

RUP disciplína Requirements

Semestrální práce

Michaela Nosková, xnosm04

14.12.2011

Obsah

1	Cíle práce	4
2	Charakteristika disciplíny Requirements	4
2.1	Začlenění disciplíny Requirements do celkového konceptu RUP	4
2.1.1	Iterativní vývoj	4
2.1.2	Aktivní správa požadavků	5
3	Workflow disciplíny Requirements	6
3.1	Analýza problému (Analyze the Problem)	7
3.2	Porozumění potřebám zainteresovaných subjektů (Understand Stakeholder Needs)	8
3.3	Definice systému (Define the System)	9
3.4	Řízení rozsahu systému (Manage the Scope of the System)	10
3.5	Upřesnění vymezení systému (Refine the System Definition)	11
3.6	Řízení změnových požadavků (Manage Changing Requirements)	12
4	Činnosti	13
4.1	Stanovení terminologického slovníku (Capture a Common Vocabulary)	13
4.2	Hledání aktérů a tvorba Use Case modelu (Find Actors and Use Cases)	13
4.3	Vytvoření Vize (Develop Vision)	14
4.4	Zjištění požadavků zainteresovaných subjektů (Elicit Stakeholder Requests)	14
4.4.1	Požadavky zainteresovaných subjektů	14
4.4.2	Scénáře užití	14
4.5	Doplnění specifikace systému (Develop Supplementary Specifications)	15
4.6	Přidělení priorit případům užití (Prioritize Use Cases)	15
4.7	Zpřesnění případů užití (Detail a Use Case)	16
4.8	Vypracování doplňujících požadavků (Detail the Software Requirements)	16
4.9	(Structure the Use-Case model)	17
4.10	Posouzení požadavků (Review Requirements)	17
5	Artefakty – popis, návaznosti na činnosti	18
5.1	Business Case	18
5.2	Změnové požadavky (Change Request)	18
5.3	Slovník pojmů (Glossary)	19
5.4	Plán Iterací (Iteration Plan)	19
5.5	Záznam o přezkoumání (Review record)	20
5.6	Seznam rizik (Risk list)	21
5.7	Dokument softwarové architektury (Software architecture document)	21

5.8	Plán vývoje SW (Software development plan)	21
5.9	Požadavky na SW (Software Requirement).....	22
5.10	Specifikace SW požadavků (Software Requirement specification)	22
5.11	Požadavky zainteresovaných subjektů (Stakeholder requests)	23
5.12	Scénáře užití (Storyboard)	24
5.13	Doplňující specifikace (Supplementary Specifications)	24
5.14	Use-Case model	24
5.14.1	Jak vyvíjet Use-Model?	25
5.14.2	Nefuncionální požadavky	25
5.14.3	Dilema Co a Jak.....	25
5.15	Návrh uživatelského rozhraní (User-Interface Prototype)	26
5.16	Vize (Vision)	26
6	Role	27
7	Závěr	27
7.1	Vlastní zhodnocení	27
8	Zdroje.....	27
9	Přílohy	28
9.1	Terminologický slovník	28

1 Cíle práce

Cílem této práce je popsat disciplínu Requirements metodiky RUP (Rational Unified Process) z manažerského pohledu tak, aby byl objasněn její význam a místo v celé metodice. Tato práce je založena na studiu samotné metodiky, snaží se charakterizovat úkoly, aktivity, artefakty a role, které se v této konkrétní části využívají.

V první části se krátce věnuji charakteristice disciplíny Requirements, uvedu celkový pohled na metodiku RUP a zasadím disciplínu Requirements do kontextu.

Ve druhé části se věnuji popisu jednotlivých aktivit v rámci workflow disciplíny Requirements, nejprve krátce obecně o workflow a dále se postupně věnuji jednotlivým aktivitám. V následující kapitole pak uvádím přehled jednotlivých úkolů (Tasks), ze kterých se sestávají samotné aktivity.

Pátá kapitola se zabývá artefakty, neboli výstupy jednotlivých úkolů. Je to nejobsáhlejší část práce, protože jsem se snažila alespoň částečně uvádět návody nebo rady k vytváření jednotlivých výstupů. Následuje krátká kapitola věnující se rolím vyskytujícím se v disciplíně a v závěru krátce uvádím osobní zhodnocení disciplíny Requirements metodiky RUP.

2 Charakteristika disciplíny Requirements

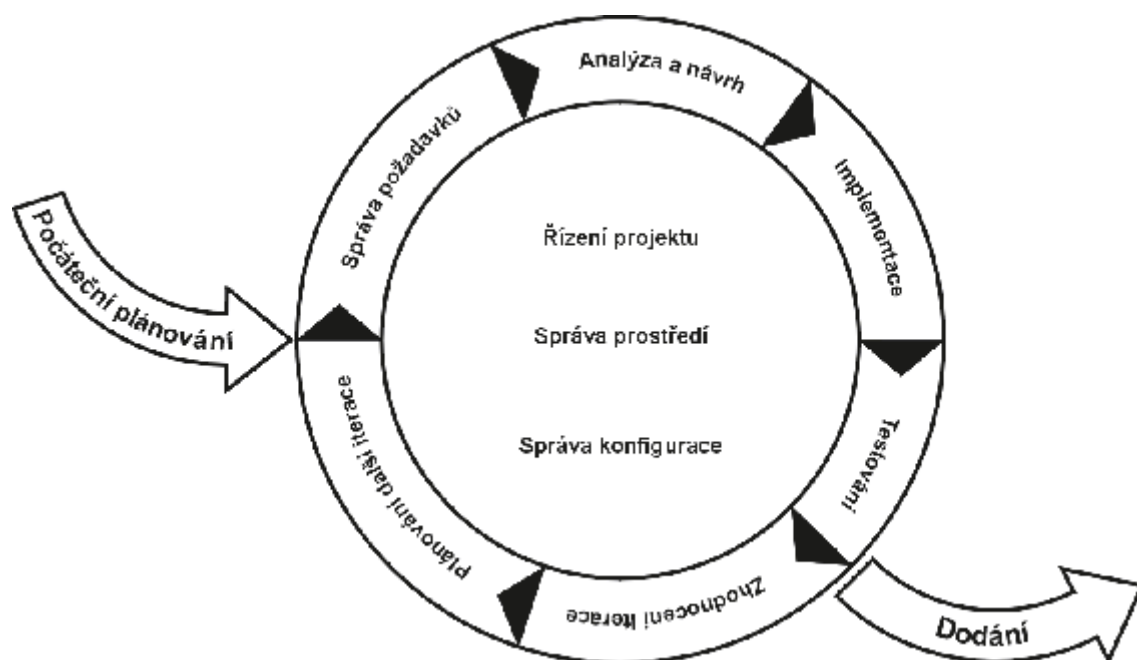
2.1 Začlenění disciplíny Requirements do celkového konceptu RUP

Základní filosofii metodiky Rational Unified Process je šest základních pravidel, tzv. “nejlepších praktik” používaných při vývoji software[1]:

1. Iterativní vývoj software
2. **Správa požadavků**
3. Architektura založená na komponentách
4. Vizuální modelování
5. Ověřování kvality software
6. Řízení změn software

2.1.1 Iterativní vývoj

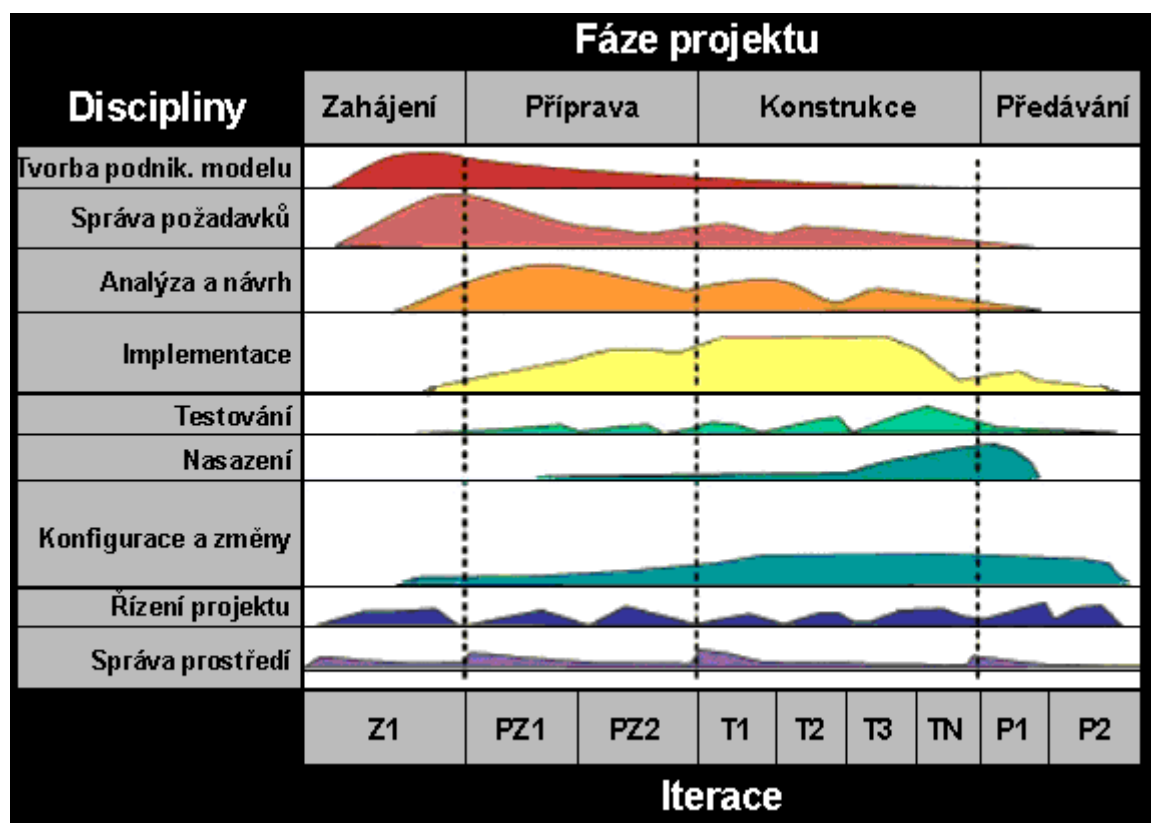
V současnosti není již možné na začátku vývoje systému přesně definovat celý problém, navrhnout řešení, provést implementaci a až na závěr - kdy je spotřebována většina přidělených finančních prostředků - celý systém otestovat. Iterativní přístup spočívá v rozdělení celého projektu na čtyři fáze, z nichž každá se skládá z několika iterací se životním cyklem typu “vodopád”. Výsledkem každé této iterace je spustitelná verze. [1]



Obrázek 1 - Životní cyklus RUP [1]

2.1.2 Aktivní správa požadavků

U vodopádového přístupu se požadavky sbírají a dokumentují pouze na začátku celého projektu, jejich změna v dalších stádiích vývoje se zpravidla nepřipouští. RUP naproti tomu využívá koncepci “aktivní správy požadavků” spočívající ve stálém kontaktu zadavatele s dodavatelem a možnosti zpřesňování a úprav požadavků v průběhu projektu. [1]



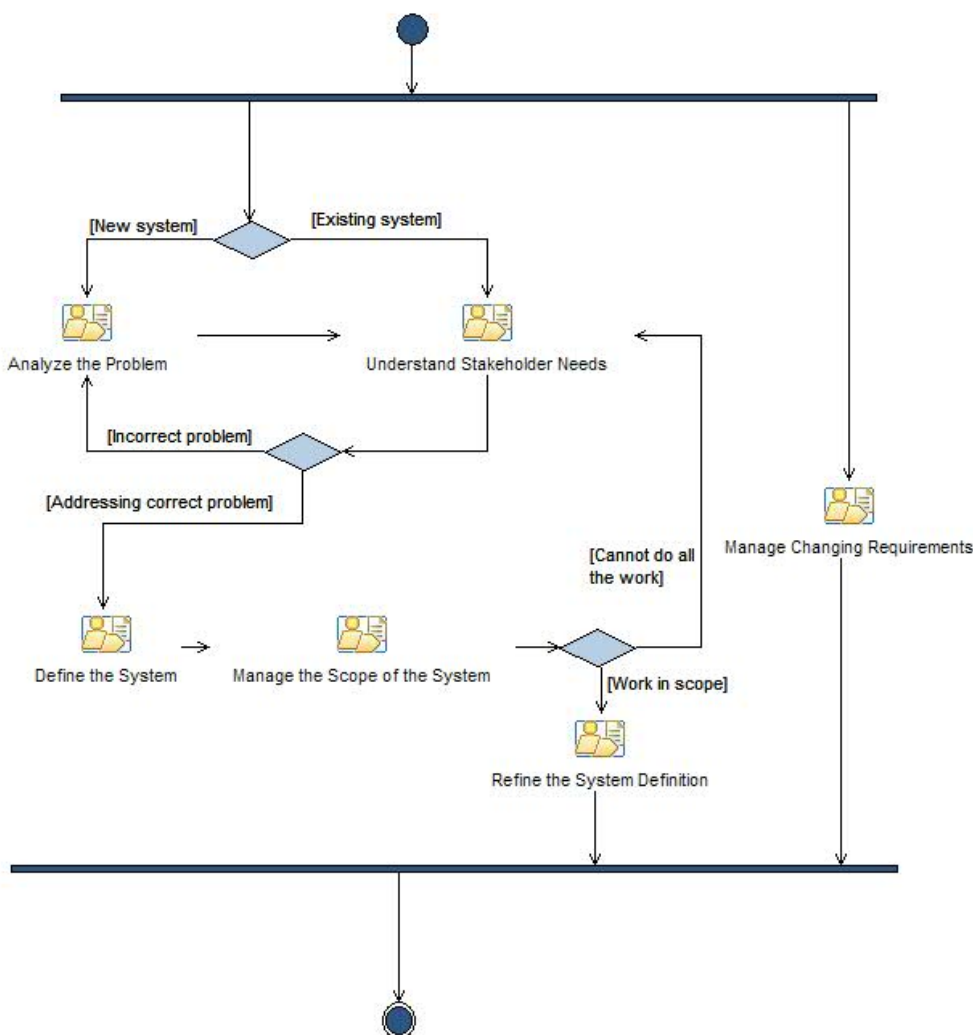
Obrázek 2 - Fáze projektu [1]

3 Workflow disciplíny Requirements

Je složen ze 6 dílčích aktivit, kde každá část je souborem jednotlivých dílčích úkolů. Ty naplňují cíle jednotlivých aktivit v rámci workflow disciplíny Requirements. Každá aktivita představuje vysokoúrovňový cíl, kterého musí být dosaženo, aby bylo možno efektivně provádět správu požadavků.

Workflow diagram ukazuje časovou posloupnost jednotlivých aktivit. V jeho počátku se nejprve rozhodujeme, zda se jedná o nový či existující systém. Pokud se jedná o nový systém, musíme nejprve analyzovat problém a poté porozumět potřebám zainteresovaných subjektů. Pokud se jedná o již existující problém, provedeme analýzu a zjistíme, zda se jednalo o správný problém. Pokud ano, pokračujeme k aktivitám definice systému a řízení rozsahu systému. Podaří-li se nám zvládnout obě tyto aktivity, můžeme přistoupit ke zpřesnění definice systému. Celým workflow disciplíny Requirements nás provází řízení změnových požadavků. Ty se vyskytují po celou dobu trvání jednotlivých aktivit.

Workflow



Obrázek 3 - Workflow disciplíny Requirements

3.1 Analýza problému (Analyze the Problem)

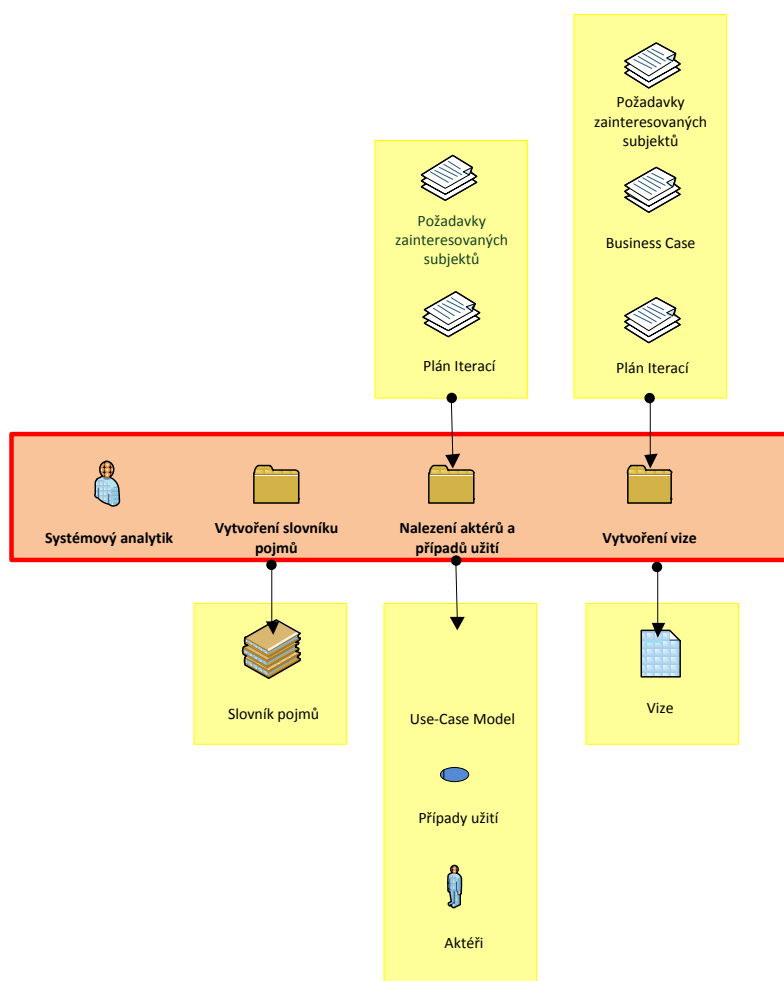
Cílem této aktivity je získat schválení k řešení daného problému. Analýza zahrnuje identifikaci zainteresovaných stran, definici hranic a identifikace omezení projektu.

První krok při analýze problému by mělo být ujištění se, že všechny zúčastněné strany považují vyřešení problému za podstatné. Abychom se vyhnuli případným nedorozuměním, je dobré si na začátku stanovit terminologický slovník, který se bude po dobu trvání projektu využívat.

Dalším nezbytným krokem je rozpoznání klíčových zainteresovaných osob, kteří se budou podílet na shromažďování požadavků.

Primárním zdrojem pro získávání informací o problému je pak Vize, ve které jsou zachyceny na vysoké úrovni uživatelské pohledy na vyvíjený systém. Ve Vizi nalezneme identifikaci klíčových funkcí, které zajišťují, že bude poskytnuto vhodné řešení.

Pro přesné vymezení projektu musí být předem dohodnuty hranice projektu. Systémový architekt identifikuje uživatele a systémy, které budou mít vliv na vyvíjený systém. Uživatelé systému mohou být reprezentováni aktéry v Use-Case modelu.



Obrázek 4 - Analýza problému

3.2 Porozumění potřebám zainteresovaných subjektů (Understand Stakeholder Needs)

Tato aktivita se snaží porozumět tomu, co zainteresované subjekty chtějí z navrhovaného řešení a definuje klíčové prvky pro řešení. Cílem této aktivity je porozumět potřebám primárních zainteresovaných subjektů a shromažďování informací o požadovaném produktu.

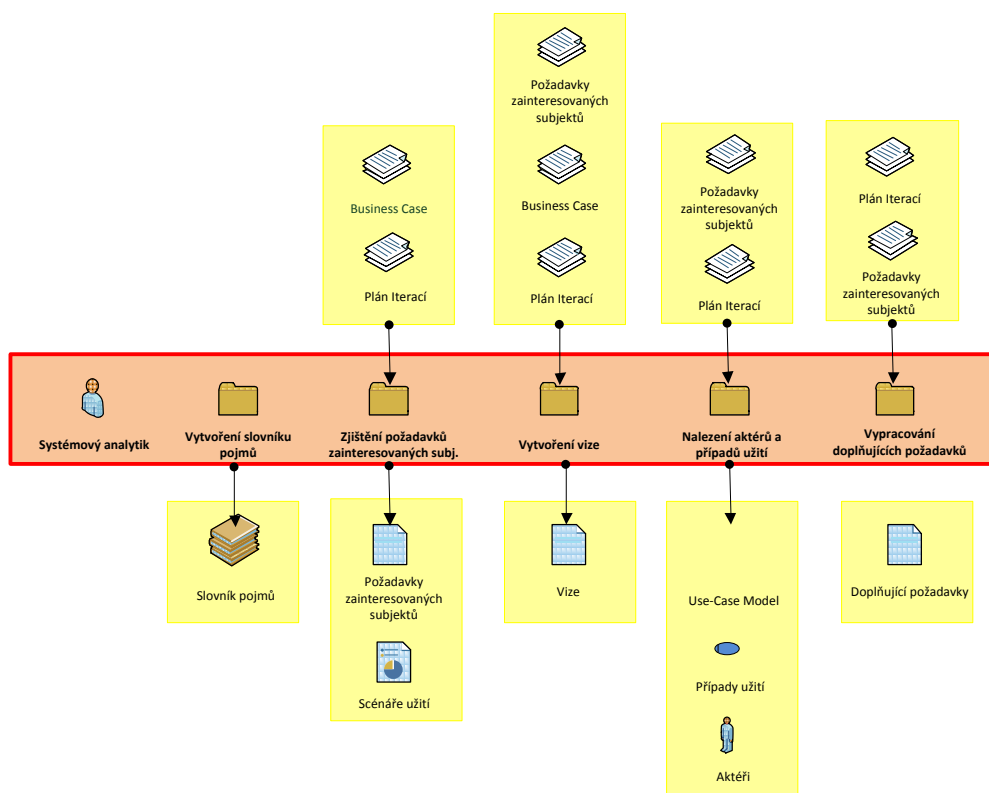
Tato aktivita se zabývá shromažďováním a získáváním informací zainteresovaných subjektů v projektu tak, abychom pochopili jejich skutečné potřeby. Shromážděné požadavky zainteresovaných subjektů lze považovat za "seznam přání", který bude použit jako hlavní vstup pro definování vyšších funkcí systému, jak je popsáno v Vizi a které se podílejí na specifikaci požadavků na software, jak je popsáno ve specifikaci požadavků na software (také use-case model případů použití a Doplnující údaje).

Hlavním cílem této aktivity je přimět zainteresované subjekty, aby pro vytváření vstupních požadavků používali rozhovory, požadavky na vylepšení a semináře. Primárním výstupem je kolekce priorit vlastností a jejich kritických atributů, které budou využity při definování systému a řízení rozsahu systému (viz definice systému (4.3), řízení rozsahu systému(4.4)).

Tyto informace mají za následek zdokonalení vize, stejně jako lepší pochopení požadavků. Také při vykonávání této činnosti můžeme začít diskutovat o funkčních požadavcích na systém z hlediska jeho případů užití a aktérů (viz Aktéři, Případy užití). Ty požadavky, které nezapadají do případů užití by mělo být dokumentováno v Doplnujících požadavcích (viz Vývoj Doplnujících požadavků).

Stejně jako jsou definovány nové požadavky, je důležité dokumentovat všechny závislosti (např. sledovatelnost) mezi těmito požadavky (viz Správa závislostí).

Dalším důležitým výstupem je aktualizovaný Slovník pojmů, který slouží k usnadnění komunikace prostřednictvím používání společného slovníku mezi členy týmu (viz Vytvoření slovníku pojmů).



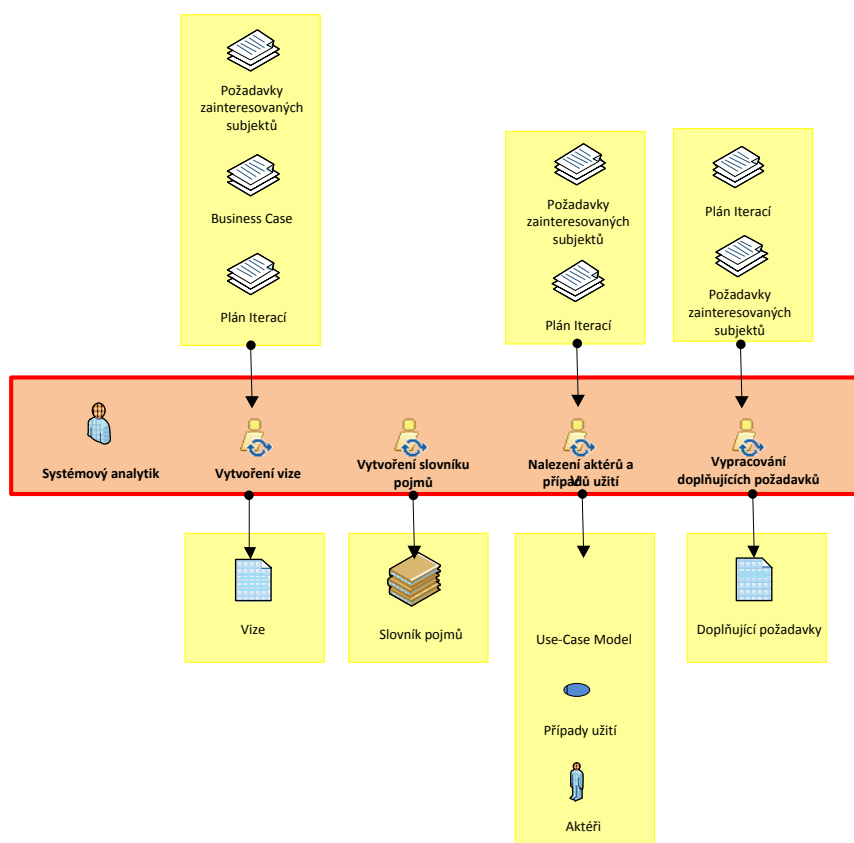
Obrázek 5 - Porozumění potřebám zainteresovaných subjektů

3.3 Definice systému (Define the System)

Cílem této činnosti je začít soustředit se na rozsah vysokoúrovňových požadavků vymezením širě detailních požadavků na systém.

Tato činnost se zabývá:

1. Přizpůsobováním projektového týmu v jejich chápání systému.
2. Provedením vysokoúrovňové analýzy výsledků z kolekce požadavků zainteresovaných subjektů
3. Zpřesněním a zdokonalením vize tak, aby zachycovala klíčové vlastnosti charakterizující systém (Vytvoření vize)
4. Upřesněním Use-Case modelu tak, aby zahrnoval hlavní rysy případů užití (Nalezení aktérů a případů užití)
5. Zpracováním osnovy non-use-case specifikace požadavků v Doplnujících požadavcích (Vypracování doplňujících požadavků)
6. Začít zaznamenávat výsledky vyvozovaných požadavků přehlednějším způsobem.
7. Slovník pojmů je také obohacen o další užívané termíny.



Obrázek 6 – Definice systému

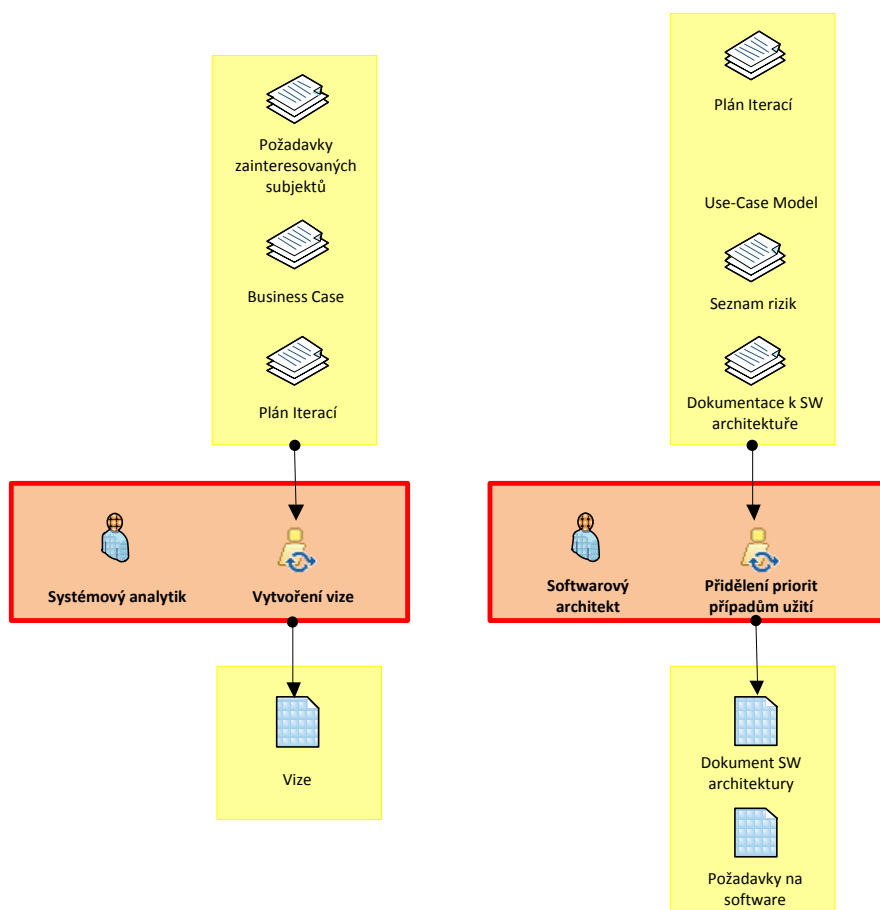
3.4 Řízení rozsahu systému (Manage the Scope of the System)

Tato činnost zajišťuje, že požadavky na systém jsou jasné a stanovuje soubor požadavků, na kterých je práce zvládnutelná při iteraci. Cílem této činnosti je, aby rozsah vyvíjeného systému byl tak zřetelný jak je to jen možné a zaměřuje se na zvládnutelnou skupinu požadavků na práci ve vývojovém cyklu.

Tato činnost se zabývá:

1. Stanovením priorit a a zpřesňováním vstupu do výběru vlastností systému a požadavků, které mají být zahrnuty v aktuální iteraci (Přidělení priorit případům užití)
2. Definováním sady scénářů chování pro jeden až více případů užití, které reprezentují určité významné centrální funkce (Přidělení priorit případům užití)
3. Definováním, jak bude zachována sledovanost včetně informace, které atributy požadavků a jaké vazby budou sledovány (Správa závislostí)
4. Zdokonalení vize o dokumentaci rozsahu řešení (Vytvoření vize)

Rozsah projektu je definován jako soubor přidělených požadavků. Řízení rozsahu projektu tak, aby odpovídal dostupným zdrojům (čas, lidé a peníze) je klíčem k řízení úspěšných projektů.



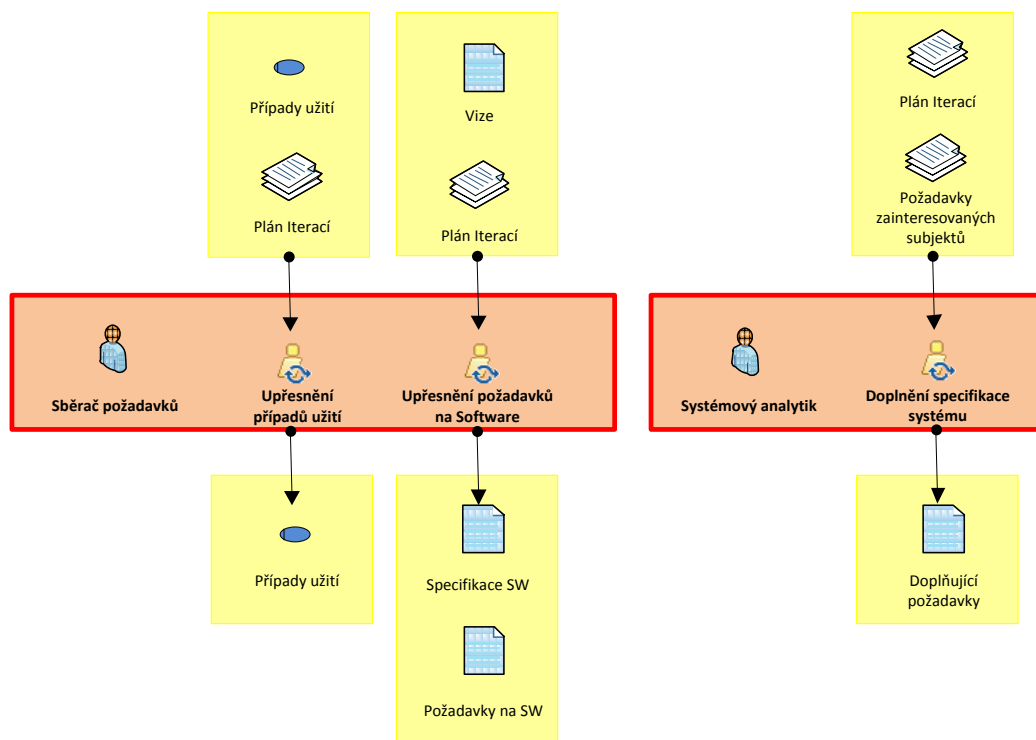
Obrázek 7 – Řízení rozsahu systému

3.5 Upřesnění vymezení systému (Refine the System Definition)

Tato činnost upřesňuje požadavky, které musí být vyvinuty v současném vývojovém cyklu. Cílem této činnosti je dále upřesnit požadavky s cílem zachytit konsensus pochopení definice systému.

Tato činnost se zabývá zpodrobňováním a zpřesňováním požadavků. Navazuje na pochopení rozsahu projektu. Jejím cílem je prioritizace klíčových vlastností systému (obvykle popsány ve Vizi), které jsou realizovatelné s pevně stanoveným rozpočtem a termínem. Výsledkem je hlubší pochopení funkčnosti systému vyjádřené v očištěných, podrobných požadavcích na specifikaci artefaktů a chování uvedených prototypů. Specifikace artefaktů může mít podobu podrobného případu užití (Upřesnění případů užití), doplňujících údajů (Vytvoření doplňujících požadavků) a v některých případech může být vytvořena formální specifikace SW (Upřesnění požadavků na SW).

Kdykoliv dojde ke změně požadavků, měly by být provedeny revize a aktualizace souvisejících atributů požadavků tak, jak je to popsáno v aktivitě řízení změnových požadavků (Řízená změnových požadavků).



Obrázek 8 - Upřesnění vymezení systému

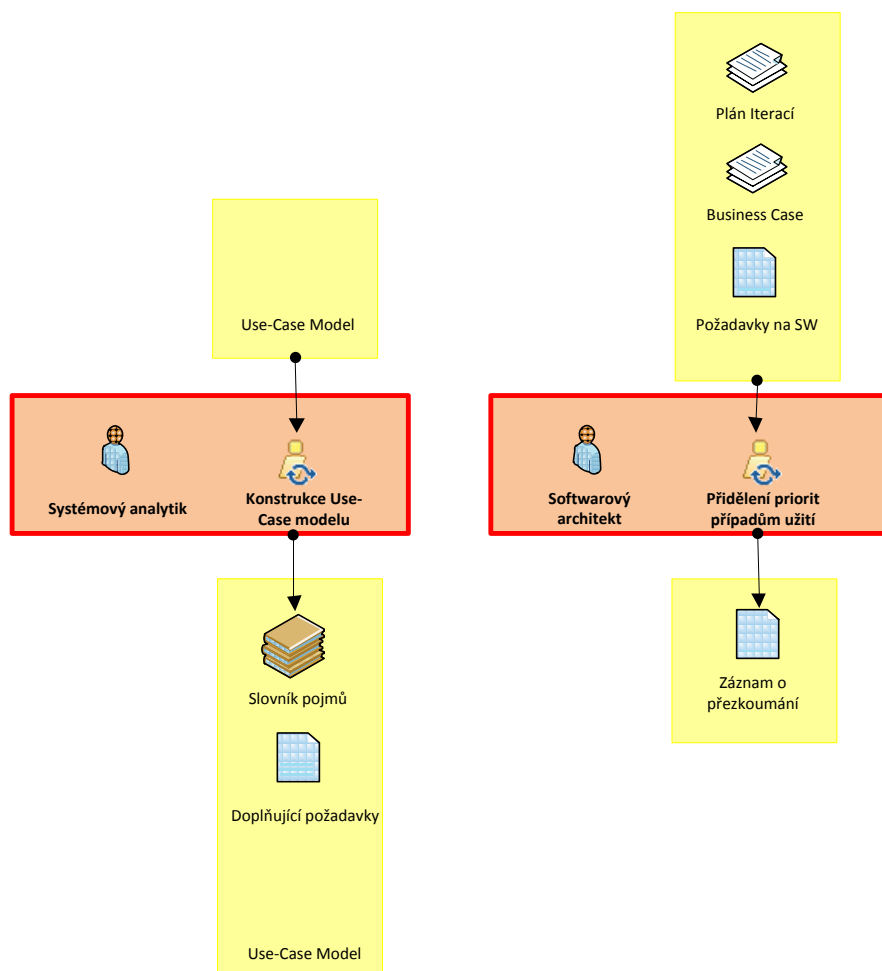
3.6 Řízení změnových požadavků (Manage Changing Requirements)

Tato činnost se zabývá změnami požadavků a hodnocením jejich celkového dopadu. Cílem této činnosti je posoudit vliv požadovaných změnových požadavků a zvládat následný dopad schválené změny.

Tato činnost se zabývá:

1. Hodnocením požadovaných změn a stanovení jejich dopadu na stávající Softwarové požadavky.
2. Strukturování use-case model s cílem zlepšit celkové řízení požadavků dokumentovány v Use Case Modelu.
3. Řízení atributů změnových požadavků a sledování vztahů.
4. Poté, ke změnám došlo, ověření, že výsledky práce odpovídají požadavkům zákazníka na systém.

Změny požadavků mají přirozeně dopad na vytvářené artefakty (např. analýzy a projektové práce, rozmístění artefaktů, atd.) Sledovatelnost vztahů je identifikována a zdokumentována v Řízení dopadu změn a je výslovně definována závislosti mezi požadavky a ostatními pracemi. Tyto vztahy jsou klíčem k pochopení dopadu změn požadavků.



Obrázek 9 - Řízení změnových požadavků

4 Činnosti

V rámci disciplíny requirements se setkáváme s množstvím dílčích úkolů, pomocí nichž naplníme workflow této disciplíny. Těchto úkolů je celkem 10, avšak vyskytují se většinou ve více aktivitách.

Těmito úkoly jsou:

4.1 Stanovení terminologického slovníku (Capture a Common Vocabulary)

Na počátku životního cyklu projektu definujeme pojmy a jejich význam ve **slovníku pojmů**. Cílem tohoto úkolu je definovat společný slovník, který lze použít ve všech popiskách systému a to zejména požadavků na software. Je vhodné využívat zavedené pojmy nebo pojmy, které jsou využívány v konkrétní situaci nebo společnosti. Tyto znalosti získáme od uživatelů, z požadavků a specifikace SW, Use Case modelu nebo Vize.

Vztahy	Povinné	Doporučené	Pomocné/Externí
Role	Systémový Analytik	-	-
Vstupy	-	Uživatel Požadavky na SW Doplňující požadavky Specifikace SW Požadavky zainteresovaných subjektů Use Case Vize	-
Výstupy	Slovník pojmů		
Výskyt	Analyze the Problem > Capture a Common Vocabulary Understand Stakeholder Needs > Capture a Common Vocabulary Define the System > Capture a Common Vocabulary		

4.2 Hledání aktérů a tvorba Use Case modelu (Find Actors and Use Cases)

V tomto úkolu jsou identifikováni aktéři a případy užití (Use Case) na podporu realizace požadavků. Identifikace aktérů a případů užití explicitně definuje rozsah systému.

Cílem této aktivity je:





- ✚ Definovat rozsah systému – co bude součástí systému a co bude řešeno mimo systém
- ✚ Definovat, kdo a jakým způsobem bude interagovat se systémem
- ✚ Přehled funkcí systému

Vztahy	Povinné	Doporučené	Pomocné/Externí
Role	Systémový Analytik	-	-
Vstupy	Plán Iterací Požadavky zainteresovaných subjektů	Business Case Slovník pojmů Vize	-
Výstupy	Aktéři Use Case model		
Výskyt	Analyze the Problem > Find Actors and Use Cases Understand Stakeholder Needs > Find Actors and Use Cases		

4.3 Vytvoření Vize (Develop Vision)

Tato úloha popisuje, jak rozvíjet celkovou vizi systému včetně problémů, které mají být řešeny, klíčových zainteresovaných osob, rozsah systému, klíčové prvky systému a možná omezení projektu.

Cílem tohoto úkolu je:

-  Specifikace problému, který je třeba řešit
-  Identifikovat zainteresované osoby
-  Definovat hranice systému
-  Popsat základní funkce systému

Vztahy	Povinné	Doporučené	Pomocné/Externí
Role	Systémový Analytik	-	-
Vstupy	Business Case Plán Iterací Požadavky zainteresovaných subjektů	Vize	-
Výstupy	Vize		
Výskyt	Analyze the Problem > Develop Vision Understand Stakeholder Needs > Develop Vision Define the System > Develop Vision Manage the Scope of the System > Develop Vision		

4.4 Zjištění požadavků zainteresovaných subjektů (Elicit Stakeholder Requests)

Vztahy	Povinné	Doporučené	Pomocné/Externí
Role	Systémový Analytik	-	-
Vstupy	Business Case Plán Iterací	Změnové požadavky Vize	-
Výstupy	Požadavky zainteresovaných subjektů Scénáře užití		
Výskyt	Understand Stakeholder Needs > Elicit Stakeholder Requests		

4.4.1 Požadavky zainteresovaných subjektů

Výsledný produkt zjišťování požadavků zainteresovaných subjektů. Obsahuje všechny typy požadavků zainteresovaných subjektů (zákazníků, uživatelů, marketingu atd.). Může také odkazovat na mnohé externí zdroje, podle kterých se musí vývoj systému řídit a které musí systém splňovat.

4.4.2 Scénáře užití

Tento dokument je logický a koncepční popis funkcionality systému pro konkrétní situaci. Obsahuje interakci mezi uživateli systému a systémem samotným. Scénář tak „vypráví konkrétní příběh“.

4.5 Doplnění specifikace systému (Develop Supplementary Specifications)

Tento úkol se zabývá požadavky, které se vztahují na konkrétní případy užití.

Jsou zde shromážděny požadavky zainteresovaných subjektů, pro které nebylo možné vytvořit Use-Case modely. Mohou zde být zachyceny jak funkční, tak nefunkční požadavky. Při plnění tohoto úkolu je důležité ujistit se, že všechny požadavky jsou uvedeny v potřebné úrovni detailnosti. Pokud jsou požadavky formálně řízeny pomocí nějakého SW na správu požadavků, je potřeba ujistit se, že je každý požadavek řádně identifikován a označen.

Vztahy	Povinné	Doporučené	Pomocné/Externí
Role	Systémový Analytik	Specifikátor požadavků	-
Vstupy	Plán Iterací Požadavky zainteresovaných subjektů	Business Case Slovník Vize	-
Výstupy	Doplňující požadavky		
Výskyt	Understand Stakeholder Needs > Develop Supplementary Specifications Define the System > Develop Supplementary Specifications Refine the System Definition > Develop Supplementary Specifications		

4.6 Přidělení priorit případům užití (Prioritize Use Cases)

V tomto úkolu jsou prioritizovány případy užití, tzn. určuje se pořadí jejich vývoje. Je tu místo, kde jsou identifikovány a prioritizovány architektonicky významné úkoly.

Cílem této aktivity je:

- ✚ Definovat vstupní výběr ze sady scénářů a případů užití, které mají být analyzovány v aktuální iteraci
- ✚ Definovat seznam scénářů, které představují významné centrální funkce systému
- ✚ Definovat soubor scénářů a případů užití, které pokrývají podstatu architektury

Vztahy	Povinné	Doporučené	Pomocné/Externí
Role	Softwarový architekt	-	-
Vstupy	Plán Iterací Seznam rizik Dokumentace SW Architektury Use-Case model	Požadavky na SW Vize	-
Výstupy	Požadavky na Software SW architektura		
Výskyt	Manage the Scope of the System > Prioritize Use Cases		

4.7 Zpřesnění případů užití (Detail a Use Case)

V tomto úkolu jsou přidány podrobnosti do specifikace případů užití.

Cílem této aktivity je:

- ✚ Popsat jeden nebo více případů užití je tok událostí dostatečně podrobně, aby na něm mohl začít vývoj software.
- ✚ Popsat do detailu případy užití tak, aby je chápali zástupci aktérů nebo zákazník.

Vztahy	Povinné	Doporučené	Pomocné/Externí
Role	Sběrač požadavků	-	-
Vstupy	Plán Iterací Případy užití	Slovník pojmů Požadavky zainteresovaných subjektů Scénáře užití Doplňující požadavky Use-Case model Vize	-
Výstupy	Případy užití		
Výskyt	Refine the System Definition > Detail a Use Case		

4.8 Vypracování doplňujících požadavků (Detail the Software Requirements)

Tento úkol popisuje, jak vypracovat doplňující požadavky na software. Účelem tohoto úkolu je sběr, zpodrobnění a zorganizování produktů práce do skupin, které kompletně popisují SW požadavky na systém.

Vztahy	Povinné	Doporučené	Pomocné/Externí
Role	Sběrač požadavků		-
Vstupy	Plán Iterací Vize	Slovník pojmů Návrh uživatelského rozhraní	-
Výstupy	Požadavky na SW Specifikace požadavků na SW		
Výskyt	Refine the System Definition > Detail the Software Requirements		

4.9 (Structure the Use-Case model)

V tomto úkolu jsou případ užití strukturovány tak, aby byly požadavky srozumitelnější a lépe udržovatelné. To zahrnuje využití standardizace mezi případy užití a aktéry a identifikace volitelného a výjimečného chování.

Účelem tohoto úkolu je:

- ✚ Najít chování případů užití, které je třeba považovat za abstraktní. Příkladem takového chování je běžné chování, volitelné chování, výjimečné chování a chování, které je třeba rozvíjet v dalších iteracích
- ✚ Nalezení nových abstrakcí aktérů, které definují role více aktérů

Vztahy	Povinné	Doporučené	Pomocné/Externí
Role	Systémový Analytik		-
Vstupy	Use-Case model		-
Výstupy	Slovník pojmů Doplňující požadavky Use-Case model		
Výskyt	Manage Changing Requirements > Structure the Use-Case Model		

4.10 Posouzení požadavků (Review Requirements)

Tato úloha popisuje, jak posuzovat požadavky pracovních produktů. Cílem posouzení požadavků je formální ověření, že výsledky požadavků odpovídají zákaznickovu pohledu na systém.

Vztahy	Povinné	Doporučené	Pomocné/Externí
Role	Technický recenzent		-
Vstupy	Business Case Plán Iterací Požadavky na SW	Aktér Změnové požadavky Slovník pojmů Specifikace SW Požadavky zainteresovaných subjektů Scénář užití Doplňující požadavky Případy užití Use-Case model Use-Case package Vize	-
Výstupy	Záznam o přezkoumání		
Výskyt	Manage Changing Requirements > Review Requirements		

5 Artefakty – popis, návaznosti na činnosti

5.1 Business Case

	Povinné	Doporučené	Externí
Vstupuje do:	Vytvoření Vize (Develop Vision)	Nalezení aktérů a případů užití (Find Actors and Use Cases)	-

Hlavním účelem tohoto artefaktu je vytvoření ekonomického plánu, nezbytného pro realizaci artefaktu Vize (Vision). Jakmile je Business Case vytvořen, je možno vyhodnotit návratnost investic (ROI). BC poskytuje argumenty pro projekt a stanovuje jeho ekonomická omezení. Na kritických milnících projektu je pomocí BC přezkoumáno, zda odpovídají odhadované výnosy a náklady a zda může projekt pokračovat.

BC by měl obsahovat následující údaje:

1. Úvod – měl by obsahovat účel projektu, jeho rozsah, definice, zkratky, reference a přehled tohoto BC
2. Popis výsledného produktu – Stručný popis vyvíjeného produktu. Měl by obsahovat název systému, příp. zkratku, která je používána. Vysvětluje, jaký problém je řešen a proč má smysl zabývat se vývojem.
3. Obchodní kontext – definuje do které oblasti je výsledný produkt namířen (např. telekomunikační služby či banky), jaká je situace na trhu a kdo jsou uživatelé. Uvádí se zde, zda je produkt vyvíjen na zakázku či se jedná o komerční produkt či zda se jedná o pokračování stávajícího projektu.
4. Cíle produktu – předběžný plán a posouzení rizik plánu. Jasně definované cíle jsou dobrým základem pro stanovení milníků projektu a řízení rizik.
5. Finanční rozpočet – Příjmy (výnosy, snížení nákladů, nehmotný majetek), Náklady (kapitál, náklady), ROI (přínosy/náklady)
6. Omezení – např. externí rozhraní, normy, certifikace, technická omezení (užití určité DB)

5.2 Změnové požadavky (Change Request)

	Povinné	Doporučené	Externí
Vstupuje do:	-	Zjištění požadavků zainteresovaných subjektů (Elicit stakeholder request)	-

Změnové požadavky se využívají k dokumentaci a sledování požadavků na změnu produktu. Umožňují zaznamenávat rozhodnutí a s příslušným posouzením zajišťují, že je brán v potaz dopad změny na produkt. Účelem je formulovat a sledovat změny a nedostatky.

Formulář změnového požadavku by měl obsahovat následující údaje:

1. Identifikace – jméno projektu, číslo požadavku, typ požadavku, název, datum, žadatel, priorit
2. Aktuální problém – popis problému, selhání, vylepšení, podmínky, za nichž byl problém zaznamenán, zdroj, náklady nebo úspory dopadem řešení problému
3. Požadované změny (žadatel) – popis požadované změny, očekávané náklady

4. Požadované změny (Hodnotitelský tým) – akce, schváleno, zamítnuto, odloženo, popis navrhované změny, postižené součásti projektu, nové funkce atd.

5. Rozhodnutí – odhad nákladů na realizaci navrhovaných změn, skutečná doba provedení, analýza, realizace, testování, dokumentace, počet postižených řádků kódu

6. Posouzení – test metod, kontrola, analýza, demonstrace, testovací platformy a případy

Změnové požadavky je nejlépe uchovávat v databázi nebo systému pro řízení změnových požadavků, aby bylo možno sledování jejich stavu (třídění dle priorit, sledování zadání a statusu dokončení). Na malé projekty postačí tabulka, ve které sledujeme změny.

5.3 Slovník pojmů (Glossary)

	Povinné	Doporučené	Externí
Role:	Systémový analytik	Systémový analytik	
Vstupuje do:	-	Nalezení aktérů a případů užití (Find Actors and Use Cases)	-
Výstup z:	Vytvoření slovníku pojmů		

Tento artefakt vymezuje důležité pojmy používané v rámci projektu. Poskytuje konzistentní soubor definice, aby nedocházelo k nedorozuměním. Členové projektu jej využívají, aby porozuměli termínům a specifikacím spojeným s projektem.

Každý slovník by měl ve svém úvodu obsahovat název, projekt, ke kterému náleží, verzi, datum, popis a autory.

5.4 Plán Iterací (Iteration Plan)

	Povinné	Doporučené	Externí
Vstupuje do:	Vytvoření vize (Develop vision) Nalezení aktérů a případů užití (Find Actors and Use Cases)	-	-

Plán iterací je časově sekvenční soubor činností a úkolů, které mají přidělené prostředky a zdroje, obsahuje úkoly v závislosti na iteraci. Je to jemnozrný plán. Plán iterací využívá vedoucí projektu a členové projektového týmu.

Pro plánování se obvykle využívá nějaký specializovaný SW (např. MS Project). Pro definici obsahu iterace, je zapotřebí: Plán Projektu, Aktuální stav projektu, seznam případů užití, které musí být dokončeny do konce iterace, seznam rizik, které je třeba řešit na konci iterace, seznam změn, které musí být začleněny do produktu. Každá iterace je uzavřena hodnocením. V plánu by měly být stanoveny hodnotící kritéria.

Plán iterací by měl obsahovat:

1. Klíčové milníky – podrobná schémata časové osy, na které je vidět kdy začíná testování, kdy beta verze, demo atd. pro každou iteraci

2. Cíle iterace – cíle a úkoly v rámci iterace, přiřazení odpovědných osob, podpůrné úkoly

5.5 Záznam o přezkoumání (Review record)

	Povinné	Doporučené	Externí
Výstup z:	Přezkoumání požadavků (Review Requirements)	-	-

Vytvářeno pro zachycení výsledků přezkoumání činnosti, v nichž je posuzován jeden nebo více projektových artefaktů. Slouží primárně pro zachycení výsledků a závěrů z kontrolní činnosti a určení jakékoliv akce vyplývající z revize.

Záznam by měl obsahovat tyto náležitosti:

1. Identifikaci projektu a typu revize - Identifikaci projektu a typu revize, například kód kontroly, požadavky na sledovatelnost revize atd.
2. Produkty práce, které budou předmětem a cílem této revize - Seznam prací, které budou předmětem tohoto přezkumu, a popis cíle revize.
3. Účastníci revize - Seznam osob, které se budou podílet na hodnocení a jejich role v průběhu jednání, např. moderátor, zapisovač, recenzent, autor.
4. Plán a lokace - Určit plán pro revizi. Uvést datum, čas a místo konání přezkumného jednání a také plán hodnotící činnosti.
5. Zjištěné problémy a doporučení pro řešení - Seznam problémů zjištěných při přezkumu. Hodnotící tým může vydávat doporučení k řešení problémů.
6. Seznam Stavů - Seznam všech akcí, jež jsou výsledkem revize. Měly by být uvedeny s identifikací vlastníka (zodpovědný za dokončení akce) a cílové datum. Tyto akce by měly vést k nápravě zjištěných problémů. Mohou zahrnovat stavy: Pokračovat v práci, vyvolání víceprací, založení změnového požadavku.
7. Záležitosti k posouzení projektového manažera - Určité problémy nebo anomálie, které by neměl řešit hodnotící tým, ale musí být předány k posouzení projektovému manažerovi.
8. Následná hodnocení - doporučení pro sledování (například, zda je nutná další revize) a jaké jsou potřeba další informace nebo data.
9. Záznam práce - Zachycuje počet hodin strávených při hodnocení, přípravě a vedení.

5.6 Seznam rizik (Risk list)

	Povinné	Doporučené	Externí
Vstupuje do:	Přidělení priorit případům užití (Prioritize Use Cases)	-	-

Slouží pro zachycení rizik, které by mohly způsobit neúspěch projektu. Seznam rizik uvádí v sestupném pořadí události, které by mohly vést k významným negativním důsledkům. Slouží jako kontaktní místo pro projektové činnosti, a tvoří základ, kolem kterého jsou seskupeny iterace.

Každý seznam rizik by měl obsahovat: Název projektu, verzi, historii revizí, datum, popis a autory.

V jeho obsahu by nemělo chybět:

1. Úvod – účel, rozsah, definice, reference
2. Přehled
3. Rizika – závažnost rizika, popis, dopady, indikátory, strategie snižování rizika, rezervní plán

5.7 Dokument softwarové architektury (Software architecture document)

	Povinné	Doporučené	Externí
Role:	Softwarový architekt	Softwarový architekt	
Vstupuje do:	Přidělení priorit případům užití (Prioritize Use Cases)	-	-
Výstup z:	Přidělení priorit případům užití (Prioritize Use Cases)		

Tento artefakt poskytuje komplexní stavební přehled o systému, pomocí několika různých architektonických názorů zobrazuje různé aspekty systému.

Dokument softwarové architektury obsahuje komplexní přehled architektury softwarového systému. Slouží jako komunikační médium mezi softwarovým architektem a dalšími členy projektového týmu.

5.8 Plán vývoje SW (Software development plan)

V tomto artefaktu jsou shromážděny všechny informace potřebné k řízení projektu. Popisuje přístup k vývoji SW a je to plán používaný na nejvyšší úrovni řízení.

Účelem plánu vývoje je shromáždit všechny informace potřebné k řízení projektu. Popisuje přístup k vývoji software na nejvyšší úrovni řízení.

Následující lidé používají software rozvojového plánu:

- * Vedoucí projektu - plánování projektu, harmonogramu a potřebné zdroje, sledování průběhu
- * Členové projektového týmu - porozumění tomu, co a kdy je třeba udělat

5.9 Požadavky na SW (Software Requirement)

	Povinné	Doporučené	Externí
Vstupuje do:	-	Vytvoření slovníku pojmů (Capture a common vocabulary)	-

Tento artefakt je specifikací pro podmínky nebo schopnosti, které musí systém splňovat.

Softwarové požadavky jsou popsány jako:

- * Schopnosti SW potřebné pro dosažení cíle
- * Schopnosti SW, které musí být splněny, aby byla naplněna smlouva, standard, specifikace nebo jiné formálně stanovené dokumentace

Požadavky na SW jsou zásadním artefaktem ve vývoji softwaru, ale v mnoha případech je pro ně typické, určitá podmnožina požadavků zůstává neúplně zdokumentována. RUP reaguje na toto znepokojení tím, že řídí vývoj softwaru v několika iteracích, což umožňuje spravovat důležité požadavky, které se objeví až v průběhu času.

Při vytváření tohoto artefaktu bychom měli zvážit různé aspekty artefaktu, včetně následujících:

- * Různých zájmových skupin a subjektů, které mohou mít své požadavky
- * Různé typy požadavků (kategorie, rozměry), které je třeba zvážit

5.10 Specifikace SW požadavků (Software Requirement specification)

	Povinné	Doporučené	Externí
Vstupuje do:	-	Vytvoření slovníku pojmů (Capture a common vocabulary)	-

Tento artefakt zachycuje požadavky na software pro kompletní systém nebo část systému.

Specifikace SW požadavků (Software Requirement specification (SRS)) se zaměřuje na sběr a organizaci všech požadavků okolí projektu. To je užitečné při sběru požadavků projektu ve formálním dokumentu.

Je důležité si uvědomit, že sběr požadavků lze najít v několika různých artefaktech a nástrojích. Z tohoto důvodu budeme shromažďovat požadavky na naše SRS v balíčku, který může být jednotný dokument nebo jako soubor různých dokumentů, ve kterých jsou uvedeny jednotlivé požadavky.

Následující lidé používají SRS:

- * Návrháři používají balíček SRS jako referenční při stanovení odpovědností, operací a atributů třídy a při nastavování tříd implementačního prostředí.
- * Realizátoři odkazují na balík SRS jako podklad při provádění tříd.
- * Project Manager se odkazuje na balíček SRS jako podklad pro plánování iterací.
- * Testeři používají balíček SRS jako podklad pro posuzování toho, co bude vyžadovat testy.

5.11 Požadavky zainteresovaných subjektů (Stakeholder requests)

	Povinné	Doporučené	Externí
Role:	Systémový analytik	Softwarový architekt	
Vstupuje do:	Vytvoření vize (Develop Vision) Nalezení aktérů a případů užití (Find Actors and Use Cases)	Vytvoření slovníku pojmů (Capture a common vocabulary)	-

Tento artefakt obsahuje všechny typy požadavků zainteresovaných stran (zákazníků, uživatelů, marketingového oddělení atd.) Může také obsahovat odkazy na všechny typy externích zdrojů, které musí brát systém v potaz.

Účelem tohoto artefaktu je zachytit všechny požadavky podané v rámci projektu a dokumentovat, jak byly tyto požadavky řešeny. Tyto informace mohou být shromažďovány v dokumentu nebo za pomoci nějakého SW nástroje a požadavky by měly být sledovány a schvalovány vedením ve schvalovacím procesu.

Příklady zdrojů zúčastněných stran jsou následující:

- * Výsledky rozhovorů se zúčastněnými stranami
- * Výsledky jednání ohledně požadavků vylákání a seminářů
- * Požadavek na změnu
- * Obchodní pravidla
- * Zákony a předpisy
- * Stávající systémy
- * Obchodní modely

Tento artefakt může být považován za dokončený, jsou-li kladně zodpovězeny následující otázky:

Byly správně nastaveny zúčastněné strany zapojené do přípravy tohoto artefaktu?

Byly pro tuto verzi systému vzaty v úvahu všechny historické požadavky?

Byl správně identifikován soubor informačních zdrojů?

Byly použity správné techniky získávání a shromažďování informací?

Odpovídá obsah tohoto artefaktu všem oblastem zájmu tohoto projektu?

Byly odpovídající požadavky zaneseny do řízení požadavků na změnu systému pro sledování?

5.12 Scénáře užití (Storyboard)

	Povinné	Doporučené	Externí
Role:	Systémový analytik		
Vstupuje do:	-	Vytvoření slovníku pojmů (Capture a common vocabulary)	-

Scénáře užití jsou logické a koncepční popisy funkcionality systému pro konkrétní situaci, včetně interakci mezi uživateli systému a systémem. Scénář "vypráví určitý příběh."

Některé příklady způsobů, kterými lze zobrazit scénáře patří:

- * Náčrtky nebo obrázky
- * Bitmapy z kreslicího nástroje
- * Výpravné kartičky
- * Snímky v programu PowerPoint
- * Snímky obrazovek (existuje-li uživatelské rozhraní nebo prototyp uživatelského rozhraní)

Je důležité si uvědomit, že hlavním účelem scénářů užití je pochopení celkového průtoku informací a interakce uživatele se systémem, ne prototyp nebo testování vzhledu uživatelského rozhraní. Scénář by se neměl vztahovat na uživatelské rozhraní (to by mělo být zahrnuto do návrhu uživatelského rozhraní (User-Interface Prototyp)).

5.13 Doplnující specifikace (Supplementary Specifications)

	Povinné	Doporučené	Externí
Role:	Systémový analytik	-	
Vstupuje do:	-	Vytvoření slovníku pojmů (Capture a common vocabulary)	-

Tento artefakt zachycuje požadavky na systém, které nejsou snadno zachyceny v požadavcích na chování systému (např. use-case specifikace).

Doplnující údaje zachycují systémové požadavky, které nejsou jinak snadno zachytitelné v případech užití use-case modelem. Mezi tyto požadavky patří:

- * Právní a regulační požadavky a aplikace standardů
- * Kvalita vlastností systému, včetně použitelnosti, spolehlivosti, výkonu a požadavků na podporu
- * Požadavky na operační systémy a prostředí, kompatibilitu s jiným softwarem a konstrukční omezení

5.14 Use-Case model

	Povinné	Doporučené	Externí
Role:	Systémový analytik		
Vstupuje do:	-	Vytvoření slovníku pojmů (Capture a common vocabulary)	-
Výstup z:	Nalezení aktérů a případů užití (Find Actors and Use Cases)		

Use-case model je model systému popisující jeho funkce a jeho okolí. Slouží jako smlouva mezi zákazníkem a vývojáři. Případy použití slouží jako sjednocující nit skrz vývojem systému. Stejný use-case model je výsledkem disciplíny Requirements a slouží jako vstup do disciplín Analysis & Design a Test.

Uživatelé a další systémy, které mohou interagovat se systémem jsou definováni jako aktéři. Aktéři pomáhají vymezit systém a poskytují jasnou představu, jaká je funkcionality daného systému. Případy užití jsou vyvinuty na základě potřeb aktérů. To zajišťuje, že systém bude splňovat očekávání uživatelů.

5.14.1 Jak vyvíjet Use-Model?

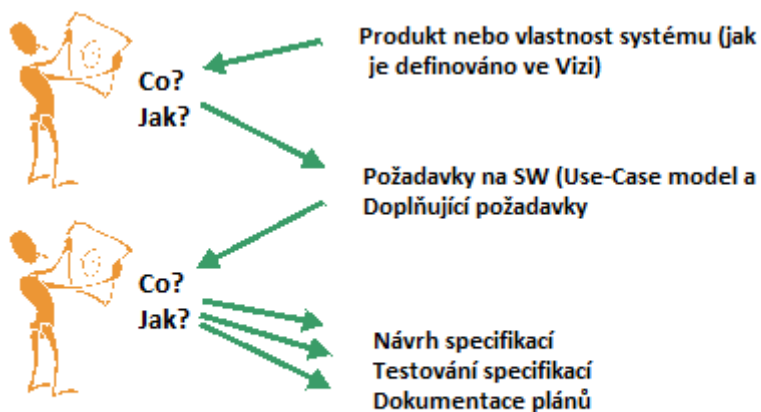
Aktéři a případy užití jsou nalezeni na základě požadavků zákazníků a potenciálních uživatelů. Jakmile jsou aktéři a případy užití nalezeny, musí být podrobně popsán každý tok událostí a jak se systém chová v jednotlivých případech užití. Use-case model je dokončen poté, co je zkontrolován (včetně popisů případů užití) vývojáři i uživateli a obě skupiny souhlasí s tím, co by měl systém dělat.

5.14.2 Nefunkcionální požadavky

Nefunkcionální požadavky jsou často klasifikovány jako použitelnost, spolehlivost, výkon a udržitelnost. Často jsou to požadavky, které stanovuje nutnost dodržování právních a regulačních požadavků. Mohou to být také konstrukční omezení vzhledem k použitelnosti s určitým OS nebo problémy s kompatibilitou. Obecně lze říci, že každý požadavek, který neumožňuje více než jeden návrh řešení by měl být považován za omezení. Mnoho nefunkcionálních požadavků se vztahuje na jednotlivé use case a jsou zachyceny ve vlastnostech use case.

5.14.3 Dilema Co a Jak

Při modelování use case je důležité zjistit, na jaké se máme pohybovat úrovni podrobnosti případů užití. Musíme rozhodnout dilema, kde je začátek vlastností systému a případů užití a kde končí případy užití a začíná design systému? Často říkáme, že případy užití a požadavky na SW by měly určovat, **co** systém dělá ale ne **jak** to dělá. Na obrázku je vidět, že cíl jedné osoby je počáteční bod osoby jiné. V závislosti na okolnostech je potřeba rozhodnout, v jakém kontextu je ono co a jak myšleno. To je třeba brát v potaz při rozhodování, který detail zařadit do use-case modelu a který již ne.



Obrázek 10 - Dilema Co a Jak

5.15 Návrh uživatelského rozhraní (User-Interface Prototype)

	Povinné	Doporučené	Externí
Vstupuje do:	-	Upřesnění SW požadavků (Detail the Software Requirements)	-

Tento artefakt je vzor uživatelského rozhraní. Používá se pro prozkoumání a / nebo ověření vzhledu uživatelského rozhraní. Významem tohoto artefaktu je prozkoumání vhodnosti navrženého uživatelského rozhraní. Pomáhá překlenout propast mezi tím, co je požadováno a popisováno a co je realizovatelné. Hlavním účelem vytvoření tohoto návrhu je možnost vyzkoušet si nanečisto vzhled UI včetně jeho použitelnosti ještě před tím, než začíná jeho vývoj. Díky tomu ušetříme náklady na vynaložený čas.

5.16 Vize (Vision)

	Povinné	Doporučené	Externí
Role:	Systémový analytik		
Vstupuje do:	-	Vytvoření slovníku pojmů (Capture a common vocabulary) Vytvoření vize (Develop Vision) Nalezení aktérů a případů užití (Find Actors and Use Cases)	-
Výstup z:	Vytvoření vize (Develop Vision)		

Vize definuje pohled zainteresovaných subjektů na produkt, který má být vyvinut, z hlediska potřeb zúčastněných stran a požadovaných vlastností. Obsahuje přehled základních požadavků a tak poskytuje základ pro podrobnější technické požadavky.

Tento artefakt poskytuje vysokoúrovňový (někdy i smluvní) podklad pro podrobnější technické požadavky, které jsou viditelné pro zainteresované subjekty. Zachycuje "podstatu" předpokládaného řešení v podobě vysokoúrovňových požadavků a konstrukčních omezení, které dávají čtenáři možnost na systému, který má být vyvinut, pohled z perspektivy. Úzce souvisí s artefaktem: Business Case. Komunikuje základní "proč a co" pro projekt a je základním měřítkem proti kterému by se všechny budoucí rozhodnutí měly potvrdit.

Vize je ze své podstaty dílem z pohledu zákazníků, se zaměřením na základní rysy systému a přijatelné úrovně kvality. Vize by měla obsahovat popis toho, jaké funkce mají být zahrnuty a i ty, které zahrnuty nebyly. Měla by také stanovit provozní kapacity (množství, dobu odezvy, přesnost), uživatelské profily (kdo bude používat systém), a mezioperační rozhraní se subjekty mimo hranice systému.

6 Role

Role	Úkoly	Artefakty
Specifikátor požadavků - Requirements Specifier	Příprava doplňující specifikace Specifikace SW požadavků Specifikace use-case	Případy užití Požadavky na SW SW specifikace
Softwarový architekt	Přiřazení priorit případům užití	Dokument SW architektury
Systémový analytik - System analyst	Stanovení společných pojmů Vytvoření plánu požadavků Vytvoření doplňující specifikace Vytvoření vize Získat požadavky zainteresovaných subjektů Určení aktérů a případů užití Vytvoření Use-Case modelu	Slovník pojmů Atributy požadavků Plán správy požadavků Požadavky zainteresovaných subjektů Scénáře užití Doplňková specifikace SW Use-Case model Vize
Technický posuzovatel - Technical reviewer	Revize požadavků	Záznam o přezkoumání

7 Závěr

7.1 Vlastní zhodnocení

Před tím, než jsem zpracovávala disciplínu Requirements v této práci, jsem si nikdy neuvědomila rozsáhlost a šíři záběru této metodiky. Na metodice jsem ocenila praktické příklady, ukázky a návody k jednotlivým aktivitám a artefaktům. Téměř u každé aktivity byl uveden seznam činností, které by měly být provedeny, aby bylo dosaženo nejefektivnějších výsledků a u artefaktů byl často nastíněn výsledný dokument a co by mělo být jeho obsahem.

Na druhou stranu na mě však disciplína Requirements (a předpokládám, že tímto způsobem budou zpracovány i ostatní disciplíny metodiky) působila velice složitě. K jejímu plnému pochopení a následné aplikace bude zapotřebí nemalého času a soustředěnosti. Soudím tak dle svých zkušeností, kdy pochopení základních principů této jedné části a jejich charakteristika mi zabralo téměř dvojnásobek času oproti prvotnímu očekávání.

V této práci jsem nemohla uvádět veškeré dostupné návody a příklady, protože mi to stanovený rozsah práce neumožňoval. Myslím si však, že pro plné pochopení principů metodiky (resp. disciplíny Requirements) jsou právě tyto části metodiky nezbytné.

Pokud bych byla součástí vývojového týmu nějakého rozsáhlého softwarového projektu, určitě bych alespoň některé (pokud ne všechny) best practises popisované v metodice RUP (v mém případě disciplíny Requirements) využila. Jsem přesvědčena, že by to vedlo ke zvýšení šance na úspěšnost projektu.

8 Zdroje

[1] <http://objekty.vse.cz/Objekty/Rup2>

[2] <https://kitscm.vse.cz/RUP/>

9 Přílohy

9.1 Terminologický slovník

název aj	název čj
Actor	Aktér
business case	business case
capture a common vocabulary	Vytvoření slovníku pojmů
detail a use case	upřesnění případů užití
Develop supplementary specification	doplnění specifikace systému
elicit stakeholder request	zjištění požadavků zainteresovaných subjektů
Glossary	Slovník pojmů
change request	Změnový požadavek
Iteration Plan	Plán iterací
Prioritize use cases	upřednostňované případy užití
Refine the System Definition	Upřesnění vymezení systému
Requirements Specifier	Specifikátor požadavků
review record	Záznam o přezkoumání
Review Requirements	přezkoumání požadavků
risk list	seznam rizik
software architect	Softwarový architekt
software architecture document	Dokumentace SW architektury
Software requirements Specification	Specifikace SW
Software requirement	Požadavky na SW
Stakeholder requests	Požadavky zainteresovaných subjektů
Storyboard	Scénář užití
Supplementary specification	Doplňující požadavky
System analyst	Systémový analytik
technical reviewer	Technický recenzent
Use Case	Případ užití
Use Case Model	Use Case Model
use case package	use case package
User interface prototype	Návrh uživatelského rozhraní
Vision	Vize