

Zarządzanie konfiguracją

Zarządzanie konfiguracją oprogramowania to zestaw czynności pozwalających kontrolować zmiany. Robione to jest poprzez identyfikację elementów, które mogą się zmieniać, ustalenie relacji pomiędzy nimi, określenie mechanizmów zarządzania wersjami.

Problemy

- Różnorodność artefaktów
 - specyfikacja wymagań
 - kod źródłowy
 - testy jednostkowe
 - prototyp
 - projekt
- Równoległa praca np. moduły: główny (jądro), drukowanie, poprawność pisowni, wyświetlanie dokumentu; także równoległa praca nad wersjami (już opublikowaną i nową)

Problemy

- Wiele wersji artefaktów
 - Identyfikacja wersji, wersja produktu a wersje artefaktów
 - Wykaz zmian pomiędzy wersjami, identyfikacja autorów zmian
 - Poszczególne artefakty nie są zmieniane synchronicznie

Narzędzia

- Darmowe
 - CVS (Concurrent Versions System), np.: FreeBSD
 - Subversion, np.: Apache, KDE, GNOME, gcc, Python
- Komercyjne
 - Microsoft SourceSafe
- Oparte o centralne repozytorium (architektura klient-serwer)
- Użytkownicy posiadają lokalne kopie artefaktów; synchronizacja na żądanie

Operacje

- Pobieranie artefaktów (checkout)
 - Wykonywane raz, na początku pracy
- Aktualizacja (update)
 - plików, katalogów, całego repozytorium
- Wysłanie zmian (commit)
 - Warto dodać komentarz do operacji
- Kolejne wersje: 1.1 => 1.2 => 1.3 ...

Operacje

- Co się dzieje w przypadku równoległej pracy nad artefaktem?
 - „up-to-date check failed!”
 - Scalanie zmian
 - Zmiany nie pokrywają się – łączymy, co może prowadzić do błędów
 - Zmiany pokrywają się – użytkownik musi zdecydować:

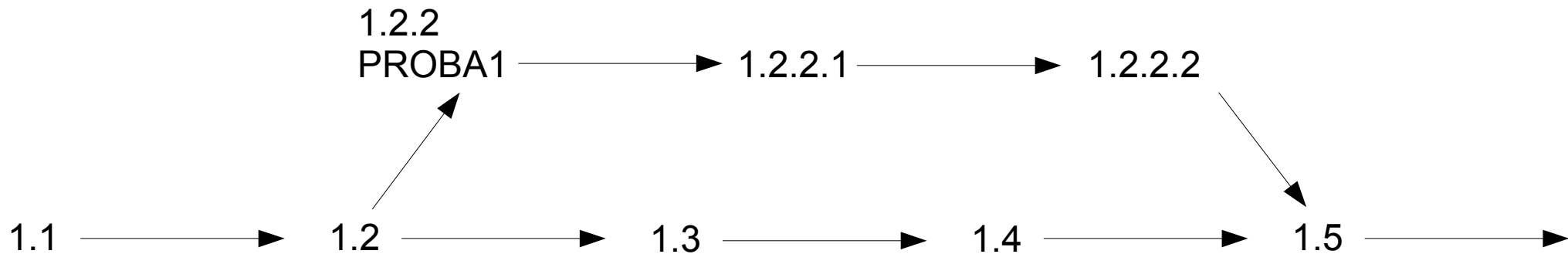
```
IF( a>0 )
<<<<<<< modul1.c
    a += 4;
=====
    a += 2;
>>>>>>> 1.6
ELSE
...
```

Etykiety (tags)

- Oznaczają zestawy plików, które wchodzą w skład pewnego wydania oprogramowania (snapshot).
- Łatwiejsze do zapamiętania, np. `RELEASE_1.0`

Gałęzie

- Utworzenie gałęzi (branch)
- Łączenie gałęzi (merge)



Przykładowa struktura plików

- bin - kod wynikowy – tylko lokalnie
- doc
 - design - UML
- images
- lib
- src
- tests

Dobre wzorce

- Główna linia – główne prace na jednej linii, wszystkie gałęzie docelowo scalamy
- Linia wydania – tworzenie nowej funkcjonalności (na początku niestabilnej) w głównej gałęzi, a jednocześnie poprawianie błędów w wydanej wersji systemu
- Gałęzie dla zadań – oddzielne gałęzie dla większych modułów