

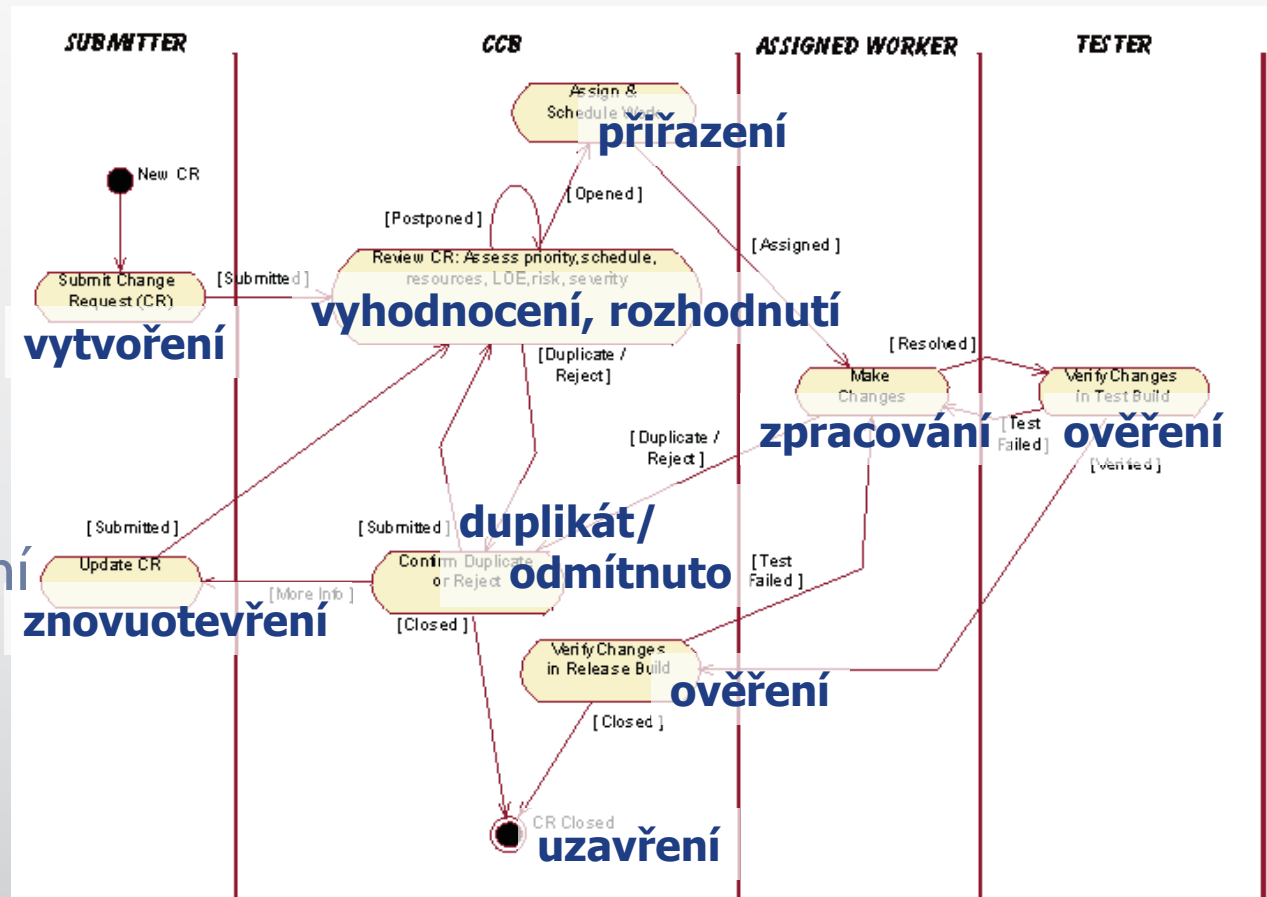
# ► Cyklus správy změn

## ► Požadavek na změnu – stavy

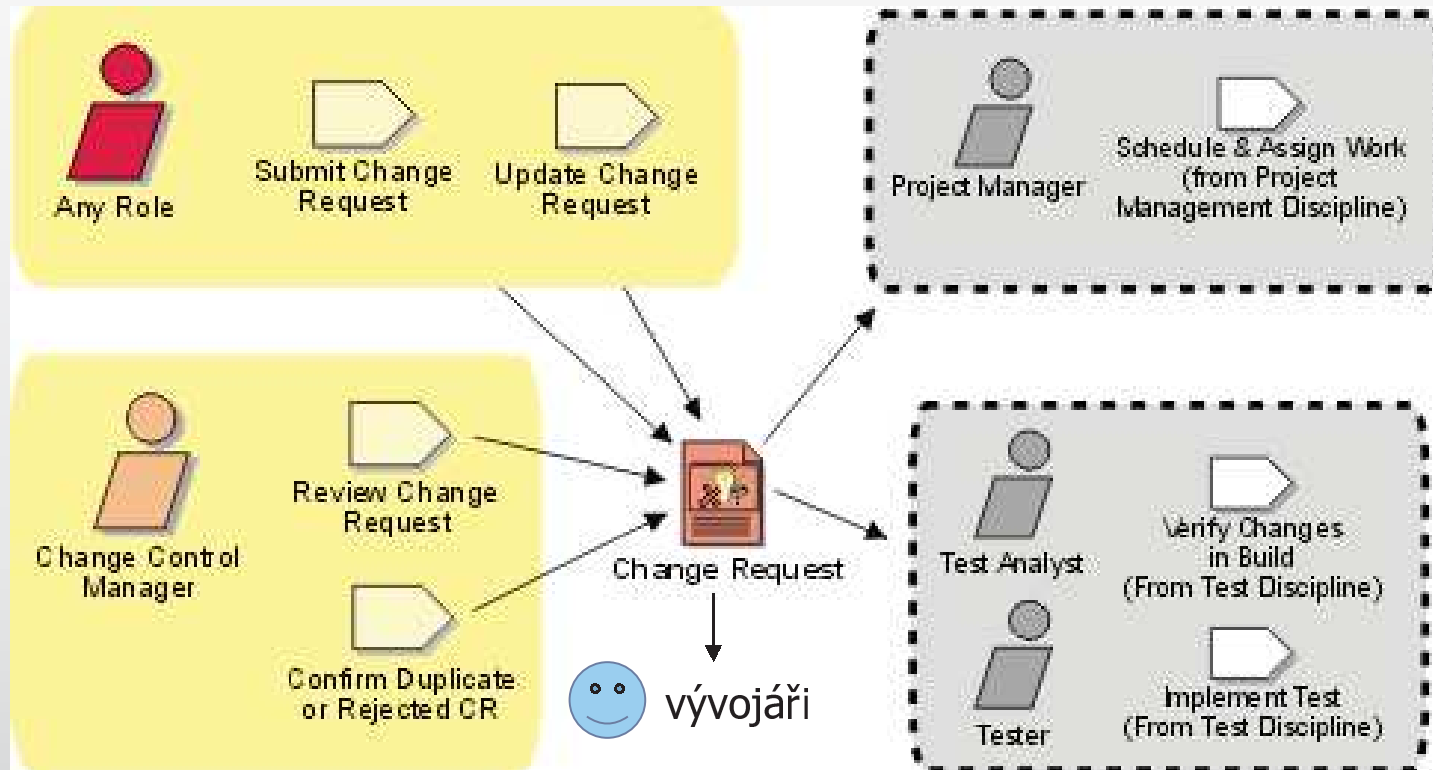
- vytvořený
- schválený
- přiřazený
- vyřešený
- ověřený
- uzavřený
- znovuotevřený
- odmítnutý

## ► Během provádění

- znovuotevření problémů
- vygenerování nových hlášení



# ► Role ve správě změn





# ▶ Určení konkrétní verze prvku

## ▶ Základní druhy verzí

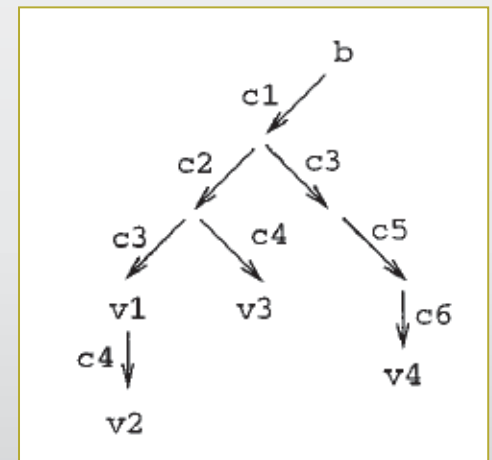
- ▶ historická podoba → **revize** („Word 6.0“)
- ▶ alternativní podoba → **varianta** („Word pro Macintosh“)

## ▶ Verzování podle stavu

- ▶ identifikují se pouze podoby prvků
- ▶ výsledná verze vznikne vhodným výběrem

## ▶ Verzování podle změn

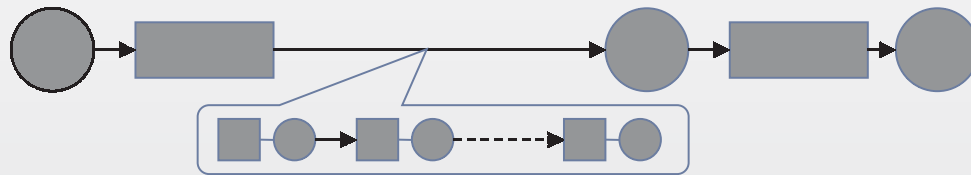
- ▶ identifikují se (také) změny prvků  
= **changeset**
- ▶ výsledná verze vznikne aplikací změn



# ► Terminologie: Codeline



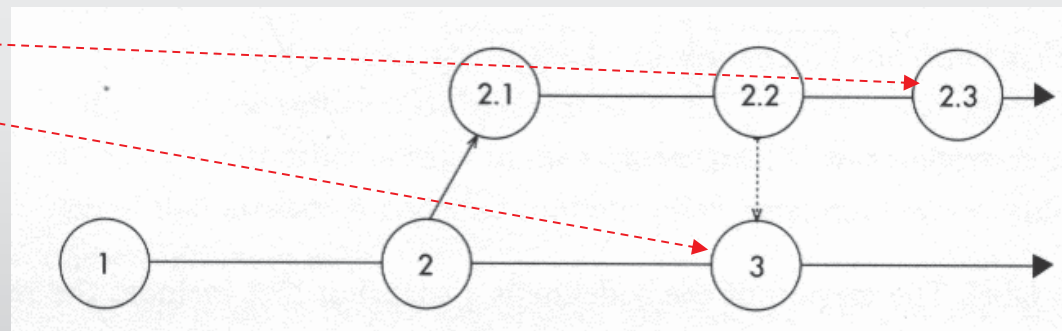
- Codeline (vývojová linie) = série podob (verzí) množiny prvků konfigurace tak, jak se mění v čase
- (extenzionální verzování podle stavu) obsahuje **commity** a/nebo changesety, větve => část grafu verzí



- vrchol codeline obsahuje nejčerstvější verzi

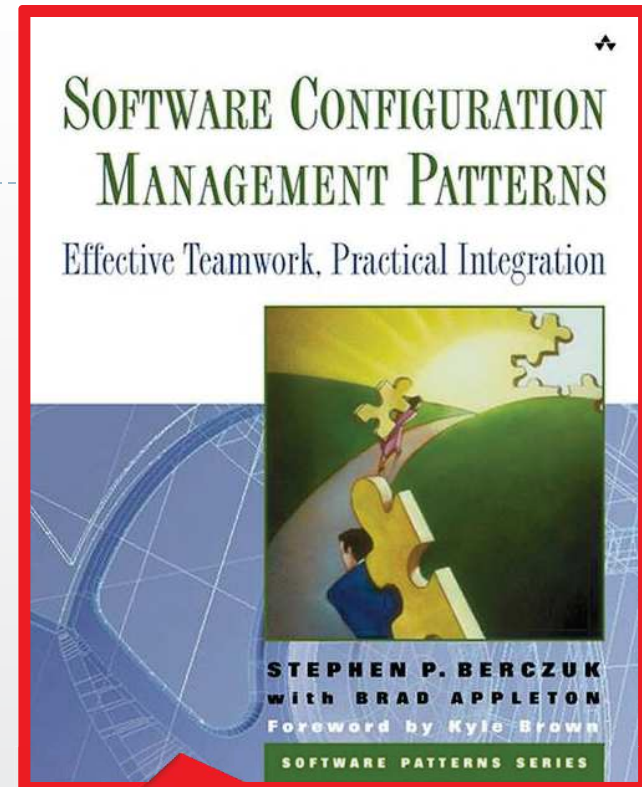
□ “HEAD”, “tip”

- má přiřazena **pravidla** práce na codeline



## ▶ **Vzory pro verzování (I)**

- ▶ Hlavní vývojová linie (*mainline*) a pravidla linie (*codeline policy*)
  - ▶ jedna codeline pro průběžný vývoj
- ▶ Stabilizační období (*code freeze*)
  - ▶ pravidla před release
- ▶ Větev pro release a jeho přípravu (*release line*)
  - ▶ místo code freeze mít samostatnou větev pro release
- ▶ Kód třetích stran na větev (*third party codeline*)
  - ▶ vlastní větev pro každý kód od externího dodavatele



viz doporučené pořadí čtení



## ▶ **Vzory pro verzování (2)**

---

- ▶ Privátní verze (*private versions*)
  - ▶ soukromé úložiště pro častější check-in
- ▶ Větev pro úkol (*task branch*)
  - ▶ složitější úpravy s většími následky dělat na větvi
  - ▶ vazba na správu změn
- ▶ Check-in pro každý úkol (*task-level commit*)
  - ▶ minimum nutného
  - ▶ po ukončení práce na jednom úkolu udělat check-in změn

**A successful Git branching model**  
<http://nvie.com/posts/a-successful-git-branching-model/>





# ▶ Vlastnosti sestavení

---

## ▶ Jedinečnost a identifikovatelnost

- PROJEKT\_v2\_build2134\_20041220T1954

- ▶ identifikátor jednoznačný, čitelný
- ▶ vytvořitelný a zpracovatelný automaticky (schema pro id)

## ▶ Úplnost

- ▶ tvoří kompletní systém, obsahuje všechny komponenty

## ▶ Konzistence

- ▶ vzniklo ze správných verzí správných komponent
  - tj. z konzistentní konfigurace

## ▶ Opakovatelnost

- ▶ možnost opakovat build daného sestavení kdykoli v budoucnu
  - se stejným výsledkem

## ▶ Dodržuje pravidla vývojové linie

- ▶ build odpovídající baseline
- ▶ zejména release má striktní pravidla



# ▶ Integrační sestavení

- ▶ Cíl: spolehlivě ověřit, že produkt jde sestavit
  - ▶ soukromý build nestačí
    - složité závislosti, špecifiká ve workspace, zjednodušení pro zrychlení
  - ▶ úplné sestavení trvá dlouho => nemůže provádět vývojář
- ▶ Postup: celý produkt (vč. závislostí) sestaven centrálně, automatizovaným a opakovatelným procesem
  - ▶ postup co nejpodobnější sestavení pro release
    - vždy „na zelené louce“ (clean full build)
  - ▶ maximální automatizace - typicky běží přes noc
  - ▶ spolehlivé mechanismy zaznamenání chyb a informování o nich
    - emailové notifikace začátek, konec, výsledek
    - web s přehledy a detaily
  - ▶ úspěšné sestavení může být označkováno ve verzovacím systému

Microsoft Windows NT 3.0 consisted of 5.6 million lines of code spread across 40,000 source files. A complete build took as many as 19 hours on several machines, but the NT development team still managed to build every day.  
– McConnell, 1996





# ▶ **Daily Build** and Smoke test



- ▶ Integrační sestavení + zkouška těsnosti
  - ▶ pravidelně 1x denně (někdy nočně)
  - ▶ výsledky okamžitě známy a reflektovány
    - nová hlášení problémů
    - opravy ihned zapracovány do kódu
  - ▶ check-in kódu, který vede k chybám, je neslušné chování
    - lehká (nebo i vážnější) sankce vhodná
- ▶ Výhody
  - ▶ malé množství změn během denních check-in

=> zvladatelné množství oprav, **včas detekce problémů „vždyt’ včera to fungovalo“**, **analýza změn kódu místo ladění – viz diskuse o sestavení**

  - ▶ pravidelný, obecně známý rytmus projektu
  - ▶ lepší morálka týmu („to nám to roste“)
- ▶ Cena: trocha disciplíny, trocha automatizace

