



**Konfigurační řízení, vývojové
prostředí, dodávka a údržba**

Tomáš Krátký

tomas.kratky@profinit.cz



Obsah

- Konfigurační řízení
- Vývojové (a jiná) prostředí
- Dodávka
- Údržba





Konfigurační řízení





Prosté cíle CM

- Identifikace elementů a verzí systému.
- Schopnost provádět bezpečně změny v systému, aniž bychom tím ohrozili systém samotný.
- Možnost paralelní práce na více verzích systému.
- Možnost paralelní práce více lidí na téže verzi systému.

! UDRŽET ŘÁD A POŘÁDEK !

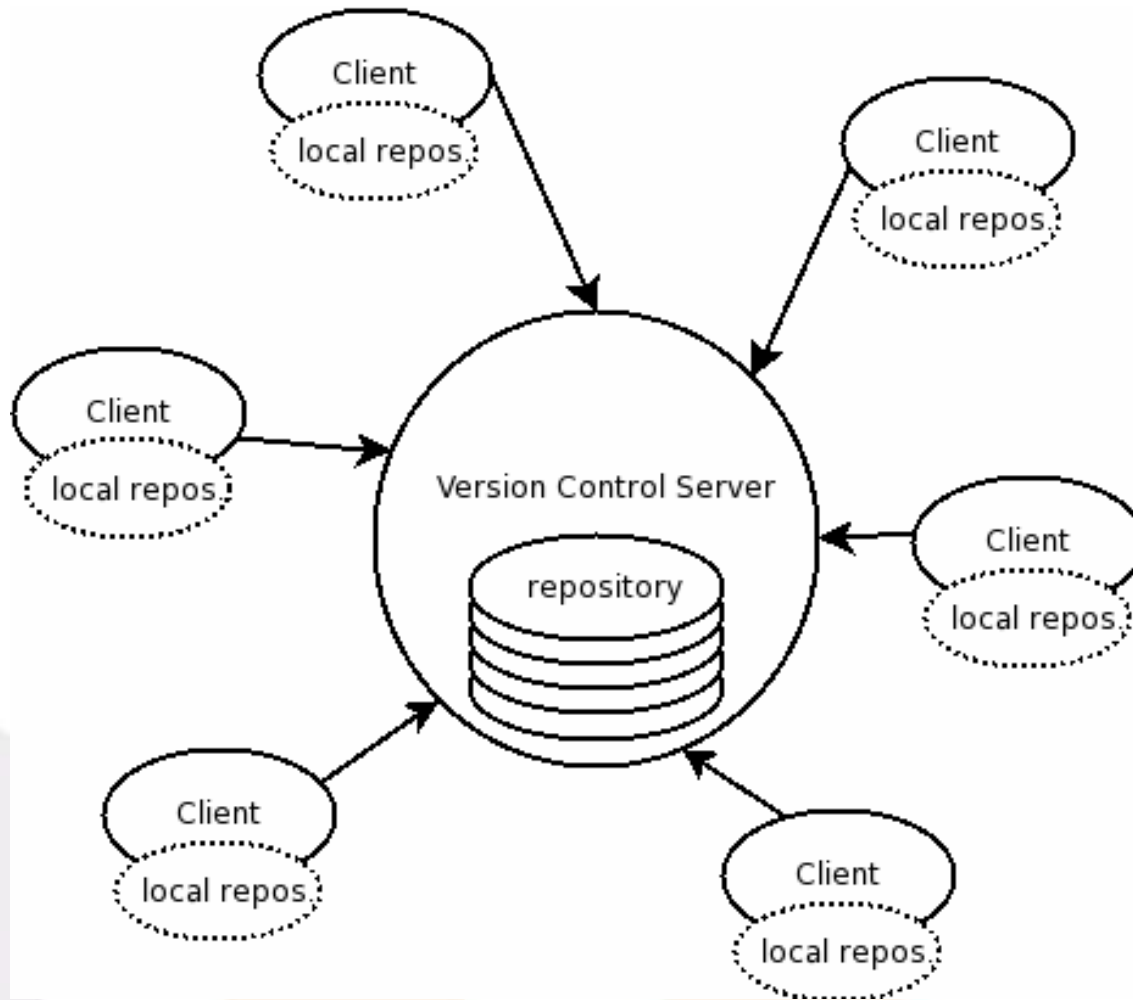


Kontrola verzí

- Identifikace konfiguračních jednotek
 - Typy konfiguračních jednotek v systému ?
 - Které spadají pod kontrolu verzí a které ne ?
 - Co se děje s těmi, které pod kontrolu verzí nespádají ?
- Technická realizace
 - CVS, SVN, ...

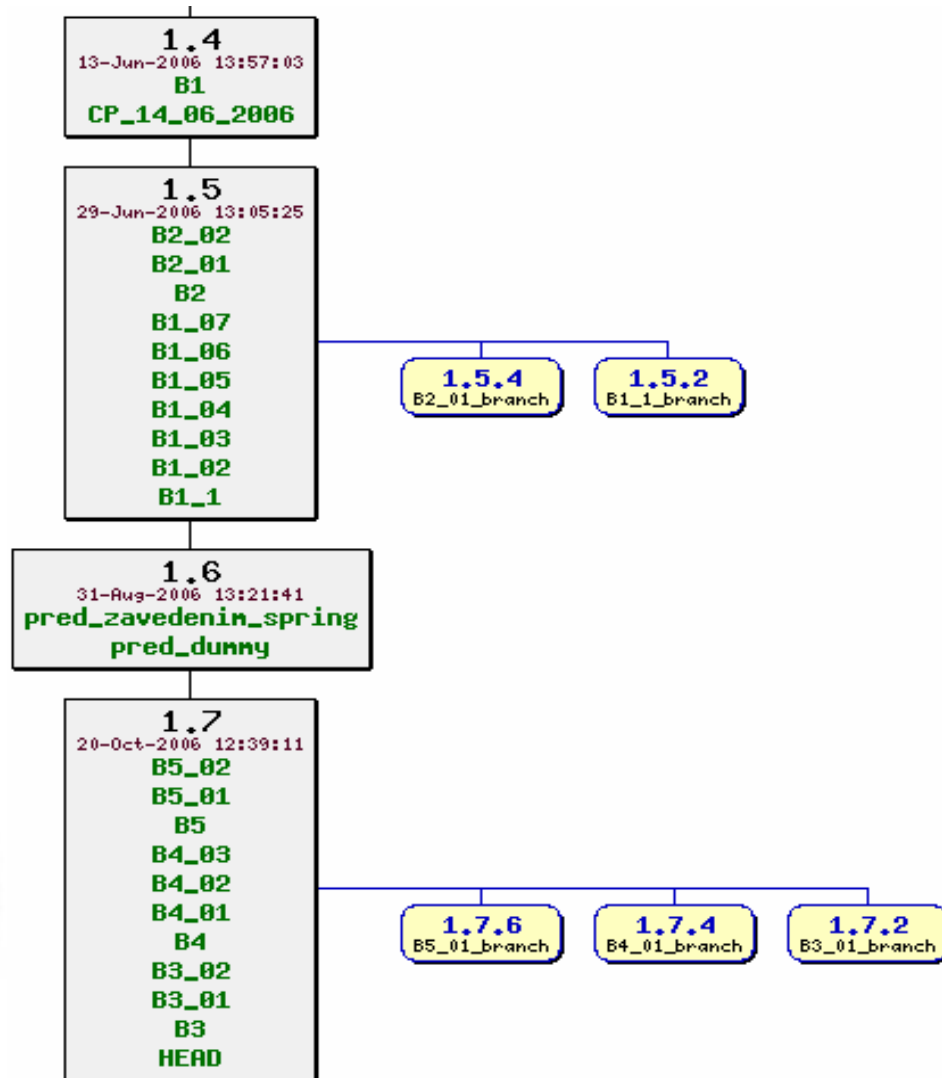
Příklad eSIPO (první část CM + úvodní stránka, kde je jaká verze systému)

Kontrola verzí v praxi



- Centrální repository
- Klienti mají své lokální kopie
- Značení jednotlivých verzí konfiguračních jednotek (jednotlivé elementy vs. napříč repository)
- Možný návrat ke kterékoli verzi dané **jednotky**
- Inteligentní merge

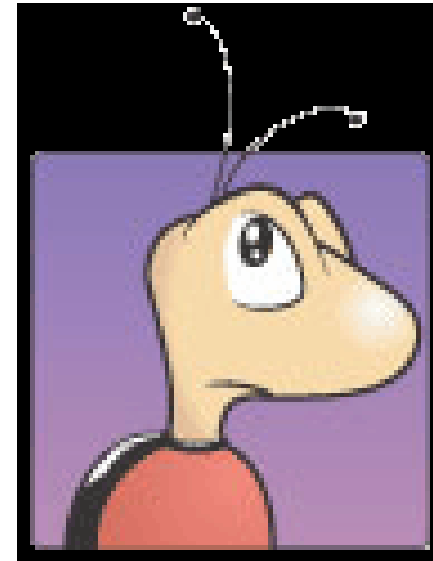
Kontrola verzí v praxi



- *Tag a Branch* jako nástroj pro označení verze systému
- Možnost práce ve více verzích systému současně
- Možnost návratu ke konkrétní **verzi systému**

- Identifikace
 - Jaké typy změn rozlišujeme ?
 - Vazba na specifikaci !
 - Vazba na kontrolu verzí !

Příklad eSIPO (kapitola Řízení změn - Identifikace)



- Technická realizace
 - Bug/Issue tracking systém – [Bugzilla](#), ...
 - Jednotná evidence všech dohod se zákazníkem (nebo i interních)

Příklad eSIPO (kapitoly Struktura až Životní cyklus + bug 6017 – iniciální vložení c#1, komunikace, soukromé poznámky, schválení c#6, log z SVN, uzavření)

Vývojové (a jiná) prostředí

profinit.
PROFESSIONALS IN IT





Hlavní motivace

- Váš systém je určen pro cílové prostředí u zákazníka, ale ...
 - musíte ho někde vyvíjet a testovat
 - před nasazením do produkce musí být někde akceptačně testován
 - ➔ musíte vytvořit několik *kopií* cílového prostředí
- V jednoduchých případech stačí
 - lokální vývojové a testovací prostředí a
 - akceptační prostředí u zákazníka.



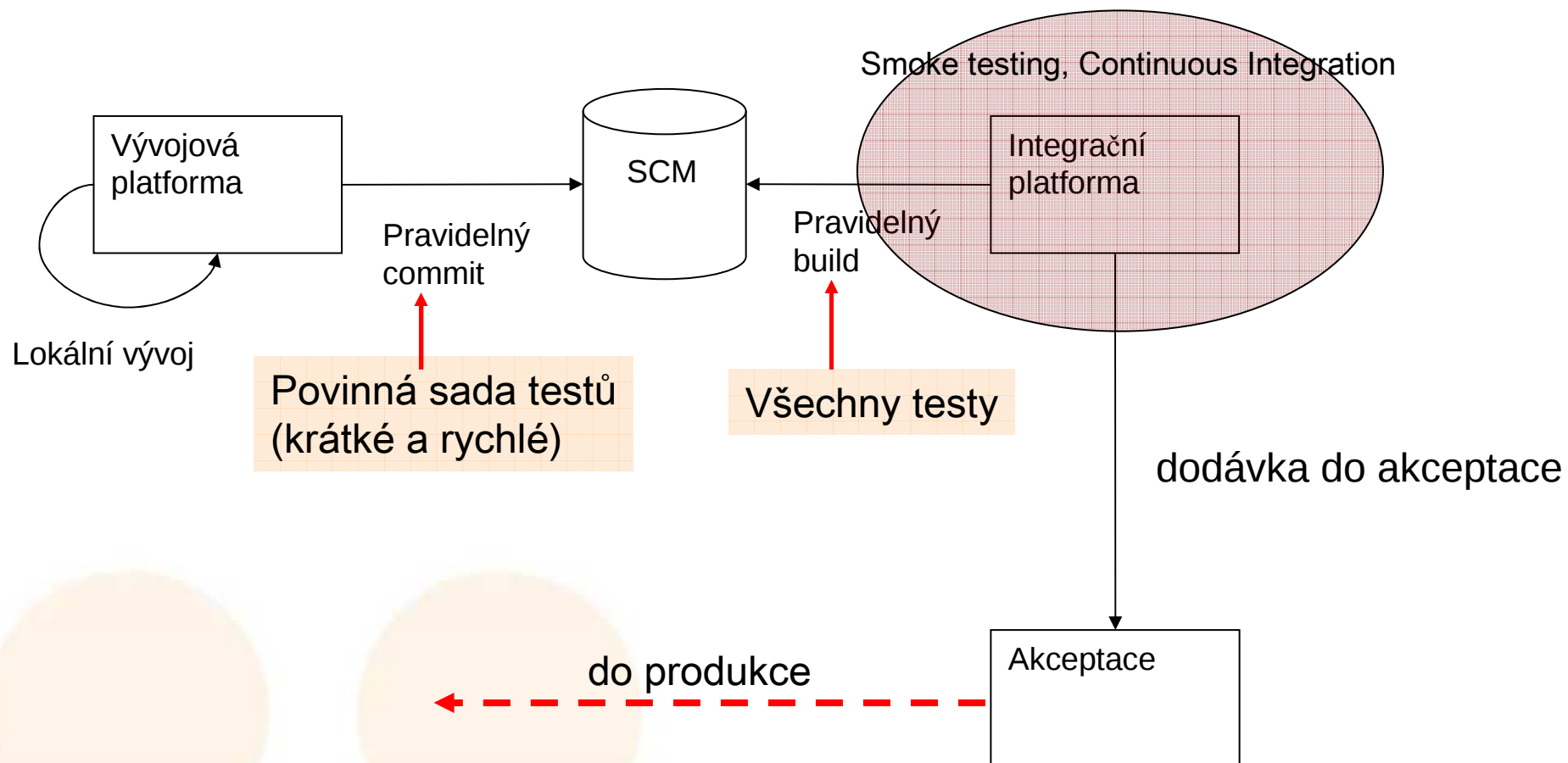
Hlavní úkoly

- Zajistit maximální shodu s produkcí
 - minimalizace rizik
 - často velmi obtížné (OS, parametry HW, nastavení SW, množství dat, ...), přesto je nutné vyvinout maximální úsilí
- Zajistit akceptační prostředí u zákazníka
- Zdokumentovat prostředí zákazníka i lokální prostředí
- Vyřešit zálohování a obnovu vývojového prostředí
 - netriviální
 - opomíjené

Příklad eSIPO (Prostředí na ČP, Vývojové prostředí)



Vývojové prostředí v praxi





Vývojové prostředí v praxi

- Typický projekt v Profinitu
 - SVN jako SCM
 - Ant jako build nástroj
 - CruiseControl jako integrační platforma
 - každodenní build, deploy, test
 - každodenní kontrola zdrojových kódů dostupnými nástroji
 - rychlá zpětná vazba v případě problémů
 - neustálá „integrace“ (Continuous integration)
- Team Foundation Server



Dodávka





Základní otázky

- Udělat dodávku znamená
 - vyrobit dodávku
 - nainstalovat dodávku
- Co když instaluje zákazník ?
 - zkusili jste si někdy připravit váš systém(program) tak, aby jej ostatní dokázali snadno nainstalovat ?
- Malá versus velká dodávka
 - kdy se vyplatí vyrobit patch ?
 - kdy se vyplatí instalace enbloc ?
- Jak často dodávat ?
 - časté dodávky → vysoké nároky na kvalitu procesu výroby a instalace dodávky, neustálá zpětná vazba od zákazníka



Automatizace

- Výroba dodávky musí být maximálně jednoduchá nebo ji nikdy nevyrobíte správně !
- Vyplatí se tuto činnost automatizovat a přesně popsat.

Příklad eSIPO (kapitola Výroba dodávky + skript make_dist.sh)

- Instalace dodávky musí být maximálně jednoduchá, pokud možno automatická !
- Součástí každé dodávky musí být jasný a přesný popis instalace.

Příklad eSIPO (kapitola Instalace dodávky + skript install.sh)



Údržba





O co při údržbě jde ?

- Údržba je jedna z fází životního cyklu projektu
 - aplikace je již předaná a (obvykle) nainstalovaná
 - běžně se používá
 - neobsahuje velké množství chyb
 - neobsahuje závažné chyby
- Servisní smlouva a běžná záruka jsou dvě rozdílné věci !
- Naším cílem je na údržbě v nejhorším případě neprodělat, v lepším případě (hodně) vydělat 😊
 - nutný velmi **efektivní proces**
 - nutné velmi **přesné odhady**



Údržba a organizace práce

- Zákazník generuje požadavky, hlásí chyby
- Dodavatel generuje nabídku (**hlídá rozsah**, odhad pracnosti)
- Zákazník rozhodne co má být realizováno, po vzájemné dohodě se určí plán dodávek
- Neplánovaně se opravují pouze závažné chyby
- Existuje jediné místo, kde je vše evidováno
- Miniwaterfall je při daném rozsahu vhodný



Údržba a organizace práce

- Příklad IS VZ US
 - Servisní smlouva
 - Omezený rozpočet zákazníka
 - Velké množství CHR známo na začátku roku, pouze několik drobností se objevilo v průběhu
 - Zákazník si vybral realizaci takových CHR, které byly důležité a stačily mu na ně peníze
- 



Údržba a testy

- U větších systému je prakticky nemožné po každé změně otestovat ručně celý systém
 - ☹️ komplexní testování se buď ignoruje nebo
 - 😊 existují regresní automatické testy.
- Důležité je správně poznat ty části systému, které budou (by mohly být) změnou ovlivněny
 - dopadová analýza → nutný dobrý přehled o systému → u větších systému velmi obtížné



Údržba a testy

● Příklad SVZ

- Jsou unit a integrační testy, nejsou testy systémové
- Testy poměrně spolehlivě odchyty chyby v DB procedurách a algoritmech, které tvoří jádro systému
- Před každou dodávkou je minimálně nutné ručně projít celou aplikaci a provést několik důležitých regresních testů (cca 4 hodiny) → automatizace by přinesla jistou úsporu



Údržba a dokumentace

- V rámci údržby upravujete systém
 - potenciálně dlouho poté, co byl vytvořen (stačí i pár měsíců)
 - aniž byste byli jeho autory

→ to je nemožné bez kvalitní dokumentace !!

Minimálně potřebujete

- kvalitní specifikaci, abyste uhlídali rozsah
- architekturu a design, abyste je mohli ctít
- kvalitní, jasné a přesné postupy, abyste se jimi mohli řídit



Údržba a dokumentace

- Příklad IS VZ US
 - Při přechodu do režimu údržby velká jednorázová investice do dokumentace. Cílem bylo
 - vytvořit přehlednou a srozumitelnou strukturu,
 - zaznamenat důležité informace a postupy,
 - konvertovat dokumentaci do podoby vyžadované zákazníkem.





Údržba a ... vše ostatní

- Jak je z předchozího vidět, v období údržby je kladen zvýšený důraz na kvalitu prakticky všech činností v rámci SW projektu (při samotném vývoji mnohdy opomíjených – dokumentace, testy, ...).

Je velmi těžké udržet pořádek.
Je velmi snadné polevit.
Je velmi snadné údržbu podcenit.
Je velmi snadné „šlápnout vedle“.





Diskuse

- Komentáře
- Otázky
- Připomínky
- Upřesnění
- Poznámky
- ...

