

# CMMI-DEV v.1.3 PA Configuration management

---

4IT421 - Zlepšování procesů budování IS

Pavel Neuman

ZS 2012/2013

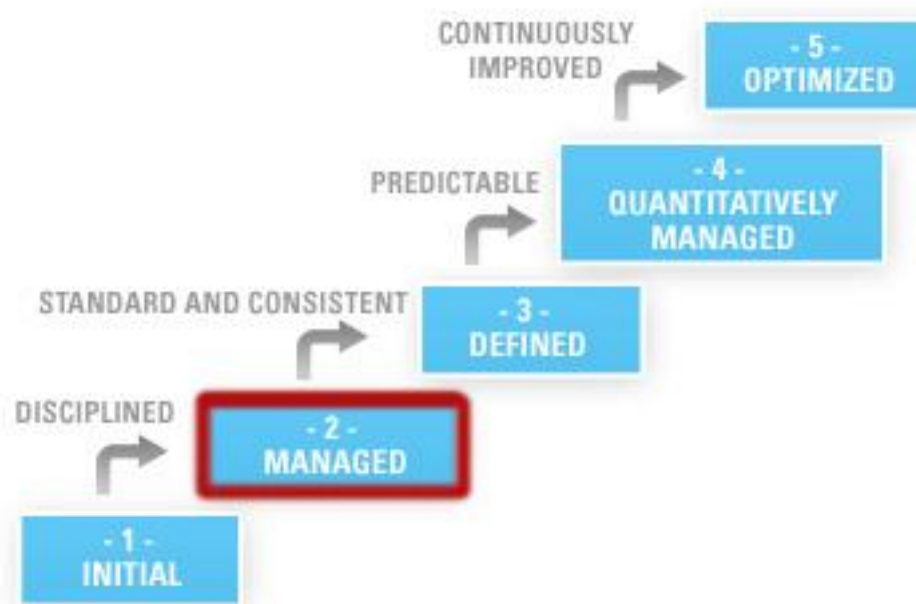
## Obsah

1. Úvod .....	2
2. Configuration Management.....	2
2.1. Úvodní poznámky .....	3
3. Specifické praktiky podle cíle .....	5
3.1. Zřízení baselines .....	5
3.1.1. Identifikace konfiguračních položek.....	5
3.1.2. Zřízení systému řízení konfigurace .....	7
3.1.3. Vytvoření či uvolnění baselines.....	8
3.2. Sledování a kontrola změn .....	9
3.2.1. Sledování požadavků na změny .....	9
3.2.2. Kontrola konfiguračních položek.....	10
3.3. Zřízení integrity.....	11
3.3.1. Zřízení záznamů o řízení konfigurace .....	11
3.3.2. Provádění konfiguračních auditů .....	12
4. Srovnání s ITIL .....	13
5. Závěr.....	14
6. Použité zdroje .....	15

## 1. Úvod

Cílem tohoto dokumentu je seznámit čtenáře s procesní oblastí Configuration management tak, jak je popsána v souboru nejlepších postupů (praktik) *Capability Maturity Model Integration for Development* (CMMI-DEV) verze 1.3.

Configuration Management neboli řízení konfigurace je podpůrná procesní oblast v druhém stupni zralosti dle CMMI.



Obrázek 1 - Stupně zralosti dle CMMI

## 2. Configuration Management

Účelem Configuration Managementu (CM) je stanovit a udržovat integritu pracovních produktů prostřednictvím identifikování konfigurace, kontroly konfigurace, sledování stavu konfigurace a konfiguračních auditů. To vede k zajištění integrity celého systému, což je žádoucí stav hned z několika důvodů:

- Změny prováděné na jakékoliv konfigurační položce v systému se řídí ustanovenými a dodržovanými postupy.
- Systém je zabezpečen proti nesprávně postupujícím vývojářům, kteří se snaží obcházet pravidla.
- Systém je zabezpečen proti útokům, u kterých hrozí poškození obsahu konfiguračních položek.

- Pracovní produkty životního cyklu jsou konzistentní v případě změny specifikací požadavků. Všechny související pracovní produkty životního cyklu jsou ověřovány, aby se rozhodlo, zda jsou u nich příslušné změny nutné.
- Na obsahu systému lze provádět periodické audity, aby bylo zajištěno, že změny v produktových komponentech jsou kompletní a správné.
- Prováděním regresního testování lze zajistit, že jsou chyby opraveny a existující funkcionality je zachována

Zavedením řízení softwarových komponent můžeme předejít nejčastějším chybám, jako např. výskytu již opravených chyb, ztrátě poslední verze zdrojového kódu, nespojitosti mezi softwarovými požadavky, dokumentací a kódem, či tomu, že nikdo vlastně neví, které moduly jsou zahrnuty ve verzi systému dodané zákazníkovi.

## 2.1. Úvodní poznámky

Procesní oblast Configuration Management zahrnuje následující činnosti:

- Identifikování konfigurace vybraných pracovních produktů, které tvoří základní linii (baseline) v daných bodech v čase
- Řízení změn konfiguračních položek
- Budování nebo poskytování specifikací pro výstavbu pracovních produktů ze systému pro řízení konfigurace
- Zachování integrity baselines
- Poskytování přesného stavu a aktuálních konfiguračních dat vývojářům, koncovým uživatelům a zákazníkům

Mezi pracovní produkty, které spadají do řízení konfigurace, patří i produkty dodávané zákazníkovi, určené interní pracovní produkty, získané produkty, nástroje a další položky používané k tvorbě a popsání těchto pracovních produktů.

Příklady pracovních produktů, které mohou být umístěny pod řízení konfigurace:

- Hardware a vybavení
- Kresby
- Specifikace produktu
- Konfigurace nástrojů
- Kód a knihovny
- Kompilátory
- Testovací nástroje a skripty
- Instalační protokoly
- Datové soubory produktu
- Technické publikace k produktu
- Plány
- Uživatelské scénáře
- Iterační rozpracovanost
- Procesní popisy

- Požadavky
- Dokumentace architektury a návrhová data
- Plány produktové řady, procesy, a hlavní aktiva

Řízení konfigurace pracovních produktů lze provádět na několika úrovních granularity. Konfigurační položky mohou být rozloženy na konfigurační komponenty a konfigurační jednotky. V této procesní oblasti se používá pouze termín „konfigurační položka“, v praktikách ale lze jeho výklad jako „konfigurační komponenta“ či „konfigurační jednotka“ považovat za správný.

Baselines poskytují stabilní základ pro kontinuální vývoj konfiguračních položek. Jako příklad můžeme uvést odsouhlasený popis produktu, který obsahuje vnitřně konzistentní verze požadavků, matice sledovatelnosti požadavků, návrh, oborově specifické položky a uživatelskou dokumentaci. Baselines jsou do systému přidávány tak, jak jsou vyvíjeny. Změny v nich a uvolnění pracovních produktů vystavených ze systému pro řízení konfigurace jsou systematicky sledovány prostřednictvím kontroly konfigurace (Configuration Control), řízení změn (Change Management) a funkcí konfiguračního auditingu v řízení konfigurace.

Tato procesní oblast se vztahuje nejen na řízení konfigurace na projektech, ale také na řízení konfigurace organizačních pracovních produktů, jako jsou normy, postupy, opětovné použití knihoven a dalších sdílených podpůrných aktiv. Pokrývá postupy pro provádění funkcí řízení konfigurace a vztahuje se na všechny pracovní produkty, které jsou umístěny pod řízení konfigurace.

Řízení konfigurace je zaměřeno na přísnou kontrolu manažerských a technických aspektů pracovních produktů, včetně dodávaného produktu nebo služby. Zahrnuje též další okolnosti pro produktové řady, kvůli sdílení hlavních aktiv mezi produkty v řadě a mezi různými verzemi hlavních aktiv a produktů.

V agilním prostředí je řízení konfigurace důležité, protože je třeba podporovat časté změny, časté vývojové verze (obvykle denně), četné základní linie a více pracovních prostorů (např. pro jednotlivce, týmy, nebo i pro párové programování). Agilní týmy mohou zabřednout, pokud organizace řízení konfigurace neautomatizuje (např. build skripty, sledování stavu, kontrola integrity) a pokud neimplementuje řízení konfigurace jako jediný soubor standardizovaných služeb. Na jeho začátku, by agilní tým měl identifikovat osobu, která bude odpovídat za zajištění, že CM bude správně implementován. Na začátku každé iterace jsou potřeby CM podpory znovu potvrzeny. CM je pečlivě integrováno do rytmu každého týmu se zaměřením na minimalizaci rozptýlení týmu, aby mohli vykonat svou práci.

## 3. Specifické praktiky podle cíle

### 3.1. Zřízení baselines

V této kapitole jsou zahrnuty specifické praktiky vedoucí k vytvoření baselines. Další kapitoly pak obsahují praktiky pro jejich dodržování (Sledování a dohled nad změnami) a pro dokumentaci a audit jejich integrity (Zřízení integrity).

#### 3.1.1. Identifikace konfiguračních položek

Identifikování konfigurace je výběrem a specifikací následujících položek:

- Produkty doručené zákazníkovi
- Určené interní pracovní produkty
- Získané produkty
- Nástroje a další hlavní aktiva pracovního prostředí projektu
- Další položky užité k vytváření a popisu těchto pracovních produktů

Konfigurační položky mohou obsahovat hardware, vybavení a další hmotná aktiva stejně jako software a dokumentaci. Dokumentace může obsahovat specifikace požadavků, dokumentaci interface a další dokumenty, které slouží k identifikaci konfigurace produktu nebo služby, jako jsou např. výsledky testů.

Konfigurační položka (Configuration Item) je entita navržená pro řízení konfigurace, může se skládat z několika souvisejících pracovních produktů, které formují baseline. Toto logické seskupení usnadňuje identifikaci a řízený přístup. Výběr pracovních produktů pro řízení konfigurace by měl být založen na kritériích stanovených během plánování.

#### Příklad pracovních produktů

1. Identifikované konfigurační položky

#### Subpraktiky

1. Výběr konfiguračních položek a pracovních produktů, které je vytvářejí, na základě zdokumentovaných kritérií.

Příklad kritérií pro výběr konfiguračních položek na odpovídající úrovni pracovních produktů může vypadat takto:

- Pracovní produkty, které mohou být využívány dvěma nebo více skupinami
- Pracovní produkty, u kterých se časem očekává změna vlivem chyb nebo změn v požadavcích
- Pracovní produkty, které jsou na sobě vzájemně závislé (např. změna v jednom nařizuje změnu v ostatních)

- Pracovní produkty kritické pro úspěch projektu

Příklad pracovních produktů, které mohou být součástí konfigurační položky, může vypadat následovně:

- Návrh
- Plány a postupy testů
- Výsledky testů
- Popis interface
- Kresby
- Zdrojový kód
- Uživatelské scénáře
- Deklarované business case, logika nebo hodnoty
- Nástroje (např. kompilátory)
- Popisy procesů
- Požadavky

## 2. Přiřazení jedinečných identifikátorů konfiguračním položkám

## 3. Specifikace důležitých charakteristik jednotlivých konfiguračních položek

Mezi charakteristiky konfiguračních položek patří např. autor, typ dokumentu nebo souboru, programovací jazyk u souborů se softwarovým kódem, minimální prodejní vlastnosti nebo účel, k němuž konfigurační položka slouží.

## 4. Určení, kdy má být každá konfigurační položka zařazena do řízení konfigurace

Příklad kritérií pro rozhodnutí, kdy umístit pracovní produkty do řízení konfigurace:

- Kdy je pracovní produkt připraven k testování
- Fáze životního cyklu projektu
- Stupeň kontroly požadovaný u pracovního projektu
- Nákladová a časová omezení
- Požadavky zúčastněných stran

## 5. Identifikace majitele zodpovědného za každou konfigurační položku

## 6. Specifikace vztahů mezi konfiguračními položkami

Zahrnutí typů vztahů (např. parent-child, závislost), které existují mezi konfiguračními položkami do struktury řízení konfigurace (např. databáze řízení konfigurace, CMDB) pomáhá při řízení efektů a dopadů změn.

### 3.1.2. Zřízení systému řízení konfigurace

Systém pro řízení konfigurace obsahuje úložná média, postupy a nástroje pro přístup k systému. Může se skládat z několika subsystémů s různou implementací, které odpovídají jednotlivým prostředím řízení konfigurace.

Systém pro řízení změn obsahuje úložná média, postupy a nástroje pro zaznamenání a přístup k požadavkům na změny.

#### Příklad pracovních produktů

1. Systém pro řízení konfigurace se sledovanými pracovními produkty
2. Postupy pro sledování přístupu k systému pro řízení konfigurace
3. Databáze požadavků na změny

#### Subpraktiky

1. Vytvoření mechanismu pro řízení vícenásobných úrovní kontroly

Úroveň kontroly je typicky založená na cílech projektu, riziku a zdrojích. Úrovně kontroly se mohou lišit v závislosti na fázi životního cyklu projektu, typu vyvíjeného systému a specifických požadavcích projektu.

Příklady úrovní kontroly:

- Nekontrolováno: Kdokoliv může provádět změny.
- Rozpracováno: Autoři kontrolují změny.
- Uvolněno: Určený správce autorizuje a kontroluje změny, relevantní zúčastněné strany jsou informovány, pokud se změny provádějí.

Úrovně kontroly sahají od neformálních, které pouze sledují změny provedené při vývoji konfiguračních položek, po formální, které využívají základní linie, které mohou být změněny pouze jako součást formálního procesu řízení konfigurace.

2. Poskytnutí kontroly přístupu k zajištění autorizovaného přístupu k systému pro řízení konfigurace
3. Uložení a získání konfiguračních položek v systému pro řízení konfigurace
4. Sdílení a přenos konfiguračních položek mezi úrovněmi kontroly v systému pro řízení konfigurace.
5. Uložení a obnova archivovaných verzí konfiguračních položek
6. Uložení, aktualizace a získání záznamů o řízení konfigurace
7. Vytvoření reportů o řízení konfigurace systémem pro řízení konfigurace



## 8. Uchování obsahu systému pro řízení konfigurace

Příklady uchovávacích funkcí systému pro řízení konfigurace:

- Záloha a obnova souborů řízení konfigurace
- Archivace souborů řízení konfigurace
- Zotavení z chyb řízení konfigurace

## 9. V případě nutnosti revize struktury řízení konfigurace

### 3.1.3. Vytvoření či uvolnění baselines

Baseline je reprezentována přiřazením identifikátoru konfigurační položce nebo souboru konfiguračních položek a přidružených entit a určitém bodu v čase. Jak se produkt nebo služba vyvíjí, může být ke kontrole vývoje a testování využito více baselines.

Hardwarové produkty by měly být, stejně jako software a dokumentace, obsaženy v baselines pro konfigurace spojené s infrastrukturou (např. software, hardware) a v přípravě na systémové testy, které zahrnují propojení mezi hardwarem a softwarem.

Běžný soubor baselines obsahuje požadavky na úroveň systému (funkční baselines), požadavky na návrh úrovní prvků systému (alokované) a definici produktu na konci vývoje/začátku produkce (produktové).

Softwarové baselines mohou být souborem požadavků, návrhu, souborů se zdrojovým kódem a přidruženého spustitelného kódu, verzovacích souborů a uživatelské dokumentace (přidružené entity), již byl přiřazen unikátní identifikátor.

#### **Příklad pracovních produktů**

1. Baselines
2. Popis baselines

#### **Subpraktiky**

1. Získání autorizace od rady pro kontrolu konfigurace (CCB) před vytvořením nebo uvolněním baselines konfiguračních položek.
2. Vytvoření nebo uvolnění baselines pouze z konfiguračních položek v systému pro řízení konfigurace
3. Dokumentace souboru konfiguračních položek obsažených v baseline
4. Zajištění dostupnosti aktuálního souboru baselines

## 3.2. Sledování a kontrola změn

Specifické praktiky u tohoto cíle slouží k udržování baselines potom, co byly vytvořeny praktikami v cíli Zřízení baselines.

### 3.2.1. Sledování požadavků na změny

Požadavky na změny se nezabývají jen novými nebo změněnými požadavky, ale také selháními a defekty v pracovních produktech.

Požadavky na změny jsou analyzovány pro určení dopadu, který bude mít změna na pracovní produkt, související pracovní produkty, rozpočet a časový plán.

#### Příklad pracovních produktů

1. Požadavky na změny

#### Subpraktiky

1. Vyvolání požadavku na změnu záznamu v databázi požadavků na změnu
2. Analýza dopadu změn a oprav navržených v požadavcích na změnu

Změny jsou vyhodnoceny aktivitami, které zajistí, že jsou konzistentní se všemi technickými a projektovými požadavky.

Změny jsou vyhodnoceny z hlediska jejich dopadu přes bezprostřední projektové nebo smluvní požadavky. Změny položky užívané v několika produktech může vést k bezodkladným otázkám a působit problémy v dalších aplikacích.

Změny jsou vyhodnoceny z hlediska jejich dopadu na plány uvolnění.

3. Kategorizace a prioritizace požadavků na změnu

Mimořádné požadavky jsou identifikovány a zpracovány pohotovostním správcem, pokud je to vhodné.

Změny jsou alokovány do budoucích baselines.

4. Přezkoumání požadavků na změnu k řešení v další baseline s relevantními zúčastněnými stranami a získání jejich souhlasu

Zřízení přezkoumání požadavků na změnu s odpovídajícími účastníky, zápis povahy každého požadavku na změnu a zdůvodnění rozhodnutí včetně kritérií pro úspěch, stručný akční plán, pokud je to vhodné, a potřeby splněné či nesplněné změnou. Provádí se činnosti požadované v povaze a výsledky jsou oznámeny relevantním zúčastněným stranám.

## 5. Sledování stavu požadavků na změnu do uzavření

Požadavky na změnu zanesené do systému by měly být zpracovány efektivně a včas. Jak je požadavek na změnu zpracován, je kritické, aby byl uzavřen náležitou odsouhlasenou činností, jakmile je to praktické. Neuzavřené činnosti vedou k nadbytečně rozsáhlým seznamům stavů, což dále vede k vyšším nákladům a zmatkům.

### 3.2.2. Kontrola konfiguračních položek

Kontrola se provádí nad konfigurací baselines pracovních produktů, zahrnuje sledování konfigurace každé konfigurační položky, schválení nové konfigurace v případě potřeby a aktualizaci baseline.

#### Příklad pracovních produktů

1. Revize historie konfiguračních položek
2. Archivy baselines

#### Subpraktiky

1. Kontrola změn konfiguračních položek napříč životním cyklem produktu nebo služby.
2. Získání vhodné autorizace předtím, než jsou změněné konfigurační položky vloženy do systému řízení konfigurace.

Autorizace může přijít např. od rady pro kontrolu konfigurace, projektového manažera, vlastníka projektu nebo od zákazníka.

3. Check-in a check out konfiguračních položek v systému pro řízení konfigurace pro zahrnutí změn způsobem, který dodržuje korektnost a integritu konfiguračních položek.

Check-in a check-out proces obsahuje např. následující kroky:

- Potvrzení, že jsou revize autorizovány
- Aktualizace konfiguračních položek
- Archivace vyměněné základní linie a získání nové
- Okomentování změn provedených na položce
- Provázání změn na související pracovní produkty jako jsou požadavky, uživatelské scénáře a testy

4. Revize za účelem zajištění, že změny neměly nežádoucí efekt na baselines (např. nedošlo k ohrožení bezpečnosti systému)

## 5. Zaznamenání změn konfiguračních položek a jejich důvodu.

Pokud je navrhovaná změna pracovního produktu akceptována, je identifikován plán zapojení změny do pracovního produktu a ostatních zasažených oblastí.

Mechanismy kontroly konfigurace mohou být uzpůsobeny kategoriím změn. Např. úvahy o schválení nemusí být tolik přísné u změn komponent, které nezasáhnou jiné komponenty.

Změněné konfigurační položky jsou uvolněny po přezkoumání a schválení změn konfigurace. Změny nejsou oficiální, dokud nejsou uvolněny.

## 3.3. Zřízení integrity

Praktiky v tomto oddílu se týkají integrity baselines, které jsou vytvořeny pomocí procesů Zřízení baselines a udržovány pomocí procesů Sledování a kontroly změn.

### 3.3.1. Zřízení záznamů o řízení konfigurace

*Vytvoření a údržba záznamů popisujících konfigurační položky.*

#### Příklad pracovních produktů

1. Historie revizí konfiguračních položek
2. Changelog
3. Záznamy požadavků na změnu
4. Stav konfiguračních položek
5. Rozdíly mezi baselines

#### Subpraktiky

1. Zaznamenání činností řízení konfigurace v dostatečném detailu, aby byl znám obsah a stav každé konfigurační položky a mohly být obnoveny předchozí verze
2. Zajištění, že relevantní zúčastněné strany znají stav konfiguračních položek a mají k němu přístup

Příklad činností komunikace stavu konfiguračních položek:

- Poskytnutí přístupových práv autorizovaným uživatelům
  - Zajištění dostupnosti kopií základních linií autorizovaným uživatelům
  - Automatické upozornění relevantních zúčastněných stran o vkládání či odebírání položek nebo o rozhodnutích týkajících se požadavků na změny
3. Specifikace poslední verze baseline
  4. Identifikace verze konfiguračních položek, které tvoří určitou baseline
  5. Popis změn mezi navazujícími baselines

6. Revize stavu a historie (např. změn, dalších činností) každé konfigurační položky, v případě potřeby

### 3.3.2. Provádění konfiguračních auditů

Konfigurační audit potvrzuje, že výsledné baselines a dokumentace odpovídají specifikovanému standardu nebo požadavku. Záznamy konfiguračních položek mohou existovat ve více databázích nebo systémech pro řízení konfigurace. V takových případech by měl konfigurační audit k zajištění přesnosti, úplnosti a konzistence informací o konfigurační položce zasahovat i do těchto databází.

Příklad typů auditu

- Funkční konfigurační audit (FCA): Audit prováděný za účelem ověření, že vývoj konfigurační položky byl úspěšně ukončen, že položka dosáhla funkcionálních a kvalitativních vlastností specifikovaných ve funkční nebo alokované baseline, a že její operativní a podpůrná dokumentace je úplná a uspokojující
- Fyzický konfigurační audit (PCA): Audit prováděný k ověření, že konfigurační položka odpovídá technické dokumentaci, která ji definuje a popisuje
- Audit řízení konfigurace: Audit prováděný k ověření, že konfigurační záznamy a položky jsou přesné, úplné a konzistentní.

#### Příklady pracovních produktů

1. Výsledky konfiguračního auditu
2. Položky činností

#### Subpraktiky

1. Posouzení integrity baselines
2. Potvrzení, že záznamy řízení konfigurace správně identifikují konfigurační položky
3. Přezkoumání struktury a integrity položek v systému pro řízení konfigurace
4. Potvrzení úplnosti, správnosti a konzistence položek v systému pro řízení konfigurace

Úplnost, správnost a konzistence obsahu systému pro řízení konfigurace se zakládá na požadavcích stanovených v plánu a na povaze odsouhlasených požadavků na změny

5. Potvrzení shody s platnými standardy a procedurami řízení konfigurace
6. Sledování položek činností od auditu do uzavření

## 4. Srovnání s ITIL

Pokud se bavíme o řízení konfigurace a souborech nejlepších praktik a standardů, nabízí se srovnání procesní oblasti Configuration Management v CMMI s ekvivalentním procesem Service Asset and Configuration Management (SACM) v ITIL v3. Velice podobně znějící názvy napovídají, že v obou souborech bude sloužit k podobnému účelu. Tím je vytvoření a zachování infrastruktury systému pomocí identifikace a kontroly klíčových prvků systému, jak již bylo zmíněno v úvodu. Hlavním rozdílem je fakt, že CMMI se zaměřuje více na aplikační vývoj, zatímco ITIL (IT Infrastructure Library) je soubor praktik pro správu IT služeb, který se soustředí na sladění IT služeb s business požadavky.

Srovnání CM a SACM na příkladech konkrétních pojmů:

	CMMI 1.3	ITIL v3
<b>Cíle</b>	<ul style="list-style-type: none"> <li>• Identifikovat konfiguraci vybraných pracovních produktů, které tvoří baselines v určitém bodě v čase</li> <li>• Kontrola změn konfiguračních položek</li> <li>• Sestavení nebo poskytnutí specifikací pro vybudování pracovních produktů ze systému pro řízení konfigurace</li> <li>• Udržování integrity baselines</li> </ul>	<ul style="list-style-type: none"> <li>• Poskytovat informace pro podporu procesů ITSM, byznys požadavků, rozhodování – autorizace změn, plánování release, řešení problémů a incidentů</li> <li>• Minimalizace problémů spojených s nesprávnou konfigurací služeb a aktiv</li> <li>• Správa komponent služeb a infrastruktury, uchovávání informací o konfiguraci</li> </ul>
<b>Configuration Item</b>	<ul style="list-style-type: none"> <li>• Specifikace požadavků, interface, architektury, návrhu</li> <li>• Plány, procedury, výsledky testů</li> <li>• Zdrojový kód</li> <li>• Nástroje (např. kompilátory)</li> <li>• Deklarované business case, logika, hodnoty</li> </ul>	<ul style="list-style-type: none"> <li>• IT služby</li> <li>• Hardware (veškeré HW prvky IT infrastruktury)</li> <li>• Software (licence)</li> <li>• Lidé</li> <li>• Budovy</li> <li>• Dokumentace</li> </ul>
<b>Baseline</b>	<ul style="list-style-type: none"> <li>• Schválený snímek systému v určitém bodě vývojového cyklu</li> <li>• Specifikace (požadavků, návrhu, ...), dokument, formálně přezkoumaný a schválený produkt, část systému</li> <li>• Veškeré změny musí být formálně schváleny</li> </ul>	<ul style="list-style-type: none"> <li>• Konfigurace služby, produktu, infrastruktury v bodě v čase</li> <li>• Zachycuje strukturu, obsah, detaily</li> <li>• Veškeré změny musí být formálně schváleny</li> </ul>
<b>Configuration Management System</b>	<ul style="list-style-type: none"> <li>• Uchovává CI, odkazy na ně a vazby mezi nimi</li> <li>• Brání neautorizovaným změnám v baseline CI</li> </ul>	<ul style="list-style-type: none"> <li>• Uchovává CI, odkazy na ně a vazby mezi nimi</li> <li>• Brání neautorizovaným změnám v baseline CI</li> </ul>

Audit	<ul style="list-style-type: none"> <li>• <b><u>Konfigurační audit</u></b></li> <li>• Vyvíjený produkt odpovídá požadavkům, standardům a smluvním podmínkám</li> <li>• Všechny komponenty produkovány, správně identifikovány a popsány, vyřešeny všechny požadavky na změny</li> <li>• <b><u>Baseline audit</u></b></li> <li>• Na konci fáze živ. cyklu</li> <li>• Kontrola úplnosti a správnosti obsahu CMS</li> <li>• Test. reporty, dokumentace verze, rozdíly mezi původní a aktuální dokumentací</li> </ul> <ul style="list-style-type: none"> <li>• Kontrola skutečného stavu ICT infrastruktury v porovnání se zdokumentovanými baselines</li> <li>• Kontrola autorizace implementace změn</li> </ul>
-------	--

## 5. Závěr

Řízení konfigurace je pro vedoucí projektu jedním z nejdůležitějších procesů, neboť poskytuje náhled do stavu produktu, kterým se projekt zabývá a umožňuje tak neustálou kontrolu nad tím, zda splňuje požadavky zákazníka. K tomu je samozřejmě nutné dodržovat určité zásady, z kterých je několik popsáno v tomto dokumentu. Jejich úspěšnou implementací do vývojové infrastruktury tak můžeme nejen získat silné zázemí pro vývoj produktů, ale zejména předejít mnoha problémům, které se vlivem špatného nebo žádného řízení konfigurace objevují.

## 6. Použité zdroje

**Software Engineering Institute. 2010. *CMMI for Development, Version 1.3*. Software Engineering Institute. [Online] 2010. [Citace: 6. 12. 2012.]**  
<http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm>.

**KASSE, Tim. *Practical Insight into CMMI®*. Second Edition. Artech House, 2008, s. 128-145. ISBN 978-1-59693-275-3.**

**RANCE, Stuart. THE STATIONERY OFFICE. *ITIL service transition*. 2nd ed. London: TSO, 2011, xii, 347 s. ISBN 978-0-11-331306-8.**