Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Studijní program: Aplikovaná informatika

Obor: Informační systémy a technologie

Porovnání RUP disciplíny Analysis & Design s odpovídajícími PA v CMMI- DEV v. 1.3 a procesy v ISO/IEC 12207

SEMESTRÁLNÍ PRÁCE

Autoři : Jan Hudák, xhudj18
Jan Čislinský, xcisj08;

2015

Abstrakt

Tato semestrální práce se zabývá porovnáním disciplíny Analysis & Design z metodiky Rational Unified Process s odpovídajícími procesními oblastmi z referenčního modelu CMMI-DEV v1.3 a také s odpovídajícími procesy z normy ISO/IEC 12207. Cílem práce je zmíněné komponenty popsat a vzájemně porovnat.

Pro naplnění tohoto cíle je nejdříve popsaná disciplína Analysis & Design z metodiky Rational Unified Process. Dále jsou identifikovány odpovídající procesní oblasti z CMMI-DEV v1.3 a procesy z ISO/IEC 12207. Jednotlivé komponenty jsou zkoumány ze čtyř různých hledisek – věcného obsahu, podrobnosti popisu, dokumentů a rolí. Na základě zjištěných výsledků je provedeno porovnání.

Klíčová slova

rational, unified, process, analysis, design, cmmi, iso, iec

Obsah

1 Úvod 1
1.1 Cíle práce 1

2 Porovnávané komponenty 2
2.1 Představení RUP 2
2.1.1 Disciplína Analysis & Design 3
2.2 Identifikace odpovídajících PA v CMMI-DEV v1.3 6
2.2.1 Requirements management 7
2.2.2 Requirements development 8
2.2.3 Technical solution 8
2.3 Identifikace odpovídajících procesů v ISO/IEC 12207 8
2.3.1 Technical processes 9
2.3.2 Software implementation processes 9

3 Porovnání komponent 11
3.1 Metodika porovnání 11
3.2 Věcný obsah 11
3.2.1 RUP Analysis & Design 11
3.2.2 CMMI-DEV v 1.3 13
3.2.3 ISO/IEC 12207 14
3.2.4 Výsledek porovnání dle kritéria 15
3.3 Podrobnost popisu 15
3.3.1 RUP Analysis & Design 15
3.3.2 CMMI-DEV v 1.3 15
3.3.3 ISO/IEC 12207 16
3.3.4 Výsledek porovnání dle kritéria 16
3.4 Role 17
3.4.1 RUP Analysis & Design 17
3.4.2 CMMI-DEV v 1.3 17
3.4.3 ISO/IEC 12207 18
3.4.4 Výsledek porovnání dle kritéria 19
3.5 Dokumenty 19
3.5.1 RUP Analysis & Design 19
3.5.2 CMMI-DEV v 1.3 21
3.5.3 ISO/IEC 12207 25
3.5.4 Výsledek porovnání dle kritéria 29
3.6 Vyhodnocení porovnání 29

4 Závěr 30

1 Úvod

Práce si bere za cíl porovnat disciplínu Analysis & Design z metodiky Rational Unified Process s odpovídajícími procesními oblastmi z referenčního modelu CMMI-DEV v1.3 a také s odpovídajícími procesy z normy ISO/IEC 12207.

Aby bylo možné provést objektivní porovnání, jsou nejpve stanoveny ony odpovídající procesní oblastni z CMMI-DEV v1.3 a také odpovídající procesy z ISO/IEC 12207. Dále jsou určena hodnotící kritéra, podle kterých je následně provedeno porovnání.

1.1 Cíle práce

1. Podat základní informace o RUP disciplíně Analysis & design.
2. Identifikovat a popsat odpovídající PA v CMMI-DEV v1.3.
3. Identifikovat a popsat odpovídající procesy v ISO/IEC 12207.
4. Výše zmíněné komponenty vzájemně porovnat.

2 Porovnávané komponenty

2.1 Představení RUP

Rational Unified Process je metodika určená k vývoji software. Lze ji považovat za rámec, který je rozširitelný a přizpůsobitelný potřebám konkrétního projektu. (Kadlec, 2003)

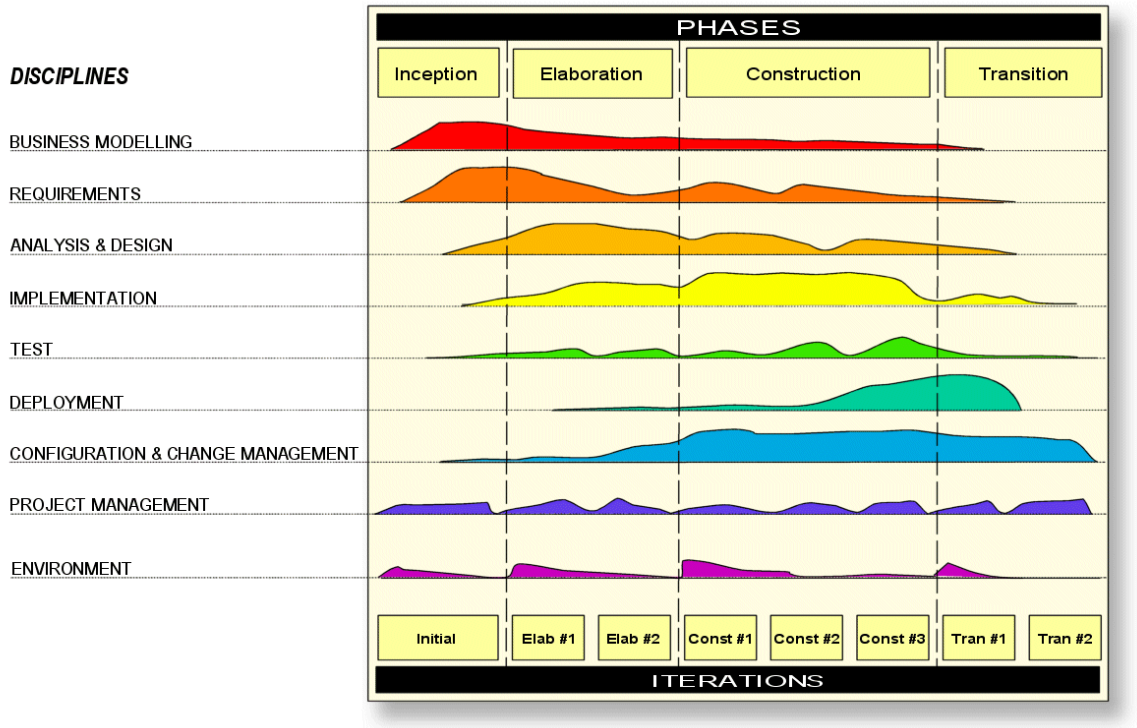
Je to standardizovaný proces vývoje software, definuje jak rozdělovat úkoly a odpovědnosti. Cílem je zajistit kvalitní software, který splňuje požadavky uživatelů na dodržení časového plánu a rozpočtu. (Kadlec, 2003; Kruchten, 2000)

Rational Unified Process je rozdělen do čtyř fází životního cyklu, jak je naznačeno na Obrázek 1 na ose X a devíti disciplín, znázorněno na ose Y na Obrázek 1. Rozlišujeme následující fáze životního cyklu.

- Inception (Zahájení) – definice rozsahu projektu a obchodního případu. Hlavním úkolem je vyjasnění a přesná definice požadavků všech zainteresovaných stran a stanovení rozsahu budoucího systému.
- Elaboration (Příprava) – podrobná analýza požadavků, formou úvodního modelu architektury, dle kterého v dalších fázích dojde k implementaci vyvíjeného systému.
- Contruction (Konstrukce) – tvorba spustitelné verze systému z modelu architektury, tj. vytvoření zdrojových kódů, programů, knihoven, modulů, potřebných datových souborů atd.
- Transition (Předání) – poskytnutí vytvořeného systému koncovému uživateli.

(Soukup, 2010)

Vývoj pomocí metodiky RUP je řízen v iteracích. Na konci každé iterace je funkční prototyp systému. Každá iterace se skládá ze všech disciplín. (Aldorf, 2008) Jinými slovy řečeno v rámci každé iterace proběhnou činnosti vázané na princip modelování, následují specifikace požadavků, analýza a návrh, implementace, testování a rozmístění (instalace). K tomu probíhá celá řada činností z podpůrných toků týkajících se správy konfigurací, řízení projektu a přípravy prostředí, ve kterém je systém vyvíjen a bude nasazen. Způsob, jakým iterace a fáze probíhají, by měl být nakonfigurován podle specifických potřeb projektu a používaných technologií. (Soukup, 2010)



Obrázek 1

Fáze a disciplíny Rational Unified Process (zdroj: Suniwe, 2008)

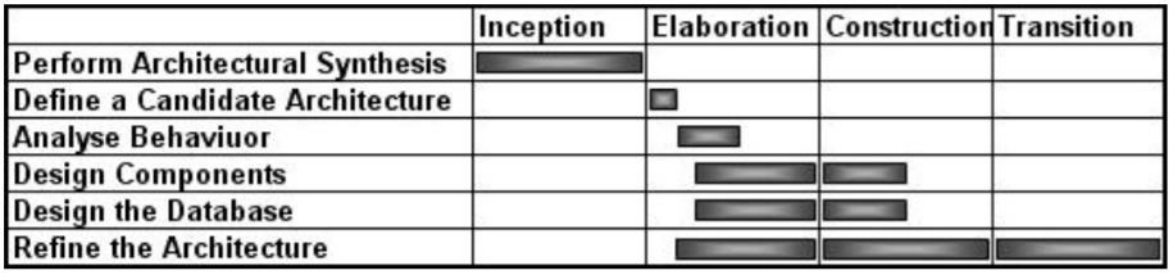
2.1.1 Disciplína Analysis & Design

Cílem disciplíny Analýza a Design je transformovat požadavky zákazníka na konkrétní návrh systému, vytvořit robustní architekturu systému a přizpůsobit návrh systému tak, aby odpovídal prostředí pro implementaci.

Disciplína Analýza a Design se dělí do šesti fází, které probíhají napříč životním cyklem Rational Unified Process, jak je ukázáno na

	Inception	Elaboration	Construction	Transition
Perform Architectural Synthesis				
Define a Candidate Architecture				
Analyse Behaviuor				
Design Components				
Design the Database				
Refine the Architecture				

Obrázek 2. V rámci fáze Inception je vykonávána fáze **Perform Architectural Synthesis**, která zjišťuje zda lze vytvořit systém s definovanými požadavky. Vytváří se architektonická analýza na základě které se usuzuje, zda bude realizace možná. (Shuja, 2007)



Obrázek 2

Rozložení disciplíny Analýza a Design v životním cyklu RUP (zdroj: Cole, Piveta a Sampaio, 2004)

Define a Candidate Architecture

Cílem této činnosti je vytvoření úvodního návrhu architektury systému. Jedná se primárně o zmapování všech podstatných okolností, jimiž je určen obsah a struktura IS. Je zamřena na celky, detaily jsou rozpracovávány v dalších iteracích. Na činnosti se podílejí softwarový architekti a designéři, kteří definují architektonické vzory, identifikují významné případy užití, provádějí use case realizaci prvků v boudoucí architekruře IS.

Hlavním výstupem je úvodní návrh architektury vyjádřený v modelu Analysis Model, který se skládá z diagramů tříd, sekvenčních diagramů, use case diagramů popisujících logické plnění funkčních požadavků. Modelování je prováděno pomocí notace UML[1].

Define a Candidate Architecture je rozdělena na dvě části, analýzu architektury a analýzu případů užití. (Soukup, 2010)

Architectural Analysis

Analýza architektury se zabývá vytvořením analytického modelu systému. Využívá zkušenosti z již existujících systémů a řešených problémů. V existujících systémech, kde je architektura dobře definována je možné tuto část vynechat. Role **Software Architekt** provádí akce k vytvoření diagramů tříd, diagramů procesů, diagrapů případů užití, atd. a je zodpovědná za výstupy prováděné práce. (Řepa, Synácek, Hamerlík a Diviš, 2006)

Use Case Analysis

Analýza případů užití slouží k zachycení dalších informací nezbytných k pochopení požadovaného vnitřního chování systému, které mohou chybět v informacích poskytnutých zákazníkem ve fázi projektu Iception. Zachycuje vztahy mezi aktéry a funkcemi systému a mezi funkcemi navzájem. (Řepa, Synácek, Hamerlík a Diviš, 2006)

Tuto činnost realizuje role Designer, jež je zodpovědná za výstup prováděné práce.

---

[1] **UML** – Unified Modeling Language je v softwarovém inženýrství grafický jazyk pro vizualizaci, specifikaci, navrhování a dokumentaci programových systémů.

Analyze Behaviuor

Chování je projevem dynamiky systému. Dynamika je schopnost vyvolat změnu systému, zejména jeho stavu. Chování prvků ve vzájemných interakcích, které vyplývají ze struktury systému, určuje chování celého systému. Je popsáno pomocí vstupních, stavových a výstupních charakteristik. (Hronek, 2007)

V této činnosti dochází k úvodnímu design uživatelského rozhraní. Až doposud se činnosti zabývaly více tzv. nefunkčními požadavky IS, v tomto kroku je však úkolem zajistit potřebnou funkcionalitu IS.

Design Components

Složitý systém je nutno popsat velkým počtem diagramů. Po celý životní cyklus Rational Unified Process je tedy nutné respektovat při modelování softwarového systému principy, umožňující vytvářené modely, obvykle reprezentované celou řadou diagramů, vhodným způsobem strukturovat a lépe tak spravovat celou dokumentaci vytvářeného produktu.

Architektura IS je obvykle velice komplexní a můžeme se na ni dívat z různých hledisek (statické vs. dynamické aspekty systému; abstraktní vs. fyzická úroveň, popis systému vs. procesy v něm probíhající). Z této komplexnosti vyplývají dva základní, částečně se překrývající pohledy. Jejich kombinace nám umožní porozumět architektonickému designu.

Těmito pohledy jsou komponentová a procesní architektura. Komponentová architektura se zaměřuje na třídy, tedy statické aspekty systému. Rozděluje systém na jasně identifikovatelné, vzájemné propojené komponenty, řeší jejich vnitřní stavy a přechody mezi nimi. Architektura procesu se naproti tomu věnuje instancím tříd (dynamické aspekty systému). Dělí systém na procesy, řeší jejich vzájemné interakce a koordinaci.

(Řepa, Synácek, Hamerlík a Diviš, 2006)

Design the Database

Návrh databáze se realizuje pomocí logického (implementačně nezávislého) modelu a následně pomocí fyzického datového modelu. Datový model obsahuje informace o reprezentaci objektů, jejich atributech, vztazích s okolními objekty a rozsazích hodnot, kterých mohou atributy nabývat.

Refine the Architecture

V průběhu celého životního cyklu Rational Unified Process, vyjma Inception, je prováděna činnost Refine the Architecture. Činnost běží paralelně s designem systému. Architektonický design je klíčovou fází vývoje informačního systému. Jeho kvalita rozhoduje o úspěchu celého projektu. Při tvorbě designu systému pracovníci vycházejí z analytických modelů předem definovaných v předchozích krocích analýzy IS.

Jádro systému vytvořené v etapě analýzy je postupně doplněno o další tři vrstvy.

- Databázi,
- uživatelské rozhraní
- distribuci.

2.2 Identifikace odpovídajících PA v CMMI-DEV v1.3

CMMI je zkratka pro Capability Maturity Model Integration – v češtině takzvaný Integrační model zralosti, který má za cíl pomoci v organizaci plánovat, definovat, implementovat, rozvíjet, hodnotit a zlepšovat procesy. Tento modle vznikl v roce 2000 v Software Engineering Institute (zkráceně SEI). Poslední verze s číslem 1.3 definuje několik variant modelu. Tato práce se zabývá variantou CMMI-DEV, která se zaměřuje na zlepšování procesů v organizacích, které vyvíjejí produkt či službu a chtějí to dělat efektivně, levně a kvalitně.

CMMI-DEV v1.3 se skládá celkem z 22 procesních oblastí ve čtyřech různých kategorií. Procesní oblast je skupina vzájemně propojených procesů, které pokud implementujeme kolektivně, tak naplní množinu cílů, jež jsou podstatné pro zlepšení dané oblasti. Tyto cíle mohou být globální a nebo specifické. Všechny definované procesní oblasti jsou znázorněny na Obrázek 3.

Category	Process Areas
Process Management	Organizational Process Focus Organizational Process Definition Organizational Training Organizational Process Performance Organizational Innovation and Deployment
Project Management	Project Planning Project Monitoring and Control Supplier Agreement Management Integrated Project Management Risk Management Quantitative Project Management
Engineering	Requirements Management Requirements Development Technical Solution Product Integration Verification Validation
Support	Configuration Management Process and Product Quality Assurance Measurement and Analysis Causal Analysis and Resolution Decision Analysis and Resolution

Obrázek 3

Procesní oblasti CMMI-DEV v1.3 (Zdroj:Gallagher a Brownsword, 2001).

Cílem této kapitoly je najít na základě oficiální dokumentace CMMI-DEV v1.3 procesní oblasti, které odpovídají RUP disciplíně Analysis & Design. Vzhledem k obsahu této disciplíny, lze relevatní procesní oblasti nalézt pouze ve skupině Engineering. Konkrétné se jedná o procesní oblasti Requirements management, Requirements Development a Technical Solution. Nelze ale vždy prohlásit, že by celý obsah procesní oblasti plně odpovídal části obsahu RUP discipníny Analysis & Design, proto je potřeba při hledání podobností věnovat pozornost jednotlivým částem obsahu dané procesní oblasti.

2.2.1 Requirements management

Účelem této procesní oblasti je spravovat požadavky na produkt a produktové komponenty s cílem zajistit soulad mezi těmito požadavky a projektovými plány a produkty. Tato procesní oblast je relevantní z hlediska svého jediného specifikého cíle s názvem Manage requirements, v rámci kterého dochází k naplňování účelu celé procesní oblasti. Činnosti v této oblasti zahrnují například správu změny požadavků, zajišťování jejicht konzistence a řízení případných opravných akcí. Zahrnuje tedy činnosti, které se vyskytují v rámci pracovních kroků Analyse behavior, Design components a Design database RUP disciplíny Analysis & Design.

2.2.2 Requirements development

Účelem této procesní oblasti je zjistit, analyzovat a stanovit zákaznické, produktové a komponentové požadavky. I přesto, že se tato oblast zabývá z velké části získáváním a komunikováním požadavků, obsahuje i aktivity jako definování produktových a komponentových požadavků pro vývoj, je tedy z části relevantní k RUP disciplíně Analysis & Design. Konkrétně je tato podobnost viditelná ve druhém specifickém cíly této procesní oblasti, který se nazývá Develop product requirements. V rámci tohoto cíle dochází k úpravě a předefinování zákaznických požadavků pro účely vývoje produktu a jeho komponent. To odpovídá některým činnostem pracovního kroku Analyze behavior z RUP disciplíny Analysis & Design. Dále se podobnost vyskytuje v rámci specifického cíle Analyze and validate requirements, kdy jsou požadavky analyzovány a validovány a dochází k rozvoji definice požadované funkcionality. Zde je podobnost s činnostmi pracovního kroku Define a candidate architecture z RUP disciplíny Analysis & Design.

2.2.3 Technical solution

Účelem této procesní oblasti je vybrat, vytvořit design a implementovat řešení, které odpovídá požadavkům. Tato oblast se tedy zaměřuje především na výběr a hodnocení variant řešení, vytváření detailních designů pro vybrané řešení a implementaci těchto designů jako produktu nebo jeho komponenty. Implementace není součástí RUP disciplíny Analysis & Design, takže relevantním je specifický cíl Select product component solution, v rámcí kterého dochází k výběru řešení produktu nebo komponenty produktu z alternativních řešení. To odpovídá činnostem pracovních kroků Define candidate architecture a Refine candidate architecture z RUP disciplíny Analysis & Design. Dalším relevantním specifickým cílem je Develop the design v rámci kterého dochází k návrhu designu řešení, který souvisí s různými pracovními kroky RUP disciplíny Analysis & Design.

2.3 Identifikace odpovídajících procesů v ISO/IEC 12207

ISO/IEC 12207 je mezinárodní standard pro procesy životního cyklu softwaru. Celý název této normy je Systems and software engineering – Software life cycle processes (Procesy v životním cyklu softwaru). Jejím cílem je stát se standardem, který definuje všechny úkoly které jsou potřeba při vývoji a udržování softwaru. Tato práce se zabývá verzí z roku 2008. Tato norma obsahuje referenční model procesů, který zahrnuje nejen procesy, které se týkají samotného vývoje, ale pokrývají celý životní cyklus softwaru.

Všechny definované procesy jsou znázorněny na Obrázek 4. Tyto procesy jsou rozděleny do dvou skupin a dále do 7 podskupin. ISO/IEC 12207 pouze definuje názvy a ukazuje aktivity,

které by se měly provádět, ale nikdy nepředepisuje jakým způsobem by měly být prováděny. Tato norma je naprosto neutrální v pojetí metod, technik, jazyků, nástrojů a organizačních struktur.

Cílem této kapitoly je najít na základě oficiální dokumentace procesy normy ISO/IEC 12207:2008, které odpovídají RUP disciplíně Analysis & Design. Relevantní procesy lze najít v podskupinách obou hlavních skupin, tedy jak v Procesech v kontextu systému i v Softwarově specifických procesech.

2.3.1 Technical processes

Procesy této skupiny se používají k definování požadavků pro systém, k transformaci těchto požadavků do efektivního produktu, řízení produktu a jeho likvidaci. Tyto procesy definují aktivity, které optimalizují benefity a redukují rizika, která vznikají z technických rozhodnoutí a akcí. V této podskupině existují z celkových jedenácti procesů dva relevantí.

- **System requirements analysis process** – Cílem tohoto procesu je transformovat požadavky zúčastněných stran do skupiny požadovaných technických systémových požadavků, kterými se bude dále řídit design systému. To zahrnuje aktivity jako specifikaci a hodnocení požadavků.
- **System architectural design process** – Cílem tohoto procesu je identifikovat, které systémové požadavky budou přiděleny jakému elementu systému. To zahrnuje aktivity jako stanovení architektury a hodnocení architektury.

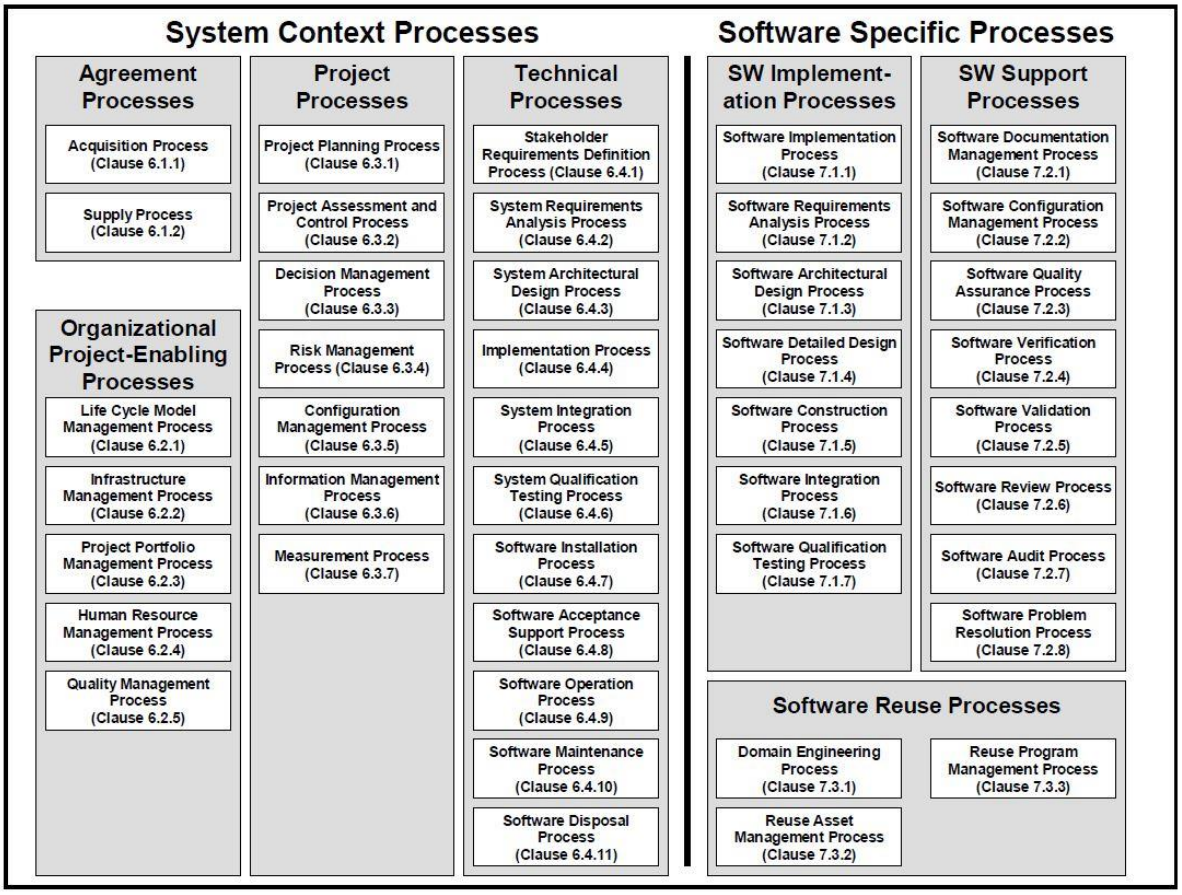
2.3.2 Software implementation processes

Procesy této skupiny se používají k produkci specifického systémového elementu (části softwaru) implementovaného do softwaru. Tyto procesy transformují specifikované chování, rozhraní a implementační omezení do implementačních akcí, které vedou k systémovému elementu, který uspokojuje požadavky odvozené ze systémových požadavků.

V této podskupině exisují z celkových sedmi procesů tři relevantní.

- **Software requirements analysis process** – Cílem tohoto procesu je stanovit požadavky na softwarové elementy systému. To zahrnuje aktivitu analýzy softwarových požadavků.
- **Software architectural design process** – Cílem tohoto procesu je poskytnout design softwaru, který má být implementován a v souladu s požadavky. To zahrnuje aktivitu design softwarové architektury.

- **Software detailed design process** – Cílem tohoto procesu je poskytnout design softwaru, který má být implementován a v souladu s požadavky a softwarovou architekturou. Na rozdíl od předchozího jmenovaného procesu je zde dosáhnuto natolik dostatečně detailního designu, který umožňují programování a testování. To zahrnuje aktivitu detailního designu softwaru.



Obrázek 4

Procesy ISO/IEC 12207 (Zdroj: ISO 12207, 2008)

3 Porovnání komponent

3.1 Metodika porovnání

K porovnání vybraných oblastí je potřeba stanovit metodu, která zajistí ohodnocení každé oblasti tak, aby bylo dosaženo objektivního výsledku. Jako metodu porovnání jsme zvolili hodnocení podle předem stanovených kritérií. Stanovili jsem čtyři kritéria – věcný obsah, podrobnost popisu, role a dokumenty. Následuje popis jednotilivých kritérií.

- **Věcný obsah** – Kritérium hodnotí věcný obsah procesů dané oblasti na základě její oficiální dokumentace. Jedná se o nejvýznamnější kritérium, jelikož se přímo zabývá účelem jednotlivých procesů a aktivitami, které jsou v rámci nich definovány.
- **Podrobnost popisu** – Kritérium hodnotí množství věnovaného textu dané problematice v oficiální dokumentaci. Zároveň také hodnotí počet úrovní, do kterých jsou aktivity a úkoly v rámci jednotlivých procesů rozepsány.
- **Role** – Kritérium hodnotí zastoupení jednotlivých rolí, které se vyskytují v rámci definovaných procesů jednotlivých oblastí.
- **Dokumenty** – Kritérium hodnotí množství a typy dokumentů, které se vyskytují v rámci definovaných procesů jednotlivých oblastí.

Vzhledem k tomu, že některé oblasti nemusejí přímo definovat například jednotlivé role nebo dokumenty, nelze tedy jednoznačně definovat konkrétní stupnice hodnot, pomocí kterých by bylo možné ohodnotit zvolená kritéria. Jedná se tedy pouze o přibližné porovnání jednotlivých oblastí na základě hledání výskytu podobných principů. Následující kapitoly se podrobněji zabývají porovnáním z hlediska každého kritéria.

3.2 Věcný obsah

Porování na základě tohoto kritéria má vzhledem k tématu této práce největší význam. Toto kritérium zkoumá věcný obsah jednotlivých zkoumaných oblastí za účelem nalezení podobností v popisu procesů.

Následuje shrnutí obsahu jednotlivých oblastí.

3.2.1 RUP Analysis & Design

Tato oblast se v rámci jednotlivých kroků zabývá následujícími aktivitami.

- Perform architectural synthesis – ověření zda je systém možné realizovat z dostupných technologií.
    - Create Proof of concept – vytvoření počátečního konceptu.
    - Asses Proof of concept – ověření počátečního konceptu.
- Define candidate architecture – vytvoření úvodního návrhu architektury.
    - Architectural Analysis – definování kandidáta na architekturu.
    - Use Case Analysis – analýza zadaných use case modelů.
- Analyze behavior – převedení popisu chování definovaný pomocí use-casů modelů do souboru prvků, na kterých může být založen design.
    - Use Case Analysis – definování tříd souvisejících s use case modely.
    - Identify Design Elements – analýza interakcí tříd za účelem identifikování prvků modelu.
    - Review of Desing – ověření konzistentního naplnění požadavků na systém.
- Design components – zpřesnění definicí prvků designu (včetně use case modelů) a vytvoření designu prvků jako komponent.
    - Use Case Design – zpřesnění use case modelů z hlediska interakcí prvků.
    - Class Design – zajistit dostatečné definice pro implementaci tříd.
    - Subsystem Design – definování chování v rámci rozhraní subsystému.
    - Review of Design – ověření konzistentního naplnění požadavků na komponenty.
- Define database – vytvoření designu databázové struktury pro ukládání persistentních tříd.
    - Class Design – definování persistentních tříd.
    - Database Design – zajištění konzistentní a efektivní práce s daty.
    - Review of Design – ověření konzistentního naplnění požadavků na databázi.
- Refine architecture – zajištění plynulého přechodu od analýzy k designu.
    - Identify Design Mechanisms – transformování mechanizmů analýzy do mechanizmů designu za respektování omezení implementačního prostředí.
    - Identifiy Design Elements – analyzování interakcí tříd analýzy k identifikování prvků designového modelu.
    - Incorporate Existing Design Elements – použití již existujících řešení.

3.2.2 CMMI-DEV v 1.3

Tato oblast se zabývá v rámci jednotlivých procesů následujícími činnostmi.

- Manage Requirements – spravování požadavků na produkt a produktové komponenty.
    - Understand Requirements – zajištění správného pochopení požadavků.
    - Obtain Commitment to Requirements – získání potvrzení požadavků od účastníků projektu.
    - Manage Requirements Changes – řízení změn požadavků v průběhu řešení.
    - Maintain Bidirectional Traceability of Requirements – zajištění obousměrné sledovatelnosti mezi požadavky a výstupy.
    - Ensure Alignment Between Project Work and Requirements – zajištění souladu projektových plánů a požadavků.
- Develop Product Requirements – vytváření produktových a komponentových požadavků z požadavků zákaznických.
    - Establish Product and Product Component Requirements – definování požadavků na produkt a jeho komponenty.
    - Allocate Product Component Requirements – alokovat požadavky pro každou komponentu produktu.
    - Identify Interface Requirements – identifikování požadavků na rozhraní.
- Analyze and Validate Requirements – analyzování a ověřování požadavků.
    - Establish Operational Concepts and Scenarios – stanovení operativních konceptů a scénářů.
    - Analyze Requirements – analyzování nezbytnosti a úplnosti požadavků.
    - Analyze Requirements to Achieve Balance – analyzování požadavků pro zajištění rovnováhy potřeb a omezení zadavatelů.
    - Validate Requirements – ověřování požadavků pro zajištění finální validace.
- Select Product Component Solution – výběr řešení produktu nebo komponenty z množiny alternativních řešení.
    - Develop Alternative Solutions and Selection Criteria – definování možných řešení a výběrových kritérií.
    - Select Product Component Solutions – výběr řešení podle výběrových kritérií.
- Develop the Design – tvorba designu produktu nebo komponent.
    - Design the Product or Product Component – tvorba designu produktu nebo jeho komponent.

- Establish a Technical Data Package – udržování balíku technických dat.
- Design Interfaces Using Criteria – tvorba designu rozhraní dle kritérií.
- Perform Make, Buy, or Reuse Analyses – zhodnocení zda by komponenty měly být vyvinuty, koupeny nebo znovupoužity na základě kritérií.

3.2.3 ISO/IEC 12207

Tato oblast se zabývá v rámci jednotlivých procesů následujícími činnostmi.

- System Requirements Analysis Process – transformování požadavků zúčastněných stran do skupiny požadovaných technických systémových požadavků, kterými se bude dále řídit design systému.
  - Requirements Specification – definování požadavků na systém.
  - Requirements Evaluation – ohodnocení požadavků dle kritérií.
- System Architectural Design Process – přiřazení systémových požadavků k prvkům systému.
  - Establishing Architecture – definování architektury systému.
  - Architectural Evaluation – ohodnocení architektury dle požadavků.
- Software Requirements Analysis Process – definování požadavků na softwarový prvek systému.
  - Software Requirements Analysis – definování a dokumentování požadavků a jejich zhodnocení dle kritérií.
- Software Architectural Design Process – vytváření a ověřování designu softwaru dle kritérií.
  - Software Architectural Design – pro každý prvek systému převést požadavky do architektury, definovat jejích rozhraní, vytvořit design databáze, vytvořit uživatelskou dokumentaci, vytvořit požadavky pro testy a celý design ohodnotit dle kritérií.
- Software Detailed Design Process – vytváření a ověřování design softwaru dle kritérií na podrobnější úrovni detailu.
  - Software Detailed Design – stejné jako předchozí jmenovaná činnost s tím rozdílem, že se pracuje na takové úrovni detailu, která umožňuje začít s vývojem a testováním.

3.2.4 Výsledek porovnání dle kritéria

Z výše uvedeného výčtu obsahu jednotlivých oblastí je zřejmé, že jsou zde pokryty stejné činnosti, aktivity a úkoly. U každého prvku je možné najít odpovídající prvek v jiné oblasti. Podstatný rozdíl je ve struktuře a míře podrobnosti, jak jsou tyto činnosti, aktivity a úkoly popsány. Tímto rozdílem se zabývá další část práce.

3.3 Podrobnost popisu

Porovnání na základě tohoto kritéria se zabývá podroností popisu procesů v rámci jednotlivých oblastí. Každá dokumentace přistupuje k popisu jiných způsobem a pro definici jednotlivých aktivit a úkolů volí i rozdílnou strukturu popisu. Tato struktura je v rámci tohoto kritéria zkoumána především z hlediska počtu úrovní popisu.

Následuje popis struktur používaných v jednotlivých oblastech.

3.3.1 RUP Analysis & Design

V této oblasti jsou pro jednotlivé kroky definovnány aktivity u kterých jsou dalé definovány kroky. Pracuje se zde tedy se dvěma úrovněmi struktury a velice obsáhlým popisem. Následuje ukázka stuktury aktivity.

Class Desing

- Steps:
    - Define Methods
    - Define Atributtes
    - Define Class dependencies
    - ...

3.3.2 CMMI-DEV v 1.3

Tato oblast popisuje procesy u jednotlivých procesních oblastí. Tyto procesy obsahují jednotlivé činnosti rozepsané na menší úkoly. Pracuje se zde tedy se čtyřmi úrovněmi struktury a velmi obsáhlým popisem. Následuje ukázka struktury procesní oblasti.

Technical solution

- Select Product Component Solution
    - Develop Alternative Solutions and Selection Criteria

- Subpractices:
  - Identify screening criteria to select a set of alternative solutions for consideration
  - Generate alternative solutions
  - ...

3.3.3 ISO/IEC 12207

Tato oblast rozepisuje jednotlivé procesy na menší aktivity, které se skládají z menších úkolů. Pracuje se zde tedy se třemi úrovněmi struktury a velice obecným popisem procesů, které vyplívá z charakteru normy. Následuju ukázka struktury procesu.

System Requirements Analysis Process

- Activities and tasks:
  - Requirements specification
    - Analyze intended use of the system to specify requirements
    - Document requirements
    - ....
  - Requirements evaluation
    - Evaluate requirements considering listed criteria
    - ...

3.3.4 Výsledek porovnání dle kritéria

Výsledek porovnání lze u tohoto kritéria vyjádřit poměrně jasně. Z hlediska počtu úrovní struktury dosáhl nejvyšší hodnoty popis v CMMI-DEV v1.3, následovaný popisem v ISO/IEC 12207, zatímco popis v RUP disciplíně Analysis & Design má nejméně složitou strukturu. Z hlediska podrobnosti popisu lze považovat RUP disciplínu Analysis & Design a CMMI-DEV v1.3 za srovnatelné, zatímco ISO/IEC 12207 je popsáno pouze velmi obecně. V rámci celkového kritéria „podrobnost popisu" dosahuje nejlepšího ohodnocení CMMI-DEV v1.3, které má obsáhlou a velmi dobře strukturovanou dokumentaci.

3.4 Role

Porovnání na základě tohoto kritéria se zabývá množstvím rolí a podrobností jejich popisu. Vzhledem k tomu, že CMMI-DEV v1.3 ani ISO/IEC 12207 role explicitně nepopisuje, role jsou odvozenu z kontextu v dokumentaci.

Následuje popis rolí používaných v jednotlivých oblastech.

3.4.1 RUP Analysis & Design

Database Designer – návrhář databáze je odpovědný za detailní návrh databáze zahrnující: tabulky, indexy, omezení, trigery, ukládací procedury a ostatní specifické databázové konstrukce potřebné pro ukládání, vyvolání či odstranění přetrvávajících objektů.

Designer – identifikuje a definuje odpovědnosti, operace, atributy a vztahy konstrukčních prvků. Zajišťuje, že design je v souladu se softwarovou architekturou, a je uveden do stavu, kdy může následovat implementace systému.

Software Architekt – má celkovou odpovědnost za řízení velkých technických rozhodnutí, ohledně architektury systému. To obvykle zahrnuje identifikaci a dokumentaci architektonicky významných aspektů systému, včetně požadavků, návrhu, implementace a nasazení tedy ze všech „pohledů“ systému.

System Analyst – vede a koordinuje požadavky elicitace vymezením funkčnosti IS.

Technical Reviewer – organizuje odpovědnost za plnění úkolů a vývoj prací na produktů ve skupinách. Je třeba zvážit i dovednosti potřebné pro úlohy a různé přístupy. Může zaměstnancům přidělovat role. Účelem Technického reviewera je, aby se dospělo k technicky dokonalejší verzi přezkoumávaného produktu, zda je v souladu doporučeními či zavedení alternativních přístupů.

Test Designer – určuje vhodné metody, nástroje a pokyny k provedení požadovaných testů.

User-Interface Designer – koordinuje návrh uživatelského rozhraní. Zahrnuje sběr požadavků na použitelnost UI a navrhuje prototyp uživatelského rozhraní v rámci těchto požadavků.

3.4.2 CMMI-DEV v 1.3

Requirements Development

Develop Customer Requirements

- *nepopisuje žádné role*

Develop Product Requirements

- *nepopisuje žádné role*

Analyze and Validate Requirements

- *nepopisuje žádné role*

Requirements Management

Manage Requirements

- *nepopisuje žádné role*

Technical Solution

Select Product Component Solutions

- *nepopisuje žádné role*

Develop the Design

- Zmiňuje v textu role **designer** a **developer**. Popis vykonávané činnosti rolí není dále rozveden, nejsou určeny zodpovědnosti ani nic dalšího.

3.4.3 ISO/IEC 12207

Technical Processes 6.4 (Technické procesy)

System Requirements Analysis Process 6.4.2 (Analýza požadavků systému)

- *nepopisuje žádné role*

System Architectural Design Process 6.4.3 (Návrhy architektury systému)

- *nepopisuje žádné role*

Software Implementation Processes 7.1 (Procesy implementace softwaru)

Software Requirements Analysis Process 7.1.2 (Analýza SW požadavků)

- V textu je zmíněna role **Implementer**, ke které je dostupný popis v sekci 4. Terms and definitions: „organization that performs implementation tasks.“ Žádné další informace nejsou k roli dostupné.

Software Architectural Design Process 7.1.3 (Návrh SW architektury)

- Opět zmínka o roli **Implementer** bez žádných dalších rozšiřujících informací.

Software Detailed Design Process 7.1.4 (Detailní návrh SW)

- *nepopisuje žádné role*

3.4.4 Výsledek porovnání dle kritéria

Výsledek porovnání je jednoznačný, až na pár zmínek nepopisuje ani CMMI-DEV v 1.3 ani ISO/IEC 12207 žádné role. Můžeme tedy říct, že popis rolí v RUP disciplíně Analysis & Design je nejrozsáhlejší jak do množství popsaných rolí, tak z hlediska podrobnosti popisu jejich odpovědností a činností, kterými se zabývají.

3.5 Dokumenty

Porovnání na základě tohoto kritéria se zabývá množstvím dokumentů a podrobností jejich popisu. ISO/IEC 12207 nedefinuje dokumenty, ale hovoří pouze o tzv. Outcomes, pokusíme se tedy namapovat tyto Outcomes na dokumenty definované v RUP disciplíně Analysis & Design, kde jsou velmi dobře popsány. CMMI-DEV v1.3 také nehovoří o dokumentech, ale pouze specifikuje tzv. Example Work Products, které lze považovat za dokumenty. Součástí této části bude tady i mapování Example Work Products na dokumenty definované v RUP disciplíně Analysis & Design.

Vzhledem k tomu, že mapování Outcomes a Example Work Products na RUP disciplínu Analysis & Design není možné provést úplně nezávisle, je toto mapování založeno na vědomostech a znalostech autora. Nelze tedy tvrdit, že tak, jak je mapování provedeno je to správně. Je tak pouze vyjádřena myšlenka autora.

Následuje popis dokumentů používaných v jednotlivých oblastech.

3.5.1 RUP Analysis & Design

Analysis Model

Model analýzy je užitečný pro lepší porozumění požadavkům dříve než se vytvoří design model. Pomocí tohoto teoretického popisu IS se dostává pracovníkům do rukou silná zbraň pro zpracování mnoha funkcionálních požadavků, přehledným srozumitelným a systematickým způsobem.

Model analýzy je složen z analytických modelů (Class diagram, Process diagram, State diagram atd.).

Navigation Map, User-Interface Prototype

Je souhrn způsobů, jakými lidé (uživatelé) ovlivňují chování strojů, zařízení, počítačových programů či komplexních systémů. Projekty s velkým a složitým uživatelským rozhraním by měly zvážit návrh uživatelského rozhraní.

Typy rozhraní:

- Grafické uživatelské rozhraní – nejrozšířenější rozhraní pro desktop,
- textové uživatelské rozhraní – s menu, tlačítky, ovládání klávesnicí a myší,
- příkazový řádek – příkazy se zadávají zápisem pomocí klávesnice,
- braillský řádek – zařízení pro převod textu do slepeckého písma.

Design Model

Upřesňuje model analýzy ve světle skutečného implementačního prostředí. Model návrhu tak představuje abstrakci zdrojového kódu, jinými slovy řečeno, reprezentuje „výkresovou“ dokumentaci určující jak bude zdrojový kód strukturován a napsán.

Design Class, Design Package

Třídy jsou základní součástí každého objektově orientovaného návrhu. Objektově orientovaný návrh je standardně používaný na většině projektů. Specifikuje množinu tříd, rozhraní a jejich vzájemné vztahy. Diagramy tříd slouží k vyjádření statického pohledu na systém.

Balíček je mechanismum obecného použití umožňující organizovat elementy do skupin s cílem redukovat složitost modelovaného systému.

Use-Case Realization

Poskytuje most z případů užití k designu. Účelem tohoto dokumentu je popsat jednotlivé případy užití a propojit pohled případů užití s logickým a datovým pohledem.

Design Subsystem

Subsystémy sdružují libovolné části modelovaného systému. Pokud se bude jednat o části, které spolu souvisejí z hlediska zajištění nějaké společné funkčnosti či účelu pak hovoříme o vytváření tzv. subsystémů.

Events

Dokument zaznamenávající posloupnost požadavků účastníka systému a reakcí systému.

Data Model

Používá se k popisu logické a případně i fyzické struktury dat.

Deployment Model

Model nasazení definuje fyzickou alokaci částí systému (jeho komponent) na fyzické zdroje (výpočetní uzly/servery).

Reference Architecture

Referenční architektura slouží k urychlení vývoje a snížení rizika opětovného použití osvědčených řešení.

Software Architecture Document (SAD)

Architektura softwaru dokumentu se používá k ucelenému architektonickému přehledu o systému. Tento přehled umožňuje porozumět systému a zachytit klíčové stavební rozhodnutí informačního systému. SAD je výsledkem procesu skládající se z – sady modelů a popisů – celého životního cyklu Rational Unified Process.

3.5.2 CMMI-DEV v 1.3

Requirements Development

Develop Customer Requirements

- Prioritized customer requirements
- Customer constraints on the conduct of verification
- Customer constraints on the conduct of validation

Develop Product Requirements

- Derived requirements
- Product requirements
- Product component requirements
- Architectural requirements, which specify or constrain the relationships among product components

Analyze and Validate Requirements

- Operational concept
- Product or product component development, installation, operational, maintenance, and support concepts
- Disposal concepts
- Use cases
- Timeline scenarios
- New requirements

- Definition of required functionality and quality attributes
- Functional architecture
- Activity diagrams and use cases
- Object oriented analysis with services or methods identified
- Architecturally significant quality attribute requirements
- Requirements defects reports
- Proposed requirements changes to resolve defects
- Key requirements
- Technical performance measures
- Assessment of risks related to requirements
- Record of analysis methods and results

Requirements Management

Manage Requirements

- Lists of criteria for distinguishing appropriate requirements providers
- Criteria for evaluation and acceptance of requirements
- Results of analyses against criteria
- A set of approved requirements
- Requirements impact assessments
- Documented commitments to requirements and requirements changes
- Requirements change requests
- Requirements change impact reports
- Requirements status
- Requirements database
- Requirements traceability matrix
- Requirements tracking system
- Documentation of inconsistencies between requirements and project plans and work products, including sources and conditions
- Corrective actions

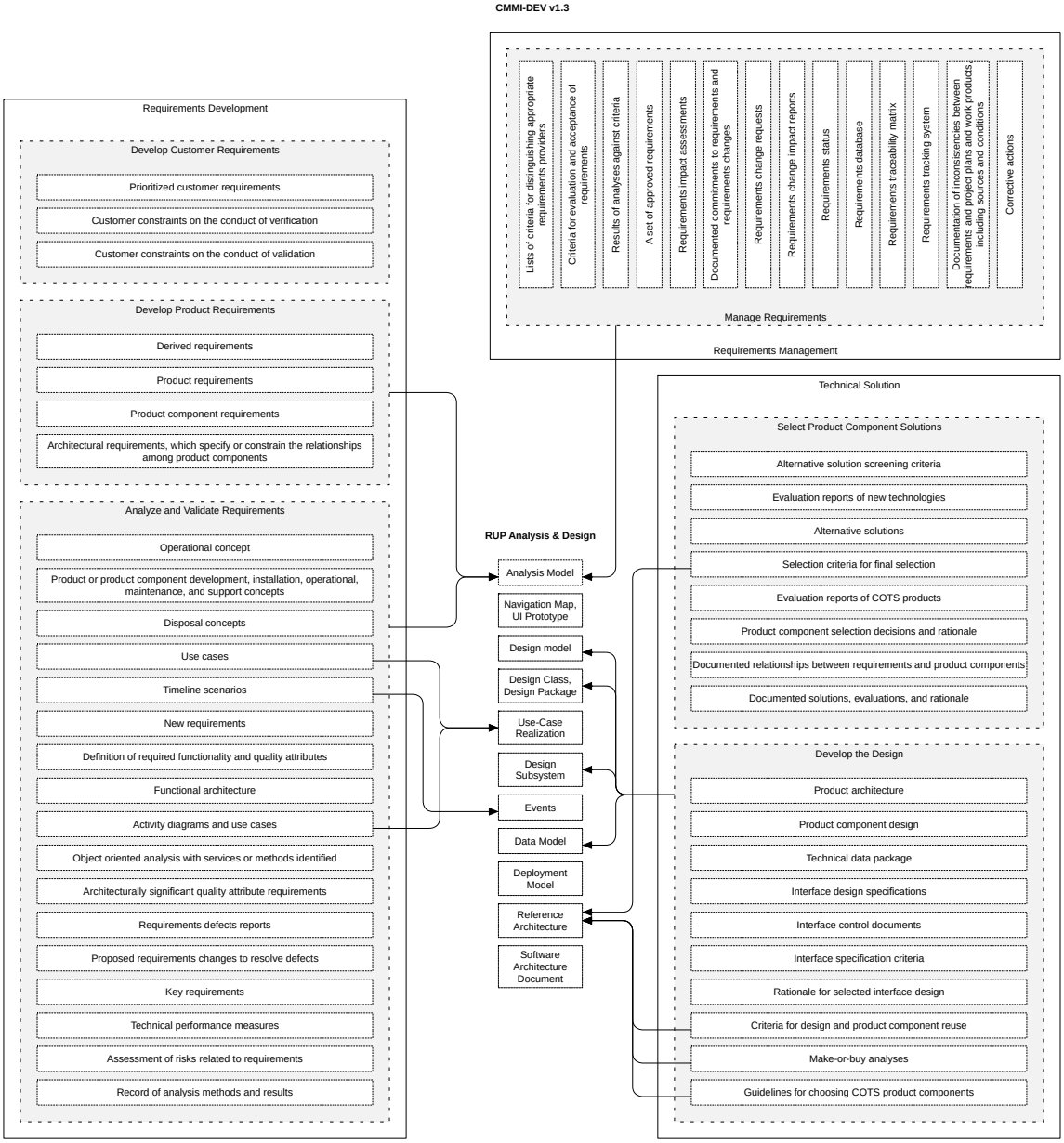
Technical Solution

Select Product Component Solutions

- Alternative solution screening criteria
- Evaluation reports of new technologies
- Alternative solutions
- Selection criteria for final selection
- Evaluation reports of COTS products
- Product component selection decisions and rationale
- Documented relationships between requirements and product components
- Documented solutions, evaluations, and rationale

Develop the Design

- Product architecture
- Product component design
- Technical data package
- Interface design specifications
- Interface control documents
- Interface specification criteria
- Rationale for selected interface design
- Criteria for design and product component reuse
- Make-or-buy analyses
- Guidelines for choosing COTS product components



Obrázek 5

Mapování CMMI-DEV v1.3 Outcomes na dokumenty v RUP Analysis & Design (zdroj: autor)

Mapování na RUP disciplínu Analysis & Design

Jak je znázorněno na Obrázek 5 CMMI-DEV v 1.3 dobře pokrývá téměř většinu popsaných dokumentů v RUP disciplíně Analysis & Design.

Dokument **Analysis Model** v RUP je velmi dobře pokryt Outcomes z procesních oblastí Develop Product Requirements, Analyze and Validate Requirements a Manage Requirements.

Pro dokumenty **Navigation Map a UI Prototype** v RUP nebyly nalezeny žádné Outcomes v CMMI-DEV v1.3. Je možné, že dokumenty tohoto typu vznikají v rámci procesní oblasti Develop the Design.

Dokument **Design Model** v RUP je ekvivalentní k Outcomes z procesní oblasti Develop the Design. Tato procesní oblast pravděpodobně pokrývá i dokumenty **Design Class** a **Design Package**, **Design Subsystem** a **Data Model**. Tyto dokumenty vycházejí z objektově orientovaného návrhu.

Dokument **Use-Case Realization** v RUP by bylo pravděpodobně možné nahradit Outcomes Use cases a Activity diagrams and use cases z procesní oblasti Analyze and Validate Requirements.

Dokument **Events** v RUP, který zaznamenává posloupnost událostí se podobá Outcome Timeline scenarios z procesní oblasti Analyze and Validate Requirements.

Pro dokument **Deployment Model** z RUP nebyla nalezena žádná shoda.

Pro dokument **Reference Architecture** z RUP byly identifikovány čtyři Outcomes, Selection criteria for final selection z PA Select Product Component Solutions a Criteria for design and product component reuse, Make-or-buy analysis a Guidelines for choosing COTS product components z procesní oblasti Develop the Design.

Pro dokument **Software Architecture Document** z RUP nebyla nalezena žádná shoda.

Pro Outcomes z procesní oblasti Develop Customer Requirements nebyly nalezeny ekvivalentní dokumenty v RUP.

3.5.3 ISO/IEC 12207

Technical Processes 6.4 (Technické procesy)

System Requirements Analysis Process 6.4.2 (Analýza požadavků systému)

Transforamce požadavků zúčastněných stran do sady technických požadavků podle kterých se vytvoří náveh systému.

- a defined set of system functional and non-functional requirements describing the problem to be solved are established;
- the appropriate techniques are performed to optimize the preferred project solution;
- system requirements are analyzed for correctness and testability;
- the impact of the system requirements on the operating environment are understood;
- the requirements are prioritized, approved and updated as needed;

- consistency and traceability are established between the system requirements and the customer's requirements baseline;
- changes to the baseline are evaluated for cost, schedule and technical impact; and
- the system requirements are communicated to all affected parties and baselined.

System Architectural Design Process 6.4.3 (Návrhy architektury systému)

Rozděluje systémové požadavky do jednotlivých skupin dle prvků systému.

- a system architecture design is defined that identifies the elements of the system and meets the defined requirements;
- the system's functional and non-functional requirements are addressed;
- the requirements are allocated to the elements of the system;
- internal and external interfaces of each system element are defined;
- verification between the system requirements and the system architecture is performed;
- the requirements allocated to the system elements and their interfaces are traceable to the customer's requirements baseline;
- consistency and traceability between the system requirements and system architecture design is maintained; and
- the system requirements, the system architecture design, and their relationships are baselined and communicated to all affected parties;
- human factors and ergonomic knowledge and techniques are incorporated in system design; and
- human-centred design activities are identified and performed.

Software Implementation Processes 7.1 (Procesy implementace softwaru)

Software Requirements Analysis Process 7.1.2 (Analýza SW požadavků)

Stanovení požadavků softwarových prvků systému.

- the requirements allocated to the software elements of the system and their interfaces are defined;
- software requirements are analyzed for correctness and testability;
- the impact of software requirements on the operating environment are understood;
- consistency and traceability are established between the software requirements and system requirements;
- prioritization for implementing the software requirements is defined;

- the software requirements are approved and updated as needed;
- changes to the software requirements are evaluated for cost, schedule and technical impact; and
- the software requirements are baselined and communicated to all affected parties.

Software Architectural Design Process 7.1.3 (Návrh SW architektury)

Vytváří návrh softwaru, který lze zpětně ověřit porovnáním s požadavky.

- a software architectural design is developed and baselined that describes the software items that will implement the software requirements;
- internal and external interfaces of each software item are defined; and
- consistency and traceability are established between software requirements and software design.

Software Detailed Design Process 7.1.4 (Detailní návrh SW)

Vytváří návrh software, který lze zpětně ověřit porovnáním s požadavky a je tak popisný z hlediska architektury, že lze na jeho základě programovat a testovat.

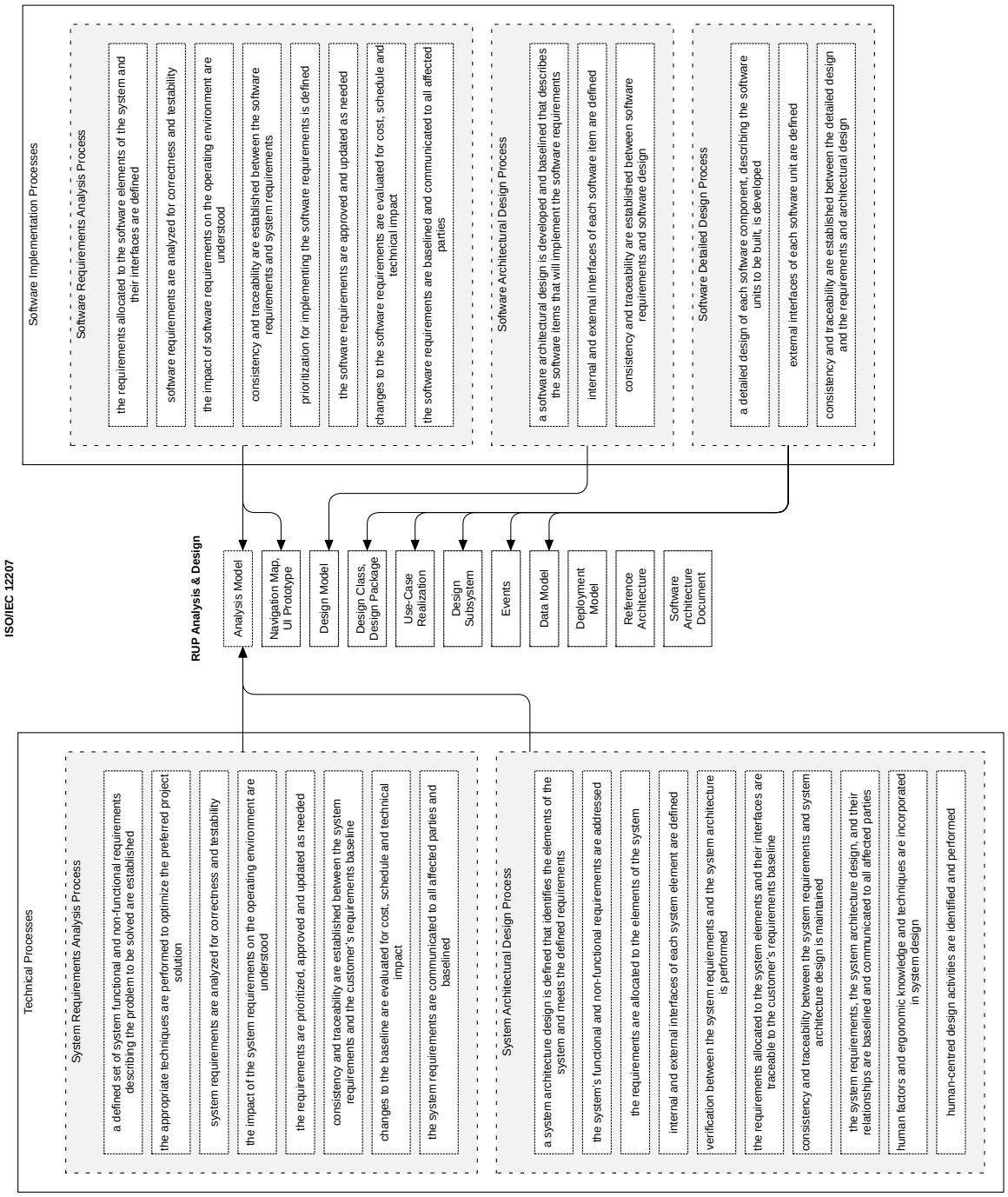
- a detailed design of each software component, describing the software units to be built, is developed;
- external interfaces of each software unit are defined; and
- consistency and traceability are established between the detailed design and the requirements and architectural design.

Mapování na RUP disciplínu Analysis & Design

Jak je znázorněno na Obrázek 6 ISO/IEC 12207 pokrývá 8 dokumentů z 11 v RUP disciplíně Analysis & Design.

Dokument **Analysis Model** v RUP lze nahradit Example Work Products z procesních skupin Technical Processes (procesy System Requirements Analysis Process a System Architectural Design Process) a Software Implementation Processes (Software Requirements Analysis Process). System Architectural Design Process řadím do Analysis Model z toho důvodu, že jedná o návrh systému na nejvyšší úrovni abstrakce, nikoliv o návrh architektury implementačně závislé na prostředí, ze kterého se vychází při realizaci.

Lze předpokládat, že dokumenty **Navigation Map** a **UI Prototype** v RUP by bylo možné pokrýt v procesu Software Requirements Analysis Process z procesní skupiny Software Implementation Processes, není to však úplně jednoznačně patrné.



Obrázek 6

Mapování ISO/IEC 12207 Example Work Products na dokumenty v RUP Analysis & Design (zdroj: autor)

Podobnost dokumentu **Design Model** v RUP s Example Work Products z procesu Software Architectural Design Process z procesní skupiny Software Implementation Processes je zřejmá.

Proces Sotware Detailed Design Process z procesní skupiny Software implementation Processes pokrývá hned několik dokumentů z RUP. Konkrétně to jsou **Design Cla**ss a **Design Package**, **Use-Case Realization**, **Design Subsystem**, **Events** a **Data model**.

Dokumenty **Deployment Model**, **Reference Architecture** a **Software Architecture Document** v RUP nejsou v ISO/IEC 12207 pokryty.

3.5.4 Výsledek porovnání dle kritéria

Ačkoliv nelze jednoznačně porovnávat dokumenty z RUP s Outcomes z CMMI-DEV v1.3 a Example Work Products z ISO/IEC 12207, pomocí mapování uvedeného výše je možné dojít k závěru, že počet vznikajících dokumentů je relativně stejný. Podrobnost popisu dokumentů nelze příliš hodnotit, vzhledem k tomu, že ani CMMI-DEV v1.3 ani ISO/IEC 12207 nenabízejí žádný popis.

3.6 Vyhodnocení porovnání

RUP disciplína Analysis & Design

Metodika obstála ve všech porovnávaných kritériích s menším nedostatkem v Podrobnost popisu, kde nejsou informace dostatečně dobře strukturovány v porovnání s CMMI-DEV v1.3 nebo ISO/IEC 12207. Na druhou stranu jako jediná detialně popisuje role a jejich zodpovědnosti.

CMMI-DEV v1.3

Referenční model pokrývá s výjimkou rolí všechna kritéria velmi dobře. Věcný obsah je dobrý, Podrobnost popisu je dokonce lépe strukturovaná jak v RUP a Outcomes, pokrývají většinu možných dokumentů. Mínus má tedy referenční model pouze za chybějící specifikaci rolí a jejich zodpovědností.

ISO/IEC 12207

Norma vyhovuje kritériu Věcný obsah velmi dobře, ovšem kritérium Podrobnost popisu příliš dobře nenaplňuje, popis jednotlivých procesů a procesních skupin nejsou v porovnání s ostatními dostatečné, neexistence rolí je také hodnocena záporně. Kritérium Dokumenty je relativně dobře naplněno, v porovnání s RUP nebo CMMI-DEV si však nevede úplně nejlépe, ba dokonce lze říci, že v tomto kritériu je z výše zmíněných nejhorší.

4 Závěr

Všechny cíle semestrální práce byly naplněny. V první části byla představena disciplána Analysis & Design z Rational Unified Process a byly identifikovány a popsány odpovídající procesní oblasti CMMI-DEV v1.3, které této disciplíně odpovídají. Dále byly identifikovány a popsány odpovídající procesy v ISO/IEC 12207.

Druhá část práce se zabývala stanovením kritérií pro porovnání, kde byla zvolena čtyři kritéria 1) věcný obsah, 2) podrobnost popisu, 3) role a 4) dokumenty. Následně byla porovnána disciplána Analysis & Design z metodiky RUP s odpovídajícími procesními oblastmi z referenčního modelu CMMI-DEV v1.3 a s procesy z normy ISO/IEC 12207.

Vyhodnocením porovnání jsme dospěli k závěru, že s vyjímkou kritéria role si relativně odpovídají, pouze norma ISO/IEC 12207 je často méně popisná, což ale vychází z její samotné podstaty. Role jsou definovány a popsány pouze v metodice RUP.

Výsledné pořadí na základě stanovených kritérií je následující.

1. RUP disciplína Analysis & Design
2. CMMI-DEV v1.3
3. ISO/IEC 12207

Zdroje

ALDORF, Filip. *Metodika RUP*. Vysoká škola ekonomická v Praze. Praha, 2008. [cit. 2015-12-06]. Dostupné z: http://objekty.vse.cz/Objekty/RUP. Diplomová práce. Vysoká škola ekonomická v Praze.

CMMI® for Development, Version 1.3: Improving processes for developing better products and services. *Carnegie Mellon* [online]. November 2010 [cit. 2015-10-19]. Dostupné z: http://www.sei.cmu.edu/reports/10tr033.pdf.

COLE, Leonardo, Eduardo Kessler PIVETA a Augusto SAMPAIO. *RUP Based Analysis and Design with Aspects.* [online]. Federal University of Santa Catarinas. 15, 2004 [cit. 2015-12-06]. Dostupné z: http://www.cin.ufpe.br/~lcn/publications/SBES2004_Cole_Piveta_Sampaio.pdf

CRAIN, Anthony. IBM. *The simple artifacts of Analysis and Design* [online]. 16 June 2004 [cit. 2015-10-19]. Dostupné z: http://www.ibm.com/developerworks/rational/library/4871.html.

GALLAGHER, Brian a Lisa BROWNSWORD. The Rational Unified Process and the Capability Maturity Model – Integrated Systems/Software Engineering. *Carnegie Mellon University: Software Engineering Institute* [online]. 2001, 2001 [cit. 2015-12-27]. Dostupné z: http://www.sei.cmu.edu/library/assets/rup.pdf

HRONEK, Jiří. Informační systémy [online]. Univerzita Palackého. Olomouc, 2007 [cit. 2015-12-06]. Dostupné z: http://phoenix.inf.upol.cz/esf/ucebni/infoSys.pdf

IBM CORP. *Rational Unified Process: RUP for Small Projects* [online]. 7.5.1. IBM, 2006 [cit. 2015-12-06]. Dostupné z: https://kitscm.vse.cz/RUP/SmallProjects/

ISO 12207. 2008. ISO/IEC 12207. *Geneva : Institute of Electrical and Electronics Engineers,* 2008. 0-7381-5664-7.

ISO/IEC 12207:2008(en): *Systems and software engineering — Software life cycle processes.* ISO [online]. 2008 [cit. 2015-10-19]. Dostupné z: https://www.iso.org/obp/ui/#iso:std:iso-iec:12207:ed-2:v1:en .

KADLEC, Václav. Rational Unified Process: základní pojmy. *Živě.cz* [online]. 5.srpna2003n. l. [cit. 2015-12-06]. Dostupné z: http://www.zive.cz/Clanky/Rational-Unified-Process-zakladni-pojmy/sc-3-a-113011/default.aspx

KRUCHTEN , Philippe. *The Rational Unified Process An Introduction, Second Edition.* Addison Wesley , March 14, 2000. Second Edition. ISBN 0-201-70710-1.

RUP Summary. *Suniwe* [online]. Staffordshire University Enterprises Ltd., 2008, 12/01/2008 15:29:18 [cit. 2015-12-06]. Dostupné z: http://projects.staffs.ac.uk/suniwe/project/rup.html

ŘEPA, Václav, Václav SYNÁCEK, Petr HAMERNÍK a Ondřej DIVIŠ. a kol. *Metodika vývoje informacního systému s pomocí nástroje Power Designer* [online]. Praha, cerven 2006 [cit. 2011-12-22]. verse 1. Dostupné z: http://opensoul.iquest.cz/forum/docs/publications/Metodika_vyvoje_IS_06_2006.pdf

SHUJA, Ahmad K. a Jochen KREBS. *IBM Rational Unified Process Reference and Certification Guide: Solution Designer* [online]. Pearson Education, 2007, 29. 12. 2007 [cit. 2015-12-06]. Dostupné z: https://books.google.cz/books?id=y54kLbokxhsC&dq=rup+perform+architectural+synthesis&hl=cs&source=gbs_navlinks_s

SOUKUP, Petr. *RUP – disciplína Analysis & Design* [online]. Vysoká škola ekonomická v Praze, 2010, 2010/11 [cit. 2015-12-06]. Dostupné z: https://doc-0k-4s-apps-viewer.googleusercontent.com/viewer/secure/pdf/7pjo76cevlu442gnb6rgim79rbbr5kq7/7j33q88p957th146idc1gndpetupokkf/1449425325000/drive/05538962632288843412/ACFrOgD9BIJBH4AF3Vih3SxgB_JXkaFozzt8lZwKAdgxPSJP8zAI43Lr8_n0Oh050pBYQwfcIM9-0I6cHEJLP1jriJAaDBeH3lX1U4wetBAFUxYLhLoYKh4Vb2buR8Y=?print=true&nonce=ccs5n5ube02a2&user=055389626322888843412&hash=ie1enluqrvprd9d5cfht31hdssrh776k