

# CMMI-DEV v. 1.3 PA Requirements Management

---

Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS	
Semestr	ZS 2012/2013
Příjmení a jméno	Vachalec Vladan
xname	xvacv17
Obor	Informační systémy a technologie
Studijní plán (C,D,E)	E

## Obsah

1. Úvod .....	3
2. Requirements management .....	4
3. Související procesní oblasti .....	6
4. Osvědčené praktiky při řízení požadavků .....	8
4.1 Pochopení požadavků .....	8
4.2 Získání dodatečných informací k požadavkům .....	9
4.3 Řízení změn požadavků .....	10
4.4 Udržování sledovatelnosti požadavků .....	10
4.5 Zajištění sladění mezi prací na projektu a požadavky .....	11
5. Závěr .....	12
6. Použité zdroje .....	12

## 1. Úvod

Vzhledem k tomu, že se práce má zabývat jednou z částí CMMI-DEV (Capability Maturity Model Integration for Development), představíme si CMMI-DEV dvěma větami a poté se již budeme detailně věnovat kapitole Requirements Management neboli Řízení požadavků.

Co je to CMMI-DEV? Jedná se o sadu best practices, které pomáhají organizacím vylepšovat a řídit procesy. Na těchto best practices spolupracují produktové týmy se zastoupením v průmyslu, veřejných institucích a Software Engineering Institutu (SEI) [CMMI-DEV,2010]. Celý koncept se skládá ze sady procesních oblastí, mezi které spadá i requirements management, které popisují jednotlivé výstupy a konkrétní kroky k dosažení cíle zavedení procesní oblasti.

To by bylo zcela ve stručnosti shrnutí CMMI a teď se již můžeme pustit do popisu jedné z jeho částí, a to procesní oblasti Řízení požadavků. Tato skupina procesů se řadí do tzv. 2. stupně zralosti procesů v rámci úrovně zralosti CMMI. To znamená, že tyto procesy jsou plánovány a vykonávány na základě policy (předem definovaných, konkrétních a určených postupů a pravidel). Tyto procesy vykonávají zkušené osoby, které mají k dispozici odpovídající zdroje k produkování odpovídajících výstupů. Nesmíme také opomenout, že procesy na 2. stupni zralosti jsou sledovány, kontrolovány a pravidelně se ověřují jejich výstupy. Porovnává se také, jestli tyto procesy odpovídají popisům procesů a jsou skutečně dodržovány best practices. Dodržování těchto popisů procesů a pravidel pak zajišťuje, že v případě stresových situací během prací na projektu odpovědné osoby ví, jak se zachovat v daných situacích a že je projekt řízen vzhledem k jeho plánům.

Management projektu je také schopen u procesů 2. stupně zralosti ověřovat a sledovat dílčí výstupy projektu definovaných popisy procesů na konci určitých bodů projektu, např. milníky, konce důležitých úkolů, konce sprintu atd.

## 2. Requirements management

Každý projekt generuje nějaké požadavky, které je potřeba řídit. Navíc zde musí existovat také požadavky, na jejichž základě projekt vzniká. U větších projektů by se mohlo stát, že se kvůli špatnému řízení požadavků nepodaří projekt zrealizovat. Lidé zainteresovaní na projektu, jak už vlastníci a nebo pracovníci by pořádně nevěděli, jaké mají být výstupy. Vzniknul by chaos, kdy by nebylo jasné, kdo je za co odpovědný, jaké požadavky jsou již zapracovány a jaké ještě čekají na zpracování. Právě proto se CMMI věnuje řízení požadavků jako nedílné součásti projektového řízení.

Vstupy, které do procesů Requirements managementu vstupují můžeme najít v další procesní oblasti CMMI-DEV, a to Requirements development. Vývoj požadavků je tak základním stavebním kamenem, díky kterému pak můžeme aplikovat procesy řízení požadavků. Je to logické, bez vytvoření požadavků bychom neměli co řídit. Takže právě tehdy, pokud jsme do našeho projektu implementovali procesy z Requirements development, které nám produkují jako své výstupy požadavky na produkt a jednotlivé komponenty produktu, pak jsme schopni využít i procesy Requirements managementu.

Na projektu se podílí z hlediska požadavků dvě důležité role, a to [CMMI-DEV,2010]: poskytovatel požadavků (requirements provider) a osoba, která požadavky přijímá (requirements reciever). Pokud projekt obdrží nějaký nový požadavek, nejdříve se tento požadavek musí zrevidovat ve spolupráci poskytovatele požadavku a osoby, která požadavky přijímá. Pokud je požadavek zrevidován, může se následně zapracovat do projektového plánu. Projekt také řídí změny požadavků, jak vznikají během prací na projektu vzešlých na základě nekonsistence v projektových plánech, dílčích výstupech projektu nebo již schválených požadavků.

Nedílnou součástí řízení požadavků je také dokumentování změn požadavků a zajištění tzv. dvousměrné sledovatelnosti (**bidirectional traceability**) mezi požadavky, změnami na požadavky, produkty a dílčími výstupy produktů [MUTAFELIJA,2009]. To například znamená, že musíme být schopni sledovat vazby mezi požadavkem a jeho výstupem. Můžeme mít požadavek „Vytvoření dokumentu úvodní studie“ a výstup dokument úvodní studie. Z hlediska našeho projektu musíme být schopni říci, na základě jakého požadavku tento dokument vznikl. A naopak musíme být schopni říci z pohledu požadavku, který výstup vznikl na základě tohoto

požadavku. Toto pravidlo by mělo platit pro všechny entity projektu. Každá entita má vazbu na nějakou jinou a tím pádem zde existuje vztah i z druhé strany.

V případě, že udržujeme činnosti projektu, změny jsou založeny na změnách existujících požadavků, designu nebo implementaci. Pokud náš projekt dodává nějaký přírůstek k stávajícímu produktu, změny mohou reflektovat vyvíjející se zákaznické potřeby nebo nové technologie. V obou případech musí být změny požadavků zdokumentovány, ať už se jedná o potřebu zákazníků nebo nové požadavky vzniklé z procesů Requirements developmentu.

Vše výše popsané platí pro projekty řízené rigorózními metodikami. Pokud bychom si chtěli Requirements management vztáhnout k agilním přístupům vývoje, musíme si výklad trochu pozměnit.

V závislosti na použité metodice agilního vývoje jsou požadavky sledovány a komunikovány v různých formách, např. u Scrumu se požadavky sledují v produktovém backlogu. Na změnách požadavků a jejich zařazování do plánu prací se obvykle podílejí všichni členové týmu. Přiřazení pracovních úkolů se děje na bázi velmi krátkých časových intervalů (dny, týdny), takže se případná změna požadavku dá zpracovat velmi rychle. Konsistence a sledovatelnost požadavků je pak zajištěna různými demy nebo schůzkami, kdy členové týmu představují výsledky své několikadenní práce [KASSE,2008].

Každý požadavek by měl být zdokumentován a obsahovat 3 části [KASSE,2008]:

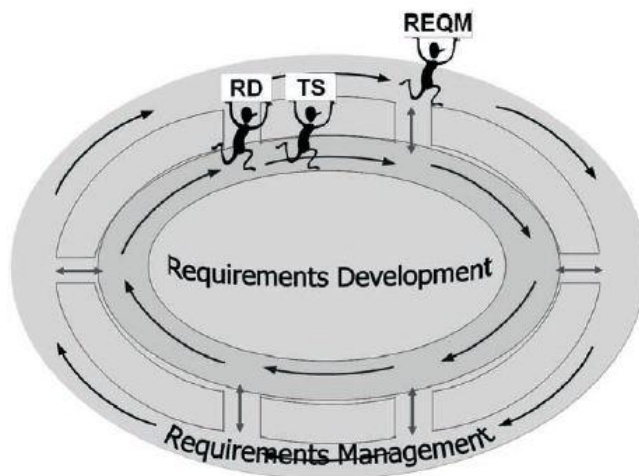
1. Přehled o požadavku
2. Ověřovací kritéria (jakým způsobem bude požadavek ověřen, s jakými riziky, jakými mechanismy)
3. Důvody (proč chceme požadavek implementovat)

Na úvod jsme si tedy vysvětlili základní důvody pro řízení požadavků, vazby mezi vstupy a výstupy a shrnuli jsme, jak je možné best practices procesy aplikovat na úrovni rigorózního způsobu řízení projektu, stejně tak jako na úrovni agilních metodik. Víme již také, do jaké úrovně zralosti patří procesy spadající do kapitoly Requirements management. Nyní se blíže podíváme, jaké procesní oblasti souvisí s námi popisovaným řízením požadavků.

### 3. Související procesní oblasti

Procesní oblasti requirements managementu nestojí samostatně bez širšího kontextu v rámci CMMI-DEV. Jsou zde i jiné oblasti, na které řízení požadavků navazuje a žije s nimi v symbióze. Zatímco se požadavky vyvíjí, analyzují, dokumentují a validují, je pro organizaci důležité tyto požadavky řídit. Řízení požadavků musí pokračovat během zbývajících částí projektu a dokonce i během fází životního cyklu produktu pokud je produkt právě implementován, optimální řešení je vybráno nebo změna na požadavek je obdržena [KASSE,2008].

Představme si nyní vztahy mezi Requirements managementem (REQM) a Requirements developementem (RD) a Technical solutions (TS) procesními oblastmi. Na obrázku 1 si můžeme všimnout, že TS a RD vyvolávají, analyzují, zpřesňují a vytvářejí více požadavků, REQM obíhá paralelně okolo, aby zajistil, že snahy RD a TS jsou řízeny již od vývoje prvního požadavku. Zatímco RD zaznamenává původní zákaznické požadavky, REQM pravidelně zachycuje tyto požadavky, identifikuje je a stará se o to, že jsou zaznamenávány všechny změny a tím i verze požadavků. Pokud vzejde z RD zákaznický požadavek, REQM se poté o požadavek postará, zvaliduje ho, opraví a nabídne vylepšenou verzi.



**Obr.1 Vztah mezi Requirements managementem a Requirements developementem [KASSE,2008]**

Jako příklad si můžeme uvést rozhodnutí o tom, že se náš produkt bude integrovat s nějakým open-source řešením a je tedy potřeba vytvořit nové rozhraní pro integraci. Toto rozhraní se stane požadavkem pro všechny komponenty produktu, které budou chtít open-source řešení

implementovat. REQM zajistí, aby byl nový požadavek zaznamenán a zvalidován a hlavně, aby se zajistila konsistence požadavku s jinými již existujícími požadavky.

REQM byl původně konfiguračním managementem požadavků na změnu požadavků. Tak to bylo dříve popisováno v CMM pro software. Nicméně REQM se stal samostatnou procesní oblastí, která indikuje důležitost ne jenom shromažďování a analyzování požadavků, ale také zajišťuje aktivní spolupráci a povědomí všech účastníků uvnitř i vně projektu o současných stavech požadavků. Musíme mít na paměti, že požadavky na změnu požadavku mohou přijít od členů projektového týmu, zákazníků nebo koncových uživatelů.

## 4. Osvědčené praktiky při řízení požadavků

Cíl procesní oblasti řízení požadavků je: **Požadavky jsou řízeny a nekonsistence s projektovými plány a pracovními produkty jsou identifikovány.**

V projektu se udržují současné a schválené seznamy požadavků během života projektu, pokud se dodržují následující doporučení [CMMI-DEV,2010]:

- Řídí se všechny změny požadavků
- Udržují se vztahy mezi požadavky, změnami požadavků, projektovými plány a produkty
- Zajišťuje se sladění požadavků s projektovými plány a produkty
- V případě nějaké nesrovnalosti se aplikují opravné kroky k zajištění konsistence požadavků s ostatními projektovými entitami

Tohoto hlavního cíle, tedy správného a konsistentního řízení požadavků dosáhneme, pokud se budeme držet dílčích cílů, které CMMI doporučuje. Tyto konkrétní best practices si popíšeme v následujících částech této kapitoli.

### 4.1 Pochopení požadavků

Cíl: **Pochopit význam jednotlivých požadavků poskytovaných různými osobami a jasný a jednoznačný výklad těchto požadavků.**

Čím dále se nacházíme v průběhu projektu a čím víc se na nás valí požadavků a změn, přidáváme k jednotlivým aktivitám nebo úkolům další požadavky. Abychom se vyhnuli nějakým nesrovnalostem v požadavcích, musí být určena pravidla, která zajistí, že se požadavky budou brát od oficiálních a relevantních poskytovatelů požadavků a budou se zaznamenávat v přesně definované struktuře a na konkrétní místo. Konkrétní osoby, které pak dostanou v rámci své práce k implementaci nějaký požadavek, musí v případě jakýchkoli nesrovnalostí konzultovat náplň, význam a smysl požadavku s osobou, která tento požadavek vytvořila. Výsledkem těchto analýz a dialogů je **množina schválených požadavků**.

Výstupy plynoucí z dodržování těchto pravidel mohou být kromě už již zmiňovaného seznamu schválených požadavků např. kritéria pro posuzování a akceptaci požadavků nebo výsledky analýz požadavků konfrontovaných s těmito kritérii.



Nyní si představíme konkrétní postupy týkající se pochopení požadavků [CMMI-DEV,2010]:

1. Určení kritérií pro rozlišení vhodných poskytovatelů požadavků.
2. Určení objektivních kritérií pro posouzení a akceptaci požadavků. Pokud bychom tyto kritéria neurčili, mohlo by se stát, že budou požadavky špatně vytvořeny, nekonsistentní a budeme muset vynakládat během projektu dodatečné náklady na změny požadavků. Typickými příklady takových kritérií může být úplnost požadavku, sledovatelnost, jasné a vhodně vymezený požadavek, požadavek přináší business value atd.
3. Analýza požadavků a tím zajištění, že požadavky splňují stanovená kritéria.
4. Dosažení porozumění požadavků společně s poskytovatelem požadavků, tak aby byli ostatní účastníci projektu schopni požadavky implementovat.

#### 4.2 Získání dodatečných informací k požadavkům

Kapitola 4.1 měla za cíl popsat jakým způsobem a jak správně porozumět požadavkům. V této části si popíšeme osvědčené postupy, které se týkají dohod a dodatečných informací k požadavkům od jednotlivých účastníků projektu, kteří se v rámci svých činností podílejí na implementaci požadavků.

**Cíl: Získat dodatečné informace k požadavkům od účastníků na projektu.**

Požadavky se vyvíjí v průběhu celého projektu. Tyto postupy zaručují, že se účastníci projektu podílejí svými připomínkami k současným a schváleným požadavkům. Tím pak mohou ovlivňovat výstupy projektu, projektové plány a činnosti. Příkladem výstupů z těchto postupů může být posouzení vlivu požadavku na ostatní entity v projektu nebo zdokumentované dodatečné informace k požadavkům nebo změnám k požadavkům. Konkrétní postupy spadající do této části řízení požadavků jsou dle [CMMI-DEV,2010] tyto:

1. Shromáždění vlivů dodatečné informace na existující požadavky. Vliv na projektové účastníky by měl být stanoven během změny požadavku nebo na začátku nového požadavku.
2. Dohodnout se na připomínkách k požadavkům a zaznamenat je. Před tím, než některý z účastníků projektu začne pracovat na požadavku, měly by být jasné domluvené a zapracované připomínky.

### 4.3 Řízení změn požadavků

**Cíl: Řídit změny požadavků jak se vyvíjí v průběhu projektu.**

V každém projektu se musí řídit změny požadavků. Postupem času při práci na projektu se mohou měnit priority činností, a tím pádem je potřeba měnit i existující požadavky. Je nezbytné, aby byly tyto změny řádně řízeny. Aby bylo možné zkoumat vliv a dopad změny požadavku na projekt, je potřeba znát zdroj od kterého změna přišla a vhodně zaznamenat racionální důvody této změny. Můžeme také požadovat, aby bylo možné sledovat vhodnost opatření proti volatilitě požadavků a rozhodnout následně, zda není nezbytné upravit přístup ke kontrole změn. Výstupy těchto postupů jsou požadavky na změny požadavku, reporty popisující dopady změny požadavku nebo třeba status požadavku. Popsané konkrétní kroky vypadají takto [CMMI-DEV,2010]:

1. Dokumentovat všechny požadavky a změny na požadavky, které patří k projektu. Jedná se jak o přiřazené požadavky projektu, tak i o požadavky vznikající v průběhu projektu.
2. Udržovat historii změn požadavků včetně důvodů, kvůli kterým bylo potřeba požadavek měnit. Udržování historie změn pomáhá sledovat volatilitu požadavků.
3. Posoudit vliv změny na požadavky z pohledu zainteresovaných subjektů. Velká změna například v architektuře se může dotknout více stakeholderů.
4. Zpřístupnit požadavky a změny na ně projektu.

### 4.4 Udržování sledovatelnosti požadavků

**Cíl: Udržovat obousměrnou sledovanost mezi požadavky a výstupy.**

O obousměrné sledovatelnosti mezi entitami jsme se již zmínili v úvodu práce a proto ji nebudeme znovu podrobně vysvětlovat. Pokud jsou požadavky dobře řízené, sledovatelnost může být založena od zdrojového požadavku do dílčích požadavků a od dílčích požadavků zpět ke zdrojovému požadavku. Taková sledovatelnost pomáhá určit, zda byly všechny zdrojové požadavky úplně zapracovány a jestli všechny dílčí požadavky mají vazbu na potřebný zdroj. Sledovatelnost požadavků také pokrývá i vztahy jiných entit, jako třeba dílčích a finálních výstupů projektu. Je potřebná i proto, že nám umožňuje přiřadit vlivy změn požadavků k činnostem a výstupům. Při vytváření sledovatelnosti musíme definovat rámec ve kterém budeme

chtít entity sledovat, definovat entity, které je potřeba sledovat a rozhodnout se, kde je potřeba horizontální nebo vertikální způsob sledovanosti. Výstupy těchto postupů mohou být matice sledovatelnosti požadavků nebo systém na sledovatelnost požadavků. To si můžeme představit jako manuálně upravované tabulky, databázi nebo jiné nástroje. Co je tedy potřeba pro sledování vazeb mezi požadavky a jinými entitami dle [CMMI-DEV,2010]?

1. Udržovat sledovatelnost vazeb mezi požadavky, aby bylo možné zdroje dílčích požadavků dokumentovat.
2. Udržovat sledovatelnost vazeb mezi požadavky k odvozeným požadavkům a sledovat jejich alokaci k pracovním produktům. Těmi produkty může být architektura, funkce, rozhraní, lidé, procesy atd.
3. Generovat matici vazeb mezi požadavky

#### **4.5 Zajištění sladění mezi prací na projektu a požadavky**

**Cíl: Zajistit, že projektové plány a výstupy projektu odpovídají požadavkům.**

Tyto best practices mají zajistit, že nevzniká nekonsistence mezi požadavky, projektovými plány a výstupy a v případě, že k nějaké nekonsistenci dojde, napraví je pomocí nápravných akcí. Výstupy těchto kroků jsou pak seznamy nekonsistencí mezi požadavky a projektovými plány a seznam nápravných akcí/kroků. Postup při dodržování těchto best practices je následovný [CMMI-DEV,2010]:

1. Přezkoumávání projektových plánů, aktivit a výstupů pro zajištění konsistence s požadavky včetně změn na tyto požadavky
2. Pokud existují, identifikovat zdroje nekonsistencí.
3. Identifikovat jakékoli změny, které by mohly být udělány do projektových plánů a výstupů na základě změn požadavků.
4. Vyvolat potřebné nápravné akce.

## 5. Závěr

Závěrem bych chtěl shrnout, jak práce popsala řízení požadavků podle CMMI-DEV. Během procesu analýzy, definice a hlavně řízení požadavků by se měly dodržovat následující postupy, které nemusí nutně zapadat jen do oblasti řízení požadavků, ale jak bylo popsáno v kapitole 3, také do procesních oblastí technického řešení a vývoje požadavků.

Je důležité, aby se identifikovaly klíčové požadavky, které mají vliv na náklady celého projektu, projektový harmonogram, rizika projektu nebo funkcionality výsledného produktu. Kromě toho se musí také identifikovat jasné metriky pro výkonost dodávaného produktu, které budou ověřovány v průběhu projektu. Neméně důležité je fáze analýzy požadavků tak, aby požadavky odpovídaly představám všech účastníků projektu a zároveň, aby bylo všem zodpovědným osobám v projektu jasné, co daný požadavek v jejich kompetenci opravdu znamená. Nesmíme zapomenout ani na řízení změn požadavků a nových požadavků v tom smyslu, že budeme přesně vědět, jaké souvislosti a jaký vliv by měla změna na celé prostředí projektu [KASSE,2008].

Procesy řízení požadavků se starají o to, abychom v projektu kontinuálně měli veškeré informace o požadavcích jak už technických, tak netechnických. Pokud v projektu implementujeme procesy z oblastí řízení požadavků, vývoj požadavků a technické řešení, pak jsme schopni efektivně řídit tyto procesy, které jsou spolu velmi úzce spjaté (viz kapitola 3).

## 6. Použité zdroje

[CMMI-DEV,2010] CMMI PRODUCT TEAM. *CMMI® for Development, Version 1.3: Improving processes for developing better products and services*. 2010. Dostupné z: <http://repository.cmu.edu/cgi/viewcontent.cgi?article=1278&context=sei>. TECHNICAL REPORT. Carnegie Mellon University.

[KASSE,2008] KASSE, Tim. *Practical insight into CMMI*. 2nd ed. Boston: Artech House, c2008, xlvii, 472 p. ISBN 978-159-6932-753.

[MUTAFELIJA,2009] MUTAFELIJA a Harvey STROMBERG. *Process improvement with CMMI v1.2 and ISO standards*. Boca Raton: CRC Press, c2009. ISBN 14-200-5283-7.