_P, PožadavkySpecifikace.ID_V, PozadavkySpecifikace.ID_SP, [Specifikace požadavků].Typ FROM [Uživatelské požadavky] INNER JOIN ([Specifikace požadavků] INNER JOIN PozadavkySpecifikace ON [Specifikace požadavků].ID_SP = PožadavkySpecifikace.ID_SP) ON ([Uživatelské požadavky].ID_V = PožadavkySpecifikace.ID_V) AND ([Uživatelské požadavky].ID_P = PožadavkySpecifikace.ID_P) GROUP BY PožadavkySpecifikace.ID_P, PožadavkySpecifikace.ID_V, PozadavkySpecifikace.ID_SP, [Specifikace požadavků].Typ HAVING (((PozadavkySpecifikace.ID_P)="0000001"));</pre>													
Výsledek:	<table border="1"> <thead> <tr> <th>ID_P</th> <th>ID_V</th> <th>ID_SP</th> <th>Typ</th> </tr> </thead> <tbody> <tr> <td>USER_REQ_0000001</td> <td></td> <td>1 SUBSYST_0000001</td> <td>komponenta</td> </tr> <tr> <td>USER_REQ_0000001</td> <td></td> <td>1 SUBSYST_0000002</td> <td>subsystém</td> </tr> </tbody> </table>	ID_P	ID_V	ID_SP	Typ	USER_REQ_0000001		1 SUBSYST_0000001	komponenta	USER_REQ_0000001		1 SUBSYST_0000002	subsystém
ID_P	ID_V	ID_SP	Typ										
USER_REQ_0000001		1 SUBSYST_0000001	komponenta										
USER_REQ_0000001		1 SUBSYST_0000002	subsystém										

Zdroj: vlastní zpracování

Příklad 2: Je potřebné zjistit kolik testů na uživatelský požadavek je zapotřebí vykonat (viz tabulka 11). Tato informace se příkládá k náročnosti jednotlivých požadavků, než dojde k závěrečnému akceptování.

Tabulka 11: Příklad 2

Příklad dotazu 2	Vyhledání počtu testů, které se musí provést pro každý uživatelský požadavek.								
<pre>SELECT [Uživatelské požadavky].ID_P, Count(PozadavkyTestCase.ID_TC) AS CountOfID_TC FROM [Uživatelské požadavky] INNER JOIN PozadavkyTestCase ON ([Uživatelské požadavky].ID_V = PozadavkyTestCase.ID_V) AND ([Uživatelské požadavky].ID_P = PozadavkyTestCase.ID_P) GROUP BY [Uživatelské požadavky].ID_P, PozadavkyTestCase.ID_P;</pre>									
Výsledek:	<table border="1"> <thead> <tr> <th>ID_P</th> <th>CountOfID_TC</th> </tr> </thead> <tbody> <tr> <td>USER_REQ_0000001</td> <td>1</td> </tr> <tr> <td>USER_REQ_0000002</td> <td>1</td> </tr> <tr> <td>USER_REQ_0000003</td> <td>2</td> </tr> </tbody> </table>	ID_P	CountOfID_TC	USER_REQ_0000001	1	USER_REQ_0000002	1	USER_REQ_0000003	2
ID_P	CountOfID_TC								
USER_REQ_0000001	1								
USER_REQ_0000002	1								
USER_REQ_0000003	2								

Zdroj: vlastní zpracování

Příklad 3: Součástí správy požadavků je také sledování propojení požadavku s jinými (viz tabulka 12). Tato informace slouží především, pokud někdo ze zúčastněných stran navrhne změnu. Vyhledáním všech závislých požadavků může zpřehlednit, jaký vliv bude mít změna na další požadavky atd. Platí, že čím vyšší závislost jednoho požadavku na jiném, tím narůstá cena výsledného projektu. Z tabulky 12 je patrné, že uživatelský požadavek USER_REQ_0000002 je závislý ve verzi 1 na další dva požadavky.

Tabulka 12: Příklad 3

Příklad dotazu 3	Vyhledání vztahu uživatelského požadavku USER_REQ_0000002 s jinými požadavky.												
<pre>SELECT Relace.ID_P, Relace.ID_P2, Relace.ID_V, Relace.ID_V2 FROM [Uživatelské požadavky] INNER JOIN Relace ON ([Uživatelské požadavky].ID_V = Relace.ID_V) AND ([Uživatelské požadavky].ID_P = Relace.ID_P) AND ([Uživatelské požadavky].ID_V = Relace.ID_V) AND ([Uživatelské požadavky].ID_P = Relace.ID_P) GROUP BY Relace.ID_P, Relace.ID_P2, Relace.ID_V, Relace.ID_V2 HAVING (((Relace.ID_P)="0000002")) OR (((Relace.ID_P2)="0000002"));</pre>													
Výsledek:	<table border="1"> <thead> <tr> <th>ID_P</th> <th>ID_P2</th> <th>ID_V</th> <th>ID_V2</th> </tr> </thead> <tbody> <tr> <td>USER_REQ_0000001</td> <td>USER_REQ_0000002</td> <td>1</td> <td>1</td> </tr> <tr> <td>USER_REQ_0000002</td> <td>USER_REQ_0000003</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	ID_P	ID_P2	ID_V	ID_V2	USER_REQ_0000001	USER_REQ_0000002	1	1	USER_REQ_0000002	USER_REQ_0000003	1	1
ID_P	ID_P2	ID_V	ID_V2										
USER_REQ_0000001	USER_REQ_0000002	1	1										
USER_REQ_0000002	USER_REQ_0000003	1	1										

Zdroj: vlastní zpracování

Příklad 4: Je důležité ošetřit, aby každý uživatelský požadavek měl přiřazen svůj akceptační test. To bylo provedeno přidělením primárních klíčů do vztahové relační tabulky PožadavkyTestCase. Všechny atributy musí být vypsány, bez toho nebude záznam proveden (viz tabulka 13).

Tabulka 13: Příklad 4

Příklad 4	Ošetření, aby každý požadavek měl přiřazen akceptační test.															
Výsledek:	<table border="1"> <thead> <tr> <th>ID_P</th> <th>ID_V</th> <th>ID_TC</th> </tr> </thead> <tbody> <tr> <td>USER_REQ_0000001</td> <td>4</td> <td>TEST_AKCEPT_0000001</td> </tr> <tr> <td>USER_REQ_0000002</td> <td>1</td> <td>TEST_AKCEPT_0000001</td> </tr> <tr> <td>USER_REQ_0000003</td> <td>1</td> <td>TEST_AKCEPT_0000001</td> </tr> <tr> <td>USER_REQ_0000003</td> <td>1</td> <td>TEST_AKCEPT_0000003</td> </tr> </tbody> </table>	ID_P	ID_V	ID_TC	USER_REQ_0000001	4	TEST_AKCEPT_0000001	USER_REQ_0000002	1	TEST_AKCEPT_0000001	USER_REQ_0000003	1	TEST_AKCEPT_0000001	USER_REQ_0000003	1	TEST_AKCEPT_0000003
ID_P	ID_V	ID_TC														
USER_REQ_0000001	4	TEST_AKCEPT_0000001														
USER_REQ_0000002	1	TEST_AKCEPT_0000001														
USER_REQ_0000003	1	TEST_AKCEPT_0000001														
USER_REQ_0000003	1	TEST_AKCEPT_0000003														

Zdroj: vlastní zpracování

Příklad 5: V opačném znění příkladu 3, lze zjistit, který požadavek není vázán na jiný (viz tabulka 14). Tím lze provést kontrolu, pokud víme, že právě požadavek USER_REQ_0000003 by měl být svázán i s jinými. Předejde se k pozdějším chybám při implementaci či akceptaci systému.

Tabulka 14: Příklad 5

Příklad dotazu 5	Vyhledání uživatelského požadavku, který není svázán se žádným jiným uživatelským požadavkem.				
<pre>SELECT Relace.ID_P2, Relace.ID_P FROM Relace GROUP BY Relace.ID_P2, Relace.ID_P HAVING (((Relace.ID_P2) Is Null));</pre>					
Výsledek:	<table border="1"> <thead> <tr> <th>ID_P2</th> <th>ID_P</th> </tr> </thead> <tbody> <tr> <td></td> <td>USER_REQ_0000003</td> </tr> </tbody> </table>	ID_P2	ID_P		USER_REQ_0000003
ID_P2	ID_P				
	USER_REQ_0000003				

Zdroj: vlastní zpracování

Příklad 6: Součástí sledovatelnosti při správě požadavků může být i přiřazení testů k požadavkům (viz tabulka 15). Zároveň se může využít i výpis jejich statusu, data a počtu změn, kterých se na daném požadavku provedlo.

Tabulka 15: Příklad 6

Příklad dotazu 6	Vyhledání uživatelského požadavku a jejich akceptačních testů a stavů a ID změny, která se provedla																									
<pre>SELECT PozadavkyTestCase.ID_P, PozadavkyTestCase.ID_TC, [Uživatelské požadavky].Status, [Uživatelské požadavky].Datum, [Uživatelské požadavky].ID_Z FROM [Uživatelské požadavky] INNER JOIN PozadavkyTestCase ON ([Uživatelské požadavky].ID_V = PozadavkyTestCase.ID_V) AND ([Uživatelské požadavky].ID_P = PozadavkyTestCase.ID_P);</pre>																										
Výsledek:	<table border="1"> <thead> <tr> <th>ID_P</th> <th>ID_TC</th> <th>Status</th> <th>Datum</th> <th>ID_Z</th> </tr> </thead> <tbody> <tr> <td>USER_REQ_0000001</td> <td>TEST_AKCEPT_0000001</td> <td>analyzovaný</td> <td>12.8.2017</td> <td>4</td> </tr> <tr> <td>USER_REQ_0000002</td> <td>TEST_AKCEPT_0000001</td> <td>přijatý</td> <td>12.8.2017</td> <td>2</td> </tr> <tr> <td>USER_REQ_0000003</td> <td>TEST_AKCEPT_0000001</td> <td>přijatý</td> <td>18.3.2017</td> <td>3</td> </tr> <tr> <td>USER_REQ_0000003</td> <td>TEST_AKCEPT_0000003</td> <td>přijatý</td> <td>18.3.2017</td> <td>3</td> </tr> </tbody> </table>	ID_P	ID_TC	Status	Datum	ID_Z	USER_REQ_0000001	TEST_AKCEPT_0000001	analyzovaný	12.8.2017	4	USER_REQ_0000002	TEST_AKCEPT_0000001	přijatý	12.8.2017	2	USER_REQ_0000003	TEST_AKCEPT_0000001	přijatý	18.3.2017	3	USER_REQ_0000003	TEST_AKCEPT_0000003	přijatý	18.3.2017	3
ID_P	ID_TC	Status	Datum	ID_Z																						
USER_REQ_0000001	TEST_AKCEPT_0000001	analyzovaný	12.8.2017	4																						
USER_REQ_0000002	TEST_AKCEPT_0000001	přijatý	12.8.2017	2																						
USER_REQ_0000003	TEST_AKCEPT_0000001	přijatý	18.3.2017	3																						
USER_REQ_0000003	TEST_AKCEPT_0000003	přijatý	18.3.2017	3																						

Zdroj: vlastní zpracování

3.6. Závěrečné shrnutí

Ve vybrané firmě XY byl identifikován problém týkající se řízení požadavků v rámci vývoje systému. Při zadání především dlouhodobého (mezi)národního projektu nastávají komplikace, jak požadavky spravovat. Velkým počtem zpracovávaných požadavků se staly činnosti dosti nepřehledné a nebyla k nim jasně přiřazena odpovědná osoba. V nejhorsích případech to způsobovalo nesrovnalosti při konečné verifikaci. Nejdříve, na základě konzultací, byl objasněn přibližný způsob fungování řízení požadavků ve vybrané firmě. Následně se vytvořil obecný návrh dle teoretických poznatků. Navržený proces se poté upravoval a rozpracovával na detailnější úrovni tak, aby ho bylo možné využívat ve firmě XY. Rozčlenily se subprocessy vytvářené během interního a externího projektu. Zároveň se přidělily do fází projektu, ve které jsou vykonávány. Činnosti, které byly vytvořeny, se přiřadily ke konkrétní roli, aby bylo jasné, kdo je za ni odpovědný. Následně se přistoupilo k řešení problému se správou požadavků. Byl vytvořen návrh databázového modelu, ve kterém je možné požadavky přidávat, odebírat, měnit a vzájemně sledovat jejich souvislosti. Požadavky jsou zároveň spojeny s testy, kterými se ověří jejich správnost. Přehled o vytvořeném řešení je zobrazen na obrázku 21.



Obrázek 21: Přehled řešení problému

Zdroj: vlastní zpracování

Pro identifikaci současného stavu byl použit Hlas zákazníka z metodického rámce Six Sigma, který probíhal formou rozhovorů. V této práci byl využit standard BPMN pro návrh procesu řízení požadavků. Zároveň jsou zde uvedeny ISO normy, které slouží jako regulátory procesu. Následně zde byly využity i praktiky, které využívá firma XY. Byly to například atributy a matice sledovatelnosti požadavků, které byly dále zakomponovány do návrhu sledovatelnosti požadavků a pomocí praktik firmy probíhaly úpravy subprocessů. Pro návrh databáze byla použita metodika datového modelování s pomocí diagramu ERD. Rozpad uživatelských požadavků na dílčí subsystémy byl vytvořen na základě V-modelu. Zmapování a navrhnutí procesu řízení požadavků probíhalo způsobem postupného řešení.

ZÁVĚR

Tato práce byla zaměřena na problematiku týkající se řízení požadavků v souvislosti s vývojem systému v rámci projektu ve vybrané firmě. Na základě použití Hlasu zákazníka z metodického rámce Six Sigma, který probíhal formou rozhovorů, bylo rozhodnuto, tento problém vyřešit, navržením procesu řízení požadavků v jednotlivých fázích projektu. Zároveň určit, jaký druh sledovatelnosti (traceability) požadavků je nutné provádět. Poté vyplynulo, ve spojení se správou požadavků, navrhnout model sledovatelnosti. Pro splnění cíle navrhnutí procesu řízení požadavků byl využit standard BPMN a ISO normy, které v procesu vystupují jako regulátory. Následně dle praktik firmy byl navržen model pro správu požadavků a byly dle nich upraveny i subprocesy. Dle metodiky datového modelování byl vytvořen návrh sledovatelnosti požadavků.

V úvodní části byly vysvětleny základní pojmy projektového managementu a systémového inženýrství. Tyto kapitoly sloužily jako obecné přiblížení problematiky, aby si čtenář lépe začlenil řízení požadavků (Requirement Engineering) do souvislostí s vývojem systému (HW i SW) v rámci projektu. Dále zde bylo vysvětleno, co je to požadavek a jaké jsou jeho vlastnosti. Byla přiblížena problematika vývoje a správy požadavků.

Na základě těchto teoretických znalostí a konzultací ve vybrané firmě, byl modelován proces řízení požadavků a jeho subprocesů. Subprocesy byly přiřazeny do jednotlivých fází projektu. Následně pro lepší přehlednost, byly rozděleny i dle typu projektu na interní a externí. Součástí práce je i návrh na sledovatelnost požadavků. V první řadě bylo vyřešeno, jaké matice sledovatelnosti jsou vhodné využívat. Jsou jimi sledovatelnost mezi požadavky jako takovými, mezi požadavky a testy, mezi požadavky a návrhem (design) systému. Dále byl vytvořen návrh modelu sledovatelnosti požadavků pomocí ER diagramu. Ten byl implementován do programu MS Access pro ověření jeho funkčnosti. Byly zde uvedeny příklady vytvořené pomocí dotazů, které by mohly napomoci při správě požadavků.

Přínos práce je zejména ve vytvoření procesu řízení požadavků pro vybranou firmu. Ten má pomoci pro vytvoření interní směrnice, a tím následně zlepšit realizaci a koordinaci projektů i pro (mezi)národní společnosti při vývoji systémů. Zároveň bylo smyslem vysvětlit a přiblížit problematiku řízení požadavků. Důvodem je především to, že během projektu mohou nastat chyby související s nedostatečnou specifikací a správou požadavků. Pokud by se chyby odhalily již v rané fázi projektu, jejich úpravy by stály až stonásobně méně, než kdyby byly objeveny až v hotovém systému. Což se samozřejmě odráží na čase předání, kvalitě, či rozpočtu projektu.

POUŽITÁ LITERATURA

- [1] Atlassian. *JIRA Software* [online]. c2017 [cit. 29. 3. 2017]. Dostupné z: <https://www.atlassian.com/software/jira>
- [2] AYBŮKE, Aurum a Wohlin CLAES. *Engineering and Managing Software Requirements*. Berlin: Springer-Verlag, 2005. 478 s. ISBN 978-3-540-25043-2.
- [3] BASHIR, Muhamamd F. a QADIR, Muhammad A. *Traceability Techniques: A Critical Study*. Faculty of Engineering and Sciences. Pakistan. [online]. 2006 [cit. 28. 1. 2017]. Dostupné z: http://www.ptidej.net/courses/log3410/summer11/Lectures/Article_7.pdf
- [4] BOČKOVÁ, HRAZDILOVÁ, Kateřina. *Projektové řízení: Učebnice*. Martin Koláček – E-knihy jedou, 2016. 470 s. ISBN: 978-80-7512-431-9.
- [5] CDC. *Unified Process Practice Guide. Requirements Traceability*. [online]. 30. 11. 2006 [cit. 20. 2. 2017]. Dostupné z: https://www2.cdc.gov/cdcup/library/practices_guides/CDC_UP_Requirements_Traceability_Practice_Guide.pdf
- [6] DOLEŽAL, Jan, Pavel MÁCHAL a Bronislav LACKO. *Projektový management podle IPMA*. Praha: Grada Publishing, 2012. 528 s. ISBN: 978-80-247-4275-5.
- [7] DOLEŽAL, Jan a kol. *Projektový management*. Praha: Grada Publishing, 2016. 424 s. ISBN 978-80-247-5620-2.
- [8] ECCAM. *Efektivnější správa požadavků*. [online]. c2017 [cit. 21. 2. 2017]. Dostupné z: http://www.eccam.com/video/effective_rm.cz.swf
- [9] FIALA, Petr. *Projektové řízení – modely, metody, analýzy*. Praha: Professional Publishing, 2004. 276 s. ISBN: 80-86419-24-X.
- [10] FREY Pete, *10 Requirement Elicitation Technique for Business Analysts*. Slideshare. [online]. c2014 [cit. 5. 2. 2017]. Dostupné z: <http://www.slideshare.net/PeteFrey1/10-requirement-elicitation-techniques-for-b-as>
- [11] GIACHETTI, Ronald E. *Design of Enterprise Systems: Theory, Architecture, and Methods*. New York: 2010. 429 s. ISBN 978-1-4398-8289-4.
- [12] HOOD, Colin, WIEDEMANN, Simon, FICHTINGER Stefan and PAUTZ Urte. *Requirements Management: The Interface Between Requirements Development and All Other Systems Engineering Processes*. Berlin: 2008. 275 s. ISBN 978-3-540-47689-4.

- [13] CHEMUTURI, Murali. *Requirements Engineering and Management for Software Development Projects*. New York: 2013. 266 s. ISBN 978-1-4614-5376-5.
- [14] IBM Rational DOORS. Rational DOORS. [online]. c2017 [cit. 29. 3. 2017]. Dostupné z: <http://www-03.ibm.com/software/products/en/ratidoor>
- [15] IIBA. International Institute of Business Analysis. *Příručka k souboru znalostí pro analýzu byznys požadavků. Verze 2.0 Rámec*. [online]. c2014 [cit. 12. 1. 2017]. Dostupné z: http://itsmf.cz/wp-content/uploads/2015/05/BABOK_Framework_CZ_V1_1.pdf
- [16] ISTQB EXAM CERTIFICATION. *What is V-model – advantages, disadvantages and when to use it?* [online]. c2017 [cit. 8. 2. 2017]. Dostupné z: <http://istqbexamcertification.com/what-is-v-model-advantages-disadvantages-and-when-to-use-it/>
- [17] KASE, Tim. *Practical insight into CMMI*. Norwood: 2008. 472 s. ISBN 978-1-59693-275-3.
- [18] KOMZÁK, Tomáš. *Řízení IT projektů pro úplné začátečníky*. Brno: Albatros Media, 2013. 205 s. ISBN 978-80-251-3791-8.
- [19] KOSSIAKOFF, Alexander, William N. SWEET, Samuel J. SEYMOUR and Steven M. BIEMER. *Systems Engineering: Principles and Practice*. New Jersey: John Wiley & Sons, 2003. 528 s. ISBN: 978-0-470-40548-2.
- [20] LOUCOPOULOS, Peri and KARAKOSTAS Vassiolios. *System Requirements Engineering*. [online]. c2015 [cit. 5. 2. 2017]. McGraw Hill. Dostupné z: https://www.researchgate.net/publication/233819114_System_Requirements_Engineering
- [21] ManagementMania. *Plán projektu*. [online]. c2017 [cit. 15. 2. 2017] Dostupné z: <https://managementmania.com/cs/plan-projektu>
- [22] NASA. *Systems Engineering Handbook*. [online]. 2007 [cit. 29. 12. 2016] Washington, D.C. NASA/SP-2007-6105 Rev1. ISBN: 978-0-16-079747-7. Dostupné z: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20080008301.pdf>
- [23] NEWTON, Richard. *Úspěšný projektový manažer*. Praha: Grada Publishing, 2008. 264 s. ISBN: 978-80-247-2544-4.

- [24] Qualität & Informatik. Requirements Management Guidebook. *Avionics Software Engineering. Software Requirements Management Working Group*. [online]. c2013-2014 [cit. 10. 2. 2017]. Dostupné z: http://itq.ch/pdf/Req_Eng_GuideBook7_Avionics1.pdf
- [25] ROSANNE, Lim. *Project Management. Top 10 main cause sof project failure*. [online]. c2016 [cit. 15. 2. 2017]. Dostupné z: <https://project-management.com/top-10-main-causes-of-project-failure/>
- [26] SAGE, Andrew P. a ROUSE William B. *Handbook of Systems Engineering and Management*. Secong Edition. New Jersey: 2009. 1504 s. ISBN 978-0-470-08353-6.
- [27] SKALICKÝ Jiří, Milan JERMÁŘ a Jaroslav SVOBODA. *Projektový management a potřebné kompetence*. Plzeň: Typos, 2010. 406 s. ISBN 978-80-7043-975-3.
- [28] SOMMERVILLE Ian. *Software Engineering 8*. Harlow: Pearson Education Limited, 2007. 840 s. ISBN: 978-0-321-31379-9.
- [29] SOMMERVILLE, I. *Softwarové inženýrství*. Brno: Computer Press, 2013. 680 s. ISBN 978-80-251-3826-7.
- [30] SOMMERVILLE, Ian. *Software Engineering, 7th edition. Requirements Engineering Processes*. [online]. c2004 [cit. 6. 2. 2017]. Dostupné z: <http://xyuan.myweb.cs.uwindsor.ca/311/Lec07.pdf>
- [31] SOMMERVILLE, Ian. *Systems, software and technology*. IEEE Standard. [online]. c2008 [cit. 7. 2. 2017]. Dostupné z: <https://ifs.host.cs.st-andrews.ac.uk/Books/SE9/Web/Requirements/IEEE-standard.html>
- [32] SQA. *ISO 9126 Software Quality Characteristics*. [online]. [cit. 9. 2. 2017]. Dostupné z: <http://www.sqa.net/iso9126.html>
- [33] SVOZILOVÁ, Alena. *Projektový management*. Praha: Grada Publishing, 2011. 392 s. ISBN 978-80-247-3611-2.
- [34] SVOZILOVÁ, Alena. *Projektový management: Systémový přístup k řízení projektů*. Praha: Grada Publishing, 2016. 424 s. ISBN: 978-80-271-9473-5.
- [35] TECH WHIRL. *Writing Software Requirements*. [online]. c2010 [cit. 8. 2. 2017]. Dostupné z: <https://techwhirl.com/writing-software-requirements-specifications/>

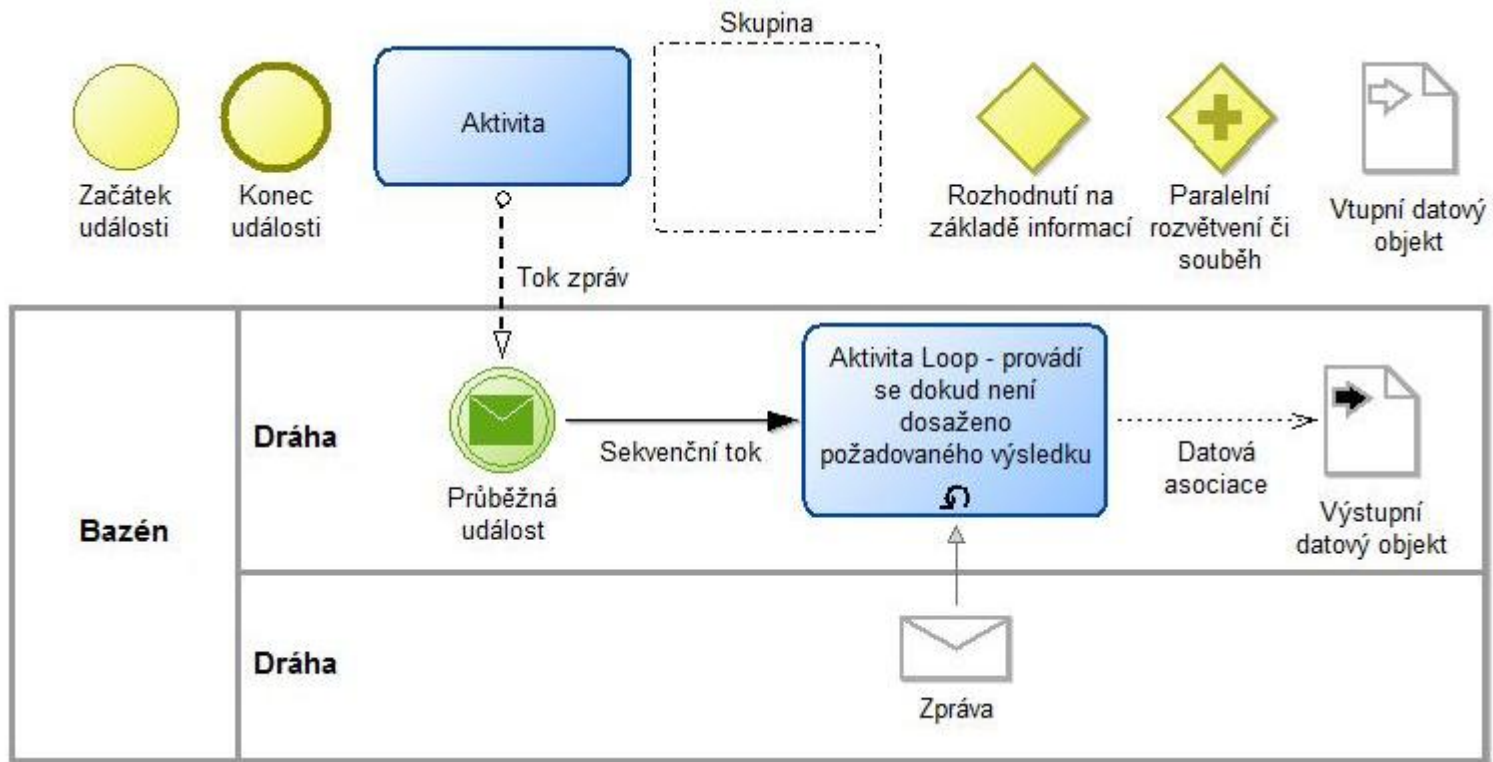
- [36] University of Virginia. Supplementary text prepared by defense acquisition. *Systems Engineering Fundamentals*. [online] 2001 [cit. 8. 2. 2017]. Dostupné z: https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-885j-aircraft-systems-engineering-fall-2005/readings/sefguide_01_01.pdf
- [37] WIEGERS, K., BEATTY, J. *Software Requirements*. Microsoft Press, 2013. 672 s. ISBN 978-0-7356-7966-5.
- [38] WIEGERS, Karl E. *More About Software Requirements: Thorny Issues and Practical Advice*. Washington: Microsoft Press, 2005. 224 s. ISBN 0-73562-267-1.
- [39] WIEGERS, Karl. *Požadavky na software*. Brno: Computer Press, 2008. 448 s. ISBN 978-80-251-1877-1.
- [40] WINKLER, Stefan a PILGRIM von Jens. *Software and System Modeling. A survey of traceability in requirements engineering and model-driven development*. Springer-Verlag: 2009. DOI 10.1007/s10270-009-0145-0.
- [41] YIN, Li, JUAN, Li, YE, Yang and MINGSHU, Li. *Requirement-Centric Traceability for Change Impact Analysis: A Case Study*. Springer-Verlag Berlin: 2008. DOI: 10.1007/978-3-540-79588-9_10.
- [42] YOUNG Ralph R. *The Requirements Engineering Handbook*. London: 2004. 263 s. ISBN: 1-58053-266-7.

SEZNAM PŘÍLOH

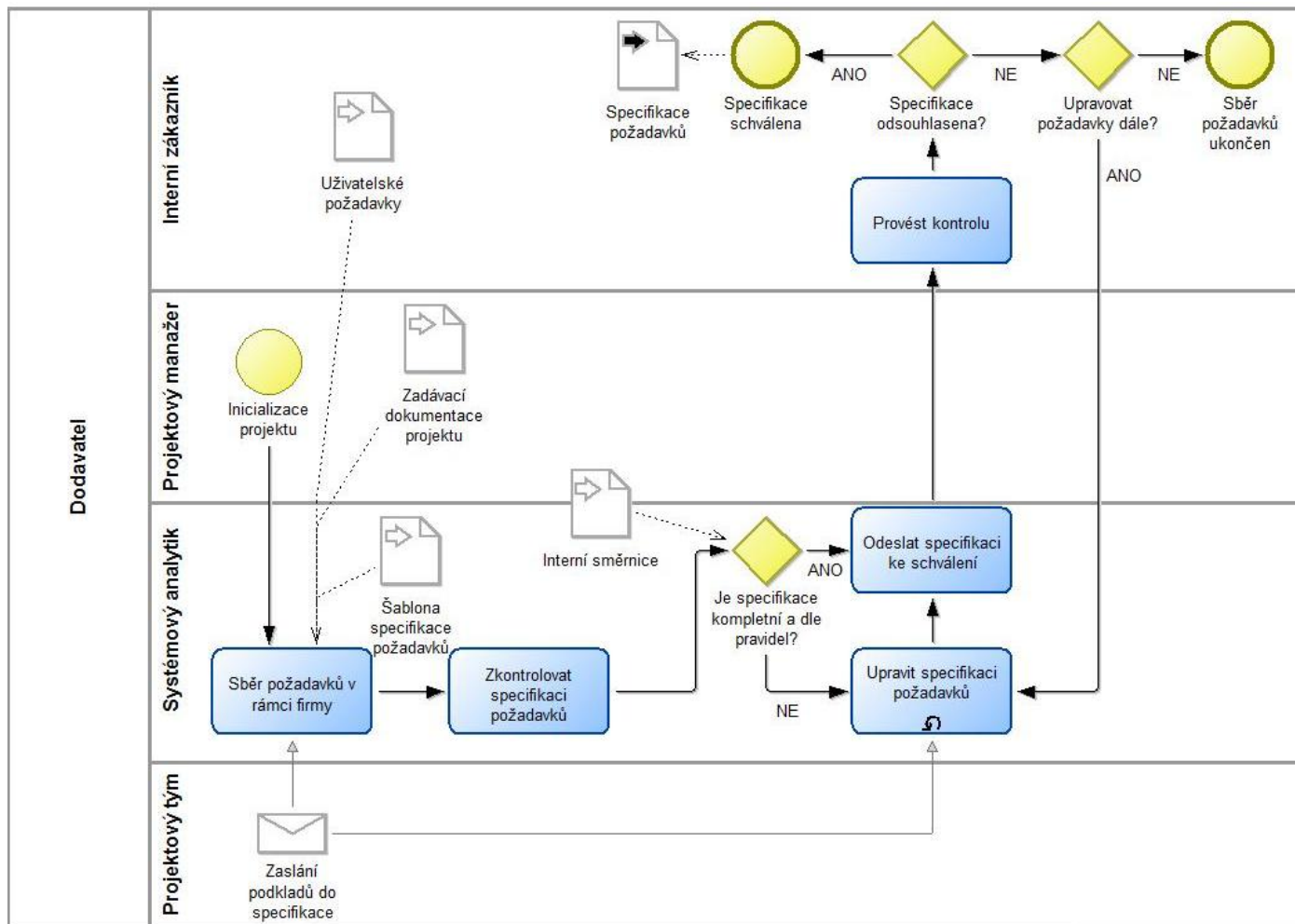
Příloha A Přehled použité notace pro modelování dle BPMN

Příloha B Modelování procesu řízení požadavků dle BPMN

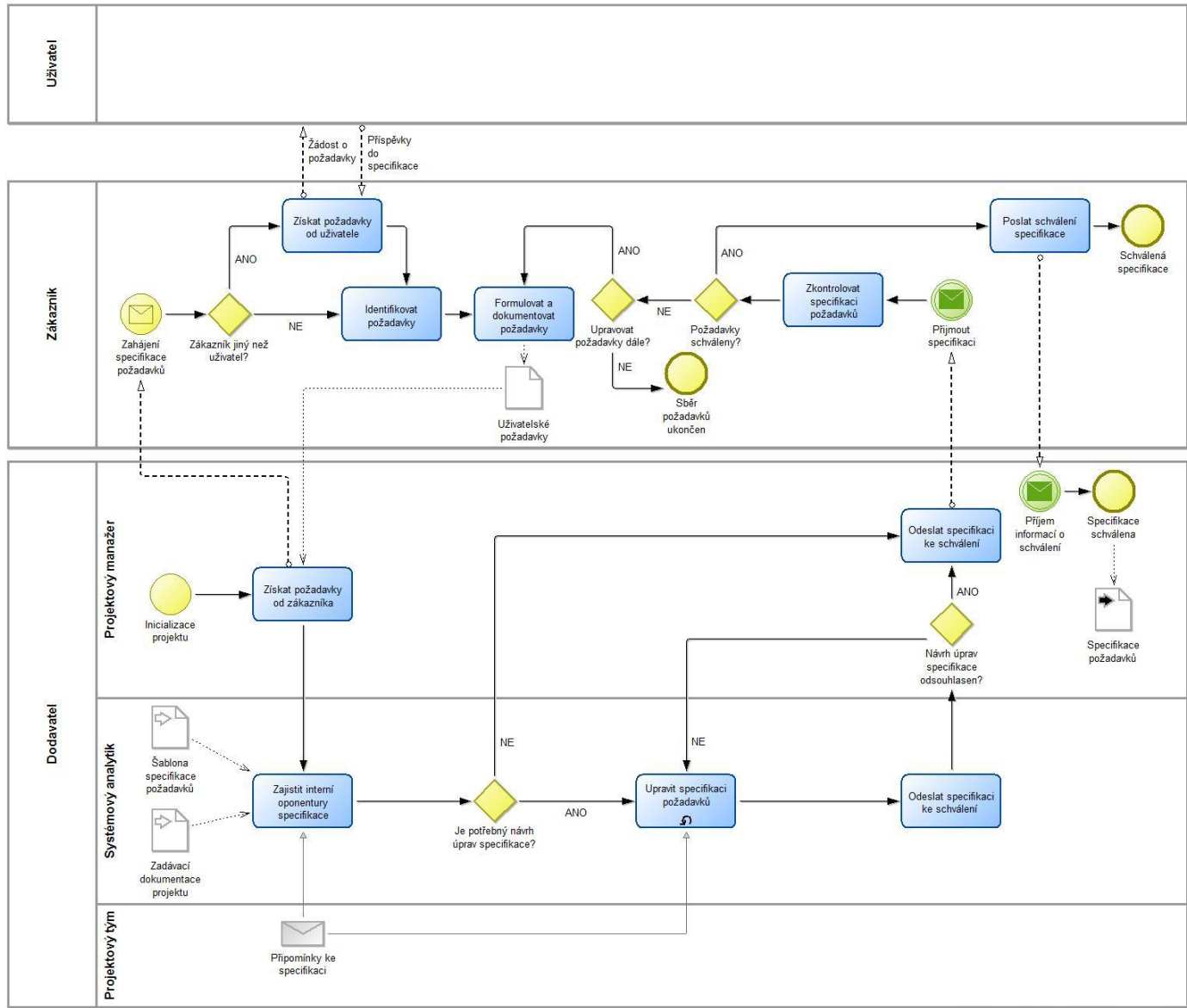
Příloha A



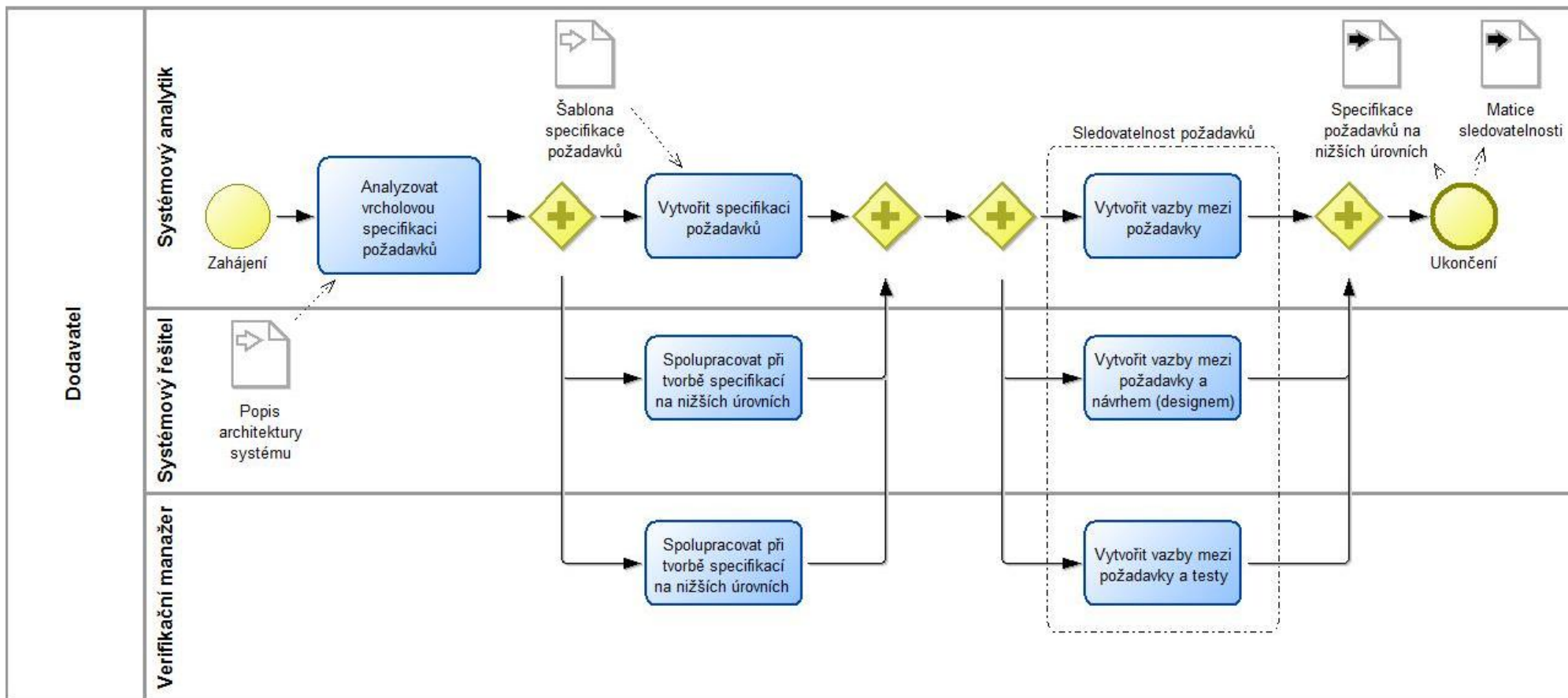
Příloha B



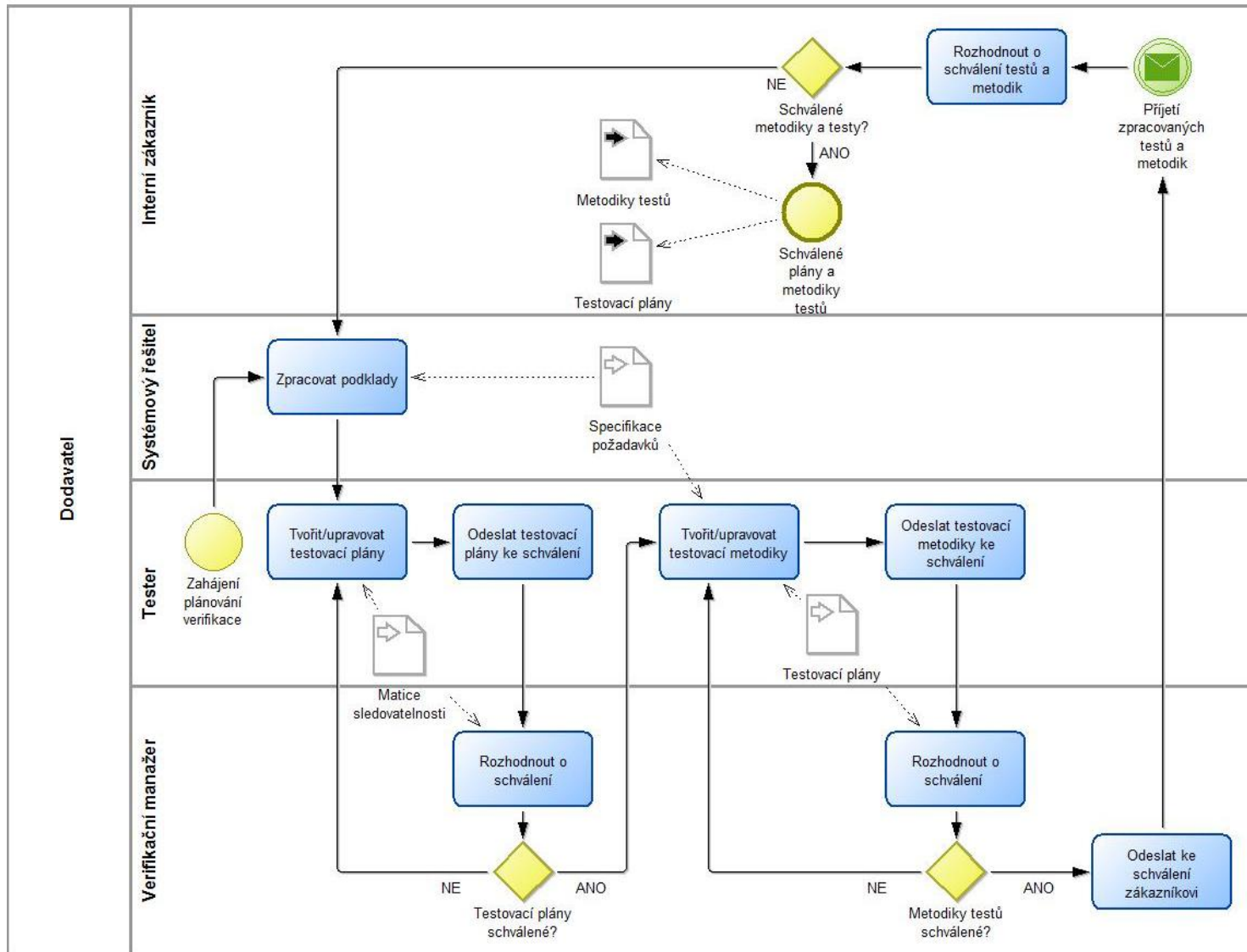
Subproces 1: Sběr požadavků probíhající v interním projektu



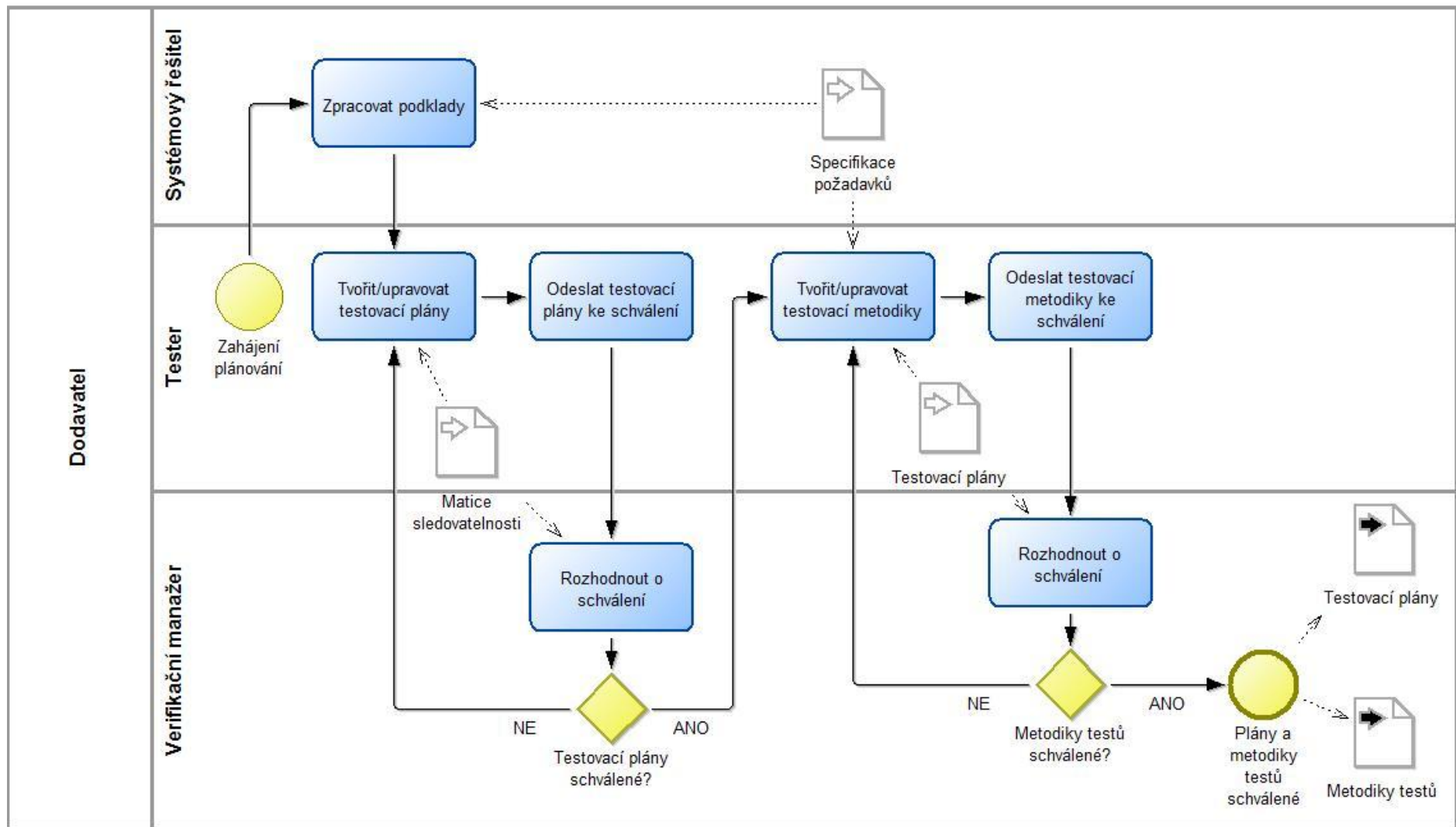
Subproces 2: Sběr požadavků probíhající v externím projektu



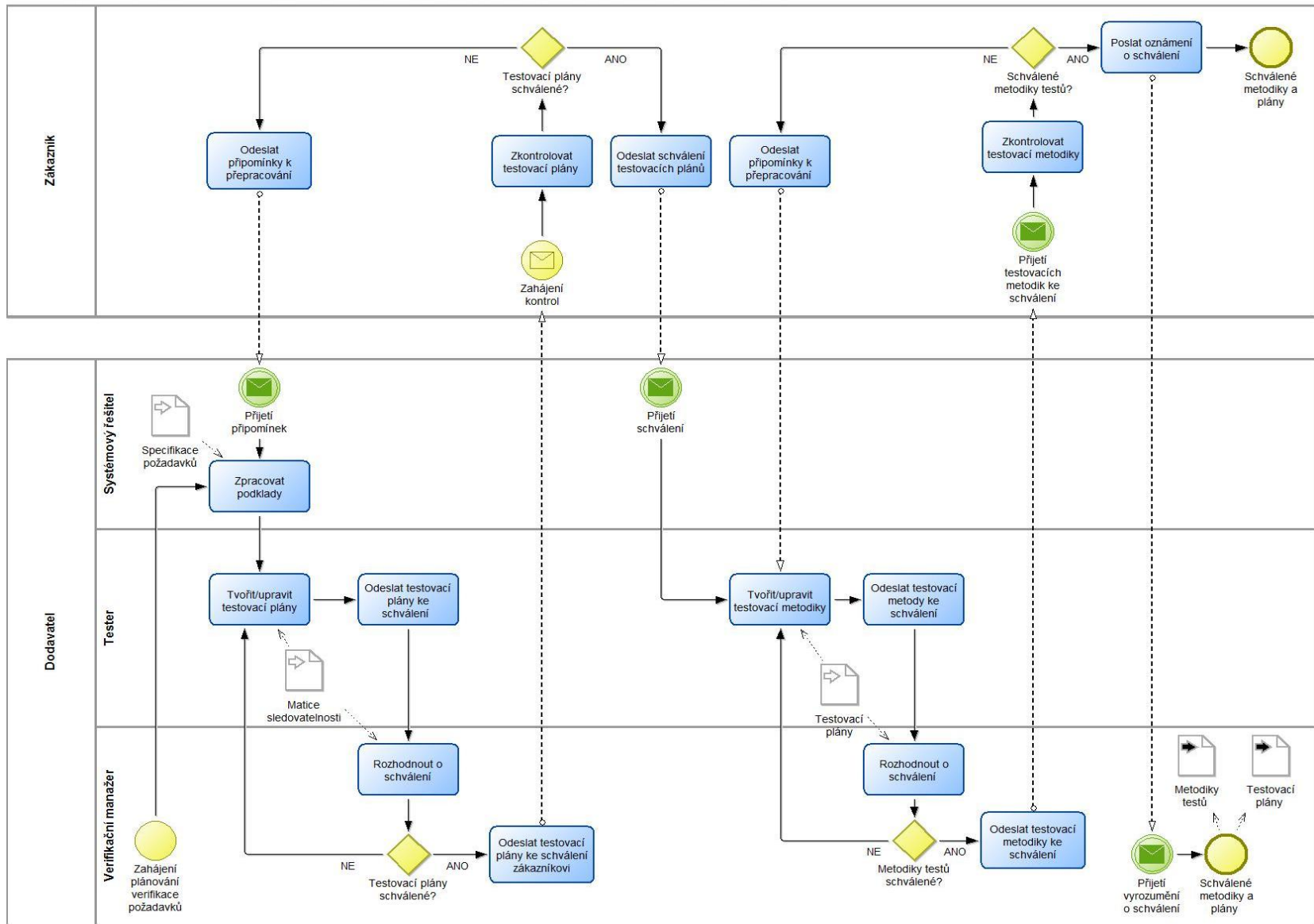
Subproces 3: Tvorba specifikace požadavků pro interní i externí projekt



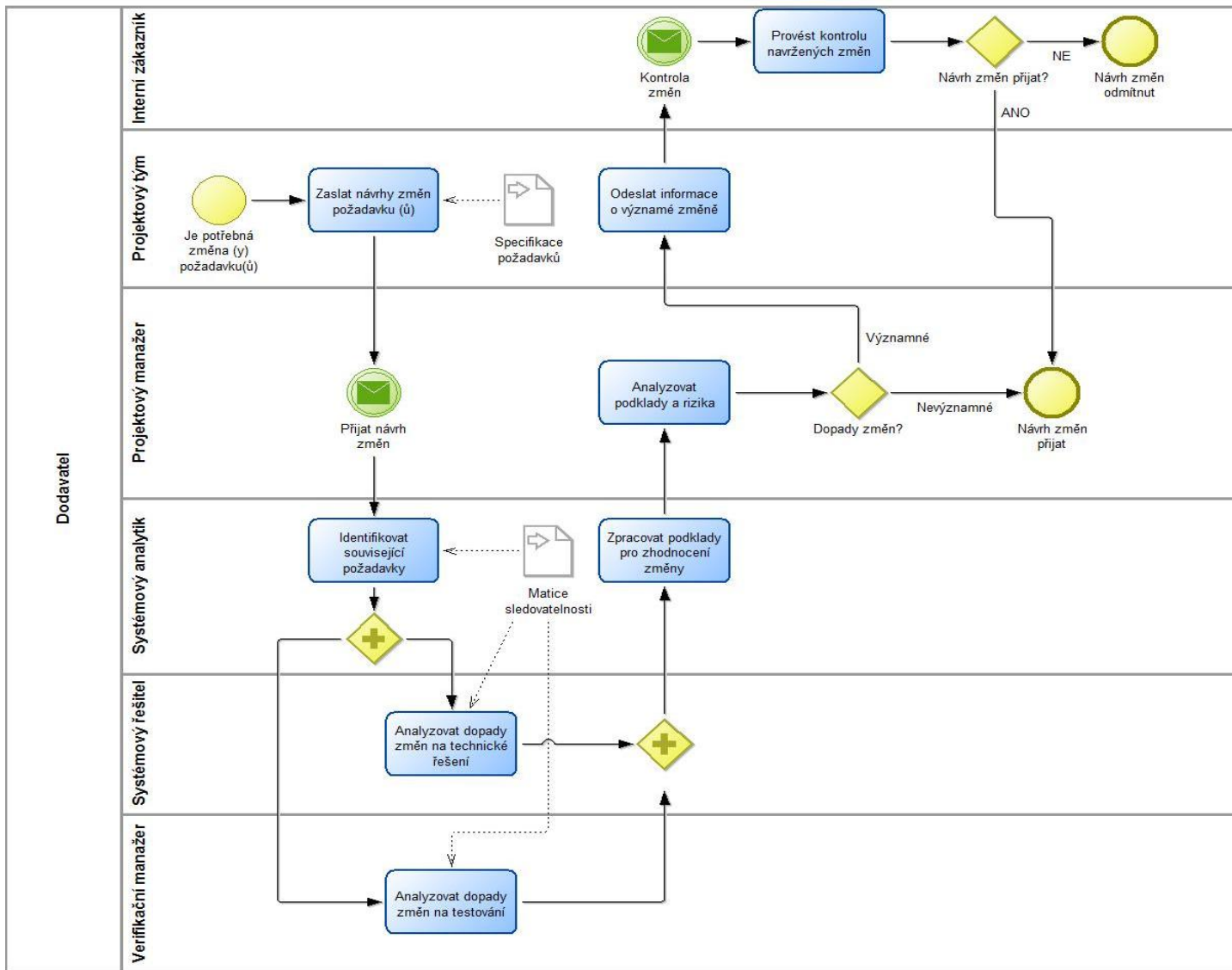
Subproces 4: Plánování verifikace požadavků pro interní projekt



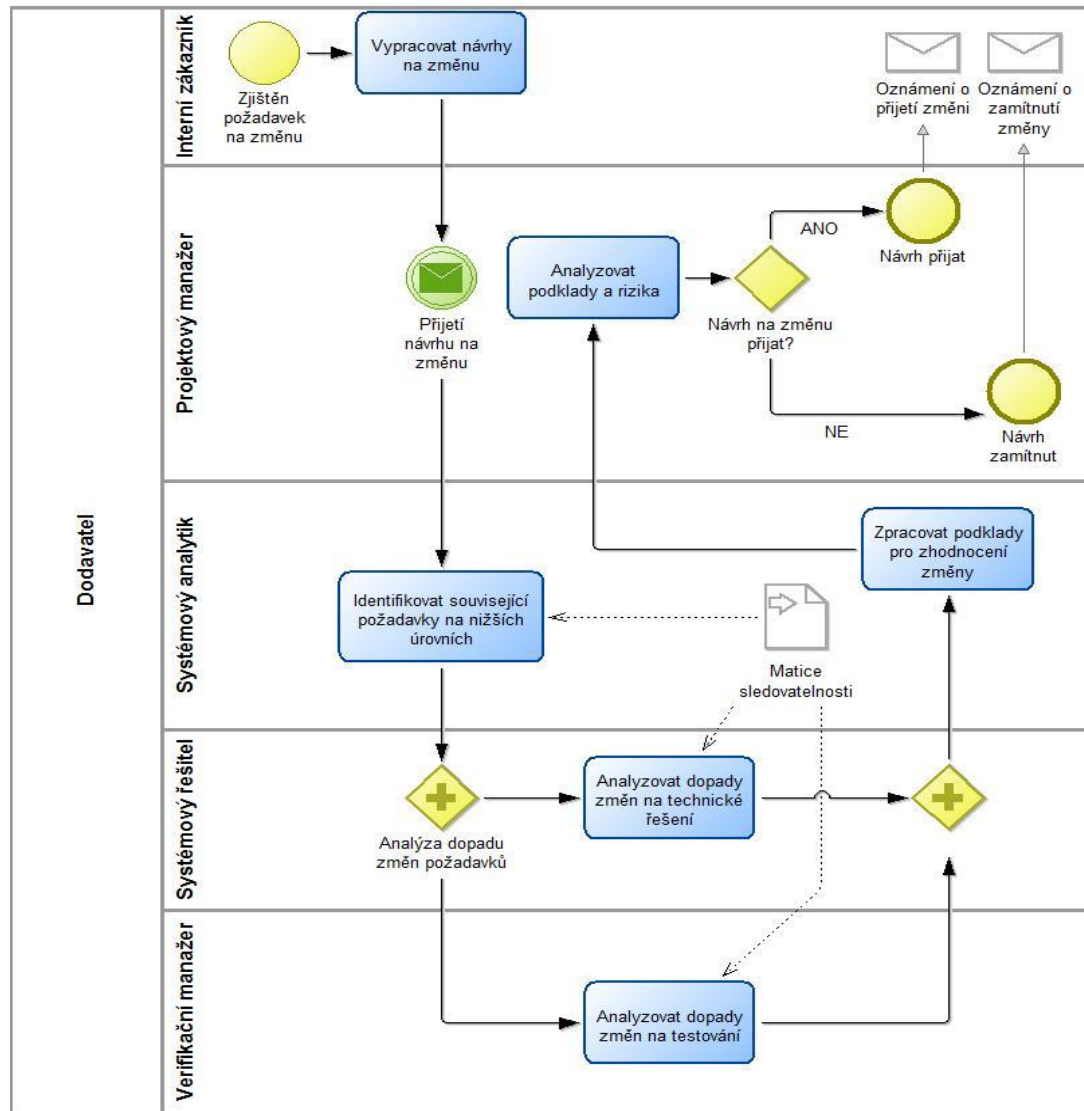
Subproces 5: Plánování verifikace probíhající interně v rámci firmy pro externí projekt



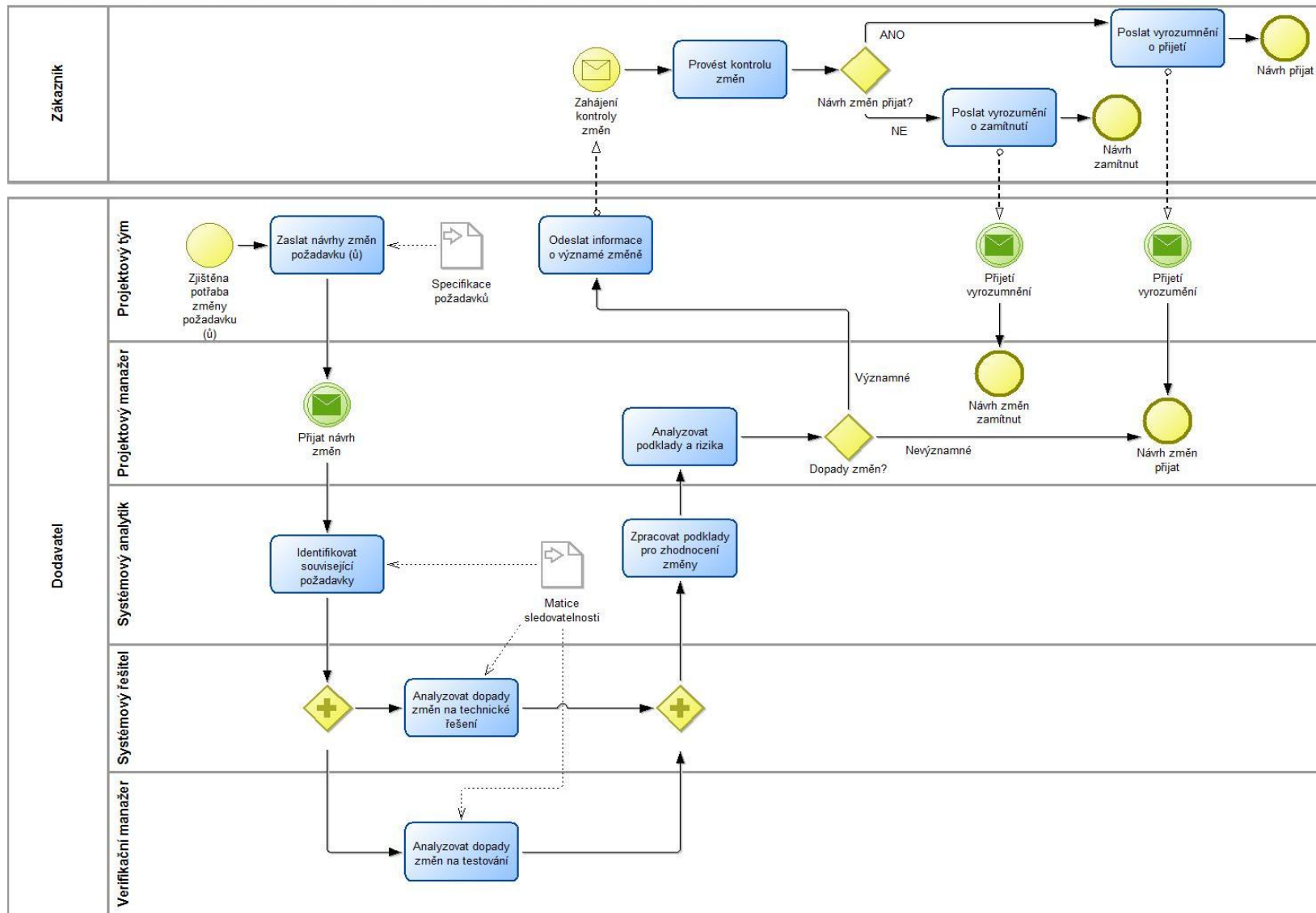
Subproces 6: Plánování verifikace se Zákazníkem v rámci externího projektu



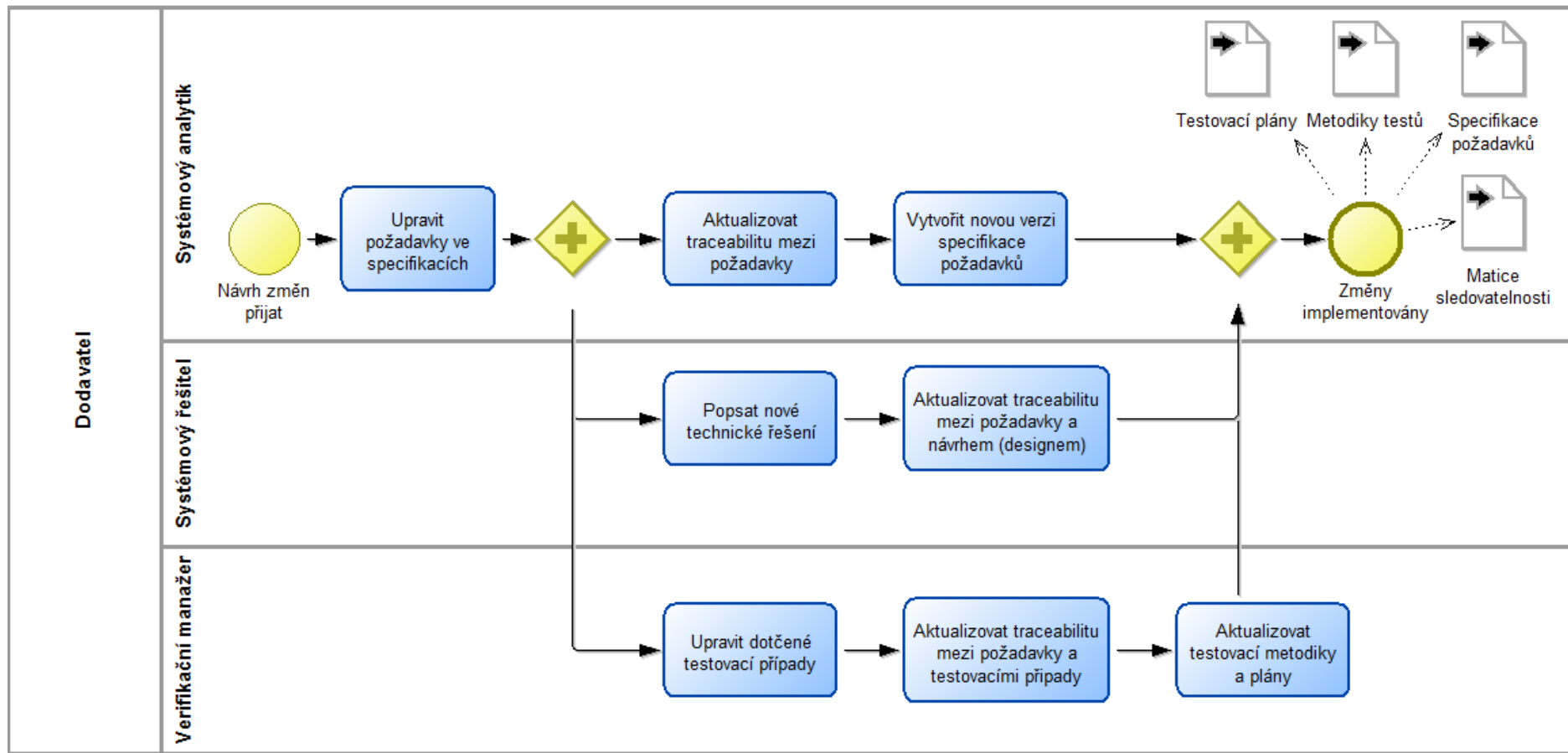
Subproces 7: Identifikace a analýza změn zjištěná Projektovým týmem v interním projektu



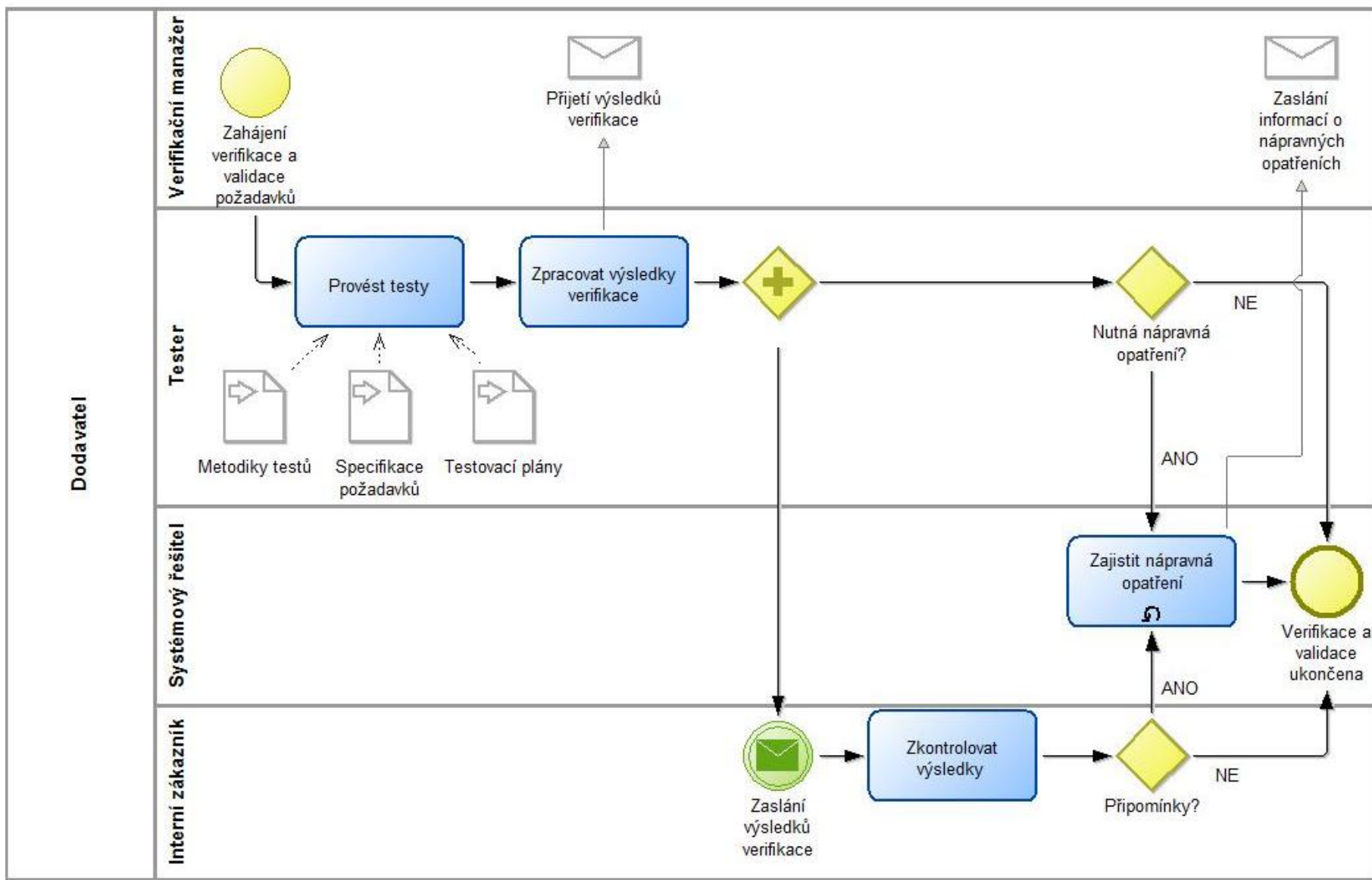
Subproces 8: Identifikace a analýza změn zjištěná Interním zákazníkem v interním projektu



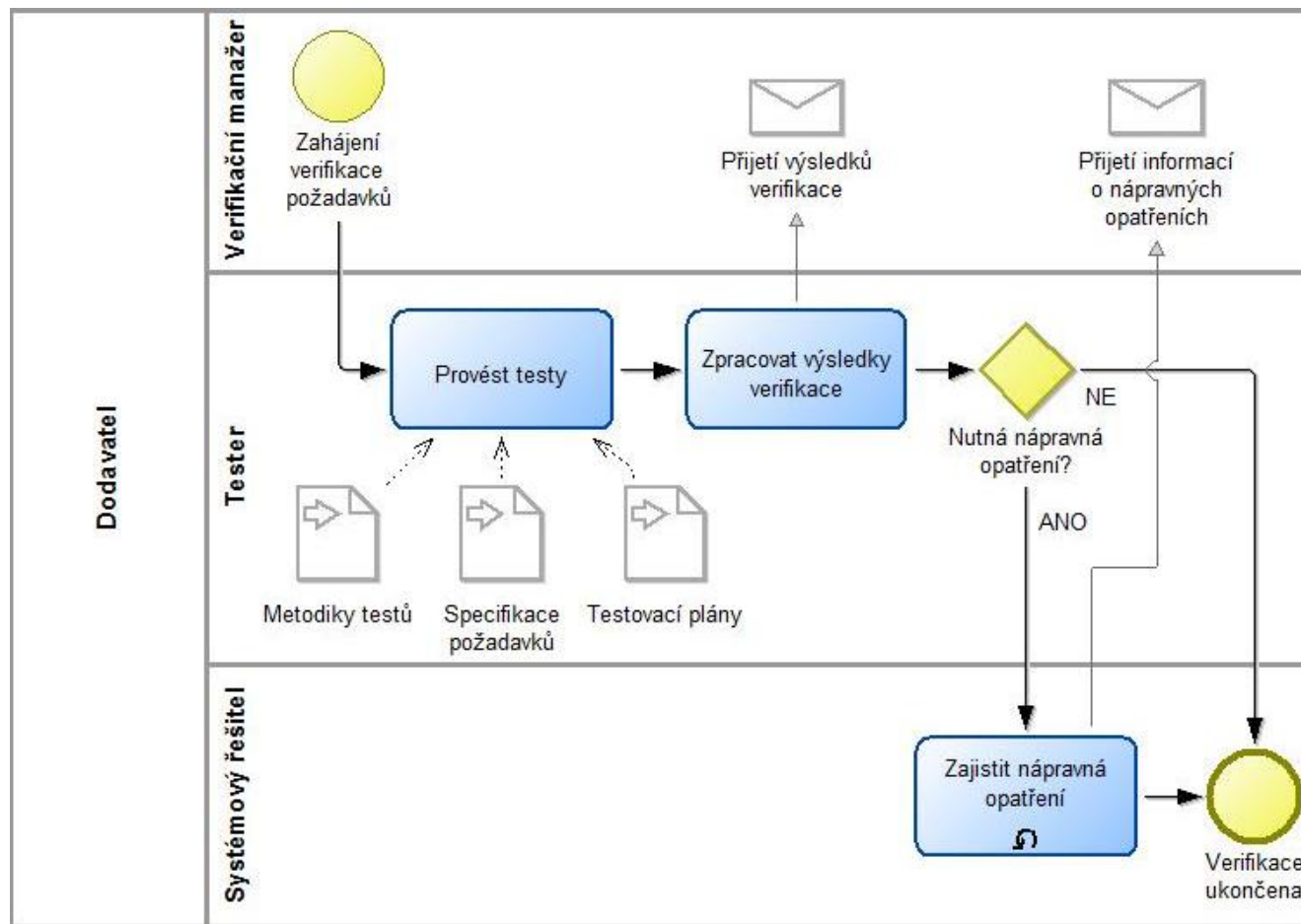
Subproces 9: Identifikace a analýza změn zjištěná dodavatelem v externím projektu



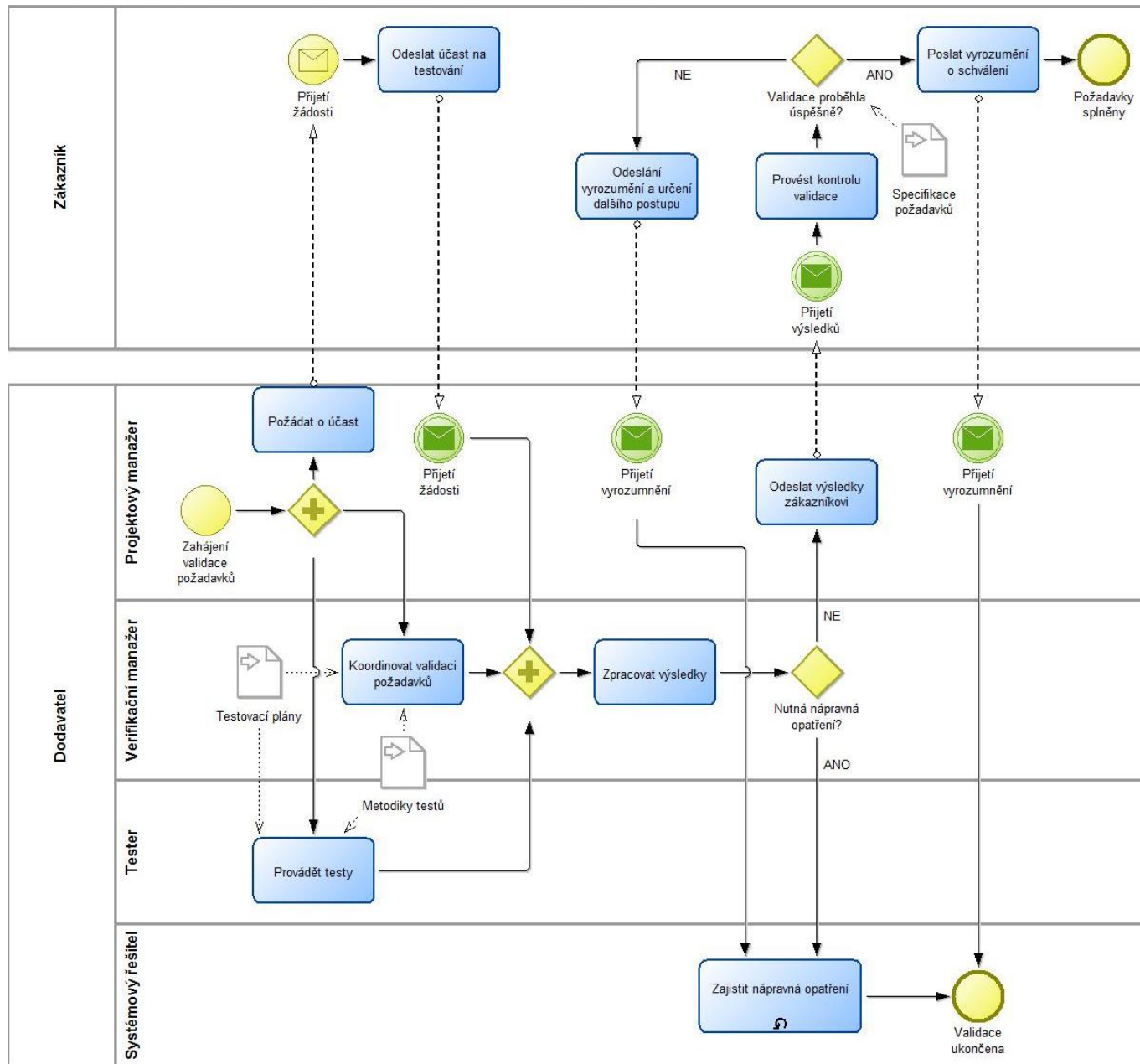
Subproces 11: Implementace změn pro interní i externí projekt



Subproces 12: Verifikace a validace požadavků pro interní projekt



Subproces 13: Verifikace požadavků probíhající interně v rámci externího projektu



Subproces 14: Validace požadavků se Zákazníkem v rámci externího projektu