

Distributed Version Control Systems

Paweł Pacana
<pawel.pacana@starware.com.pl>



Systemy Kontroli Wersji

- lokalne
 - RCS (1980)
- centralne
 - CVS (1990)
 - SVN (2000)
- rozproszone
 - GNU Arch (2001)
 - Monotone (2003)
 - Darcs (2004)
 - Baazar (2005)
 - Git (2005)
 - Mercurial (2005)

Subversion

- dzielone zdalne repozytorium
 - single point of failure
 - wymagany aktywny dostęp sieciowy
 - jak podzielić się zmianami z pominięciem centralnego serwera?
- operacje sieciowe
 - niezbędne
 - powolne
- branching – łatwo utworzyć, ale...
 - bolesny merge
 - ręczne śledzenie mergowanych rewizji
 - kosztowny - ogranicza eksperymentowanie
- zapis (commit) = publikacja

Subversion – Merge (1)

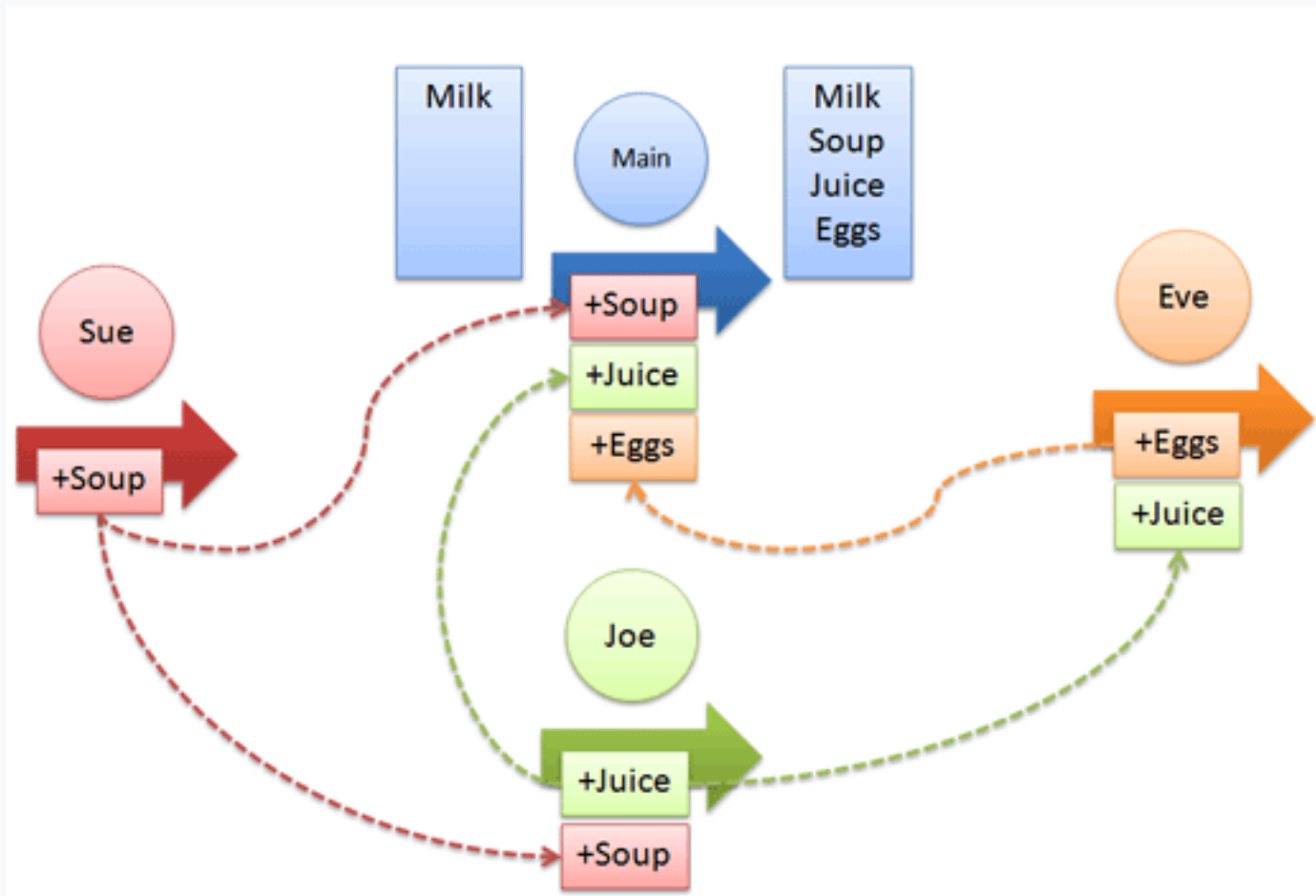
*„The bottom line is that Subversion’s merge-tracking feature has an **extremely complex internal implementation**, and the `svn:mergeinfo` property is the only window the user has into the machinery. Because the feature is **relatively new**, a numbers of **edge cases** and possible **unexpected behaviors** may pop up.”*

– Version Control with Subversion

Subversion – Merge (2)

- ręczne śledzenie rewizji
 - łatwo o pomyłkę
- merge po zmianie nazw
- svn update
 - niejawny merge
 - konflikty w nieskomitowanym working copy
 - jak wrócić do poprzedniego stanu
 - wymuszanie natychmiastowego rozwiązania

DVCS – Distributed VCS

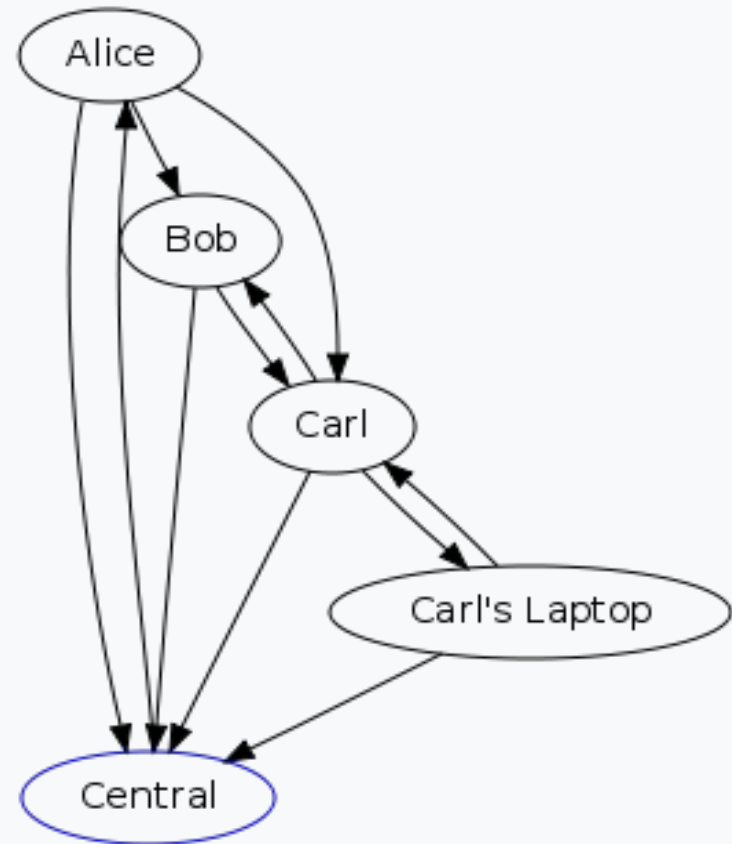


DVCS - Zalety

- rozproszony model
 - dowolny workflow
 - wymiana zmianami w dowolnych konfiguracjach
 - każde lokalne repozytorium to pełna kopia
 - tyle backupów ile deweloperów
- operacje lokalne do ekstsremum
 - sieć nie jest wymagana
 - najczęstsze operacje szybkie
- większa granularność
 - nowe operacje:
 - push
 - commit kiedy chcesz
 - push kiedy gotowe
 - pull
 - zaciąga zmiany z zewnątrz, nie implikuje merge!
 - tworzy osobną gałąź od ostatniego wspólnego changesetu

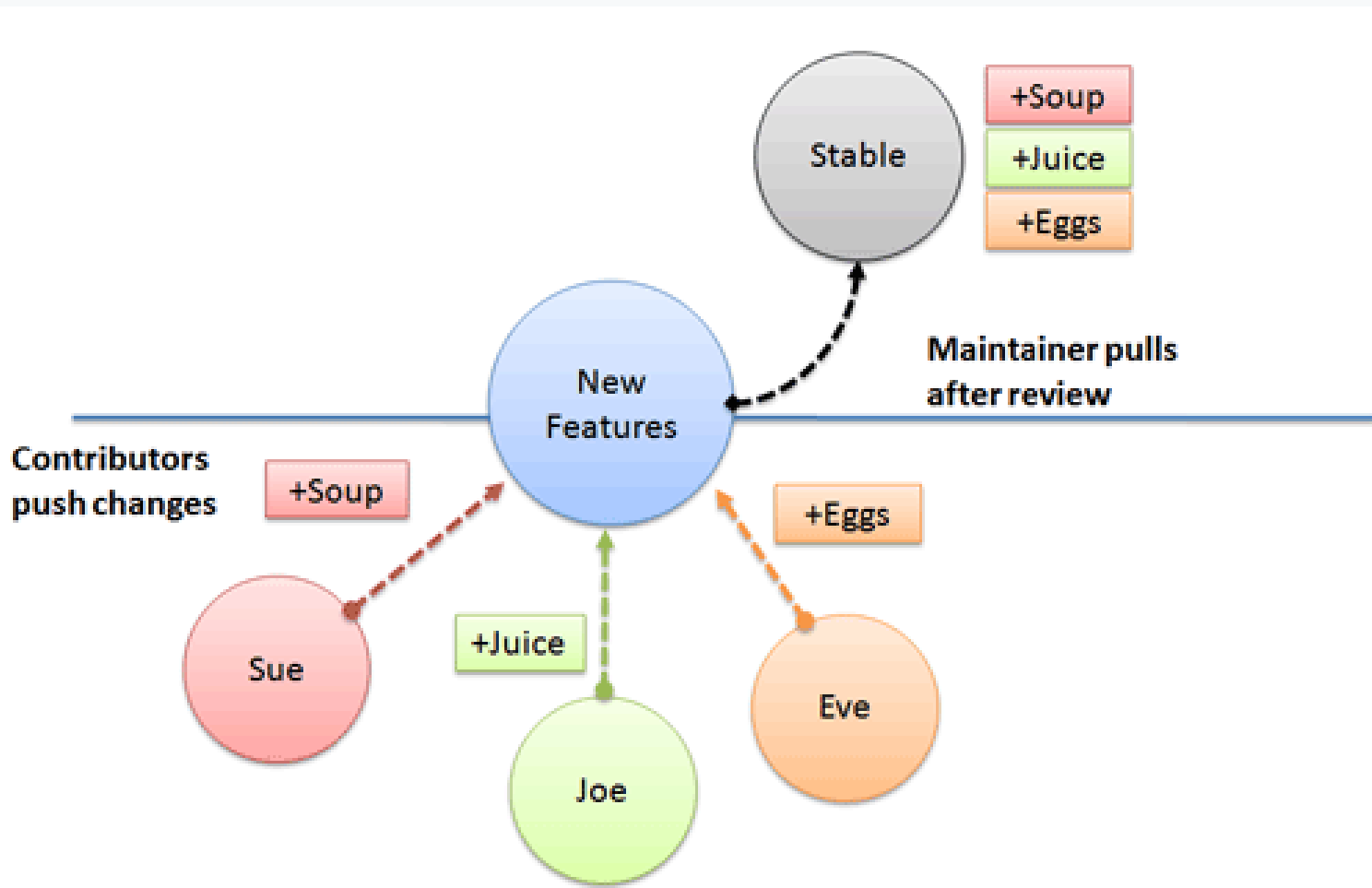
DVCS - Workflows

- lone developer
- shared push
 - jak centralny, ale bez ograniczeń
- pull only
 - hierarchie (Linux Kernel)
- ...



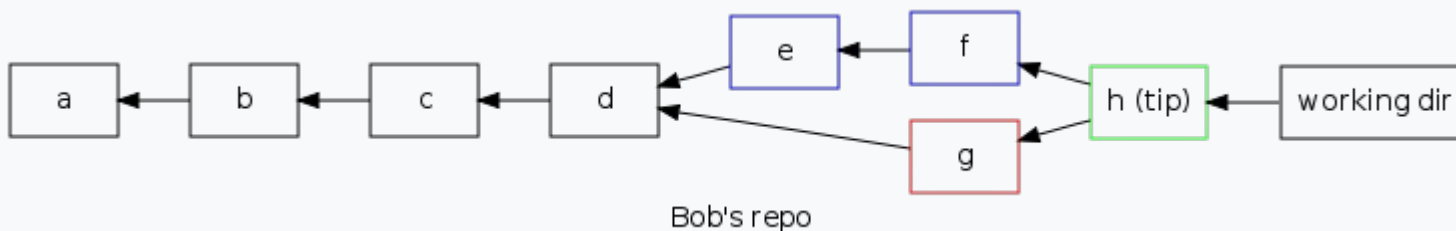
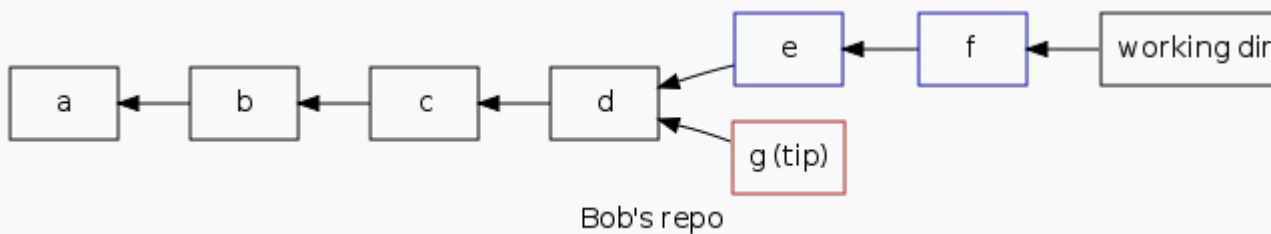
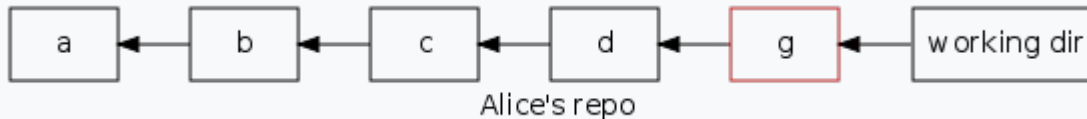
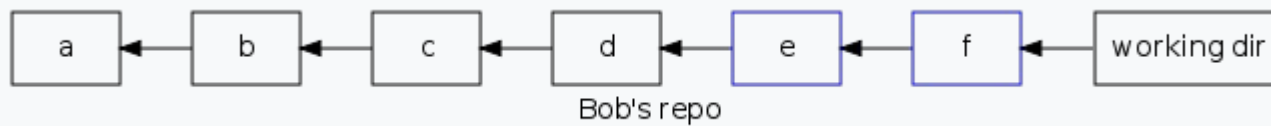
A Mercurial Network

DVCS – Push/Pull



DVCS – DAG

- historia zmian jako Directed Acyclic Graph



DVCS - Merge

- branch & merge
 - codzienne operacje w DVCS
 - muszą być proste
 - zachęcają do eksperymentowania
 - pull zaciąga zmiany ale ich nie merguje
 - zwykle tworzy nową gałąź
 - jawny merge łączy gałęzie
 - może powodować konflikty
 - łatwo się wycofać
 - merge nie łączy niegotowych zmian*
- rozwiązywanie konfliktów
 - no silver bullet

DVCS – Dobre Obyczaje

- Sync Early, Sync Often
 - częsty pull = mniejsza szansa na trudny merge
 - sporadyczny push = codebomb!
- Merge on your side
 - wiesz **najlepiej** jak zrobić merge na **twoim** kodzie
- Nie zmieniaj historii gdy jest publiczna!
 - zmiany wypuszczone w świat są **nie do cofnięcia**
 - możliwe duplikacje

DVCS – Wady

- duże pliki binarne
- partial clone
 - pobieranie wąskiego podrzewa repozytorium
- shallow clone
 - pobieranie tylko ostatniej historii repozytorium
- holy war
 - Git vs. Mercurial

DVCS – Happy End

Pytania?

Paweł Pacana
<pawel.pacana@starware.com.pl>

