

Vysoká škola ekonomická v Praze



# POROVNÁNÍ RUP DISCIPLÍNY REQUIREMENTS S ODPOVÍDAJÍCÍMI PA V CMMI-DEV v.1.3 A PROCESY V ISO/IEC 12207

4IT421 ZLEPŠOVÁNÍ PROCESŮ BUDOVÁNÍ INFORMAČNÍCH SYSTÉMŮ

AUTOŘI: BC. JAN KNETTIG, BC. ONDREJ ZLOCHA  
DEN ODEVZDÁNÍ: 1.1.2016

#### ABSTRAKT:

Práce si klade za úlohu porovnat disciplínu Requirements z metodiky Rational Unified Process, s odpovídajícími procesními oblastmi z modelu CMMI-DEV v.1.3 a procesy z normy ISO/IEC 12207. V první části práce jsou představené jednotlivé výše uvedené metodiky resp. modely a v druhé části je provedeno porovnání na základě vybraných kritérií. Porovnání je vykonáno na základě následujících kritérií: Rozsah dané metodiky, role v dané metodice, Podrobnost popisu procesů dané metodiky a předpokládané výstupy v dané metodice. V závěrečné části práce je vyhodnocen výsledek provedeného porovnání.

#### KLÍČOVÁ SLOVA:

RUP Requirements, Porovnání, Requirements Analysis, Requirements Management, Správa požadavků

# OBSAH

|                                                                    |           |
|--------------------------------------------------------------------|-----------|
| <b>ÚVOD</b>                                                        | <b>4</b>  |
| <b>RUP</b>                                                         | <b>5</b>  |
| <b>OBEZNĚ O RUP</b>                                                | <b>5</b>  |
| <b>DISCIPLÍNA REQUIREMENTS JAKO SOUČÁST RUP</b>                    | <b>5</b>  |
| AKTIVNÍ SPRÁVA POŽADAVKŮ                                           | 6         |
| <b>VLASTNOSTI KVALITNÍHO POŽADAVKU</b>                             | <b>6</b>  |
| <b>CMMI DEV 1.3</b>                                                | <b>8</b>  |
| <b>OBEZNĚ O CMMI DEV 1.3</b>                                       | <b>8</b>  |
| <b>VÝVOJ POŽADAVKŮ (REQUIREMENTS DEVELOPMENT)</b>                  | <b>9</b>  |
| 1. VYVINUTÍ ZÁKAZNICKÝCH POŽADAVKŮ (DEVELOP CUSTOMER REQUIREMENTS) | 9         |
| 2. VYVINUTÍ POŽADAVKŮ NA PRODUKT (DEVELOP PRODUCT REQUIREMENTS)    | 10        |
| 3. ANALÝZA A OVĚŘENÍ POŽADAVKŮ (ANALYZE AND VALIDATE REQUIREMENTS) | 11        |
| <b>SPRÁVA POŽADAVKŮ (REQUIREMENTS MANAGEMENT)</b>                  | <b>13</b> |
| <b>ISO/IEC 12207</b>                                               | <b>15</b> |
| <b>OBEZNĚ O ISO/IEC 12207</b>                                      | <b>15</b> |
| SOFTWARE REQUIREMENTS ANALYSIS PROCESS                             | 15        |
| PŘEDPOKLÁDANÝ VÝSLEDEK ÚSPĚŠNÉ IMPLEMENTACE                        | 15        |
| <b>POROVNÁNÍ DISCIPLÍN</b>                                         | <b>17</b> |
| <b>ZVOLENÁ KRITÉRIA PRO POROVNÁNÍ</b>                              | <b>17</b> |
| ROZSAH                                                             | 17        |
| ROLE                                                               | 17        |
| PODROBNOST POPISU PROCESU                                          | 17        |
| VÝSTUPY                                                            | 18        |
| <b>RUP</b>                                                         | <b>19</b> |
| <b>CMMI-DEV v.1.3</b>                                              | <b>20</b> |
| <b>ISO/IEC 12207:2008</b>                                          | <b>21</b> |
| <b>ZÁVĚR</b>                                                       | <b>22</b> |
| <b>ZDROJE</b>                                                      | <b>23</b> |

## ÚVOD

Práce si klade za úlohu porovnat disciplínu Requirements z metodiky Rational Unified Process, s odpovídajícími procesními oblastmi z modelu CMMI-DEV v.1.3 a procesy z normy ISO/IEC 12207. V první části práce jsou představené jednotlivé výše uvedené metodiky resp. Modely, a to konkrétně zejména jejich zpracování oblasti Requirements a v druhé části je provedeno jejich porovnání na základě vybraných kritérií. Porovnání je vykonáno na základě následujících kritérií: Rozsah dané metodiky, role v dané metodice, Podrobnost popisu procesů dané metodiky a předpokládané výstupy v dané metodice. Při vytváření kritérií pro hodnocení byl použit jako základ systém METES, který byl vyvinut k evaluaci metodik. Z tohoto systému byla vybrána a přizpůsobena některá kritéria, která byla vyhodnocena jako relevantní pro porovnání oblasti Requirements v různých metodikách. Na základě hodnot definovaných pro jednotlivá kritéria jsou následně přiřazovány hodnoty pro jednotlivé metodiky na základě zjištění, která provedli autoři této práce. V závěru práce je uvedeno celkové porovnání uvažovaných metodik na základě celkového součtu bodů, které jednotlivé metodiky získaly při hodnocení výše uvedených kritérií.

# RUP

## OBEZNĚ O RUP

Rational Unified Process (dále jen RUP) je metodika vývoje software vytvořena společností Rational Software Corporation. RUP vychází z kolekce osvědčených praktik a postupů při vývoji software. Je možné ji použít pro jakýkoliv rozsah projektu a díky vysoké rozsáhlosti RUP je vhodné přizpůsobit metodiku specifickým potřebám.

## DISCIPLÍNA REQUIREMENTS JAKO SOUČÁST RUP

- **Disciplína** - Kolekce příbuzných aktivit, které se vztahují ke stejné oblasti zájmu v rámci celého projektu.

Základem metodiky RUP je šest základních částí, tzv. „Best practises“ využívaných při vývoji software:

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

Zdroj: (1)

Cílem Requirements disciplíny je popsat, co by měl systém zvládat a umožnit vývojářům a zákazníkovi shodnout se na tomto popisu požadavků. Z toho vyplývají následující účely:

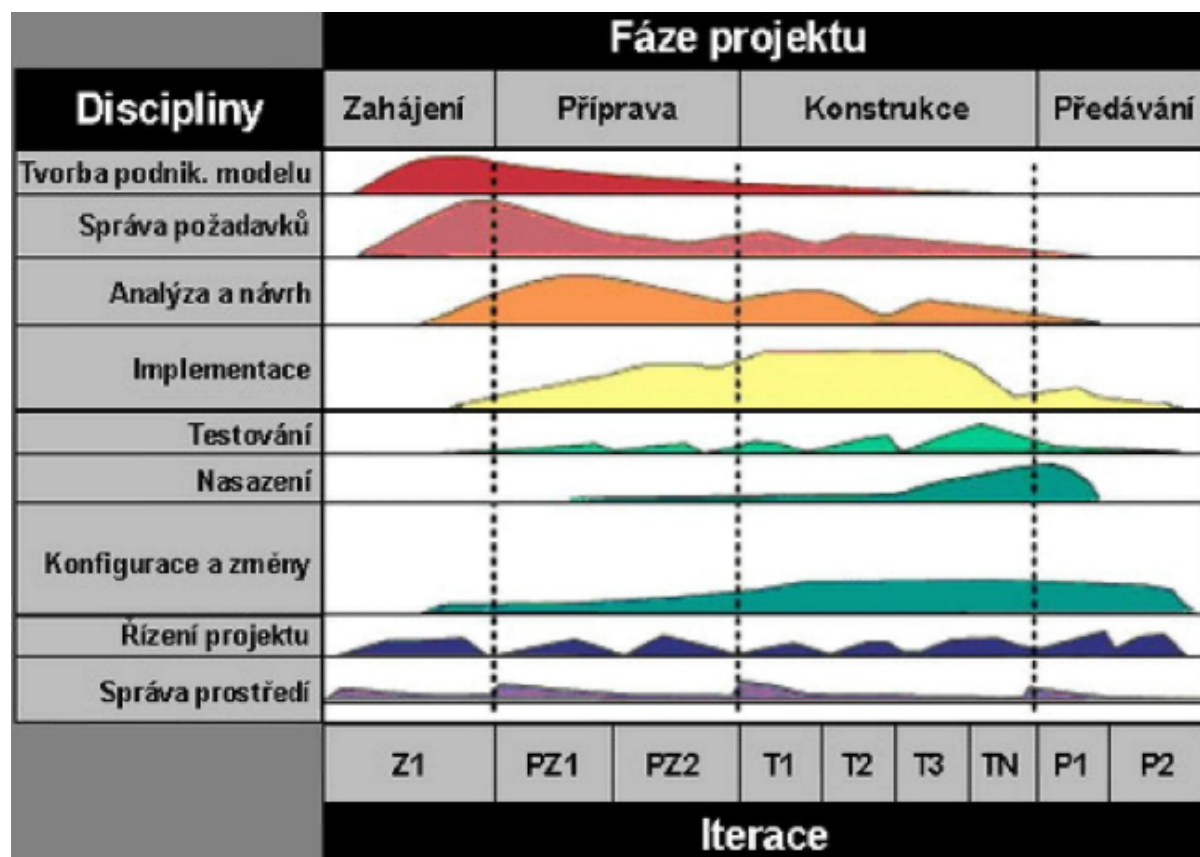
- Založit a udržovat závaznou dohodu mezi realizačním týmem, zákazníkem a zaujatým osobám (stakeholders) o tom, co má systém dělat.
- Poskytnout vývojářům lepší pochopení systémových požadavků.
- Definovat a vyznačit hranice systému.
- Poskytnout základy pro plánování technického obsahu jednotlivých iterací.
- Poskytnout základy pro odhad finanční a časové náročnosti vývoje systému.
- Definovat uživatelské rozhraní systému se zaměřením na potřeby a cíle jeho uživatelů.
- **Základní role** v disciplíně Requirements:
  - Specifikátor požadavků (Připravuje doplňkovou specifikaci, Specifikuje softwarové požadavky, Specifikuje UseCase)
  - Softwarový architekt (Priorizuje UseCase)
  - Systémový analytik (Identifikuje užívané pojmy, Připravuje plán správy požadavků, Připravuje vizi, Zpravuje závislosti, Identifikuje aktéry a UseCase)
  - Technický posuzovatel (Posuzuje požadavky)

Zdroj: (2)

- **Předpokládané výstupy:** Slovník pojmů, Aktéři, Use Case Model, Vize, Požadavky zainteresovaných subjektů, Scénáře užití, Doplňující požadavky, Požadavky na software, Softwarová architektura

## AKTIVNÍ SPRÁVA POŽADAVKŮ

U tradičního (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. Díky tomu má disciplína Requirements svou úlohu v podstatě v každé ze čtyř fází projektu. (Znázorněné na následujícím obrázku).



Obrázek 1: Fáze projektu dle RUP, Zdroj: (7)

## VLASTNOSTI KVALITNÍHO POŽADAVKU

V metodice RUP jsou definovány i základní vlastnosti požadavků, které by měl každý kvalitní požadavek splňovat:

- Korektnost - Požadavek musí být pravdivý ohledně toho, co má systém dělat.
- Kompletnost - Požadavek musí popisovat všechny požadované aspekty systému a zároveň všechny požadavky musí popisovat celkový systém.
- Konzistence - Požadavek nesmí být v konfliktu s jiným požadavkem.
- Jednoznačnost - Požadavek musí mít právě jednu interpretaci.
- Ověřitelnost - Požadavek musí být ověřitelný, či měřitelný.

- Pochopitelnost - Požadavek musí být pochopitelný nejenom uživateli ale i vývojáři
- Dohledatelnost - Původ anebo důvod vzniku požadavku musí být dohledatelný.
- Potřebnost - Pokud systém vyhovuje požadavkům uživatelů bez toho, aby byl nějaký požadavek realizován, tak nebyl potřebný.
- Uskutečnitelnost - Požadavek musí být realizovatelný v rámci dohodnutého rozpočtu a termínu.
- Neredundantnost- Mezi požadavky nesmí existovat duplicity.
- Řiditelnost dle důležitosti a stability - Požadavky musí být řiditelné podle klientské, či vývojářské důležitosti anebo stability.

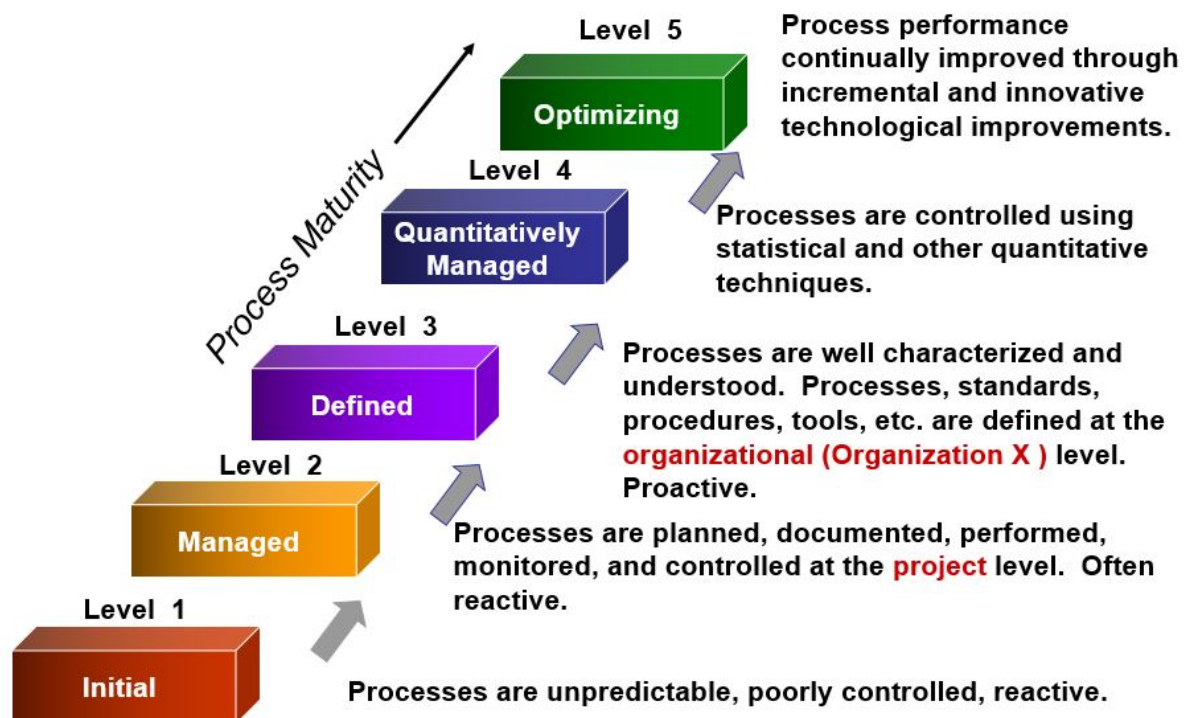
Zdroj: (2)

## CMMI DEV 1.3

### Obecně o CMMI DEV 1.3

#### Capability Maturity Model Integration

“CMMI je model kvality organizace práce určený pro vývojové týmy. Definuje procesní oblasti, které musí tým realizovat a cíle, kterých musí v každé oblasti dosahovat.” (3)



Obrázek 2: Definice stádia zralosti procesu dle CMMI. Zdroj: (4)

Osvětlení jednotlivých stádií:

1. Počáteční (Initial): Týmy na této úrovni definované procesy nevykonávají nebo vykonávají pouze částečně
2. Řízená (Managed): Je stanoveno řízení projektů a činnosti jsou plánovány
3. Definovaná (Defined): Postupy jsou definovány, dokumentovány a řízeny
4. Kvantitativně řízení (Quantitatively Managed): Produkty i procesy jsou řízené kvantitativně.
5. Optimalizující (Optimizing): Tým soustavně optimalizuje své činnosti

Zdroj: (3)

CMMI problematiku požadavků (requirements) rozděluje na 2 části:

1. Vývoj požadavků (Requirements Development)
2. Správa požadavků (Requirements Management)

## Vývoj požadavků (Requirements Development)

Cíl: Zjistit, Analyzovat a Stanovit požadavky zákazníka na produkt a produktové komponenty

### Specific Goal and Practice Summary

#### SG 1 Develop Customer Requirements

SP 1.1 Elicit Needs

SP 1.2 Transform Stakeholder Needs into Customer Requirements

#### SG 2 Develop Product Requirements

SP 2.1 Establish Product and Product Component Requirements

SP 2.2 Allocate Product Component Requirements

SP 2.3 Identify Interface Requirements

#### SG 3 Analyze and Validate Requirements

SP 3.1 Establish Operational Concepts and Scenarios

SP 3.2 Establish a Definition of Required Functionality and Quality Attributes

SP 3.3 Analyze Requirements

SP 3.4 Analyze Requirements to Achieve Balance

SP 3.5 Validate Requirements

*Obrázek 3: Etapy vývoje požadavků dle CMMI, Zdroj: (5)*

### 1. Vyvinutí zákaznických požadavků (Develop Customer Requirements)

Potřeby, očekávání, omezení a rozhraní jsou sesbírány od zaujatých osob (stakeholders) a přeloženy do zákaznických požadavků.

Stakeholders (zaujaté osoby) jsou *zákazníci, koncoví uživatelé, dodavatelé, tvůrci, testéři, výrobci, logistická podpora*.

CMMI nedefinuje žádné role spojené přímo s oblastí vývoje či správy požadavků. Je tomu tak možná proto, že na definici požadavků se podílejí všechny zainteresované strany.

Požadavky by měly být jasně definovány a porozuměny, což je iterativní proces během celého života projektu.

#### 1.1 Zjištění potřeb(Elicit Needs)

*Zjistit potřeby, očekávání, omezení a rozhraní pro všechny fáze produktového života.*

Příklady technik použitých k zjištění potřeb dle CMMI:

- Dotazníky, interview, scénáře získané od koncových uživatelů
- Prototypy a modely
- Brainstorming
- Průzkum trhu
- Případy užití
- Uživatelské příběhy
- Průzkumy spokojenosti zákazníků

Výstup:

- Výsledky aktivit získávání požadavků

## **1.2 Transformace potřeb do požadavků (Transform needs into requirements)**

*Transformace potřeb, očekávání, omezení a rozhraní do prioritizovaných zákaznických požadavků.*

Výstupy:

- Prioritizované zákaznické požadavky

## **2. VYVINUTÍ POŽADAVKŮ NA PRODUKT (DEVELOP PRODUCT REQUIREMENTS)**

Zákaznické požadavky jsou zjemněny a propracovány, aby byly vyvinuty požadavky na produkt a jeho komponenty.

Požadavky na produkt a jeho komponenty se zabývají potřebami spojenými s každou životní fází produktu.

### **2.1 Stanovení požadavků na produkt a jeho komponenty (Establish Product and Product Component Requirements)**

*Stanovení a údržba požadavků na produkt a jeho komponenty, které jsou založeny na zákaznických požadavcích.*

Výstupy:

- Požadavky na produkt
- Požadavky na produktové komponenty
- Architektonické požadavky, které specifikují nebo omezují vztahy mezi produktovými komponentami

### **2.2 Přiřazení požadavků na produktové komponenty (Allocate Product Component Requirements)**

*Přiřazení požadavků pro každou produktovou komponentu.*

Výstupy:

- Přehled alokace požadavků
- Designová omezení
- Vztahy mezi získanými požadavky

### **2.3 Identifikace požadavků na rozhraní (Identify Interface Requirements)**

*Jsou identifikována rozhraní mezi funkcemi (nebo mezi objekty nebo dalšími logickými entitami).*

Výstupy:

- Požadavky na rozhraní

### 3. ANALÝZA A OVĚŘENÍ POŽADAVKŮ (ANALYZE AND VALIDATE REQUIREMENTS)

Požadavky jsou analyzovány a ověřovány, aby bylo zajištěno, že výsledný produkt bude fungovat dle očekávání v použitém prostředí.

Zvažované faktory: proveditelnost, potřeby mise, cenová omezení, potenciální velikost trhu, strategie získání trhu

#### 3.1. Stanovení konceptů a scénářů provozu (Establish Operational Concepts and Scenarios)

*Koncepty a scénáře provozu slouží k naplánování fungování produktu a jeho komponent bez ohledu na technickou stránku věci.*

Výstupy:

- Provozní koncept
- Vývoj produktu nebo produktové komponenty, instalace, provoz, správa a podpůrné koncepty
- Koncept zrušení
- Případy užití
- Časové scénáře

#### 3.2 Definice atributů vyžadované funkcionality a kvality (Establish a Definition of Required Functionality and Quality Attributes)

*Funkcionalita a atributy kvality definovány pomocí funkční analýzy – analýza scénářů užití s cílem popsat, co má produkt dělat.*

Výstupy:

- Definice vyžadovaných funkčních a kvalitativních atributů
- Diagramy aktivity a případy užití
- Funkční architektura

#### 3.3 Analýza požadavků (Analyze Requirements)

*Analýza požadavků, aby bylo zjištěno, zda jsou potřebné a dostačující.*

Výstupy:

- Reporty chyb v požadavcích
- Navrhnuté změny v požadavcích
- Klíčové požadavky
- Metriky technického výkonu

### **3.4 Analýza požadavků pro zajištění rovnováhy (Analyze Requirements to Achieve Balance)**

*Analýza požadavků pro dosažení rovnováhy mezi potřebami a omezeními zaujatých osob.*

Výstupy:

- Posouzení nebezpečí spojených s požadavky

### **3.5 Ověření požadavků (Validate Requirements)**

*Ověřit požadavky, aby bylo zajištěno, že výsledný produkt bude fungovat, jak bylo zamýšleno, v konečném uživatelském prostředí*

Příklady technik použitých k ověření:

- Analýza
- Simulace
- Prototypování
- Demonstrace

Výstupy:

- Záznam o metodách analýzy a výsledky

## SPRÁVA POŽADAVKŮ (REQUIREMENTS MANAGEMENT)

Cíl: „Spravovat požadavky na produkt a produktové komponenty a zajistit soulad mezi těmito požadavky, plány projektu a jeho výstupy.“ (5)

Požadavky jsou spravovány a jsou identifikovány nekonzistence vůči projektovým plánům a výstupům.

Aktuální a odsouhlasený seznam požadavků udržován pomocí:

- Zpracovávání všech změn do požadavků
- Udržování vztahů mezi požadavky, projektovými plány a výstupy
- Zajišťování souladu mezi požadavky, projektovými plány a výstupy
- Dělání opravných opatření

### 1.1 Porozumění požadavkům (Understand Requirements)

*Je třeba vyvinout porozumění významu požadavků na základě konzultace s jejich tvůrci*

Příklady kritérií pro posouzení a akceptaci požadavku:

- Jasně a správně uveden
- Kompletní
- Konzistentní mezi sebou
- Unikátně identifikovaný
- Ověřitelný
- Sledovatelný
- Dosažitelný
- Spojený s hodnotou businessu
- Identifikovaný jako priorita pro zákazníka

Výstupy:

- Seznam kritérií pro rozlišení vhodných poskytovatelů požadavků
- Kritéria pro posouzení a přijetí požadavků
- Seznam schválených požadavků

### 1.2 Získání závazku k požadavkům (Obtain Commitment to Requirements)

*Je důležité zajistit si souhlas s požadavky a jejich změnami u těch, kteří je vykonávají*

Výstupy:

- Posouzení dopadu požadavků
- Dokumentované závazky vůči požadavkům a jejich změnám

### **1.3 Správa změn požadavků (Manage Requirements Changes)**

*V průběhu projektu může dojít ke změnám. Je důležité znát zdroj a důvod změny a být schopen posoudit její důležitost.*

Výstupy:

- Požadavek na změnu požadavků
- Zpráva o dopadu změny v požadavku
- Status požadavku
- Databáze požadavků

### **1.4 Zachování obousměrné dohledatelnosti požadavků (Maintain Bidirectional Traceability of Requirements)**

*Slouží k zajištění, že všechny hlavní požadavky i všechny jejich úpravy jsou implementovány a je možné dohledat jejich zdroj a vazbu na produkty*

Výstupy:

- Matice dohledatelnosti požadavků
- Sledovací systém pro požadavky

### **1.5 Zajištění souladu mezi projektem a požadavky (Ensure Alignment Between Project Work and Requirements)**

*Vzniklé inkonzistence mezi požadavky, projektovými plány a produkty je třeba objevit a zahájit opravné akce*

Výstupy:

- Dokumentace inkonzistencí mezi požadavky, plány projektu a výstupy, včetně zdrojů a podmínek
- Opravná opatření

# ISO/IEC 12207

## OBECEŇ O ISO/IEC 12207

Je to mezinárodní standard pro procesy životního cyklu software, jeho cílem je být standardem, který definuje všechny úlohy požadované pro vývoj a údržbu software. (6)

V této práci jsme pracovali s novější verzí z roku 2008, která má název

***ISO/IEC 12207:2008 Systems and software engineering – Software life cycle processes***

Co se týče rozsahu uvedeného dokumentu, je to přibližně 140 stran včetně obsahu a dodatečných tabulek.

Z procesů, které popisuje, jsme pro naši práci z hlediska vhodnosti zvolili ***Software Requirements Analysis Process***, který se nejvíce přibližuje svým obsahem k disciplíně Requirements z metodiky RUP. Nutno poznamenat, že samotný rozsah tohoto procesu je v porovnání s ostatními srovnávanými procesy výrazně nižší, tím ho i při závěrečném srovnání nebude možné porovnat na základě všech požadovaných kritérií.

## *SOFTWARE REQUIREMENTS ANALYSIS PROCESS*

Zařaduje se mezi procesy nižší úrovně procesů implementace softwaru. Jeho primárním účelem je založit požadavky pro všechny části systému. Na rozdíl od ostatních dvou srovnávaných modelů nemá přesnou definici rolí, požadovaných vstupů ani výstupů. Popisuje jenom požadovaný „výsledek“ úspěšné implementace tohoto procesu. Z těchto hlavních předpokládaných výsledků uvedeme jenom některé primární:

- Všechny požadavky jsou přiřazeny jednotlivým částem systému a jejich rozhraní jsou definována
- Požadavky jsou analyzované z hlediska správnosti a testovatelnosti
- Konzistence a sledovatelnost je zajištěna mezi softwarovými a systémovými požadavky
- Požadavky jsou schvalované a je možné je měnit podle potřeby
- Tyhle případné změny jsou vyhodnoceny z hlediska nákladů, časového plánu a technického vplyvu na systém

## *PŘEDPOKLÁDANÝ VÝSLEDEK ÚSPĚŠNÉ IMPLEMENTACE*

Tenhle proces dále stručně popisuje jenom tři základní úlohy, které by implementátor softwaru měl dodržet. Jsou to:

- Implementátor by měl založit a zdokumentovat požadavky softwaru, zahrnující i specifikace vlastností kvality, popsané v několika bodech. Sem patří např. (Funkčnost a schopnost systému, výkon, popis prostředí, požadovaná kvalifikace pro práci, bezpečnostní předpisy....)

- Implementátor by měl vyhodnotit požadavky na základě uvedených kritérií. Výsledky tohoto vyhodnocení by měly být zdokumentované. Kritéria zahrnují sledovatelnost, externí a interní konzistenci, testovatelnost, uskutečnitelnost samotného designu software a uskutečnitelnost práce a následné údržby software.
- Poslední popsany bod uvádí, že by implementátor měl poskytnout a provádět následné vyhodnocení v souladu s detailně popsáním procesem „Software Review Process“, který je také součástí normy ISO/IEC 12207 (6)

# POROVNÁNÍ DISCIPLÍN

## ZVOLENÁ KRITÉRIA PRO POROVNÁNÍ

### *ROZSAH*

Kritérium *rozsah* definujeme jednoduše jako množství stran, na kterých je hodnocená oblast Requirements v jednotlivých metodikách popsána.

- 0: Oblast není vůbec popsána
- 1: Oblast popsána na nejmenší počet stran
- 2: x
- 3: Oblast popsána na středním počtu stran
- 4: x
- 5: Oblast je z hodnocených metodik popsána na nejvíce stranách

### *ROLE*

Kritérium *role* porovnává, zda jsou pro oblast Requirements v jednotlivých metodikách definovány role a jak podrobně jsou definovány.

- 0: Role vůbec nejsou zmíněny
- 1: Role jsou naznačeny jako vyplývající ze zaujatých osob (stakeholders)
- 2: x
- 3: Role jsou vyjmenovány
- 4: Role jsou vyjmenovány a popsány
- 5: Role jsou definovány – je jasné určeno, čím se která role zabývá v každé fázi oblasti Requirements

### *PODROBNOST POPISU PROCESU*

V definici kritéria *podrobnost popisu procesu* vycházíme z jeho formulace v rámci systému METES. Jedná se tedy o kritérium, které hodnotí, jak podrobně je oblast Requirements v rámci jednotlivých metodik zpracována.

- 0: Oblast není definována v rámci metodiky
- 1: Je definován cíl oblasti, oblast popsána
- 2: Popsány výstupy oblasti
- 3: Popsány navíc činnosti
- 4: Popsány navíc role
- 5: Krom cíle, výstupů, činností a rolí jsou rovněž definovány metriky a omezující podmínky

## VÝSTUPY

Kritérium *výstupy* porovnáváme jako absolutní počet výstupů definovaných v rámci oblasti Requirements v dané metodice.

Platí zde podobný systém hodnocení jako pro hodnocení kritéria *role*. Hodnoty tedy jsou:

- 0: Žádné výstupy nejsou definovány
- 1: V rámci hodnocených metodik je počet definovaných výstupů nejmenší
- 2: x
- 3: V rámci hodnocených metodik je počet definovaných výstupů střední
- 4: x
- 5: V rámci hodnocených metodik je počet definovaných výstupů nejvyšší

## RUP

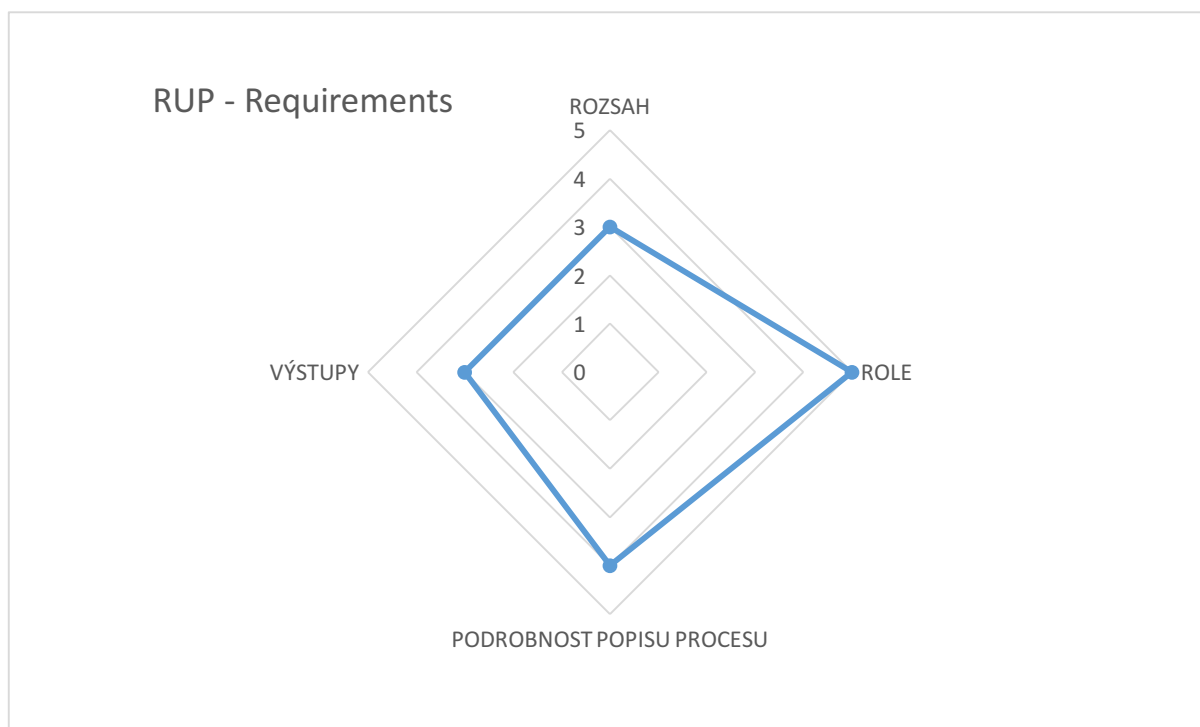


Diagram 1: RUP – hodnoty kritérií

### Komentář k diagramu:

Z výše uvedeného grafu je zřejmé, že disciplína Requirements má v metodice RUP skutečně velké a rozsáhlé zpracování. V popisu procesu získala 4 body, je jasně definován cíl, má popsané role, činnosti rozdělené do jednotlivých skupin podle fáze projektu a výstupy jsou zdefinovány také podle konkrétních činností procesu. Co se týče samotných výstupů, dostala 3 body, je totiž jejich počet ve srovnání s ostatními metodikami střední. V rámci všech činností v rámci procesu jsme jich sečetli 9. Plný počet bodů získává v kritériu za popis jednotlivých rolí, vzhledem k tomu že každá role má jednoznačně určenou úlohu v každé části procesu, je detailně popsána a má definované i předpokládané výstupy své práce. Za rozsah v rámci celé metodiky získala 3 body, což představuje střední rozsah v porovnání s ostatními metodikami, v tomto případě má nejrozsáhlejší dostupný dokument pro část Requirements rozsah přibližně 10 stran.

## CMMI-DEV v.1.3

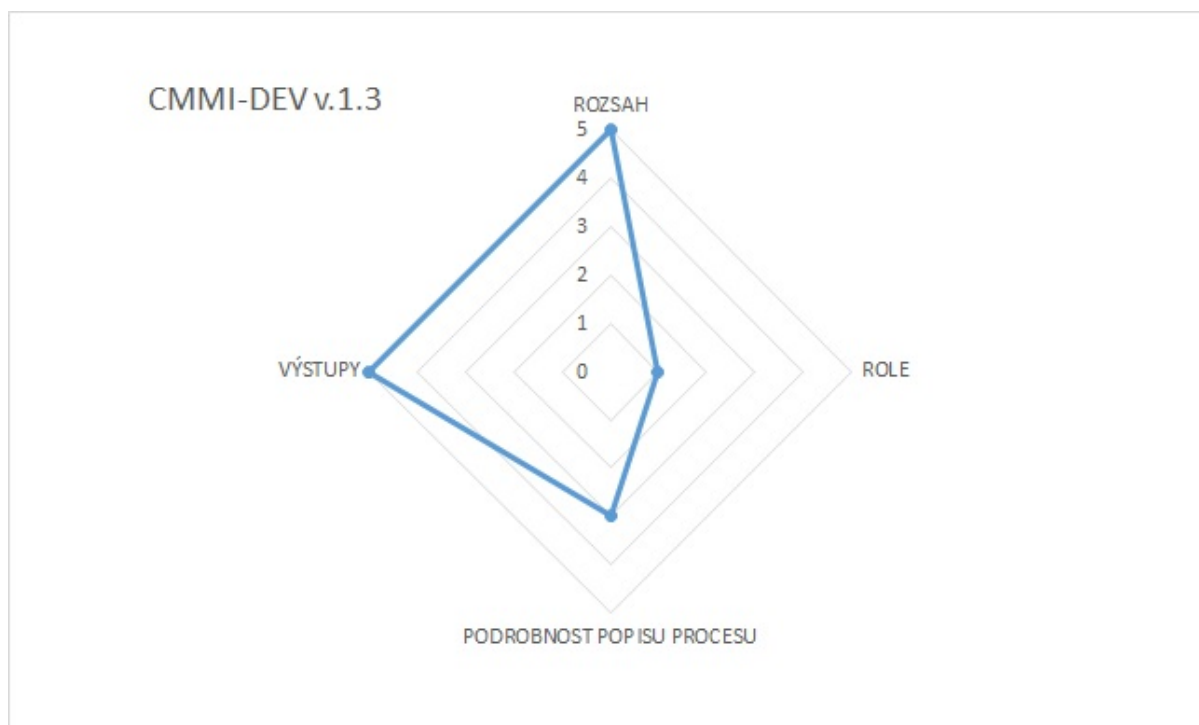


Diagram 2: CMMI - hodnoty kritérií

### Komentář k diagramu:

V kritériu rozsah dosahuje CMMI nejvyšší hodnoty 5, jelikož je oblast Requirements popsána na 22 stranách, což je největší rozsah s uvažovaných metodik. U kritéria role to pak je pouze 1, jelikož role nejsou specificky definovány, jen teoreticky vycházejí z vyjmenovaných stakeholderů. Podrobnost popisu je na úrovni 3, jelikož ačkoli je zde vysoký rozsah, chybí role, metriky a omezující podmínky. Výstupy naopak dosahují nejvyšší hodnoty 5, protože celkově lze napočítat v této metodice až 43 vyjmenovaných výstupů.

## ISO/IEC 12207:2008

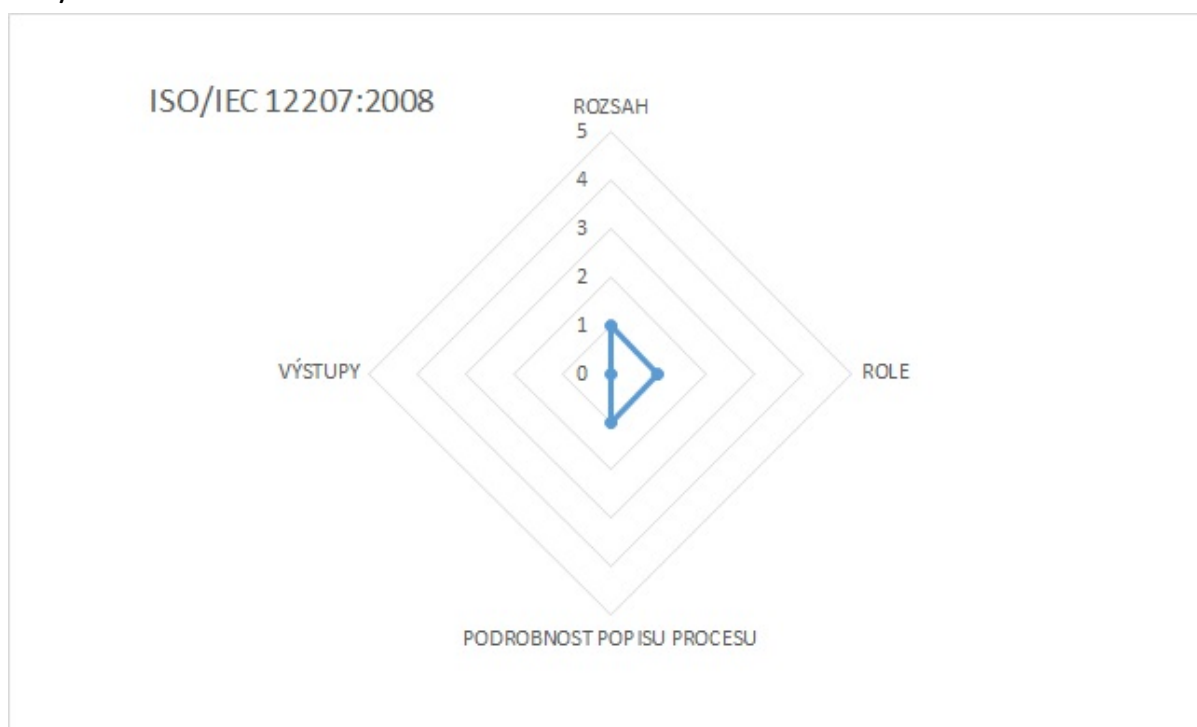


Diagram 3: ISO/IEC 12207:2008 – hodnoty kritérií

### Komentář k diagramu:

Z tvaru a velikosti grafu je zřejmé, že ve srovnání s ostatními metodikami má ISO 12207:2008 nejmenší zastoupení pro správu požadavků. Získala jenom 1 bod v kritériích role, podrobnost i rozsah. Rozsah je pro Requirements Analysis Process pouze necelé 2 strany, role nejsou nijak blíže definovány – uvádí se pouze implementátor, který požadavky analyzuje. Co se týče podrobnosti popisu procesu, nedefinuje konkrétní role ani činnosti, ani žádné specifické výstupy. Uvádí pouze obecný cíl (výsledek) tohoto procesu. Z tohoto důvodu má z hlediska výstupů 0 získaných bodů.

## ZÁVĚR

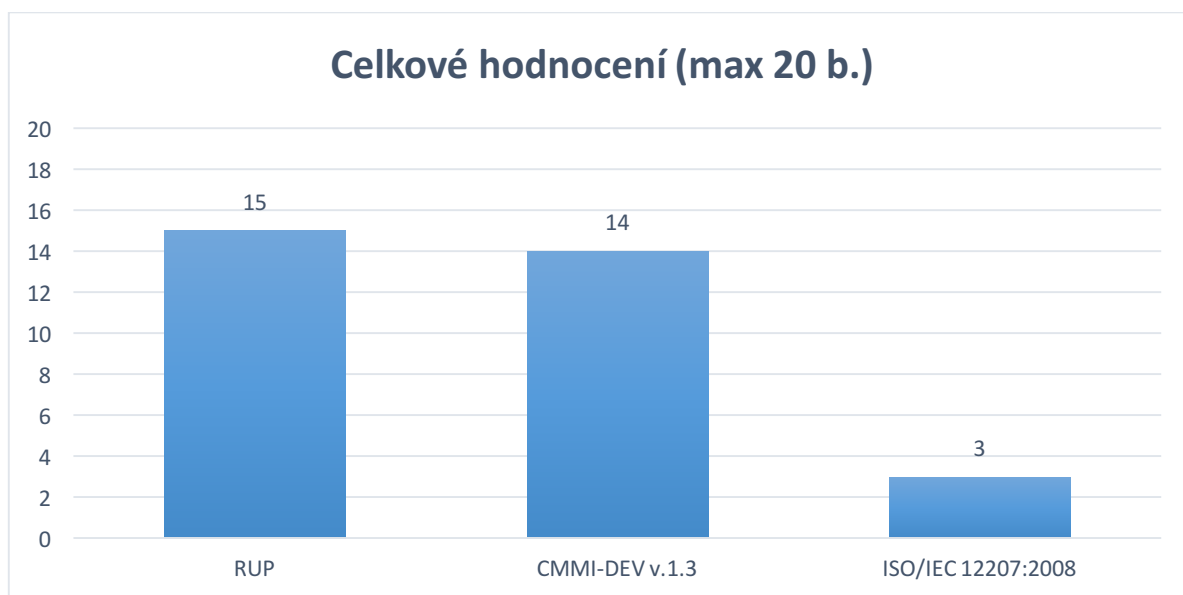


Diagram 4: Součet hodnocení pro jednotlivé metodiky ze všech kritérií

Výše uvedený diagram Součet hodnocení pro jednotlivé metodiky ze všech kritérií nám udává jasný závěr plynoucí z porovnání oblasti Requirements (požadavky) u metodik RUP, CMMI-DEV v.1.3 a ISO/IEC 12207:2008. Hodnoty byly získány tak, že body získané v jednotlivých kritériích byly pro jednotlivé metodiky sečteny. Z výsledků vyplývá, že nejlépe, a tedy nejkomplexněji, oblast Requirements zpracovává metodika RUP s celkem 15 body z 20 možných. CMMI je však jen v těsném závěsu se 14 body. Převaha RUP nad CMMI byla způsobena tím, že CMMI postrádá oproti RUP definice rolí a přiřazení činností k nim. A to navzdory tomu, že její celkový rozsah je větší. ISO je v této oblasti natolik obecné a neobsažné, že v porovnání se zbylými 2 metodikami dalece zaostává v komplexnosti poskytnutých informací a proto získalo velice nízký počet bodů ve všech jednotlivých disciplínách, a tedy i v závěrečném součtu.

## ZDROJE

1. Rational Software. *Rational Unified Process - Best Practices for Software Development Teams*. [Dokument] 2011.
2. Staš, Jakub. *Správa požiadaviek ako disciplína RUP*. [Semestrální práce] Praha : autor neznámý, 2011.
3. CMMI. *Wikipedia.org*. [Online] <https://cs.wikipedia.org/wiki/CMMI>.
4. CMMI. *UMSL.edu*. [Online] <http://www.umsl.edu/~sauterv/analysis/Fall2013Papers/Lanjewar/CMMI.html>.
5. Institute, Software Engineering. *CMMI for Development, Version 1.3*. [PDF document] 11 2010.
6. Institute of Electrical and Electronics Engineers. *ISO/IEC 12207 IEEE*. [Dokument] New York : autor neznámý, 2008.
7. RUP – Rational Unified Process. *testovanisoftware.cz*. [Online] <http://testovanisoftware.cz/manualni-testovani/modely-zivotniho-cyklu-software/rup/>.