

**Slovenská technická univerzita**

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4

---

**Metodika manažmentu verzií artefaktov  
pomocou nástroja Subversion**

**Bc. Ivan Šimko**

---

Študijný program: Softvérové inžinierstvo

Predmet: Manažment projektov softvérových a informačných systémov

Ak. rok: 2011/12

Ročník: 1

# 1 Úvod

Účelom metodiky je opis manažmentu verzií artefaktov softvérového projektu a procesov s ním spojených. Procesy sú vytvorené pre použitie v kombinácii s nástrojom Subversion.

Opisované sú dve úrovne abstraktnosti. Úroveň z vyššieho pohľadu je zachytená v kapitole 4 a predstavuje celkový pohľad na správu verzií artefaktov. Dostupný je prehľad procesov, grafické znázornenie ich návazností ako aj bližší popis každého procesu. Z tejto úrovne bol vybraný jeden proces, ktorý bližšie opisuje kroky vývoja a úpravy artefaktu pomocou nástroja Subversion (kapitola 5). Diagramy opisujúce následnosť procesov sú v kapitole 6.

## 2 Použité pojmy

### **Subversion**

voľne dostupný nástroj na centralizovaný systém manažmentu verzií

### **Repozitár**

Centrálne úložisko všetkých spravovaných artefaktov. Obsahuje všetky artefakty spojené s projektom, ich kompletnú históriu ako aj všetky vedľajšie, hlavné vývojové a produkčné verzie systému.

### **Artefakt**

Jedna konfigurovateľná jednotka nástroja na manažment verzií. V prípade nástroja Subversion je to akýkoľvek súbor.

### **Vedľajšia vetva**

Predstavuje modifikácie artefaktov hlavnej vývojovej vetvy projektu. Obsahuje vykonané neotestované zmeny. Každá jednotlivá funkcionálna z návrhu sa vyvíja vo svojej vedľajšej vetve.

### **Hlavná vývojová vetva**

Predstavuje relatívne stabilnú verziu vývoja projektu. Obsahuje zlúčenie otestovaných a verifikovaných vedľajších vetiev. Používa sa na vykonávanie akceptačných testov a uchovávanie produkčnej verzie projektu.

### **Lokálna pracovná verzia (working copy)**

Kópia verzie projektu umiestnená na klientskej strane. Obsahuje viaceré zmeny artefaktov vykonané vývojárov. Používa sa na vykonávanie a odosielanie zmien do centrálného úložiska.

### 3 Roly účastníkov

Rola	Zodpovednosť
Projektový manažér	Identifikovanie požiadaviek na systéme Spoluúčasť na identifikácii artefaktov
Vývojár	Spoluúčasť na identifikácii artefaktov Vývoj a úprava artefaktov Proces riešenia konfliktov
Manažér podpory vývoja	Spoluúčasť na identifikácii artefaktov Inštalácia nástroja na sledovanie zmien Konfigurácia repozitára a klientov na jeho použitie Proces začlenenia vývojovej vetvy do hlavnej vývojovej vetvy Proces riešenia konfliktov Proces nasadenia projektu
Manažér kvality	Proces testovania a verifikácie úprav artefaktov Proces nasadenia projektu

### 4 Procesy manažmentu verzií artefaktov

Manažment verzií softvérových artefaktov zahŕňa tieto procesy:

1. Proces identifikácie artefaktov
2. Proces inštalácie nástroja na sledovanie zmien
3. Proces konfigurácie repozitára
4. Proces vývoja
  - 4.1. Proces vytvorenia novej vedľajšej vývojovej vetvy
  - 4.2. Proces vývoja/úpravy artefaktu (detailný opis v kapitole 5)
  - 4.3. Proces testovania a verifikácie vykonaných úprav
  - 4.4. Proces začlenenia vedľajšej vetvy do hlavnej vetvy vývoja
5. Proces riešenia konfliktov
6. Proces nasadenia projektu

Diagramy opisujúce následnosť procesov sú v kapitole 6.

#### 4.1 Proces identifikácie artefaktov

<i>Vstup</i>	<i>požiadavky vytváraného systému</i>
<i>Výstup</i>	<i>identifikované artefakty ktoré chceme sledovať</i>
<i>Role používateľov</i>	<i>Projektový manažér, Vývojár, Manažér podpory vývoja</i>

Cieľom procesu je dohodnutie sa jednotlivých členov procesu na presnom určení všetkých konfigurovateľných častí projektu, ktorých verzie chceme sledovať. Príkladom je zdrojový kód modulu, špecifikácia požiadaviek alebo akceptačný test. Zúčastnené role tohto procesu sú projektový manažér, vývojár a manažér podpory. Každá rola má k výsledku procesu prispieť svojimi poznatkami v danej oblasti (napríklad požiadavky na produkt, spôsob a prostredie vývoja

systému alebo možnosti nástrojov na manažment verzií).

Úspešné ukončenie pokračuje procesom inštaláciou nástroja na sledovanie zmien(kapitola 4.2).

## 4.2 Proces inštalácie nástroja na sledovanie zmien

<i>Vstup</i>	<i>identifikované artefakty</i>
<i>Výstup</i>	<i>nainštalovaný vhodný nástroj na centrálnom serveri</i>
<i>Role používateľov</i>	<i>Manažér podpory vývoja</i>

Zodpovednosť za tento proces nesie manažér podpory vývoja. Jeho hlavnými úlohami sú:

- voľba vhodného nástroja na základe identifikovaných artefaktov
- inštalácia nástroja na centrálny server
- zabezpečenie proti výpadkom a strate údajov (redundancia, záloha repozitára)

Výsledkom procesu je nainštalovaný nástroj na centrálnom serveri. Po úspešnom ukončení procesu nasleduje proces jeho konfigurácie (kapitola 4.3).

## 4.3 Proces konfigurácie repozitára

<i>Vstup</i>	<i>identifikované artefakty, nainštalovaný nástroj na serveri</i>
<i>Výstup</i>	<i>nakonfigurovaný centrálny repozitár na serveri, nakonfigurovaní klienti na použitie centrálného repozitára</i>
<i>Role používateľov</i>	<i>Manažér podpory vývoja</i>

Hlavnú zodpovednosť za tento proces nesie manažér podpory vývoja. Jeho úlohou je na základe identifikovaných artefaktov správne vytvoriť a nakonfigurovať centrálny repozitár. Jeho povinnosti sú:

- vytvorenie centrálného repozitára pre projekt
- určenie vhodnej organizácie artefaktov v repozitári
- prepojenie repozitára s nástrojom na manažment úloh
- nastavenie prístupových práv pre všetkých potrebných používateľov
- konfigurácia klientov

Výsledkom procesu je pripravené celkové prostredie na prácu s nástrojom na manažment verzií artefaktov projektu.

## 4.4 Proces vývoja

<i>Vstup</i>	<i>analýza a návrh zmeny artefaktu (prípadne nového artefaktu)</i>
<i>Výstup</i>	<i>upravený artefakt začlenený do hlavnej vývojovej vetvy</i>
<i>Role používateľov</i>	<i>Manažér podpory vývoja, Vývojár, Manažér kvality</i>

Tento proces predstavuje samotný vývoj jednotlivých častí projektu na základe vytvoreného návrhu.

Viacero procesov tohto typu môže existovať paralelne, pričom sa neovplyvňujú. Proces sa ďalej delí na menšie podprocesy a jeho výsledkom je začlenenie celkovej otestovanej novej úpravy do hlavnej vývojovej vetvy projektu. Zodpovedné role majú bližšie určené úlohy v jednotlivých podprocesoch.

#### 4.4.1 Proces vytvorenia novej vedľajšej vývojovej vetvy

<i>Vstup</i>	<i>analýza a návrh zmeny artefaktu</i>
<i>Výstup</i>	<i>nová vedľajšia vývojová vetva</i>
<i>Role používateľov</i>	<i>Vývojár</i>

Úlohou procesu je vytvorenie novej vedľajšej vývojovej vetvy, teda prostredia v ktorom sa vyvíja jedná nová funkcionality podľa návrhu. Zabráňuje predčasným konfliktom a tým umožňuje paralelnú prácu viacerých vývojárov na rozličných funkcionality v jednom vývojovom cykle.

Nová vetva je vytváraná vývojárom podľa metodiky tvorenia vetví. Názov vetvy sa skladá z troch nasledovných častí:

- typ úlohy v nástroji na manažment úloh projektu (*task*, alebo *bug*)
- číslo úlohy v nástroji na manažment úloh
- stručný opis

Nástroj Subversion činnosť tvorenia vetvy umožňuje kopírovaním hlavnej vývojovej vetvy do zložky predstavujúcej novú vetvu. Umiestnenie zložky je určené podľa konfigurácie aktuálneho repozitára vykonanej v jeho procese konfigurácie (kapitola 4.3). Meno tejto zložky je identické s názvom vetvy.

Úspešné vykonanie procesu predstavuje začatie nového procesu vývoja a úpravy artefaktu (kapitola 4.4.2).

#### 4.4.2 Proces vývoja/úpravy artefaktu

<i>Vstup</i>	<i>artefakt a návrh zmeny z repozitára</i>
<i>Výstup</i>	<i>upravený artefakt podľa návrhu vložený do centrálného repozitára vo vedľajšej vývojovej vetve</i>
<i>Role používateľov</i>	<i>Vývojár</i>

Vývoj artefaktu má na starosti vývojár. Proces opisuje samotný vývoj softvéru a jeho častí. Zmeny by mali byť vykonávané po čo najmenších funkčných celkoch. V jednej vedľajšej vývojovej vetve môže tento proces vykonávať viacero vývojárov súčasne. Počas vykonávania jednotlivých krokov tohto procesu môžu nastať konflikty v zmenách artefaktov. V tomto prípade je nutné pokračovať procesom riešenia konfliktov (kapitola 4.5).

Podrobný opis jednotlivých krokov procesu je uvedený v časti 5 tejto metodiky. Po úspešnom ukončení procesu nasleduje proces testovania a verifikácie vykonaných zmien (kapitola 4.4.3).

### 4.4.3 Proces testovania a verifikácie vykonanej úpravy

<i>Vstup</i>	<i>upravená neotestovaná verzia artefaktu v repozitári</i>
<i>Výstup</i>	<i>výsledok testovania artefaktu – rozhodnutie o začlenení do hlavnej vývojovej vetvy</i>
<i>Role používateľov</i>	<i>Manažér kvality</i>

Zodpovednosť za proces nesie manažér kvality. Jeho úlohou je otestovanie vykonaných úprav artefaktov podľa metodiky testovania. Na základe výsledku testovania artefaktu je spustený jeden z nasledujúcich procesov:

- neúspešné testovanie : proces vývoja/úpravy artefaktu (kapitola 4.4.2)
- úspešné testovanie: proces začlenenia úpravy do hlavnej vetvy vývoja (kapitola 4.4.4)

Proces zabezpečuje integritu údajov v hlavnej vývojovej vetve a garantuje že všetky úpravy vložené do hlavnej vetvy spĺňajú všetky podmienky a testy platné pre dané typy artefaktov.

### 4.4.4 Proces začlenenia vedľajšej vetvy do hlavnej vetvy vývoja

<i>Vstup</i>	<i>úspešne otestované zmeny artefaktu vo vedľajšej vývojovej vetve</i>
<i>Výstup</i>	<i>začlenenie vývojovej vetvy do hlavnej vetvy vývoja, uzavretie vývojovej vetvy</i>
<i>Role používateľov</i>	<i>Manažér podpory vývoja</i>

Zodpovedná používateľská rola za proces je manažér podpory vývoja. Jeho úlohou je zabezpečenie správneho zlúčenia vedľajšej vývojovej vetvy vývoja funkcionality s hlavnou vývojovou vetvou projektu. Proces predstavuje ukončenie vývoja jednej funkcionálnej časti a uzatvára jeho vedľajšiu vývojovú vetvu.

Počas procesu môže nastať konflikt úprav vykonaných artefaktov. V tomto prípade je potrebné pokračovať procesom riešenia konfliktov (kapitola 4.5).

## 4.5 Proces riešenia konfliktov

<i>Vstup</i>	<i>pracovný adresár s nevyriešenými konfliktami zmien</i>
<i>Výstup</i>	<i>pracovný adresár s vyriešenými konfliktami zmien</i>
<i>Role používateľov</i>	<i>Vývojár; Manažér podpory vývoja</i>

Počas práce s repozitárom a hlavne počas synchronizácie s centrálnym repozitárom môžu nastať konflikty v lokálnych a vzdialených vykonaných zmenách. Nástroj Subversion neumožňuje uchovávanie konfliktných stavov artefaktov do centrálného úložiska, preto ich je potrebné riešiť lokálne.

Ak konflikt nastane počas vykonávania procesu synchronizácie so vzdialeným repozitárom (krok procesu úpravy artefaktu opísaný v kapitole 5.7), zodpovedný za odstránenie konfliktu je vývojár ktorý synchronizáciu vykonával. Výsledkom procesu je v tomto prípade odstránený konflikt v lokálnom pracovnom adresári spomínaného vývojára.

V prípade výskytu konfliktu počas procesu začlenenia úpravy do hlavnej vývojovej vetvy

(kapitola 4.4.4) zodpovednosť za tento proces ostáva manažérovi podpory vývoja. V tomto prípade je výsledkom správne pokračovanie procesu začlenenia a teda zabezpečenie integrity hlavnej vývojovej vetvy.

## 4.6 Proces nasadenia projektu

<i>Vstup</i>	<i>akceptačné testy, rôzne verzie všetkých artefaktov</i>
<i>Výstup</i>	<i>splnenie akceptačných testov uchovaný stav systému v jednom zvolenom čase</i>
<i>Role používateľov</i>	<i>Manažér testovania, Manažér podpory vývoja</i>

Produkčná verzia softvéru je verzia prezentovaná zákazníkovi a musí spĺňať všetky akceptačné testy. Zodpovedná rola za túto časť je manažér testovania, vykonávaná je podľa metodiky testovania a vykonáva sa typicky po každom ukončení vývojového cyklu.

Po úspešnom testovaní je nutné vytvoriť a uchovať aktuálny obraz systému, keďže hlavná vývojová vetva môže opisovať nefunkčný, nestabilný a neotestovaný stav systému. Túto činnosť má na starosti manažér podpory vývoja.

Nástroj Subversion uchováva produkčný stav systému ako kópiu hlavnej vývojovej vetvy v zložke určenej podľa organizácie projektu. Tento proces je analogický k procesu tvorenia vedľajších vývojových vetví. Výsledkom procesu je vytvorená zložka na správnom mieste, ktorá obsahuje najnovší produkčný stav systému. Táto zložka je pomenovaná podľa aktuálnej verzie softvéru.

## 5 Detailný opis procesu vývoja softvérového artefaktu pomocou nástroja Subversion

### 5.1 Vybratie vývojovej vetvy

Úpravy artefaktov sú vykonávané vo vedľajších vetvách. Pred vykonávaním akýchkoľvek zmien nad artefaktom si musí vývojár prepnúť svoju pracovnú verziu adresára na vedľajšiu vývojovú vetvu venovanú danej úlohe.

Ak táto vetva existuje (na úlohe sa už pracovalo), prepnutie je uskutočnené nasledovne:

```
$ svn switch ^/<meno_projektu>/branches/<meno_vetvy>
```

V prípade neexistujúcej vedľajšej vývojovej vetvy pre vykonanie úprav je nutné jej vytvorenie, ktoré zabezpečuje proces vytvorenia novej vedľajšej vývojovej vetvy (kapitola 4.4.1)

### 5.2 Pridávanie zmeny na sledovanie nástrojom

Nástroj Subversion na manažment verzií artefaktov vyžaduje presné určenie zmien ktoré má sledovať a zaznamenávať v centrálnom repozitári. Toto určenie zmien je umožnené na súborovej úrovni. Pre pridanie nového súboru na sledovanie ako aj pridanie zmeny v sledovanom súbore slúži nasledujúci príkaz:

```
$ svn add <cesta_k_súboru>
```

### 5.3 Odobranie sledovania artefaktu nástrojom

Používaný nástroj na manažment verzií Subversion umožňuje odobranie sledovaného artefaktu. Po odobratí už nebude súčasťou projektu a pri ďalšej synchronizácii z centrálnym úložiskom bude odstránený. História artefakt však nemožno odstrániť, teda všetky verzie artefaktu pred jeho odstránením ostanú uchované v centrálnom úložisku. Príkaz na vykonanie tejto operácie je nasledovný:

```
$ svn delete <cesta_k_súboru>
```

### 5.4 Synchronizácia so vzdialeným repozitárom

Počas práce s artefaktami je nutné synchronizovať lokálnu pracovnú verziu z centrálnym vzdialeným úložiskom. Častá synchronizácia predchádza tvoreniu zbytočných konfliktov a umožňuje jednoduchšiu paralelnú prácu viacerých vývojárov.

Pre synchronizáciu s centrálnym úložiskom je nutné mať vybranú zvolenú vývojovú vetvu (kapitola 5.1). Následne je možné vykonať príkaz na synchronizáciu:

```
$ svn update
```

Pri operácii synchronizovania so vzdialeným centrálnym úložiskom môžu nastať konflikty v lokálnych a vzdialených zmenách artefaktov. V tomto prípade je nutné pokračovať procesom riešenia konfliktov (kapitola 4.5).

### 5.5 Odoslanie zmien do centrálného úložiska

Po vykonanej úprave artefaktu je potrebné odoslať zmeny do centrálného úložiska. Tieto zmeny by mali byť čo najmenšie a funkčné. Časté odosielenie zmien umožňuje iným vývojárom vidieť postup práce ako aj jednoduchšie a skoršie riešenie konfliktov.

Pred odoslaním je nutné vykonať procesy pridania zmien a/alebo odobrania sledovania artefaktu nástrojom, ukončený proces synchronizácie so vzdialeným nástrojom a nemať žiadne konflikty v lokálnej pracovnej kópii vývojovej vetvy (kapitoly 5.2, 5.3 a 5.4, ).

Po splnení podmienok je možné odoslanie vykonaných zmien do centrálného úložiska príkazom:

```
$ svn commit -m '<sprava_vysvetlujuca_zmeny>'
```

Vysvetľujúca správa musí obsahovať typ úlohy ktorá sa vykonávala podľa manažmentu správy úloh projektu (*task* alebo *bug*) ako aj číslo úlohy. Ak odosielená skupina zmien ukončuje vykonávanie danej úlohy je nutné pripísať kľúčové slovo *fixes* pred určenie typu úlohy. Týmto je úloha v systéme na manažment správy označená ako vyriešená a čaká na vykonanie procesov testovania a verifikácie. Správa taktiež musí obsahovať stručný ale presný opis vykonaných zmien. Formát výslednej vysvetľujúcej správy je nasledovný:

```
[fixes] <typ_ulohy> #<cislo_ulohy> <sprava>
```

Príklad odoslania zmien:

```
$ svn commit -m 'fixes task #138 vykonany refactoring modulov AgentMove, AgentTrainer'
```



## 6 Diagramy znázorňujúce interakcie procesov

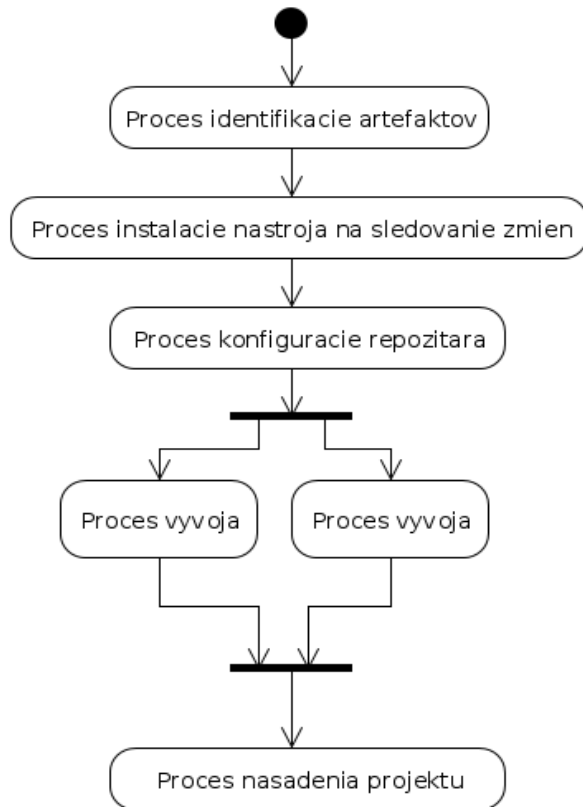


Diagram 1.: Znázornenie procesov manažmentu verzii

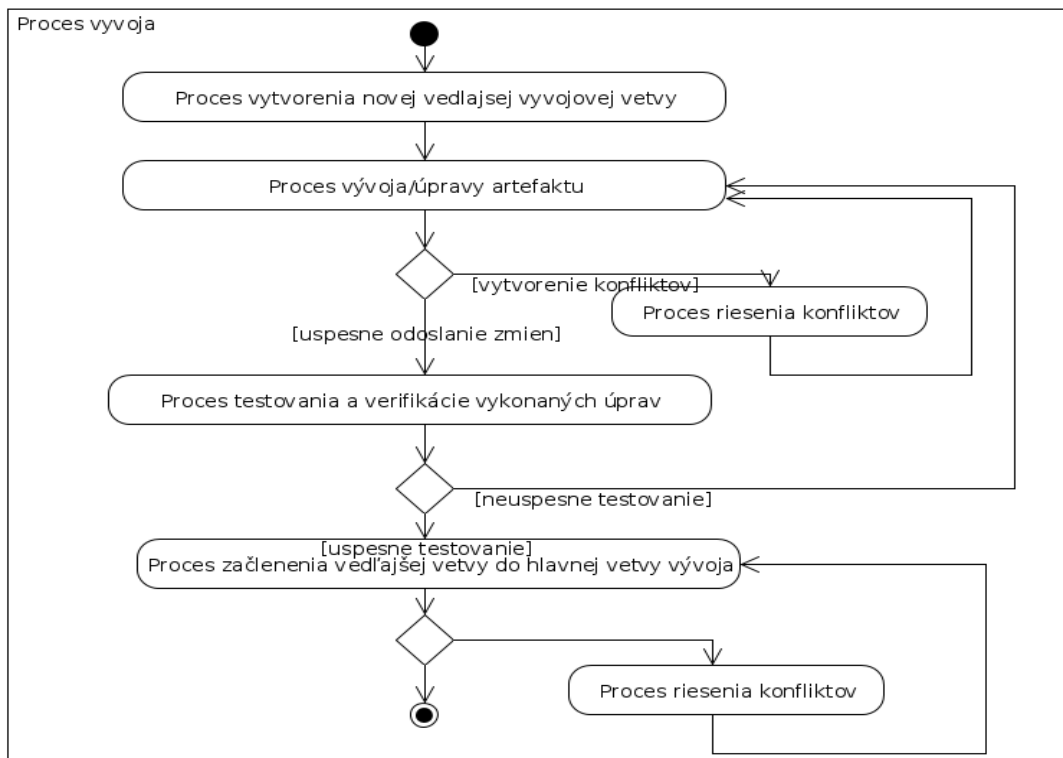


Diagram 2.: Znázornenie podprocesov procesu vývoj artefaktu