

# Implementační balíček

## Správa verzí

### Základní profil

---

#### Poznámky:

Tento dokument je duševním vlastnictvím autorovy společnosti. Informace obsažené v tomto dokumentu však mohou být volně používány. Distribuce všech částí tohoto dokumentu je povolena pro nekomerční využití pouze při zahrnutí následujícího oznámení:

© Thai Industrial Standard Institute

Komerční využití tohoto dokumentu je přísně zakázáno. Tento dokument je distribuován pro zlepšení výměny technických a vědeckých informací.

Tento materiál je vytvořen tak, jak je. Autor (autoři) neposkytuje (neposkytují) záruky jakéhokoli druhu, ať už vyjádřené nebo implicitní, a to v jakékoli záležitosti včetně, ale nejen, záruky vhodnosti pro účely obchodu, exkluzivity nebo výsledků získaných při použití materiálu.

Záměrem procesů popsaných v tomto implementačním balíčku není zamezení použití ani odrazení od použití dalších procesů, které mohou být pro velmi malé podniky užitečné.

<b>Autor</b>	Sanyakorn Buasung, Thai Industrial Standard Institute, Thailand
<b>Editoři</b>	C. Y. LAPORTE – École de Technologie Supérieure (ETS), (Canada) ANA VAZQUEZ – 5th level, (México)
<b>Datum vytvoření</b>	23. října 2008
<b>Poslední update</b>	15 prosince 2012
<b>Stav</b>	Potvrzená specifikace
<b>Verze</b>	1.4

Verze 1.4

## Předchozí verze

Datum (rrrr-mm-dd)	Verze	Autor	Modifikace
2008-04-11	0.1	S. Buasung	Vytvoření dokumentu.
2008-05-05	0.2	S. Buasung	Aktualizace kapitol 1, 2, 3, příloh A a D.
2008-09-17	0.3	S. Buasung	Stanovení rozsahu balíčku Správa verzí pro základní profil.
2008-09-18	1.0	S. Buasung	Finální verze – Připraveno pro posouzení
2008-09-22	1.1	C Laporte	Drobné úpravy
2008-09-28	1.2	S. Buasung	Přepracováno na základě posudků
2008-10-23	1.3	S. Buasung	Aktualizace přílohy A.2 a přílohy D.1
2009-08-14	1.4	C Laporte	Aktualizace dle nové šablony a celková revize

## Zkratky

Zkratka	Definice
DP	Deployment Package (implementační balíček) - soubor artefaktů vyvinutých pro usnadnění implementace sady postupů, vybraného rámce, ve velmi malých podnicích.
VSE	Very Small Entity (velmi malý podnik) – podnik, organizace, oddělení nebo projekt do 25 lidí.
VSEs	Very Small Entities (velmi malé podniky)
<detaily>	<detaily>

## Obsah

<b>1. Technický popis</b>	<b>4</b>
Účel dokumentu	4
Proč je správa verzí důležitá?	4
<b>2. Definice</b>	<b>5</b>
Obecné termíny	5
Specifické termíny	5
<b>3. Vztahy s ISO/IEC 29110</b>	<b>7</b>
<b>4. Popis procesů, činností, úkolů, kroků, rolí a produktů</b>	<b>9</b>
Úkoly	9
Plánování & nastavení úložiště	9
Identifikace verze	10
Kontrola změn	10
Role & artefakty	12
<b>5. Šablony</b>	<b>13</b>
Strategie správy verzí	13
Rozvržení úložiště projektu	13
Identifikace verzí - <a href="http://www.tbs-sct.gc.ca">www.tbs-sct.gc.ca</a>	14
Šablona Žádosti o změnu obsahu	14
<b>6. Příklad životního cyklu aktivit</b>	<b>15</b>
Příklad životního cyklu správy verzí	15
<b>7. Kontrolní seznam</b>	<b>16</b>
<b>8. Nástroje</b>	<b>17</b>
<b>9. Odkazy na další standardy a modely</b>	<b>18</b>
Matice odkazů ISO 9001	18
Matice odkazů ISO/IEC 12207	18
Matice odkazů CMMI	18
<b>10. Odkazy</b>	<b>19</b>
<b>11. Formulář pro hodnocení</b>	<b>20</b>

## 1. Technický popis

### **Účel dokumentu**

Tento implementační balíček (DP) podporuje základní profil, jak je definováno v ISO/IEC 29110 Část 5-1: Příručka pro řízení a implementaci. Implementační balíček je soubor artefaktů vyvinutých pro usnadnění implementace sady postupů, vybraného rámce, ve velmi malých podnicích (VSE). Implementační balíček není procesní referenční model (tj. není normativní). Prvky typického implementačního balíčku jsou: popis procesů, činnosti, úkoly, role a produkty, šablona, kontrolní seznam, příklady, reference a odkaz na normy a modely, a nástroje.

Obsah dokumentu je zcela *informativní*.

Tento dokument byl vytvořen Sanyakorn Buasungem (TISI) za jeho oficiální účasti na ISO JTC1/SC7/WG24.

### **Proč je správa verzí důležitá?**

System správy verzí poskytuje úložiště pro evidenci změn ve zdrojovém kódu a souvisejících artefaktech. Je to místo, kde máme přehled o tom, jak se software změnil, kdy se změnil a kdo ho změnil.

Úspěch projektu je vysoce závislý na efektivním systému správy verzí. Správa verzí se stala nezbytnou pro softwarové organizace všech velikostí, včetně VSEs, protože softwarový projekt produkuje během svého průběhu množství položek, včetně různých dokumentů, programů, dat a manuálů. Všechny tyto položky mohou být snadno změněny kdykoliv v průběhu projektu.

Typické výhody správy verzí:

- Správa verzí uchovává historii změn. Je zachována historie pro každý artefakt, což pomáhá členům týmu určit, proč byla provedena určitá rozhodnutí, co bylo vyzkoušeno v minulosti a kdo provedl změny. Je také důležité zajistit, aby všichni členové týmu měli vždy přístup k nejaktuálnější verzi a mohli lehce najít, nebo přejít na starší verze.
- Správa verzí zajišťuje koordinaci týmů a zlepšuje komunikace mezi členy týmu, správa verzí umožňuje komukoliv číst a kopírovat projektové zdroje, ale pouze ověření nebo autorizovaní členové týmu mohou provádět jejich aktualizace.
- Vzhledem k umožnění skupinové spolupráce členů týmu na jednom artefaktu, ať ve stejném čase nebo sekvenčně, správa verzí pomáhá zajistit, že jeden člen týmu nepřepíše práci jiného člena týmu a že práce všech členů týmu jsou konzistentní a kompatibilní.
- Zlepšuje produktivitu vývojového týmu a kvality práce, protože každý má přehled o všech aspektech projektu.
- Se správou verzí lehce zvládneme řízení uvolňování softwaru a zvýšíme zákaznickou spokojenost.

## 2. Definice

V této kapitole najde čtenář dvě sady definic. První definuje pojmy používané ve všech implementačních balíčcích, tj. obecné termíny. Druhá termíny používané v tomto implementačním balíčku, tj. specifické termíny.

### **Obecné termíny**

**Proces:** soubor vzájemně souvisejících nebo vzájemně působících činností, který přeměňuje vstupy na výstupy [ISO/IEC 12207].

**Činnost:** sada soudržných úkolů procesu [ISO/IEC 12207].

**Úkol:** požadovaná, doporučená nebo přípustná činnost, přispívající k dosažení jednoho nebo více výstupů procesu [ISO/IEC 12207].

**Pod-úkol:** když je úkol složitý, je rozdělen na pod-úkoly.

**Krok:** V implementačních balíčcích je úkol rozložen na posloupnost kroků.

**Role:** definovaná funkce, kterou má plnit člen projektového týmu, jako je testování, plnění, kontrolování, kódování. [ISO/IEC 24765]

**Produkt:** údaj, který může být vytvořen (není povinné) jedním nebo několika úkoly. (*např. návrhový dokument, zdrojový kód*).

**Artefakt:** informace, která není uvedena v ISO/IEC 29110 část 5, ale může pomoci VSE během provádění projektu.

### **Specifické termíny**

**Záloha:** **(1)** systém, komponenta, složka, procedura nebo osoba schopná nahradit, nebo pomoci obnovit primární položku v případě poruchy nebo externě způsobné katastrofy **(2)** vytvořit nebo určit systém, komponentu, soubor, proceduru nebo osobu jako náhradu [ISO/IEC 24765]

**Potvrzená specifikace:** specifikace nebo produkt, který byl formálně posouzen, a bylo dohodnuto, že bude sloužit jako základ pro další vývoj a může být změněna pouze prostřednictvím formálních postupů. [ISO/IEC 12207:2008]

**Požadavek na změnu:** žádost o rozšíření nebo omezení rozsahu projektu, změnu politiky, procesů, plánů nebo procedur, úprava nákladů nebo rozpočtu nebo revidování plánů. Žádosti o změnu mohou být přímé nebo nepřímé, zahájeny interně nebo externě a právně nebo smluvně pověřené nebo volitelné. Pouze formálně zdokumentované požadavky na změnu jsou

Verze 1.4

---

zpracovány a pouze schválené požadavky na změnu jsou prováděny. [PMBOK® Guide – Third Edition]

**Kontrola změn:** identifikování, dokumentování, schvalování nebo zamítání a kontrolování změn potvrzené projektové specifikace. [PMBOK® Guide – Third Edition]

**Zavádění:** integrace změn provedených vývojářem v jeho otevřeném zdrojovém kódu do sdíleného kódu přístupného prostřednictvím systémového úložiště pro správu verzí. [ISO/IEC 24765]

**Konfigurační položka (CI):** entita v rámci konfigurace, která splňuje funkci koncového použití a která může být jednoznačně identifikována v daném referenčním bodě. [ISO/IEC 12207:2008]

**Konfigurační management:** disciplína uplatňující technické a administrativní vedení a dohled na: identifikování a zdokumentování funkčních a fyzických vlastností konfigurační položky, řídit změny těchto vlastností, zaznamenává a hlásí změny zpracování a realizace stavu a ověřuje soulad se stanovenými požadavky. [ISO/IEC 24765]

**Analýza dopadů:** identifikace celého systému a softwarových produktů, které žádost o změnu ovlivňuje a vývoj odhadu zdrojů potřebných k dosažení změny. [ISO/IEC 24765]

**Obnova: (1)** obnovení systému, programu, databáze, neb jiného systémového zdroje do podoby, ve které se mohou provádět požadované funkce **(2)** klonování clusteru po jeho selhání nebo odstranění. [ISO/IEC 24765]

**Uvolněná verze:** konkrétní verze konfigurační položky, která je k dispozici pro konkrétní účel (např. testovací verze). [ISO/IEC 12207:2008]

**Úložiště: (1)** sbírka všech softwarově souvisejících artefaktů, které patří do systému, místo/formát, ve kterém je taková sbírka uložena. [ISO/IEC 24765]

**Softwarová konfigurace:** Ucelený soubor softwarových produktů zahrnující:

- Specifikaci požadavků
- Návrh softwaru
- Sledování záznamů
- Komponenty
- Software (jednotka, produkt, položka)
- Testovací případy a zkušební postupy
- Hlášení chyb
- Návod k použití
- Uživatelská dokumentace softwaru
- Dokumentace údržby

[ISO/IEC TR 29110-5-1]

**Správa verzí:** zřízení a vedení potvrzených specifikací a identifikace a řízení změn potvrzených specifikací, aby bylo možno vrátit se do předchozí úrovně. [ISO/IEC 24765]

### 3. Vztahy s ISO/IEC 29110

Tento implementační balíček zahrnuje činnosti související se správou verzí ISO technická zpráva ISO/IEC 29110 část 5-1 pro velmi malé podniky (VSEs) – Základní profil [ISO/IEC29110].

V této kapitole najde čtenář seznam procesů, činností, úkolů a rolí projektového řízení (PM) a softwarové implementace (SI) z kapitoly 5, které se přímo vážou k tomuto tématu. Téma je detailně popsáno v následující kapitole.

- **Proces: Projektové řízení**
- **Činnost: PM.1 Plánování projektu**
- **Úkoly a role:**

Úkoly	Role <sup>1</sup>
PM.1.10 Zdokumentujte strategii správy verzí v projektovém plánu.	PM, TL
PM.1.15 Vytvořte, nebo připravte za pomoci strategie správy verzí.	PM, TL

- **Proces: Projektové řízení**
- **Činnost: PM.2 Realizace projektového plánu**
- **Úkoly a role:**

Úkoly	Role <sup>2</sup>
PM.2.5 Provedte zálohu podle strategie správy verzí.	PM
PM.2.6 Pokud je to nutné, provedte obnovení projektového úložiště pomocí jeho zálohy.	PM

- **Proces: Implementace softwaru**
- **Činnost: SI.2 Analýza softwarových požadavků**
- **Úkoly a role:**

Úkoly	Role <sup>3</sup>
SI.2.7 Začleňte specifikaci požadavků a * uživatelskou dokumentaci softwaru k potvrzené specifikaci softwarové konfigurace.	TL

- **Proces: Implementace softwaru**
- **Činnost: SI.4 Konstrukce softwaru**
- **Úkoly a role:**

<sup>1</sup> Role jsou definovány v následující kapitole. Role jsou také definovány v normě ISO/IEC 29110 část 5-1

<sup>2</sup> Role jsou definovány v následující kapitole. Role jsou také definovány v normě ISO/IEC 29110 část 5-1

<sup>3</sup> Role jsou definovány v následující kapitole. Role jsou také definovány v normě ISO/IEC 29110 část 5-1

Verze 1.4

Úkoly	Role <sup>4</sup>
SI.2.7 Začleňte specifikaci požadavků a * uživatelskou dokumentaci softwaru k potvrzené specifikaci softwarové konfigurace.	TL

- **Proces: Implementace softwaru**
- **Činnost: SI.5 Softwarová integrace a testování**
- **Úkoly a role:**

Úkoly	Role <sup>5</sup>
SI.5.11 Začleňte software, sledování záznamu, testovací report, návod k použití a uživatelskou dokumentaci k softwaru do softwarové konfigurace jako součást potvrzené specifikace.	TL

- **Proces: Implementace softwaru**
- **Činnost: SI.6 Dodání produktu**
- **Úkoly a role:**

Úkoly	Role <sup>6</sup>
SI.6.6 Provedte dodávky dle dodacích pokynů.	TL

**Poznámka:**

- Úkoly jsou uvedeny postupně v kapitole 3.1, pořadí však nepředstavuje žádný životní cyklus (tj. detailní úkoly mohou být organizovány buď sekvenčně, nebo iterativně). Čtenář najde v kapitole 3.3 některé příklady činností objednáni v rámci některých životních cyklů.
- Neexistují žádná pravidla týkající se přesného formátu artefaktu (např. specifikace požadavků mohou být dokumentovány a řízeny v dokumentu aplikace Word, nebo v tabulkovém procesoru Excel, nebo pomocí webového nástroje).
- Každý z kroků popsaných níže musí být přizpůsoben kontextu a organizaci projektu. Důvodem je snížit rizika spojená s nedostatkem řízení konfigurace pro VSE.

Náročnost každého kroku se bude lišit v závislosti na velikosti projektu (malý nebo velký systém) od několika člověkohodin po několik člověkodnů nebo člověkotýdnů.

Mezi hlavní úkoly správy verzí patří:

- Plánování a nastavení úložiště
- Identifikace verzí
- Kontrola provedených změn

<sup>4</sup> Role jsou definovány v následující kapitole. Role jsou také definovány v normě ISO/IEC 29110 část 5-1

<sup>5</sup> Role jsou definovány v následující kapitole. Role jsou také definovány v normě ISO/IEC 29110 část 5-1

<sup>6</sup> Role jsou definovány v následující kapitole. Role jsou také definovány v normě ISO/IEC 29110 část 5-1



## 4. Popis procesů, činností, úkolů, kroků, rolí a produktů

### Úkoly

#### Plánování & nastavení úložiště

<b>Cíle:</b>	Vytvořit strategii správy verzí a nastavit prostředí pro provádění správy verzí.
<b>Odůvodnění:</b>	Vzhledem k tomu, že každý projekt má rozdílné vlastnosti, je třeba přizpůsobit postup správy verzí pro každý jednotlivý projekt a je třeba se ujistit, že všichni účastníci projektu vědí, jak bude prováděno řízení správy verzí. Úložiště se používá k ukládání a řízení konfigurace softwaru (zdrojové kódy a související artefakty).
<b>Role:</b>	Projektový manažer Technický vedoucí
<b>Artefakty:</b>	Strategie správy verzí Projektové úložiště
<b>Kroky:</b>	1. Vytvoření strategie správy verzí 2. Vytvoření úložiště
<b>Popis kroků:</b>	<p><b>Step 1. Vytvoření strategie správy verzí:</b> Projektový manažer vytvoří strategii správy verzí. Strategie správy verzí je popsána v projektovém plánu. (viz typická tabulka obsahu v příloze A)</p> <p><b>Step 2. Vytvoření úložiště:</b> Technický vedoucí projektu vytvoří úložiště a to konfiguruje.</p> <ul style="list-style-type: none"> <li>• Vytvořte nové úložiště</li> <li>• Vytvořte sdílený prostor nebo složky (např. pracovní složky, sílené složky)</li> <li>• Nastavte přístupy do úložiště</li> <li>• Zajistěte mechanismy pro ukládání, načítání, zálohování a obnovování položek.</li> <li>• Zajistěte školení členům týmu a zúčastněným stranám o: strategii správy verzí, úložišti, procedurách a nástrojích.</li> <li>• Proveďte pravidelné zálohování úložiště</li> <li>• Ověřte, že můžete úspěšně provádět zálohování</li> </ul> <p><i>Tip:</i> K dispozici jsou open-source nástroje, které podporují vytváření úložišť a správu verzí. (viz příloha D)</p>

Verze 1.4

**Identifikace verze**

<b>Cíle:</b>	Definovat uvolněné verze, položky a systém číslování dokumentů pro verze a varianty položek.
<b>Odůvodnění:</b>	Identifikace verzí je vyžadována pro všechny položky potvrzené specifikace (včetně zdrojových souborů, objektových modulů, uvolněných verzí a dokumentací)
<b>Role:</b>	Technický vedoucí
	Projektový tým
<b>Artefakty:</b>	Softwarová konfigurace
	Potvrzené specifikace nebo uvolněná verze
<b>Kroky:</b>	1. Definování položek identifikace
	2. Provedení uvolnění identifikace
<b>Popis kroků:</b>	<p><b>Krok 1. Definování položek identifikace:</b></p> <p>Projektový manažer a technický vedoucí vyberou produkty / položky, které budou kontrolovány.</p> <p>Typické položky zahrnují specifikaci požadavků, projektové dokumentace, zdrojový kód, testovací dokumenty, projektové plány, uživatelské příručky, školící materiály, zadávací dokumentace, testovací záznamy, podpůrné nástroje a další, které budou součástí dodávky.</p> <p>Technický vedoucí přiřadí jedinečné identifikátory položkám, které spadají pod správu verzí na základě předdefinované konvence a číselných schémat.</p> <p><b>Krok 2. Provedení uvolnění identifikace:</b></p> <p>Technický vedoucí získá povolení před vytvořením nebo uvolněním potvrzených specifikací konfiguračních položek.</p>

**Kontrola změn**

<b>Cíle:</b>	Sledovat a řídit změny konfiguračních položek / potvrzených specifikací
<b>Odůvodnění:</b>	Mnoho úsilí při vývoji softwaru zahrnuje skupinová spolupráce členů týmu, častokrát geograficky oddělených, pracujících souběžně na vzájemně závislém softwaru, vyvíjeném pomocí mnoha iterací a zaměřeném na různé produkty a platformy. Snadno lze ztratit přehled o tom, co se změnilo a proč a jak zapadají jednotlivé části

Verze 1.4

	do sebe. Výsledky mohou mít vážný dopad na náklady, harmonogram a kvalitu.
<b>Role:</b>	Žadatel
	Projektový manažer
	Technický vedoucí
	Projektový tým
<b>Artefakty:</b>	Požadavek na změnu
	(Aktualizace) Softwarová konfigurace
<b>Kroky:</b>	1. Kontrola změn
	2. Provedení změn v CIs
<b>Popis kroků:</b>	<p><b>Krok 1. Kontrola změn:</b></p> <p>Jakékoli změny položek potvrzené specifikace musí být zdokumentovány, vyhodnoceny, schváleny nebo zamítnuty odpovědnou osobou a sledovány, aby mohly být uzavřeny projektovým manažerem.</p> <p><b>Krok 2. Provedení změn v CIs:</b></p> <p>Schválené požadavky na změny jsou realizovány pomocí definovaných postupů. Tento úkol může být podporován nástroji pro správu verzí, které umožňují provádět kontrolu verzí / změn.</p> <ul style="list-style-type: none"> <li>• Pro provedení úprav zkopírujte CIs z projektového úložiště.</li> <li>• Upravte CIs ve Vašem "pracovním adresáři".</li> <li>• Verifikujte provedené změny v CIs</li> <li>• Umístěte upravené CIs do projektového úložiště ("sdíleného adresáře")</li> </ul> <p><i>Tip:</i> Nezapomeňte zaznamenat popis změn při jejich zavádění.</p>

**Role & artefakty**

<b>Role</b>	<b>Definice</b>
Projektový manažer	Vyvíjí strategii správy verzí, identifikuje položky konfigurace a uvolňování a schvaluje požadavky na změnu.
Projektový tým	Vyvíjí produkty / pracovní produkty a provádí změny podle schválených požadavků na změnu.
Žadatel	Potvrzuje požadavky na změnu, může jím být zákazník nebo projektový tým.
Technický vedoucí	Vytváří, zálohuje a obnovuje projektové úložiště, řídí změny a uvolňování potvrzených specifikací.

**Tabulka 1 Definice rolí**

<b>Artefakty</b>	<b>Definice</b>
Požadavek na změnu	Viz definice.
Projektové úložiště	Úložiště konfigurací softwaru, jejich změn, potvrzených specifikací, uvolněných verzí; obsahuje přístupové a kontrolní mechanismy; zálohovací a obnovovací mechanismy
Úložiště	Kombinace souborů, které se spolu stávají konfigurační položkou. Je distribuováno mimo projekt.
Softwarová konfigurace	Viz definice
Strategie správy verzí	Dokument strategie implementace správy verzí. Je součástí projektového plánu

**Tabulka 2 Definice artefaktů**

## 5. Šablony

Šablony představují příklad použití tohoto implementačního balíčku, měly by však být přizpůsobeny Vašemu projektu.

### **Strategie správy verzí**

Strategie správy verzí, která může být součástí plánu řízení projektu, popisuje následující oblasti:

- Identifikace nástrojů pro práci s úložištěm a způsob práce s nimi
- Umístění a způsob přístupu k úložišti
- Popis způsobu verzování
- Popis způsobu zálohování a obnovení požadované verze
- Popis ukládání, manipulace, archivace a vyhledávání verzí

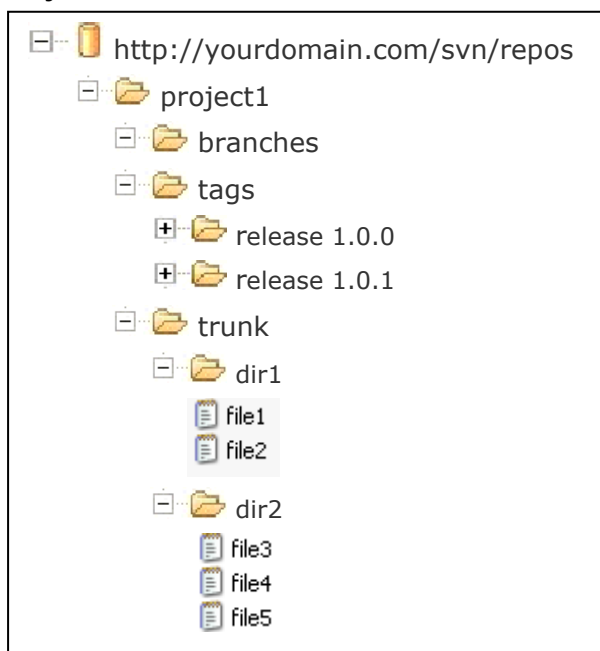
### **Rozvržení úložiště projektu**

Příklad úložiště pro správu verzí pomocí Subversion (SVN)

#### **Úložiště (Repository)**

Pro správu verzí pomocí Subversion je doporučováno zvolit nezávisle na druhu projektu rozvržení úložiště s kořenovým adresářem (nejvýše postavená složka, která obsahuje data vztahující se k projektu), a poté vytvořit 3 podadresáře vztahující se ke kořenovému adresáři:

- **trunk**, je podadresář, kde probíhá samotný vývoj software;
- **branches**, je podadresář, kde je možné vytvářet různě pojmenované větve při hlavním vývoji software; a
- **tags**, je podadresář, ve kterém se nachází soubory, které se již nezmění. Zpravidla se jedná o finální verze software.



Poslední pracovní verze je umístěna v podadresáři /trunk/.  
Uvolněné verze jsou umístěny v podadresáři /tags/.

Verze 1.4

---

## **Identifikace verzí - [www.tbs-sct.gc.ca](http://www.tbs-sct.gc.ca)**

### ***Položka (Item)***

---

- Každá položka má vlastní název, v daném projektu jedinečný, neopakující se
- Každá položka obsahuje historii provedených změn
- Každá položka má přiřazený identifikátor verze, který může mít 2 formy:
  - Celé číslo, počáteční verze se označuje číslem 1 a každá další verze je označována následujícím celým číslem.
  - Desetinné číslo, které označuje pořadí uvolněné verze. Používá se např., pokud dokumentace souvisí s verzí produktu.
- Každá položka by dále měla obsahovat: popis položky, datum vytvoření/pořízení a důvod vytvoření/pořízení.

### ***Uvolněná verze (Release)***

---

Všechny uvolněné verze se označují ve formátu: **M.F.I**

Kde

**M** = Update důležité funkcionality (Major feature update), počáteční číslice: "1"

**F** = Update doplňkové funkcionality, případně oprava chyb (Minor feature or bug fix)

Označení první verze: "0"

**I** = Vnitřní upgrade (Internal upgrade), Označení první verze: "0".

Označení tedy bude např. ve tvaru: (Uvolněná) verze 2.1.0.

- Každé pole se označuje číslicí, která se s další verzí zvyšuje o jedničku.
- Pole označující vnitřní upgrade může být vynecháno, pokud nedošlo k vnitřní změně.
- Opravy (patch) mohou být vydány jako součást nové uvolněné verze, musí však být označeny přidáním písmena na pozici F nebo I pole. Příklad: Oprava 2.1A.0.
- Uvolněné verze jsou sestavovány z kombinace položek, které samy mají vlastní identifikátory verzí (viz. výše).

## **Šablona Žádosti o změnu obsahu**

Převzato z ISO/IEC 15504.5 Change Request's WP Characteristics

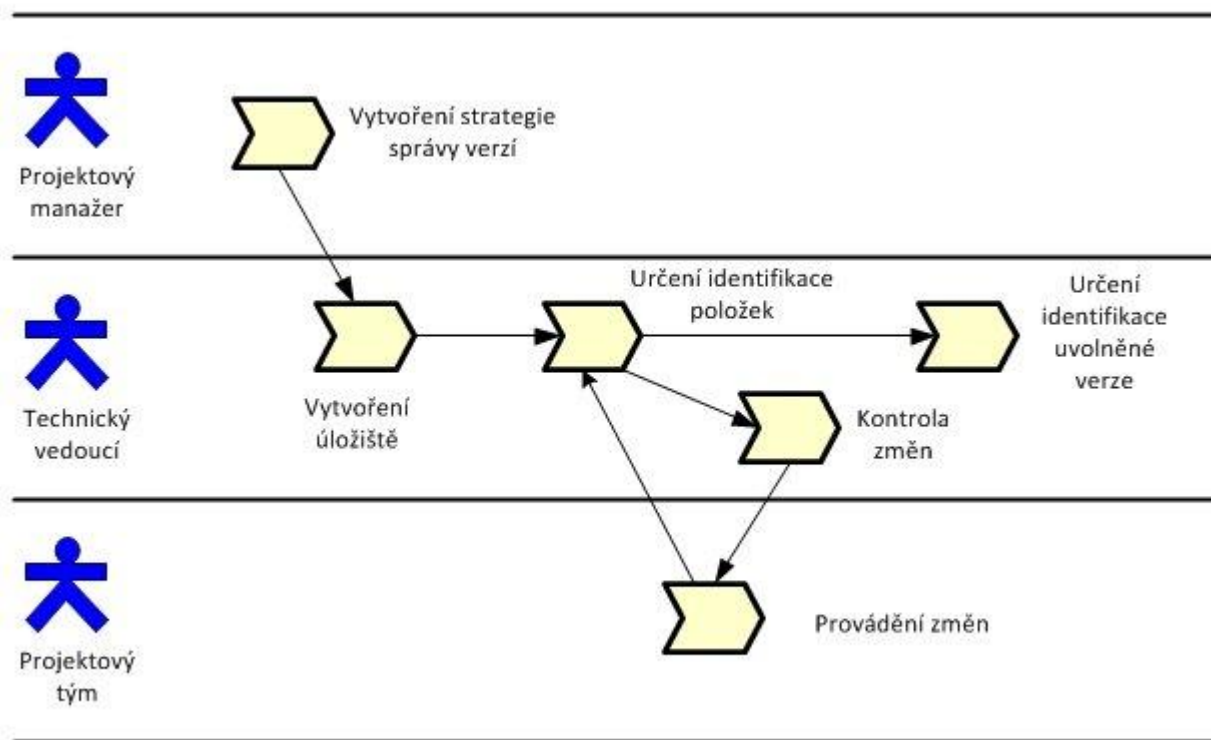
1. Číslo Žádosti o změnu
2. Účel změny
3. Stav Žádosti o změnu
4. Informace o žadateli
5. Ovlivněný systém/systémy
6. Vliv na operace probíhající v již existujícím systému/systémech.
7. Vliv na související dokumentaci
8. Závažnost žádosti
9. Termín provedení změny

## 6. Příklad životního cyklu aktivit

**Upozornění:** Tato sekce obsahuje grafické znázornění příkladů postupu verzování. Tyto příklady jsou poskytnuty jako pomoc čtenáři při implementaci vlastního postupu verzování odpovídajícímu kontextu a omezení jeho IT projektu.

### Příklad životního cyklu správy verzí

Tento diagram slouží pouze pro ilustraci – pro tvorbu obdobných diagramů použijte SPEM v aplikaci Microsoft Visio (<http://www.pa.icar.cnr.it/cossentino/FIPAmeth/docs/SPEM.vss>).



Obrázek 1 Příklad životního cyklu správy verzí

Verze 1.4

---

## 7. Kontrolní seznam

Bude dopracován



## 8. Nástroje

Příklady verzovacích systémů, které jsou k dispozici zdarma

- **Subversion** [<http://subversion.tigris.org/>] - Subversion (SVN) slouží ke správě souborů a adresářů v průběhu času. Strom souborů je umístěn do centrálního úložiště.
- **TortoiseSVN** [<http://tortoisesvn.tigris.org/>] - Klient Subversion pro operační systém Windows. Je k dispozici zdarma. Tutoriál k používání TortoiseSVN je k dispozici na adrese: [http://merlin.fit.vutbr.cz/wiki/index.php?title=SVN\\_tutori%C3%A1l&redirect=no](http://merlin.fit.vutbr.cz/wiki/index.php?title=SVN_tutori%C3%A1l&redirect=no)
- **SmartSVN** [<http://www.syntevo.com/smartsvn/index.html>] - Grafický klient pro Subversion. Výkonný a uživatelsky jednoduchý.
- **CVS** [<http://ximbiot.com/cvs/>] - Concurrent Versions System. Jeden z prvních verzovacích systémů. Dnes se spíše používá nástupce Subversion.
- **SmartCVS** [<http://www.syntevo.com/smartcvs/index.html>] - Inovativní multiplatformní CVS klient. Obsahuje užitečné funkce pro práci s adresářem a soubory, zobrazování transakcí, prohlížení verzí v úložišti, apod. Velmi jednoduché, intuitivní ovládání. SmartCVS se zaměřuje na denní úkoly, použitelnost a není limitován dostupnými CVS příkazy.

Verze 1.4

## 9. Odkazy na další standardy a modely

Tato kapitola poskytne informace o souvislostech Implementačního balíčku Správy verzí s normou ISO/IEC a modelem zralosti CMMI verze 1.2, organizace Software Engineering Institute (CMMI<sup>®7</sup>).

Poznámky:

- tato část je poskytována pouze pro informační účely
- v každé tabulce jsou uvedeny pouze úkoly, na které se vztahuje tento implementační balíček
- tabulky používají následující konvence:
  - Plné pokrytí (Full Coverage) = F
  - Částečné pokrytí (Partial Coverage) = P
  - Žádné pokrytí (No Coverage) = N

### **Matice odkazů ISO 9001**

<b>Ustanovení normy ISO 9001</b>	<b>Pokrytí F/P</b>	<b>Název úkolu a krok</b>	<b>Komentář</b>
Bude dopracováno			

### **Matice odkazů ISO/IEC 12207**

<b>Ustanovení normy ISO/IEC 12207</b>	<b>Pokrytí F/P</b>	<b>Název úkolu a krok</b>	<b>Komentář</b>
Bude dopracováno			

### **Matice odkazů CMMI**

<b>Cíle/Praktiky CMMI V1.2</b>	<b>Pokrytí F/P</b>	<b>Název úkolu a krok</b>	<b>Komentář</b>
Bude dopracováno			

<sup>7</sup> Capability Maturity Model, CMMI je registrován u U.S. Patent and Trademark Office a je ochrannou známkou Carnegie Mellon University.

Verze 1.4

## 10. Odkazy

Hodnota	Reference
ISO/IEC 29110	Software Engineering – Lifecycle Profiles for Very Small Entities (VSEs) – Part 5-1: Management and Engineering Guide - Basic VSE Profile
ISO/IEC12207	ISO/IEC 12207:2008 Systems and software engineering – Software life cycle processes.
ISO/IEC15504	ISO/IEC 15504.5:2006 Process Assessment – An exemplar Process Assessment Model.
ISO/IEC 24765	ISO/IEC 24765, Slovník systémového a softwarového inženýrství. <a href="http://pascal.computer.org/sev_display/index.action">http://pascal.computer.org/sev_display/index.action</a>
SWEBOK	ISO/IEC TR 19759:2005 Software Engineering – Guide to the Software Engineering Body of Knowledge (SWEBOK).
Treasury Board of Canada Secretariat	Version Identification Procedure, <a href="http://www.tbs-sct.gc.ca/emf-cag/config-mgmt/cm020-eng.asp">http://www.tbs-sct.gc.ca/emf-cag/config-mgmt/cm020-eng.asp</a>

Verze 1.4

## 11. Formulář pro hodnocení

### Implementační balíček - Správa verzí – Verze 1.4

Vaše zpětná vazba umožní tvůrcům vylepšit, rozvíjet a opravovat implementační balíčky. Jakékoliv komentáře, návrhy či připomínky jsou vítány.

#### 1. Jak spokojen(á) jste s obsahem tohoto implementačního balíčku?

Velmi spokojen(á)    Spokojen(á)    Ani spokojen(á), ani nespokojen(á)    Nespokojen(á)    Velmi nespokojen(á)

#### 2. Jak hodnotíte pořadí, ve kterém jsou kapitoly uspořádány?

Velmi spokojen(á)    Spokojen(á)    Ani spokojen(á), ani nespokojen(á)    Nespokojen(á)    Velmi nespokojen(á)

#### 3. Jak hodnotíte strukturu/vzhled tohoto implementačního balíčku?

Velmi spokojen(á)    Spokojen(á)    Ani spokojen(á), ani nespokojen(á)    Nespokojen(á)    Velmi nespokojen(á)

#### 4. Přejde Vám některá kapitola zbytečná? (popište prosím)

#### 5. Jaká témata v tomto implementačním balíčku postrádáte? (popište prosím)

- Navrhované téma:
- Zdůvodnění navrhovaného tématu:

#### 6. Našli jste v tomto balíčku chybu/chyby? (zaznamenejte prosím)

- Popis chyby:
- Pozice chyby (kapitola #, téma #, tabulka #) :

#### 7. Jiné komentáře, připomínky:

#### 8. Doporučil(a) byste tento implementační balíček kolegům pracujícím v malých a středních podnicích?

Rozhodně ano    Spíše ano    Nevím jistě    Spíše ne    Určitě ne

#### Pokud nechcete zůstat anonymní, můžete zanechat kontakt

- Jméno: \_\_\_\_\_
- e-mailová adresa: \_\_\_\_\_

Vyplněný formulář zašlete na: [claude.y.laporte@etsmtl.ca](mailto:claude.y.laporte@etsmtl.ca), [Avumex2003@yahoo.com.mx](mailto:Avumex2003@yahoo.com.mx)

Anglický originál lze stáhnout zde: <http://profs.etsmtl.ca/claporte/English/VSE/Deploy-Pack/DP-Version%20Control-v1.4.doc>