

2013

Vysoká škola ekonomická
Fakulta informatiky a
statistiky obor Informační
systémy a technologie

xpavt27, xkulb00

[SEMESTRÁLNÍ PRÁCE – CMMI & AGILNÍ METODIKY]

1 Obsah

2	Úvod	1
2.1	Vymezení tématu práce a důvod výběru tématu	1
2.2	Cíle práce a způsob jejich dosažení	1
3	Základní pojmy	2
3.1	CMMI	2
3.2	Agilní metodiky	4
3.3	Rozdíl mezi rigorózními a agilními metodikami	6
4	Proč je vhodné kombinovat CMMI a agilní metodiky	7
4.1	Problémy agilních metodik, které mohou být zmírněny přístupy CMMI	10
5	Implementace CMMI za využití agilních metodik	12
5.1	Implementace CMMI	12
5.2	Implementace agilních metodik	13
5.3	Implementace CMMI pomocí agilních přístupů	13
5.4	Studie – Implementace CMMI pomocí agilních metodik	14
6	Závěr	17
6.1	Shrnutí dosažených výsledků	17
6.2	Naplnění cíle práce	18
6.3	Náměty na další činnost	18
7	Bibliografie	19

2 Úvod

2.1 Vymezení tématu práce a důvod výběru tématu

Tématem této semestrální práce je propojování CMMI (Capability Maturity Model Integration) s agilními metodikami a agilním vývojem. CMMI, který lze přeložit jako model způsobilosti a zralosti, a který slouží ke zlepšování procesů ve firmě, se v současné době stále častěji diskutuje v kombinaci právě s agilními metodikami a agilním přístupem. Tento fakt přináší nové možnosti ve zlepšováním procesů a proto je vhodné se jím zabývat.

2.2 Cíle práce a způsob jejich dosažení

Cílem práce je tedy identifikovat výhody, popřípadě nevýhody daného spojení. Pro dosažení vytyčeného cíle byla provedena rešerše různých zdrojů, na jejímž základě byly vybrány relevantní dokumenty, sloužící jako teoretický základ. Skrze takto nalezené zdroje byl proveden informační průřez tak, aby čtenáři poskytl co nejucelenější pohled na danou problematiku.

3 Základní pojmy

3.1 CMMI

Jak již bylo zmíněno v kapitole „Úvod“ CMMI je model sloužící ke zlepšování zralosti a způsobilosti procesů, umožňuje tedy zvýšit výkonnost organizace a její schopnost dosahovat stanovených obchodních cílů. Stěžejní myšlenka tohoto modelu spočívá v kvalitě procesů, která udává kvalitu systému a produktu. CMMI je tedy definována jako sada best practices a cílů zlepšování procesů, které organizace využívají pro vyhodnocení a zlepšení jejich procesů (Novák, 2013). Základy tohoto modelu vznikaly již v 80. letech 20. století konkrétně v SEI (Software Engineering Institute) a společnosti IBM, kde se při softwarovém vývoji využívaly principy statické kontroly jakosti (quality) a komplexního řízení jakosti (total quality management, TQM). Tento model byl původně zaměřen pouze na oblast softwarového inženýrství, nicméně v průběhu 90. let 20. století došlo k určitému zobecnění a dnešní verze modelu může být využita v mnoha odvětvích (Kučera, 2008). V současné době existují 3 varianty modelu ve verzích 1.3 (CMMI Institute, 2013):

- CMMI for Acquisitions:
 - Model poskytuje návod organizacím, které spravují dodavatelský řetězec k zisku a integraci produktů a služeb k uspokojení potřeb zákazníka.
- CMMI for Development:
 - Model je určen pro zlepšování procesů v organizacích vyvíjejících produkty. CMMI-DEV poskytuje návod na zlepšení účinnosti, efektivity a kvality jejich práce při vývoji produktů.
- CMMI for Services:
 - Model poskytující návod organizacím, které zakládají, řídí a poskytují služby, které uspokojují potřeby zákazníků a koncových uživatelů.

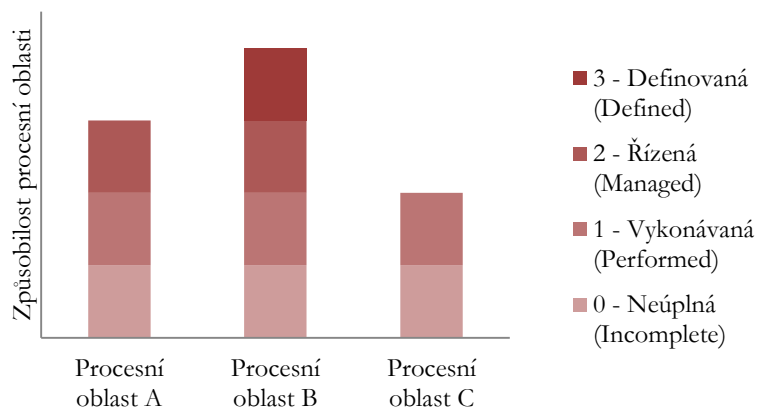
Míra zralosti či způsobilosti procesu je vždy vyjádřena pomocí definovaných úrovní sloužících k popisu vývoje cest doporučených pro organizace, které chtějí zlepšit jejich procesy využívané k získávání, rozvoji a dodávání produktů a služeb.

Také tak mohou tyto úrovně sloužit jako výstup hodnocení procesů v rámci tzv. SCAMPI systému (Standard CMMI Appraisal Method for Process Improvement). Důvody proč společnosti podstupují toto hodnocení, jsou obvykle následující (Wikipedia, 2013):

- Odlišení, jak dobře jsou procesy organizace nastaveny v porovnání s CMMI best practises a identifikace oblastí, které vyžadují zlepšení,
- sdělení externím zákazníkům a dodavatelům, jak dobře jsou procesy organizace nastaveny v porovnání s CMMI best practises,
- splnění smluvních požadavků jednoho nebo více zákazníků.

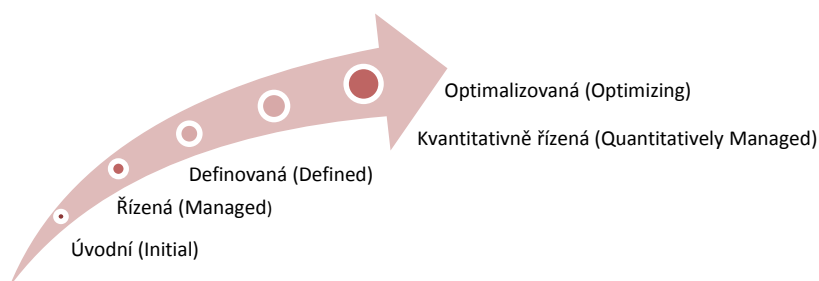
Výše 3 zmíněné varianty modelu podporují jak úrovně zralosti, tak i způsobilosti.

Capability levels nebo také úrovně způsobilosti slouží organizaci k soustředění svého úsilí o procesní zlepšení na konkrétní procesní oblast. Dané oblasti může být přiřazena úroveň způsobilosti 0 až 3. Po definování procesní oblasti, je dalším krokem rozhodnutí, které uvádí, v jaké míře plánuje organizace procesy (spojené s vybranou procesní oblastí) zlepšovat, tedy jaké úrovně chce dosáhnout. Dosažení určitých úrovní v organizaci, pak může být, pomocí grafu, zachyceno následujícím způsobem. (CMMI Institute, 2013)



Obrázek 2.1 - Příklad úrovní jednotlivých procesních oblastí

Maturity levels neboli úrovně zralosti poskytují inscenaci procesů pro celkové zlepšení napříč celou organizací a to od úrovně 1 až 5. Toto zlepšení zahrnuje dosažení cílů procesních oblastí, na každé úrovni zralosti. Procesní oblasti mohou být seskupeny dle úrovně zralosti. (CMMI Institute, 2013)



Obrázek 2.2: Úrovně zralosti procesů.

Jednotlivé úrovně vychází ze CMMI-DEV, V1.3 a byly převzaty z reportu vydaným Software Engineering Institute (Software Engineering Institute, 2010)

3.2 Agilní metodiky

(Buchalceová, 2005) v své knize „Metodiky vývoje a údržby informačních systémů“ uvádí, že neustále se zvyšující nároky na podniky, s čímž souvisí časté změny technologií a ekonomického prostředí, v sobě odráží sílící požadavky společností na co nejrychlejší zavedení IS/ICT. Aby bylo možné tyto požadavky plně uspokojit je nutná změna používaných metodik, především pak přístup k vývoji a zavádění, se kterým dané metodiky pracují. Původní, tzv. rigorózní, metodiky již nejsou v současné době, kvůli zmíněným měnícím se podmínkám, schopny pokrýt veškeré požadavky a nároky klientů, z tohoto důvodu se začínají prosazovat takové metodiky, které umožňují vytvořit řešení velmi rychle a pružně jej přizpůsobovat měnícím se požadavkům a prostředí. Tyto relativně nové metodiky jsou označovány jako agilní, jedná se soubor různých metodik, které postupně vznikaly od druhé poloviny 90. let. Agilní metodiky stojí na myšlence, která uvádí, že jedinou cestou, jak prověřit správnost navrženého systému, je vyvinout jej (nebo jeho část) co nejrychleji, předložit zákazníkovi a na základě zpětné vazby jej upravit. Ačkoli se jednotlivé agilní metodiky od sebe určitým způsobem liší, tak všechny jsou postaveny na stejných principech a hodnotách. V roce 2001 byl podepsán „Manifest agilního vývoje software“ a byla vytvořena „Aliance pro agilní vývoj software“.

Zmíněný „Manifest agilního vývoje software“ uvádí:

„Odhalili jsme lepší způsob vývoje software, sami jej používáme a chceme pomoci i ostatním, aby jej používali. Z toho pohledu dáváme přednost:

- individualitám a komunikaci před procesy a nástroji,
- provozuschopnému software před obsažnou dokumentací,
- spolupráci se zákazníkem před sjednáváním kontraktu,
- reakci na změnu před plněním plánu.“

Na základě těchto základních hodnot bylo stanoveno deset stěžejních principů agilních metodik:

- Nejvyšší prioritou je včas a kontinuálně dodávat software, který zákazníkům přináší hodnotu
Tento princip předpokládá, že pro zákazníka nejsou důležité dokumenty, diagramy či integrace stávajících systémů. Stěžejní se tak pro zákazníka stává jistota, že v každé iteraci obdrží fungující software a že poskytovaná funkcionalita splní jeho očekávání a potřeby.
- Změnu požadavků je možné provést i v pozdějších fázích vývoje, protože tím může zákazník získat konkurenční výhodu
Cílem je realizovat změny efektivně a řídit rizika negativních dopadů. Za stěžejní lze pokládat realizaci krátkých iterací (dva týdny až dva měsíce), na jejichž konci je dodáván fungující software. Ačkoli je iterativní vývoj prosazován též rigorózními metodikami, tak agilní přístupy se soustředí na zkrácení cyklu dodání přírůstků.
- Uživatelé a vývojáři spolupracují denně na projektu
Agilní způsob vývoje předpokládá, že není možné dohodnout a podepsat veškeré požadavky již na začátku projektu. Proto jsou v úvodu definovány požadavky pouze hrubě a jsou vyladěny, zdrobňovány nebo měněny až v průběhu projektu.

- Motivovaní jedinci, kteří mají vytvořeny podmínky pro práci a mají podporu vedení, jsou klíčovým faktorem úspěchu projektu
Princip zdůrazňuje dostatečnou motivovanost zainteresovanost lidí, kteří se na projektu podílí, jelikož jsou to vždy lidé, na kterých projekty stojí.
- Nejeфекtivnějším způsobem přenosu informací v rámci vývojového týmu je osobní komunikace
Není nutné tvořit obsáhlou dokumentaci k projektu, aby jej bylo možné dostatečně pochopit. Přímá komunikace je efektnější a šetří též čas a tedy i náklady.
- Primární mírou úspěchu je fungující software
Princip se snaží vyjádřit, že Ani plnění plánu, ani existující dokumentace návrhu nezajistí úspěch projektu.
- Agilní procesy předpokládají „zdravý“ vývoj
Termínem „zdravý“ vývoj je myšlena snaha vyvarovat se přehnané zátěži (práce přesčas, o víkendech) členů projektového týmu a vymezit pracovní prostor (cca 40 hodin týdně), kdy je tým schopný zůstat v dobré kondici.
- Perfektní technické řešení i návrh
Ačkoli jsou agilní metody v rychlosti a flexibilitě podobné rychlému vývoji aplikací (RAD Rapid Application Development), tak zásadně se liší v kvalitě návrhu a řešení. Agilní přístupy zdůrazňují kvalitu návrhu, která je klíčová pro realizaci změn.
- Zásadním požadavkem je jednoduchost řešení, tj. umění maximalizovat množství neudělané práce
- Nejlepší architektury, požadavky a návrhy vznikají ze samoorganizujících se týmů
Velmi důležitá je kreativita lidí, častá komunikace a přizpůsobování metodiky

Mezi agilní metodiky lze zahrnout následující:

- Dynamic Systems Development Method (DSDM),
- Adaptive Software Development (ASD),
- Feature–Driven Development (FDD),
- Extrémní programování (Extreme Programming, XP),
- Lean Development,
- Scrum,
- Crystal metodiky,
- Agilní modelování (Agile Modeling).

3.3 Rozdíl mezi rigorózními a agilními metodikami

(Popelka, 2009) ve své bakalářské práci s názvem „Srovnávací analýza metodik vývoje software“ popisuje hlavní rozdíly mezi rigorózními a agilními metodikami a to podle zvolených hledisek porovnání:

	Rigorózní metodiky	Agilní metodiky
Proces vývoje software	Přesně a podrobně definovaný proces, lze ho opakovat.	Empirický proces, není možné ho opakovat, ale vyžaduje adaptaci.
Změny a požadavky	Minimalizace změn (sběr požadavků předem a plánování předem), změny jsou součástí řízení změn.	Akceptace změn (přírůstkové shromažďování požadavků, plánování pro iteraci) přehodnocení požadavků, snaha o umožnění změn (v závislosti na nových znalostech).
Zapojení zákazníka na řízení projektu	Bez zásahu. Nedůvěra v zákazníka - Po podpisu smlouvy a vymezení dokumentu specifikace požadavků v počáteční fázi, zákazníka zajímá až konečné řešení.	Participace a spolupráce - Zákazník se aktivně zapojuje do celého projektu, může měnit priority funkcí při každé iteraci.
Kvalita	Více zaměřeny na kvalitu vlastních procesů, než na výsledek pro zákazníka.	Zaměření na priority zákazníka, na užitnou hodnotu pro zákazníka, tedy na výslednou kvalitu produktu.
Způsob a rozsah řešení	Podrobný popis procesů a činností, složité řešení. Při vývoji obsaženy všechny funkce, snaha o zabudování budoucích požadavků.	Eliminace nepotřebných činností, jednoduché řešení. Při vývoji zahrnuty jen potřebné funkce, snaha o minimalizaci, žádné začlenění budoucích požadavků.
Rozložení teamového know-how	Každý člen týmu má přiřazený činnost, v rámci role, kterou zastává. Požadavek na specializaci zaměstnanců. Převládá písemná forma komunikace (dokumentace).	Sdílení znalostí v týmu, spolupráce (kooperace) a aktivní komunikace v rámci týmu, společné řešení problémů, řízení a integrace znalostních toků.
Lidský faktor	Pohled na lidi jako na sekundární faktor, procesy doplňovány rozsáhlou dokumentací. Jedinci jsou nahraditelní.	Lidé jako primární a klíčový faktor úspěchu projektu. Využití schopností a dovedností jedinců (individualit), důraz na znalosti, kvalifikaci.
Způsob vývoje	Vodopádový životní cyklus, iterativní s dlouhými iteracemi.	Přírůstkový (inkrementální) vývoj s velmi krátkými iteracemi.

Tabulka 3-1: Porovnání agilních a rigorózních metodik. Zdroj: (Popelka, 2009)

Ačkoli se tato práce zaměřuje na Agilní metodiky a CMMI, tak vzhledem k tomu že CMMI patří do rigorózních přístupů, je dle autorů práce důležité, poukázat na výše zmíněné rozdíly. Toto srovnání totiž napomůže k pochopení další části textu, která se snaží odpovědět na otázku „Proč a zdali vůbec je vhodné kombinovat CMMI a agilní metodiky?“

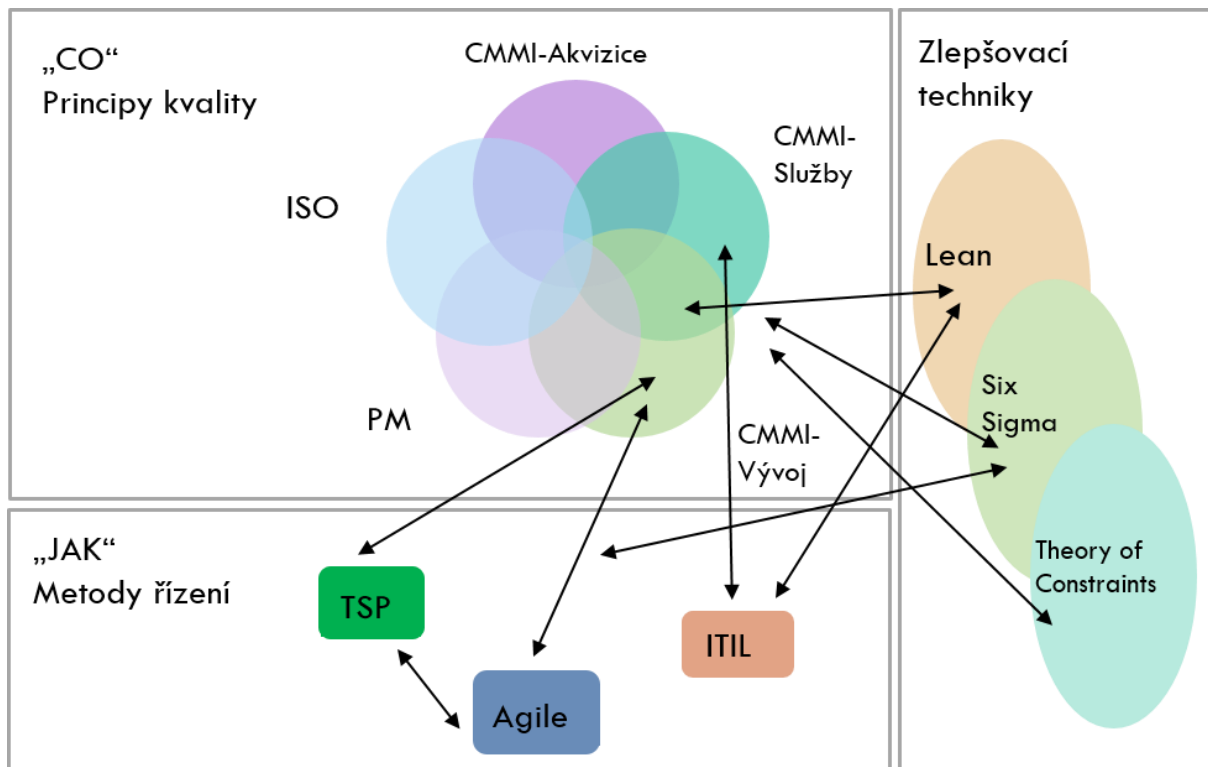
4 Proč je vhodné kombinovat CMMI a agilní metodiky

Mnoho společností, které se rozhodli přijmout a do svého fungování integrovat agilní přístupy, se zároveň zajímá například o PMI's OPM3 (Organizational Project Management Maturity Model), ISO (International Organization for Standardization) nebo CMMI (Capability Maturity Model® Integration). Jak uvádí (Shelton, 2008) firmy, které se pokoušely zároveň implementovat agilní metodiky a PMI model nebyly úspěšné, naopak dvě velké společnosti, které Cindy Shelton pozorovala, byly obě schopné úspěšně implementovat agilní metodiky i CMMI zároveň. Shelton uvádí, že velkým problémem je fakt, že skupina implementující CMMI vidí konkurenci ve skupině implementující agilní metodiky. Naopak by tyto dva přístupy měly fungovat spolehlivě vedle sebe, a co více, spolupracovat. Jelikož oba přístupy umožňují vyvíjet software vysoké kvality a to v co nejkratší možné době.

Zajímavé je, že CMMI bylo vytvořeno k řešení některých stejných problémů, kvůli kterým byly též vytvořeny agilní postupy. Problémem zůstává, že vzhledem k tomu, že obě tyto metodiky přinášejí dva rozdílné přístupy tak každý z těchto „přístupů“ má svou komunitu loajálních příznivců. Ti, co kritizují CMMI, prosazují názor, že je až příliš založeno na přehnané byrokracii a také dává přednost procesům, před samotnými výrobky, což narušuje dohodnuté time-to-market požadavky, které jsou pro dobře fungující byznys stěžejní. Na druhé straně kritici agilních přístupů namítají, že agilní přístupy přinášejí nedostatečnou kontrolu a v dosažených výsledcích nejsou zdokumentovány veškeré změny, implementované v průběhu projektu, což přináší obecný chaos.

Pokud se ovšem zaměříme čistě na oblast softwarového inženýrství, není toto striktní oddělení metodik vhodné a nepřispívá k maximální efektivitě. Důvodem, proč tyto odlišné „tábory“, příznivců ať už agilních přístupů nebo CMMI, odmítají připustit vzájemnou koexistenci, je podle Jeffreyho Daltona (ExecutiveBrief, 2009), prezidenta společnosti Broadsworld, především nedostatečná znalost těchto metodik (příznivci CMMI neznají podrobně agilní přístupy a naopak). Jelikož přínosy koexistence těchto dvou přístupů jsou značné a to od těch potencionálních až k dramatickému zlepšení celého byznysu.

Příklad toho jak se tyto přístupy mohou vzájemně doplňovat je obsažen v projektové rovině, kdy CMMI se soustředí na otázku „CO“ jednotlivé projekty opravdu dělají a zajišťují, kdyžto agilní metodiky pomáhají s otázkou „JAK“ v projektu vytvářet produkty (viz obrázek č. 3.1). Obecně lze konstatovat, že CMMI je v podstatě souhrn tematických oblastí, které je třeba řešit v případě zlepšování procesů, na druhou stranu agilní přístupy jsou podmnožinou témat a určují, jak mohou v určitých situacích fungovat. Jinými slovy CMMI poskytuje technické systémové postupy, které napomáhají k tomu, aby byly agilní metodiky využívány také na projektech většího rozsahu. Agilní metodiky poskytují návody k vývoji softwaru, které chybí u CMMI best practises.



4-1 Vzájemná podpora CMMI a agilního přístupu

Tabulka 3.1 představuje konkrétní rozdíly mezi CMMI a agilními přístupy (Glazer, a další, 2008):

Implementace agilních metodik do společnosti by měla být přizpůsobena aktuální úrovni zralosti procesů. Ideálním časem pro přijetí agilních postupů je označován stav, kdy jsou procesy ve společnosti na úrovni 3 dle pohledu CMMI. V případě, že CMMI level je nižší než právě 3, postrádá organizace stabilní procesy využívané při realizaci projektů. Ve společnostech mající CMMI na úrovni 1 záleží procesní změny především na konkrétním uživateli nebo události, lze tedy říci, že jsou především intuitivního charakteru. Agilní projekty mohou být za i za této situace úspěšné, většinou ovšem nejsou opakovatelné. Organizace tak nemají stabilní prostředí a nemusejí tak znát nebo rozumět všem prvkům, které toto prostředí tvoří. Celkový úspěch v organizacích s touto úrovní CMMI tedy závisí především na institucionální znalosti a kompetencí jednotlivých lidí ve společnosti, stejně tak jako na míře úsilí vynaloženého týmem. Při dosažení druhého jsou již některé procesy opakovatelné, nelze ještě ovšem říci, že pro všechny projekty. V případě, že jsou agilní metody a opatření do organizace implementovány na této úrovni, poskytují strukturu a disciplínu potřebnou pro posun na CMMI level 3. S tím také souvisí fakt, že implementace agilních přístupů na druhé úrovni je dražší, na druhou stranu se tímto ukáže, jak jsou tyto postupy přínosné. Pokud je naopak CMMI na úrovni vyšší než 3 omezuje výhody agilních přístupů.

Oblast	CMMI	Agilní přístupy
Zaměření	Soustředí se především na organizaci jako celek.	Soustředění je směřováno na projekt a projektový tým. Agilní přístupy mohou tým/projekt do jisté míry izolovat od celé organizace.
Řízení	Management je zde naprosto stěžejní a zajišťuje úspěch projektu.	Manager zde má roli spíše určitého „kouče“ a nejedná se tedy o způsob jako u CMMI příkaz-kontrola. Manager především pomáhá odstraňovat překážky bránící rozvoji.
Důvěra	CMMI spadá do tzv. „low-trust“ prostředí, které se vyznačuje především vysokým rizikem selhání.	Patří do „high-trust“ prostředí a předpokládá, že tým funguje nejlépe, pokud je složený z jednotlivců, kteří jsou odborníky v různých oblastech a kooperují spolu.
Plánování	Ačkoli rigorózní metodiky plánují především na začátku projektu téměř celý projekt, u CMMI toto nemusí platit. Využívá způsob, při kterém dochází k podrobnému naplánování pouze do další iterace.	Agilní metodiky plánují také především na začátku každé iterace a soustředí se na změny požadavků zadavatele. Podle agilních přístupů není možné celý projekt naplánovat již v počátku.
Předpoklady trhu	CMMI se uplatňuje zejména u vyspělých trhů, ve kterých již inovace procesů hraje klíčovou roli při úspěchu.	Agilní metodiky jsou využívány u začínajících dosud ne jasně vymezených trhů.
Návrh	Předpokládá, že architekturu je možné navrhnout již na začátku projektu.	Architektura na začátku navržena jen hrubě a je dolaďována až v průběhu projektu.
Perspektiva	Dlouhodobé projekty	Krátkodobé až střednědobé projekty
Hodnocení	Procesy jsou hodnoceny na základě tzv. metody SCAMPI	Úspěch lze konstatovat pouze při pohledu na výsledky projektu.
Náklady na selhání	Vzhledem k tomu, že se společnosti s uplatňováním CMMI soustředí na dlouhodobé a drahé projekty je zde riziko selhání poměrně vysoké.	Krátkodobé, ne příliš drahé projekty jsou označovány jako nízko rizikové.

Tabulka 3.1: CMMI vs. Agilní přístupy

4.1 Problémy agilních metodik, které mohou být zmírněny přístupy CMMI

Agilní hodnoty, principy a prakticky se primárně soustředí na projektovou úroveň a projektový tým a jejich objekty do značné míry závisí na individualitách. CMMI cíle a praktik předpokládají, že organizace chce své procesy široce propagovat a to bez časového a prostorového omezení. Proto CMMI poskytuje organizační infrastrukturu a také mechanismy k

- zachování informace a znalosti v průběhu času,
- strukturování a poskytování kritérií k rozhodování,
- posílení a normalizaci risk managementu,
- metodické aplikaci technických přístupů,
- specifikaci soustředění na procesní zlepšování)
- specifikaci technických postupů.

CMMI také zahrnuje praktiky pro normalizaci procesů na projektové i organizační úrovni. Právě tyto praktiky jsou ve společnostech využívající agilní přístupy často přehlíženy a to kvůli faktu, že agilní přístupy jsou velmi často aplikovány pouze v krátkodobých projektech

Agilní metodiky nevyužívají výhodu kvantitativních procesů, mohou existovat námitky, že není potřeba kvantitativní optimalizace v případě, že proces probíhá s dostatečnou frekvencí pravidelností, takže přínosy kvantitativního modelu zde existují – to platí u projektů: projekty zcela odlišné od ostatních, krátkodobé projekty s minimálním úsilím – ovšem projekty, na kterých se podílí více než 1-2 lidé, trvají měsíce, si již kvantitativní procesy žádají. Ve skutečnosti mohou být projekty popsány, stabilizované, normalizované a optimalizované pouze s několika iteracemi nebo sprinty. CMMI se u větších procesů soustředí na subprocessy. Opatření jsou pak prováděna tam kde to má obchodní smysl (Glazer, 2010)

Příklady jsou:

- Continuous integration.
- Test-driven development.
- Sprint/iteration planning.
- Refaktoring
- Pair programming.

Problémy CMMI, které mohou být zmírněny Agilními přístupy

Agilní metodiky zdůrazňují, že je důležitější soustředit se na individuality a komunikaci, fungující produkty, spolupráci se zákazníkem, reakce na změny nežli na procesy, nástroje, zbytečně obsáhlou dokumentaci a dodržování prvního plánu.

Pod hrozbou možné ztráty dosažené CMMI úrovně mnoho společností nerado experimentuje a mění své procesy, jelikož dosažení určité CMMI úrovně je velmi nákladné a to jak časové tak také finančně, manažeři proto nechtějí riskovat ztrátu dosažené úrovně.

Díky agilním metodikám mohou společnosti implementovat CMMI, aniž by se museli stát nejdříve odborníky ve zlepšování procesů – Snaha o vyšší úroveň se tak změnila na snahu o výsledky, jelikož společnost dosáhne vyšší úroveň svým vlastním způsobem. (Glazer, 2010)

5 Implementace CMMI za využití agilních metodik

V předchozí kapitole byly popsány důvody, proč je vhodné kombinovat CMMI s agilními přístupy. V této kapitole je nastíněn pohled na CMMI a agilní metodiky ještě z jiného úhlu. Agilní metodiky nejen, že se s CMMI nevylučují, ale mohou pomoci dokonce dosahovat vyšších úrovní CMMI.

5.1 Implementace CMMI

Jak již bylo popsáno dříve, CMMI je model sloužící ke zlepšování zralosti a způsobilosti procesů v podniku. Jestliže se tedy podnik chystá implementovat CMMI, chystá se upravovat mnoho interních podnikových procesů. Cílem je implementovat CMMI do takové míry, aby nebyl narušen chod firmy a hlavně v dlouhodobém hledisku implementace byla pro firmu přínosem. CMMI neobsahuje přímo pracovní postupy, které by se daly jednoduše převzít. Je třeba ho chápat spíše jako soubor „best practices“. Modelem se tak je vhodné řídit, ale ne dogmaticky přejímat veškerá doporučení i za cenu snížení efektivity v podniku. Je třeba najít určitý poměr převzatých doporučení a *zdravého rozumu*.

Toto dobře vystihuje (PDQM.cz, 2013):

„ Model CMMI je tak obsáhlý a pracovních postupů a výstupů v něm navrhovaných je tolik, že kdyby je normální komerčně fungující softwarová firma všechny převzala, do roka nejspíš zkrachuje.“

Nemá zkrátka smysl snažit se za každou cenu převzít co nejvíc postupů z CMMI, hlavním cílem je zachování jak kontinuity byznysu, ale i jeho specifické konkurenční výhody. Zkrátka, kdyby všechny firmy implementovaly úplně stejné praktiky, byly by všechny stejné. Nemusely by být v každém odvětví desítky nebo stovky firem, stačila by vždy jedna.

Implementaci CMMI lze podle (Quality Institute of America, 2009) rozdělit do několika kroků:

- Porozumění modelu
- Příprava k posouzení
- Zřízení procesně – zlepšovací infrastruktury, tedy týmu odborníků
- Příprava k rozvoji a samotný rozvoj vyspělého procesu
- Ustanovení procesní dokumentace
- Nezávislé posuzování procesní oblasti
- Posouzení

Je zde vidět, že nasadit CMMI není jednoduchý proces. Kromě samotné úpravy některé procesní oblasti zahrnuje i například tvorbu dokumentace nebo vyčlenění speciálního týmu v podniku. Stejně tak je důležité, aby se zamýšleným změnami byli seznámeni všichni zaměstnanci.

5.2 Implementace agilních metodik

Nasazení agilních metodik do podniku je na první pohled jednodušší než implementace CMMI. Už kvůli tomu, že, jak bylo uvedeno v kapitole 3, agilní metodiky zasahují „pouze“ do procesní oblasti, ne do samotné struktury a fungování podniku. Při nasazování agilních postupů jde ale z velké části o změnu přístupu jednotlivých členů (nejen) vývojového týmu. (Agilemethodology.org, 2008)

Nasazení agilní metodiky znamená velkou změnu pro jedince a zaměstnanci tak musí být ke změně výrazně motivováni. Stejně tak poté – nasazená agilní metodika bez motivovaných zaměstnanců nemůže fungovat. Už například kvůli samořídícím se týmům a vůbec velké odpovědnosti přenesené z manažera na každého z členů týmu (např. u Scrumu).

Některé agilní metodiky nutí programátory radikálně měnit způsob jejich práce – jsou to například extrémní programování s praktikou párového programování, nebo Test Driven Development, kde je vývojář nucen nejdříve psát test a až poté samotnou funkcionalitu. Další velkou změnou, kterou agilní metodiky do podniku přinášejí, je mnohem více komunikace. Ať v týmu, v podobě denních meetingů u Scrumu, nebo se zákazníkem, který má být podle Extrémního programování členem vývojového týmu.

Je důležité si uvědomit, že přestože mají agilní metodiky dopad na menší oblast, jejich implementace není jednoduchou záležitostí práce kvůli tomu, že si jednotliví členové týmu musejí zvyknout na zcela jiný styl práce. Dále je třeba také uvažovat, že ne všechny typy osobností se hodí pro práci v agilním týmu.

5.3 Implementace CMMI pomocí agilních přístupů

(PDQM.cz, 2009) uvádí, že CMMI se vhodně doplňuje s metodikami pro řízení projektů, jako jsou například Scrum. Důvodem je fakt, že model CMMI model nepředepisuje ani nepopisuje konkrétní interní procesy nebo rozdělení úkolů mezi jednotlivé pracovníky. CMMI stanovuje cíle, která podporuje doporučeními, jak jich dosáhnout. Přináší obecný rámec a souhrn činností, které jsou pro kvalitní práci potřebné

„ It can be extremely time-consuming and expensive to climb the CMMI ladder. But agile can change this! Working in short iterations makes the cycle times much shorter and the learning opportunities are 10 to 100 times more frequent! “

(Amsjo, 2008) tedy říká, že „šplhání po žebříku CMMI“ může být extrémně časově náročné a drahé. Ale agilita tohle může změnit. Práce v krátkých iteracích celý cyklus mnohonásobně zkracuje a nabízí mnohonásobně více příležitostí k učení.

Na implementaci CMMI tedy lze nahlížet jako na jakýkoliv jiný projekt. A na práci na projektu lze použít agilní postupy. Jednotlivé cíle CMMI a agilní praktiky se nevyklučují – naopak. Lze tedy „namapovat“ jednotlivé cíle a procesní oblasti na jednotlivé praktiky agilních metodik. (Hurtado, 2006)

Agilní praktiky tak přímo pomáhají dosahovat stanovených cílů a vyplňují tak „mezeru“, kterou CMMI neobshuje, tedy konkrétní pracovní postupy nebo interní procesy.

5.4 Studie – Implementace CMMI pomocí agilních metodik

V roce 2006 byla provedena v Latinské Americe studie mezi softwarovými společnostmi zaměřená právě na implementaci CMMI pomocí agilních metodik. (Hurtado, 2006)

Tato geografická oblast je charakteristická velkým množstvím malých a středních softwarových společností. Jedná se tedy o trh, který se rychle rozrůstá a zatím není příliš vypělý. Snahou v této lokalitě tedy je zvyšovat kvalitu pomocí dosahování standardů, které společností mohou pomoci dosáhnout dvou cílů:

- Zlepšení image společnosti v rámci mezinárodního trhu a tím otevření nových možností
- Zefektivnění vlastního výhoje a s tím snížené například administrativní náklady

Většina firem v dané lokalitě se v počátku rozhodla pro proces vývoje softwaru implementovat agilní metodiky. Důvodem pro nasazení agilních metodik zde bylo nejčastěji zachování konkurenčních výhod konkrétních společností spolu s nízkými vstupními náklady.

Uvedená studie se zabývá právě implementací CMMI, jakožto snahy o standardizaci, pomocí agilních metodik, které zde již jsou rozšířené. Autoři se zde snažili stanovit požadavky pro dosažení certifikace CMMI na úrovni 2 a 3.

Základem studie je předpoklad, že každé generické, tedy obecné praktice, chceme-li cíli, náleží i konkrétní praktika, chceme-li, metoda. Tzv. vyžadovaná komponenta popisuje požadavky, které organizace musí splnit aby dosáhla uspokojivé úrovně na dané procesní oblasti. Tzv. Očekávaná komponenta popisuje, co musí organizace implementovat, aby požadavků dosáhla.

K jednotlivým procesním oblastem tak byly na základě této úvahy přiřazeny konkrétní praktiky agilních metodik, které se jeví být v dané oblasti užitečné. Nejprínosnější metodiky napříč jednotlivými procesními oblastmi shrnuje tabulka 4.2

Procesní oblast	Nejvíce přínosné metodiky
Zajištění jakosti produktů a procesů	FDD, Crystal
Měření a analýza	XP, ASD, Scrum
Plánování projektů	Scrum, XP
Monitorování a řízení projektů	XP, Scrum

Tabulka 5-1: Přínosné metodiky v jednotlivých procesních oblastech. Zdroj: (Hurtado, 2006)

V oblasti **Zajištění jakosti produktů a procesů** vynikala metodika Feature Driven Development, například kvůli individuálnímu vlastnictví funkcionalit. V situaci, kdy si každý

vývojář zodpovídá za *své* funkcionality nebo jejich skupiny, se dá mnohem lépe kontrolovat kvalita. V oblasti **Měření a analýza** lze vidět největší přínos u metodiky Extrémní programování, například díky zastoupené roli tzv. Trackera. Další přínosnou metodikou je zde Adaptive Software Development s praktikou tzv. Time Boxů. V oblasti **Plánování projektů** hrají silnou roli praktiky Scrumu, například denní meetingy, retrospective meetingy případně sprint backlogy. Scrum má plánovací techniky velmi dobře propracované a je zde tak velkým přínosem. Stejně tak tomu je u procesní oblasti **Monitorování a řízení procesů**. Problematickou procesní oblastí se ukázalo být **Řízení konfigurací**, kde nebyly nalezeny žádné agilní praktiky, které by zde výrazně pomáhaly.

Procesní oblast Řízení požadavků je zde detailně popsána, shrnutí nabízí tabulka 4.1.

Metodika	Přínos
XP	Přítomnost zákazníka na projektu, zpětná vazba
Scrum	Denní meetingy, product backlog
FDD	Seznam funkcionalit
ASD	Tolerance změn
DSDM	Návratnost změn

Tabulka 5-2: Metodiky a jejich přínos v implementaci. Zdroj: (Hurtado, 2006)

V procesní oblasti Řízení požadavků bylo identifikováno několik metodik, které tuto oblast mohou napomoci zlepšovat. Jsou jimi například Extrémní programování, Scrum, Feature Driven Development, Adaptive Software Development a Dynamic Systems Development Methodology.

Přínos metodiky **Extrémní programování** spočívá především v přítomnosti zákazníka na projektu. Lze tak průběžně získávat zpětnou vazbu a zapracovávat tak včas jeho požadavky. Metodika **Scrum** ve vztahu k řízení požadavků napomáhá nejvíce pravidelnými denními meetingy, které pomáhají produkt kontinuálně zlepšovat. Lze tak každý den reagovat na případné měnící se požadavky a zařazovat je do časového harmonogramu. Dalším důležitým přínosem metodiky Scrum je sestavení tzv. product backlogu, tedy souhrnu veškerých požadavků na daný produkt, spolu s jejich prioritami. V případě nového požadavku lze stanovit prioritu a následně ho zařadit. S tím souvisí přínos metodiky **Feature Driven Development**, která stanovuje seznam skupin funkcionalit, potažmo jednotlivých funkcionalit. Je pak jednoduché do takového seznamu průběžně zařazovat další, s postupem času vyvstalé funkcionality. Metodika **Adaptive Software Development** přispívá v této procesní oblasti spíše svou filozofií – metodika ASD chápe změnu jako příležitost, nazývá se také metodikou pro odvážné. Což v důsledku znamená, že se vývojáři změn nebojí a zařazovat nové požadavky jim nečiní potíže. Souvisí s tím i idea metodiky **Dynamic Systems Development Methodology**, která říká, že veškeré provedené změny jsou návratné. V praxi to například může znamenat to, že pokud zákazník není spokojen se změnou kterou požadoval, lze produkt vrátit do původního stavu. Opět tak odpadá strach přijímat zákaznické požadavky.

Výše uvedené „namapování“ metodik a praktik by mělo pomoci dosáhnout 72% úrovně zralosti 2 a 60% úrovně zralosti 3.

Případová studie

Pilotním projektem implementace CMMI byla společnost Unisoft. Jedná se o malý softwarový podnik o devíti zaměstnancích, který na začátku projektu neměl žádnou z procesních oblastí CMMI na uspokojivé úrovni. Na základě poznatků získaných ve studii zde byla implementována procesní oblast Řízení požadavků na úrovni zralosti 2. K implementaci byly použity praktiky metodik Extrémní programování a Scrum. Ve výchozím stavu byla identifikována úroveň naplnění cílového stavu 32%.

Použití konkrétních praktik bylo následující:

- Podle metodiky Extrémní programování se zavedlo definování požadavků pomocí user stories.
- Ze Scrumu byl použit životní cyklus a seznam požadavků, oboje však ve zjednodušené formě.
- Na základě Extrémního programování byl zákazník více „vtažen“ do průběhu projektu
- Místo vedoucího projektu byla stanovena role Product Owner (kterého popisuje Scrum)
- Product Owner je zodpovědný za přijímání/odmítání změn

Nasazením výše uvedených praktik bylo dosaženo 84% úrovně 2 dané procesní oblasti. Což je lepší výsledek než autoři studie předpokládali. Další zaznamenaná odchylka byla, že se jednotlivé praktiky nenasazovaly separátně, v praxi se ukázalo, že se vzájemně doplňují a tak byly nasazeny najednou.

V průměrném podniku se tedy ukázalo, že pomocí agilních praktik lze dosáhnout lepších výsledků než se očekávalo. Na druhou stranu každý proces je unikátní, protože závisí na každém podniku – na jeho cílech, struktuře, atd. Ale je zřejmé, že i malý podnik může dosáhnout certifikaci CMMI.

6 Závěr

6.1 Shrnutí dosažených výsledků

Jedni tvrdí, že agilita je chaos. Druzí tvrdí, že CMMI je byrokracie. Jeden z přístupů je byrokratický, druhý anti-byrokratický. Nyní je ale zřejmé, že dva na první pohled rozdílné přístupy, CMMI a agilní metodiky, mohou fungovat pospolu. A co víc, jejich kombinace se zdá být velmi přínosnou. Principy CMMI a agility se totiž hned na druhý pohled nerozcházejí, ale mohou se velmi dobře doplňovat. Jako hlavní důvod toho, že se dva odlišné přístupy mohou doplňovat je fakt, že CMMI a agilita mají odlišné pole působnosti. CMMI je orientovaný více komplexně, agilita převážně projektově.

Agilní metodiky se soustředí spíše na daný okamžik a projekt, jejich slabou stránkou je to, že nezahrnují komplexnější pohled na podnik, jeho chod a koncepci. A zde je právě prostor pro CMMI. CMMI kromě jiného zahrnuje praktiky pro normalizaci procesů nejen na projektové, ale i organizační úrovni

CMMI vnese do podniku určitou standardizaci postupů, měřitelnost procesů, atd. Zákazník tak má možnost usoudit, jak kvalitní produkt společnost dokáže vyvinout, může zjistit, na jaké úrovni má společnost procesní oblasti, které jsou pro něj klíčové. To jsou pro něj důležitá vodítka.

Na druhé straně agilní metodika umožní rychlý vývoj, časté dodávky software a schopnost reagovat na změny požadavků. Zákazník tak má „pocit, že se něco děje“, zkrátka v co nejkratší době je představena první verze produktu a je možné okamžitě zapracovávat další požadavky, případně operativně upravovat vznikající produkt. Dochází tak k výraznému ošetření rizika, že zákazník produkt uvidí až ve finální verzi a v tu chvíli zjistí, že očekával něco jiného.

Samotné agilní praktiky je dokonce možné využít i k dosahování vyšších úrovní CMMI. Jak je zřejmé z kapitoly 4, agilní metodiky obsahují praktiky, které výrazně pomáhají zkvalitňovat vývoj, komunikaci na projektu a v podstatě celou práci projektového týmu. Právě díky tomu je možné pomocí agilních metodik dosahovat vyšších úrovní CMMI. Metoda, jakou se tak může dík je například přiřazení ke konkrétním cílům procesních oblastí jednotlivé, odpovídající, praktiky agilních metodik.

Z uvedené případové studie vyplývá, že není problém ani pro malé podniky získat certifikaci CMMI. Znovu to tedy potvrzuje fakt, že CMMI nemusí být nutně příliš robustní a rigorózní a není důvod, aby se malé podniky z tohoto důvodu CMMI obávaly. Implementace CMMI pomocí agilních praktik se tak jeví jako velmi prospěšná zejména pro malé podniky.

6.2 Naplnění cíle práce

Cílem práce bylo poskytnout pohled na uvedenou problematiku, tedy CMMI v kombinaci s agilními metodikami. Byla provedena rešerše literatury, na jejíž základě byly nalezeny a představeny možnosti kombinace přístupů. Cíl práce byl tedy naplněn.

6.3 Náměty na další činnost

Bylo by například přínosné uvesti případovou studii podniknout i v prostředí České republiky. Případně by jistě bylo zajímavé zjistit, kolik procent společností, které mají certifikaci CMMI, mají zároveň implementované agilní metodiky... Prostoru pro další práci je v této oblasti jistě mnoho.

7 Bibliografie

Agilemethodology.org. 2008. Agile methodology, development methodologies. *What is agile methodology?* [Online] 2008. [Citace: 30. 11 2013.] <http://agilemethodology.org/>.

Amsjo, Geij. 2008. ScrumAlliance. [Online] 2008. [Citace: 03. 12 2013.] <http://www.scrumalliance.org/community/articles/2008/july/agile-and-cmmi-better-together>.

Buchalcevová, Alena. 2005. *Metodiky vývoje a údržby informačních systémů*. Praha : Grada, 2005. ISBN 80-247-1075-7.

CMMI Institue. 2013. CMMI Levels. *CMMI Institute*. [Online] 2013. [Citace: 5. listopad 2013.] <http://cmmiinstitute.com/cmmi-solutions/cmmi-appraisals/cmmi-levels/>.

CMMI Institute. 2013. Solutions. *CMMI Institute*. [Online] 2013. [Citace: 13. říjen 2013.] <http://cmmiinstitute.com/cmmi-solutions/>.

ExecutiveBrief. 2009. CMMI and Agile: Opposites Attract. *ExecutiveBrief - Technology Management Resource for Business Leaders*. [Online] únor 2009. [Citace: 4. listopad 2013.] <http://www.executivebrief.com/cmmi/agile-cmmi/P2/>.

Glazer, Hillel. 2010. Love and Marriage: CMMI and Agile Need Each Other. [Online] leden 2010. [Citace: 20. listopad 2013.] <http://www.crosstalkonline.org/storage/issue-archives/2010/201001/201001-Glazer.pdf>.

Glazer, Hillel, a další. 2008. CMMI® or Agile: Why Not Embrace Both! *Software Engineering Institute*. [Online] listopad 2008. [Citace: 20. listopad 2013.] http://resources.sei.cmu.edu/asset_files/TechnicalNote/2008_004_001_14924.pdf.

Hurtado, A., Cecilia M. 2006. Implementing CMMI Using a Combination of Agile Methods. [Online] 2006.

Kučera, Martin Bc. 2008. *Zvyšování zralosti testovacích procesů v IT společnosti*. Praha : Diplomová práce. Vysoká škola ekonomická, Fakulta informatiky a statistiky, Katedra informačních technologií, 2008.

Novák, Adam. 2013. CMMI: Vylepšete váš inovační proces I. *Mámnápad.cz*. [Online] 25. září 2013. [Citace: 13. říjen 2013.] <http://www.mamnapad.cz/2013/09/25/cmmi-vylepsete-vas-inovacni-proces-i/>.

PDQM.cz. 2013. Otázky k zavedení a posouzení shody s modelem CMMI. *PDQM*. [Online] 2013. [Citace: 25. 11 2013.] <http://www.pdqm.cz/Blog/CMMI-FAQ.html>.

—. 2009. PDQM.cz. Modely CMMI a ITIL a způsoby jejich implementace. [Online] 2009.

Popelka, Vladimír. 2009. *Srovnávací analýza metodik vývoje*. Praha : Bakalářská práce. Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky, Vyšší odborná škola informačních služeb, 2009.

Quality Institute of America. 2009. CMMI Implementation Steps. *Quality Institute of America*. [Online] 2009. [Citace: 02. 12 2013.] <http://www.qi-a.com/cmmiImpl.htm>.

Shelton, Cindy. 2008. Agile and CMMI: Better Together. *SCRUM ALLIANCE*. [Online] 9. červenec 2008. [Citace: 23. říjen 2013.] <http://www.scrumalliance.org/community/articles/2008/july/agile-and-cmmi-better-together>.

Software Engineering Institute. 2010. CMMI® for Development, Version 1.3. *Software Engineering Institute*. [Online] listopad 2010. [Citace: 5. listopad 2013.] <http://www.sei.cmu.edu/reports/10tr033.pdf>.

Wikipedia. 2013. Capability Maturity Model Integration. *Wikipedia - The free encyclopedia*. [Online] 11. říjen 2013. [Citace: 13. říjen 2013.] http://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration#Appraisal.