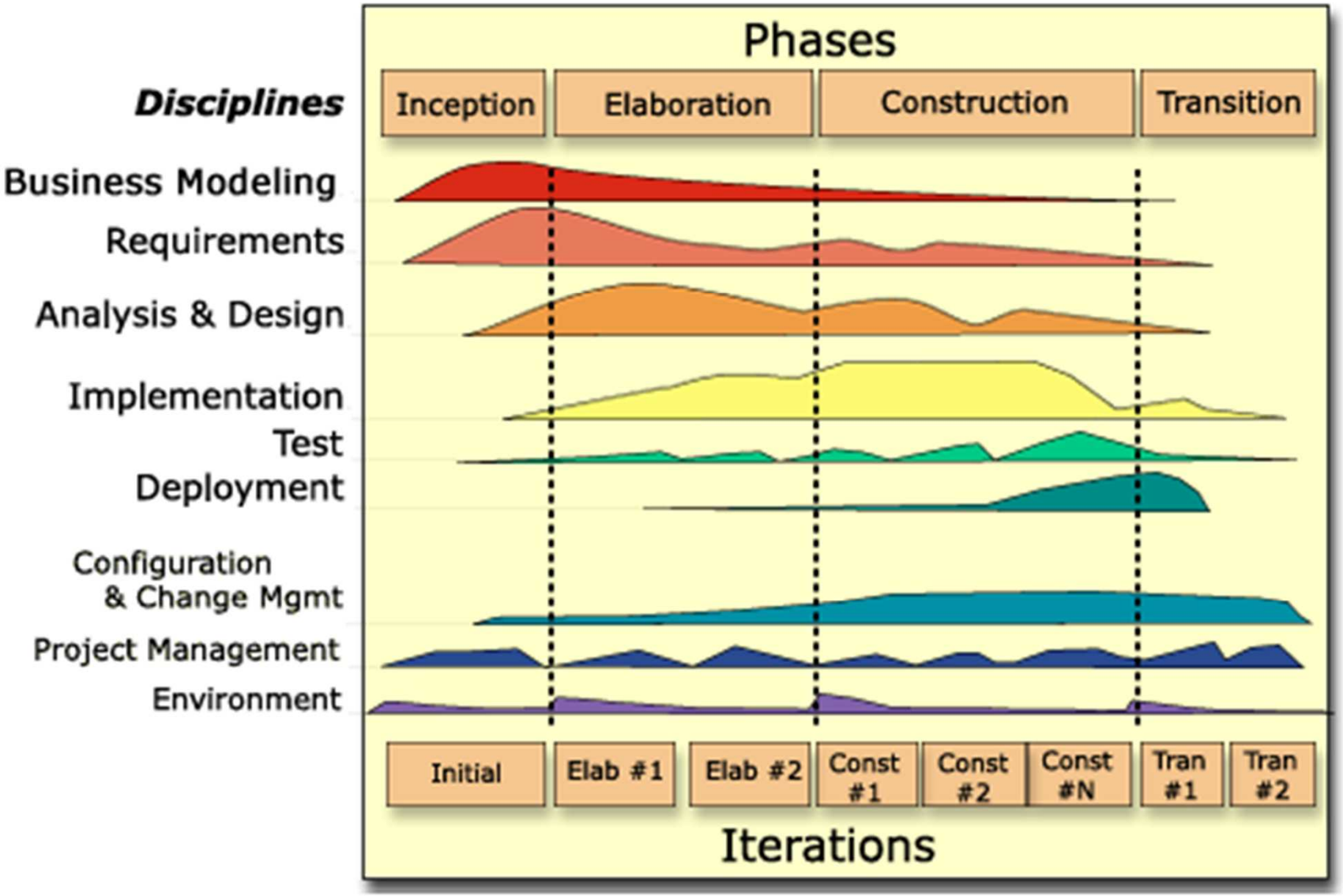
Inżynieria Oprogramowania

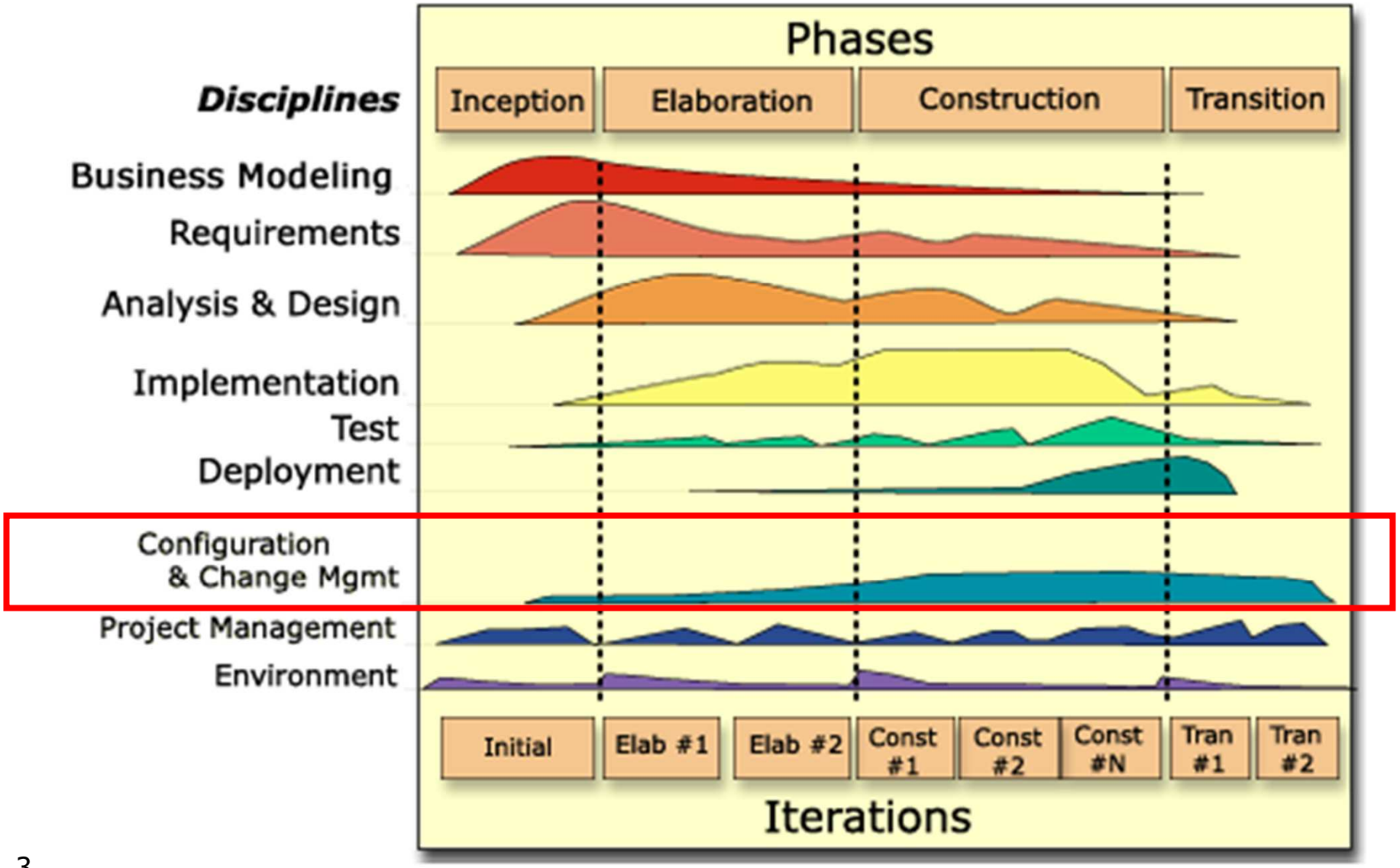
Zarządzanie konfiguracją

Wydział Matematyki, Informatyki i Mechaniki
Uniwersytet Warszawski
www.mimuw.edu.pl/~dabrowski

Rational Unified Process



Rational Unified Process

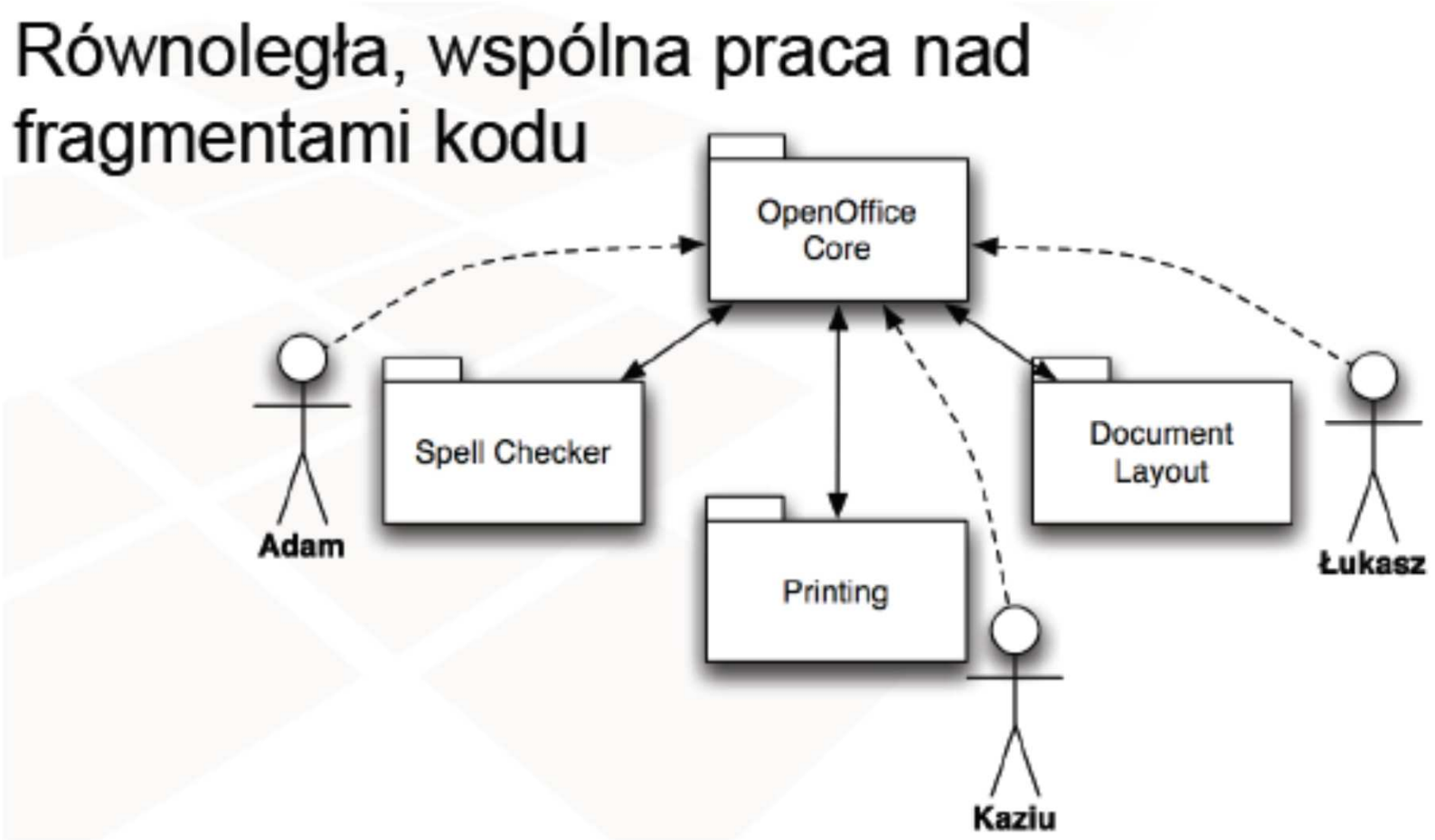


Jakie produkty powstają przy produkcji oprogramowania?

- Specyfikacja (funkcjonalna, pozafunkcjonalna)
- Projekt, model
- Prototyp
- Kod źródłowy
- Testy (jednostkowe, użytkownika)
- Wydania (binaria)
- Instrukcja instalacji
- Lista zmian (w stosunku do wydania poprzedniego)
- Lista znanych błędów
- Podręczniki (użytkownika, administratora)
- Harmonogram
- Notatki robocze zespołu
- …

- Różne typy/formaty!

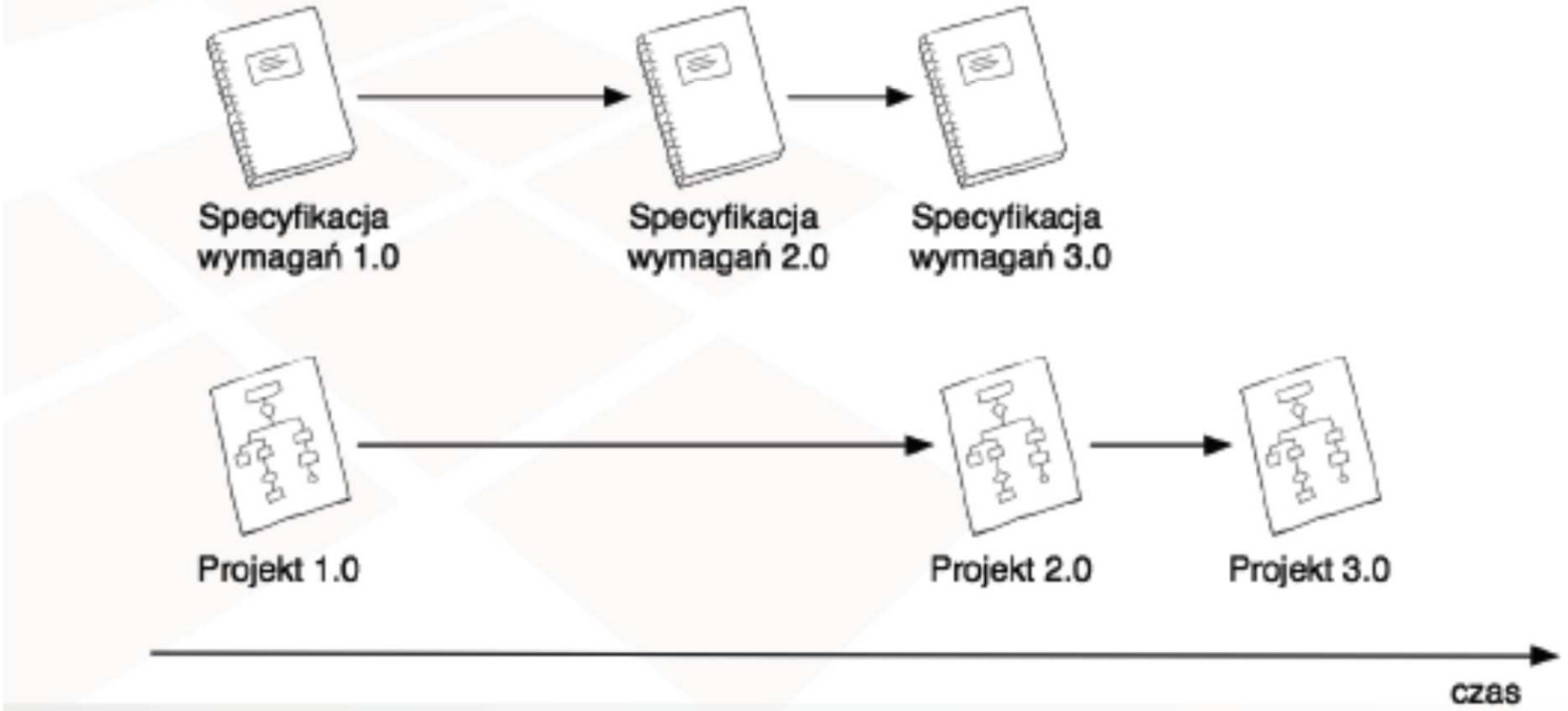
Co jest wyzwaniem?



Co jest wyzwaniem?

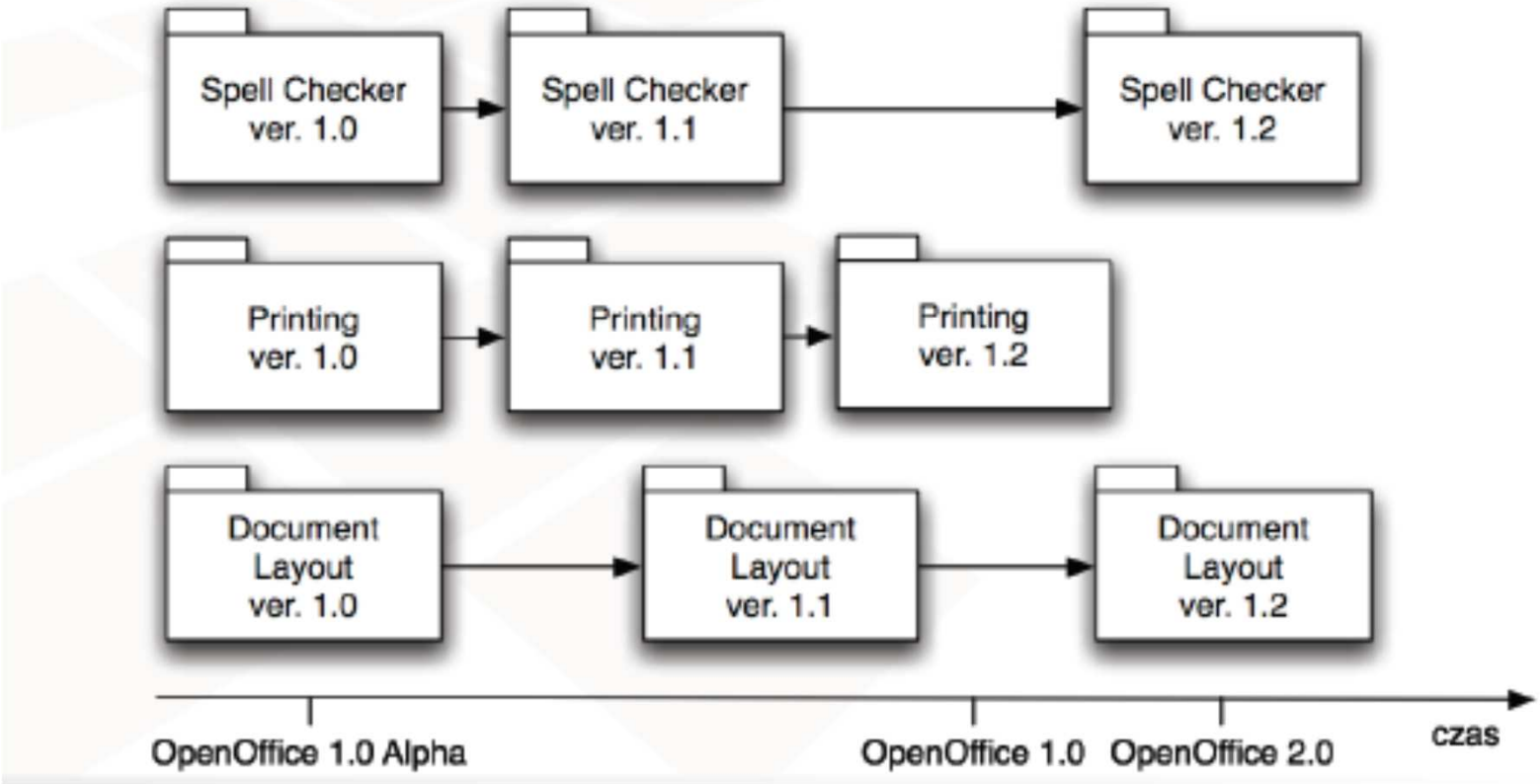
Wiele wersji artefaktów:

– identyfikacja, przechowywanie



Co jest wyzwaniem?

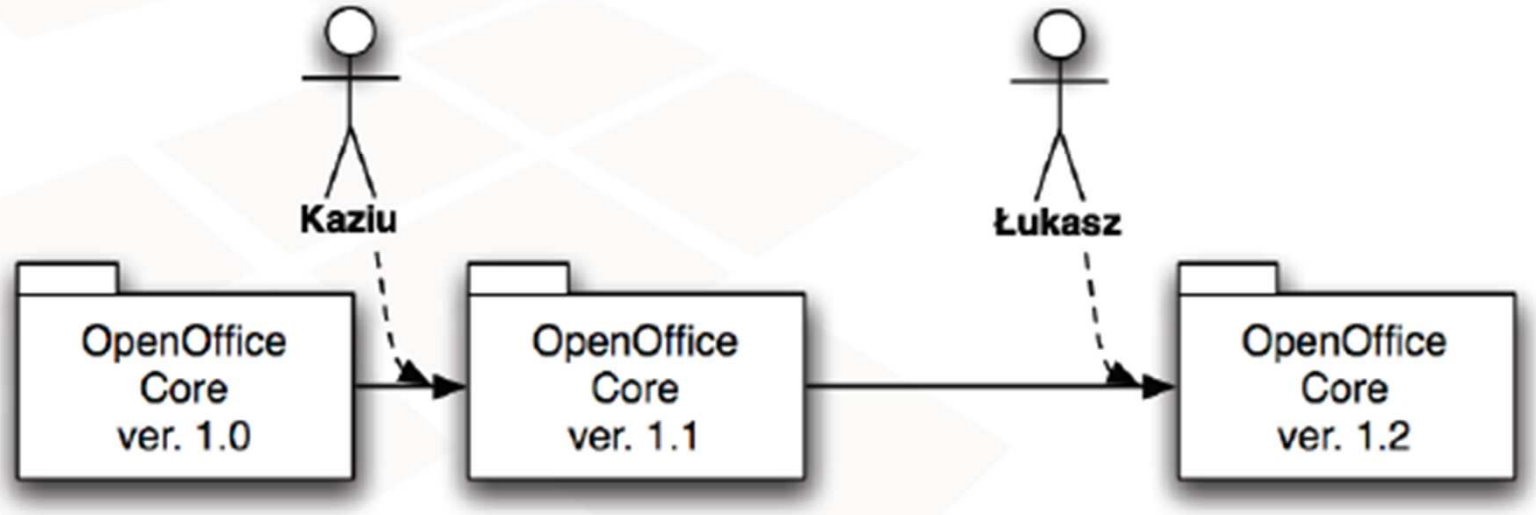
Wersje artefaktów, a wersje produktu



Co jest wyzwaniem?

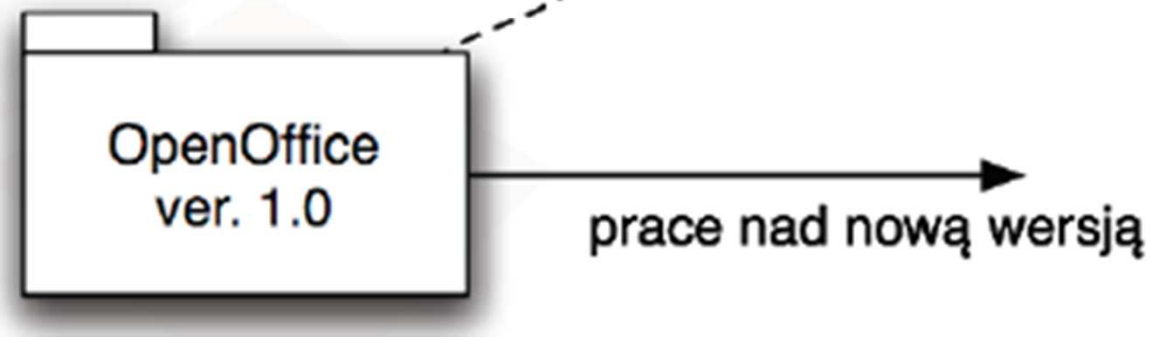
Analizowanie historii:

– Kto? Kiedy? Jaka zmiana?



Co jest wyzwaniem?

Praca nad nową wersją systemu i poprawianie błędów w starej



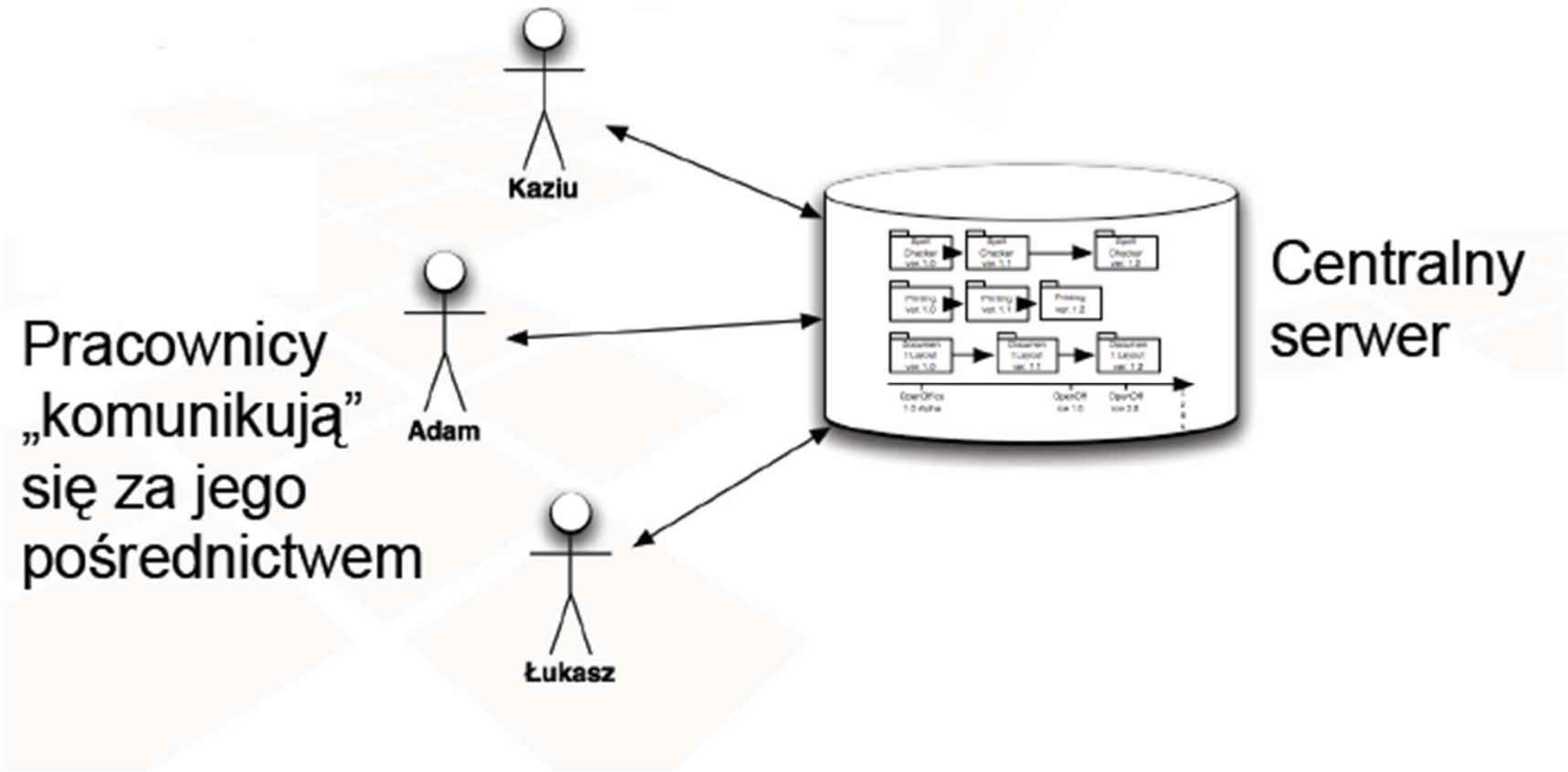
Czego oczekujemy od systemu zarządzania konfiguracją?

1. Każdy komponent oprogramowania jest jednoznacznie identyfikowany
2. Oprogramowanie jest zbudowane ze spójnego zestawu komponentów
3. Zawsze wiadomo, która wersja komponentu oprogramowania jest najnowsza
4. Zawsze wiadomo, która wersja dokumentacji pasuje do której wersji komponentu oprogramowania
5. Komponenty oprogramowania są zawsze łatwo dostępne
6. Komponenty oprogramowania nie zostaną utracone
   - np. wskutek nieuwagi użytkownika
7. Zmiany oprogramowania nie zaginą
   - np. wskutek jednoczesnych sprzecznych aktualizacji
8. Każda zmiana oprogramowania jest zatwierdzona i udokumentowana
9. Zawsze istnieje możliwość powrotu do poprzedniej wersji
10. Historia zmian jest przechowywana
    - np. możliwe odtworzenie kto i kiedy zrobił zmianę, i jaką zmianę

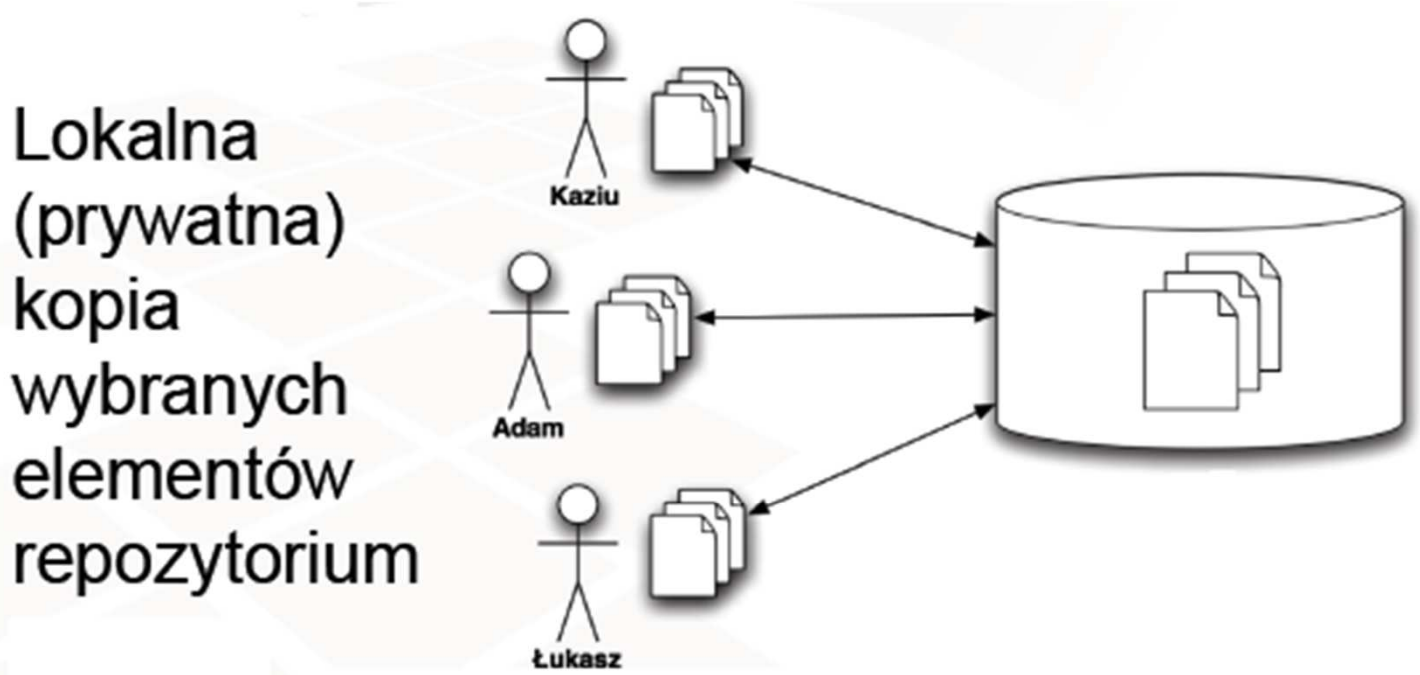
Jakie działania w tym celu realizujemy?

- Kontrola wersji
- Zarządzanie zmianą
- Budowanie wydań

Kontrola wersji

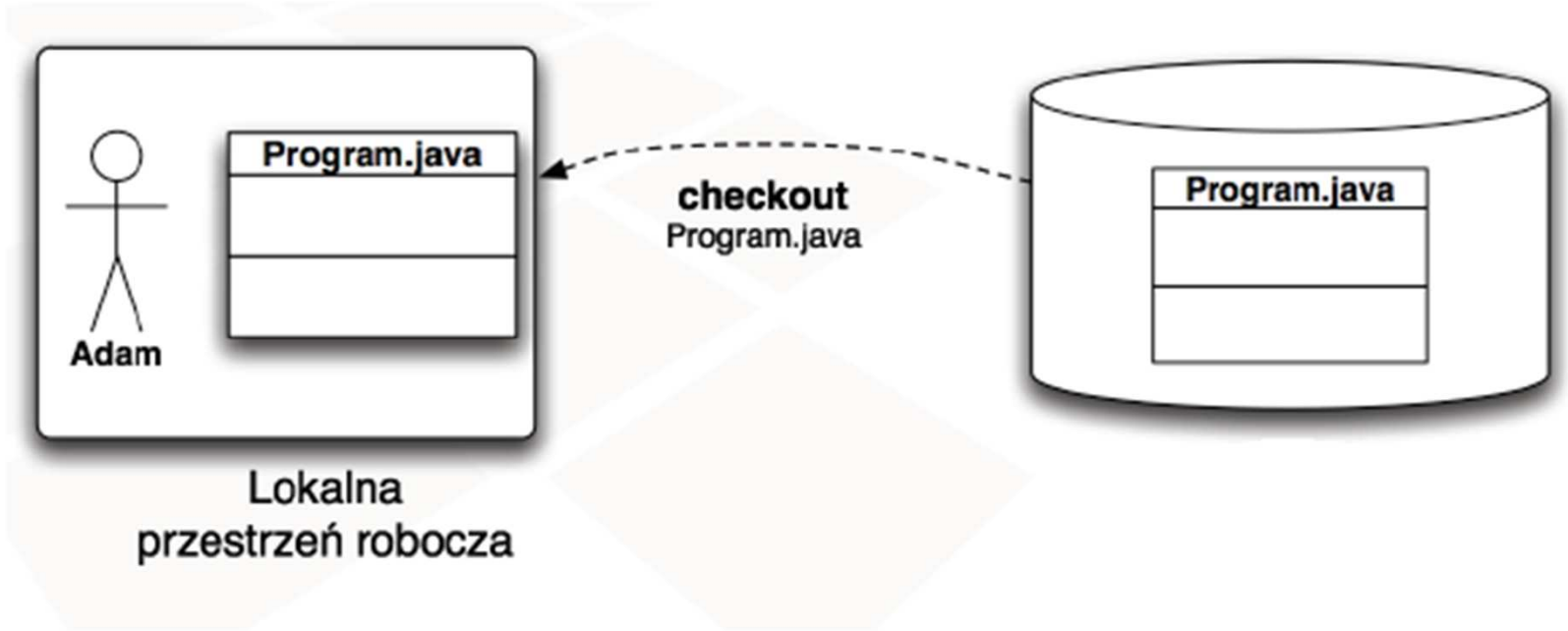


Kontrola wersji

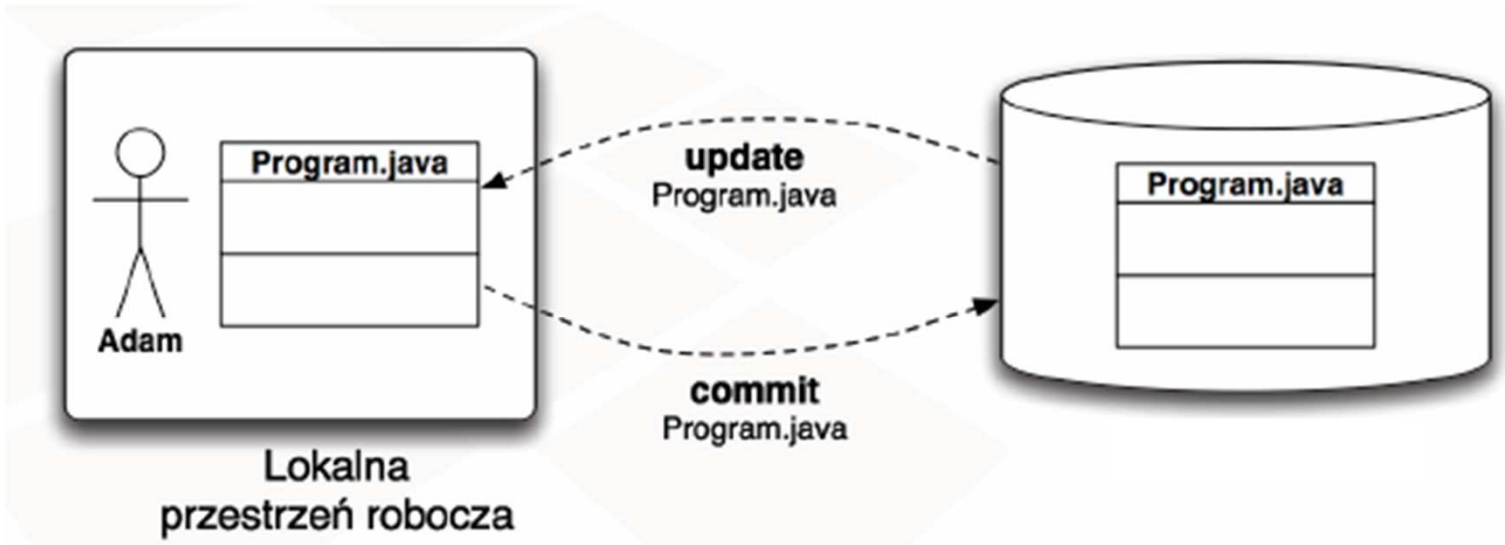


Zmiany
wprowadzane lokalnie
synchronizowane na żądanie

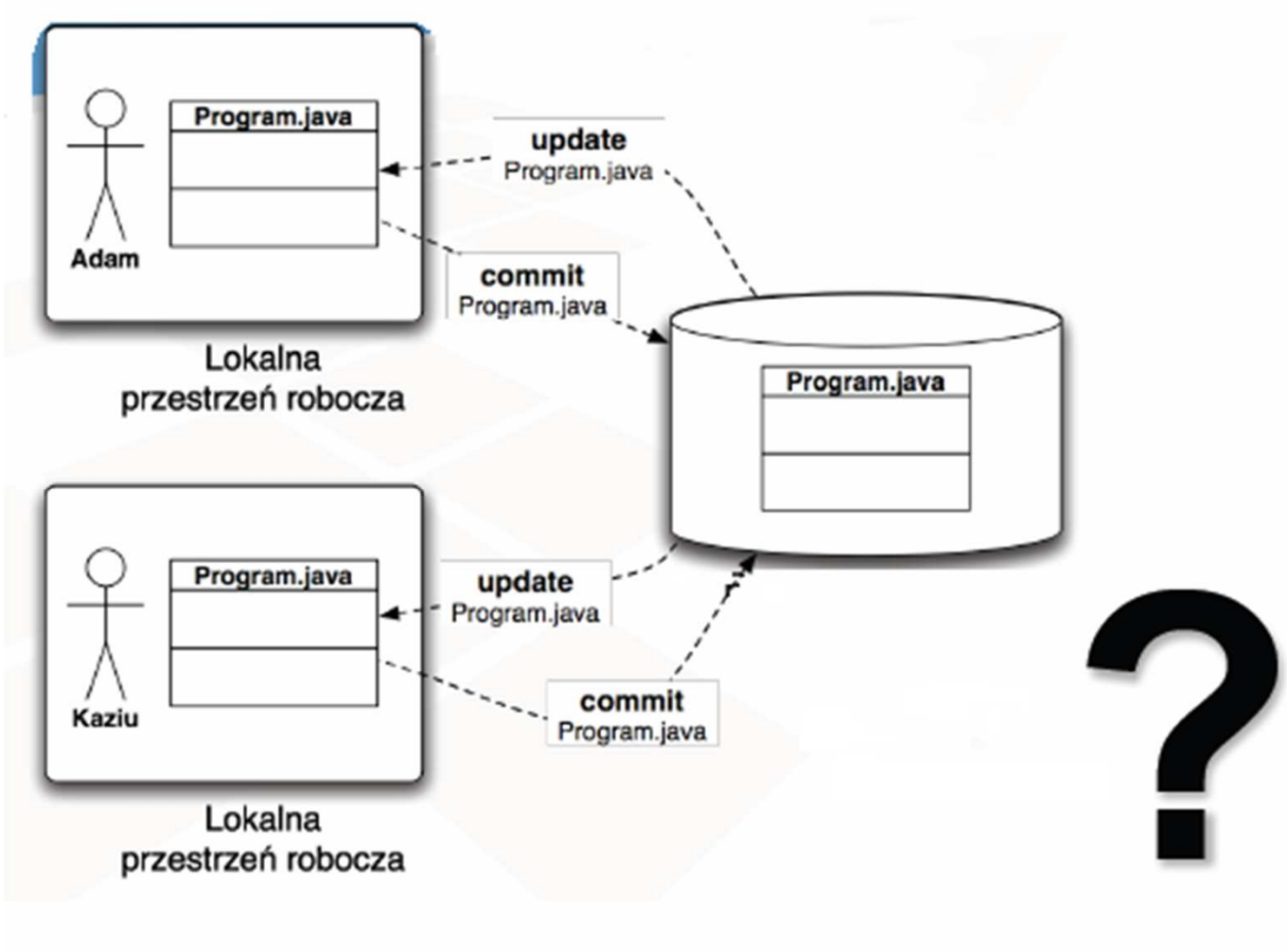
Kontrola wersji – „checkout”



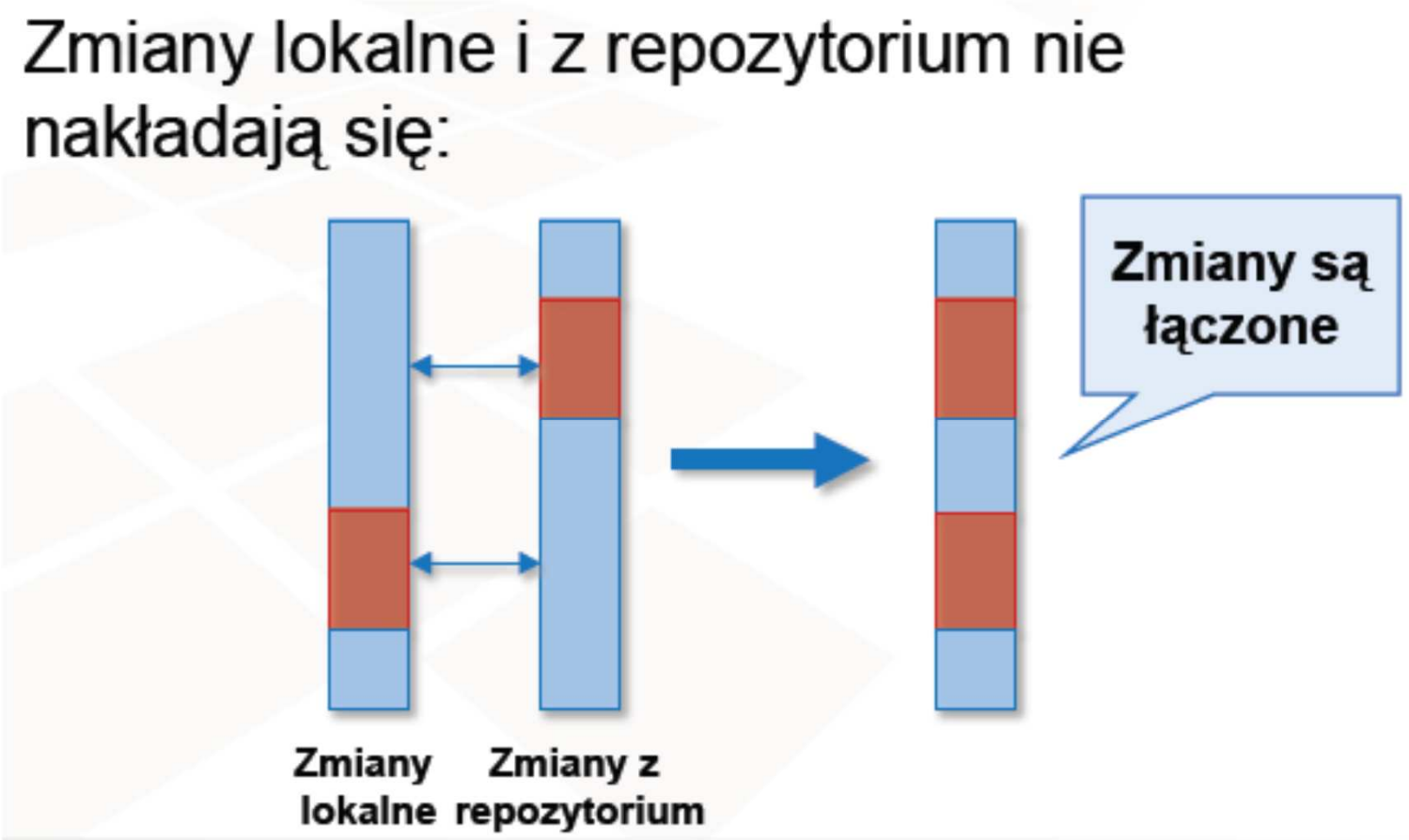
Kontrola wersji – „commit”, „update”



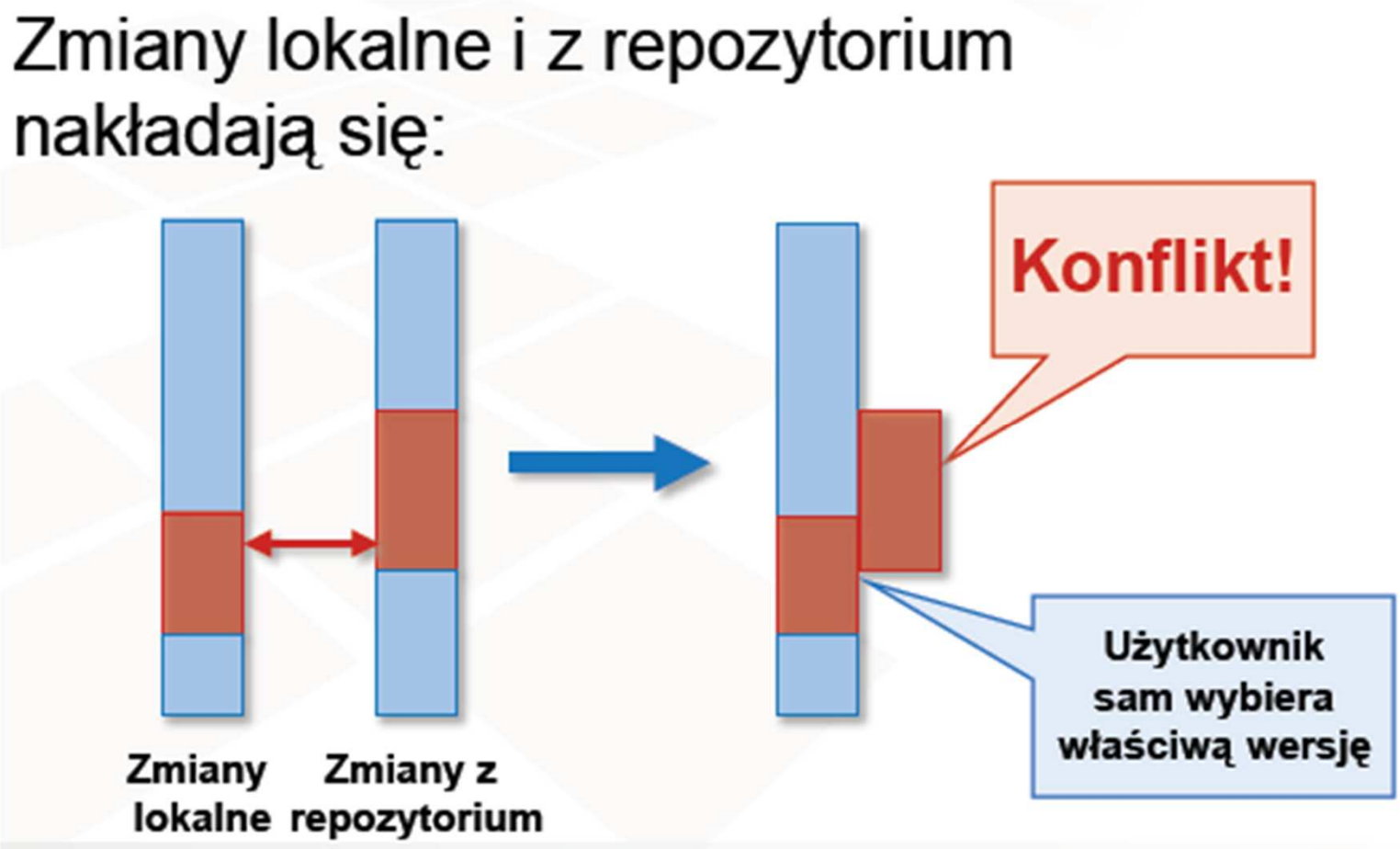
Kontrola wersji – problemy



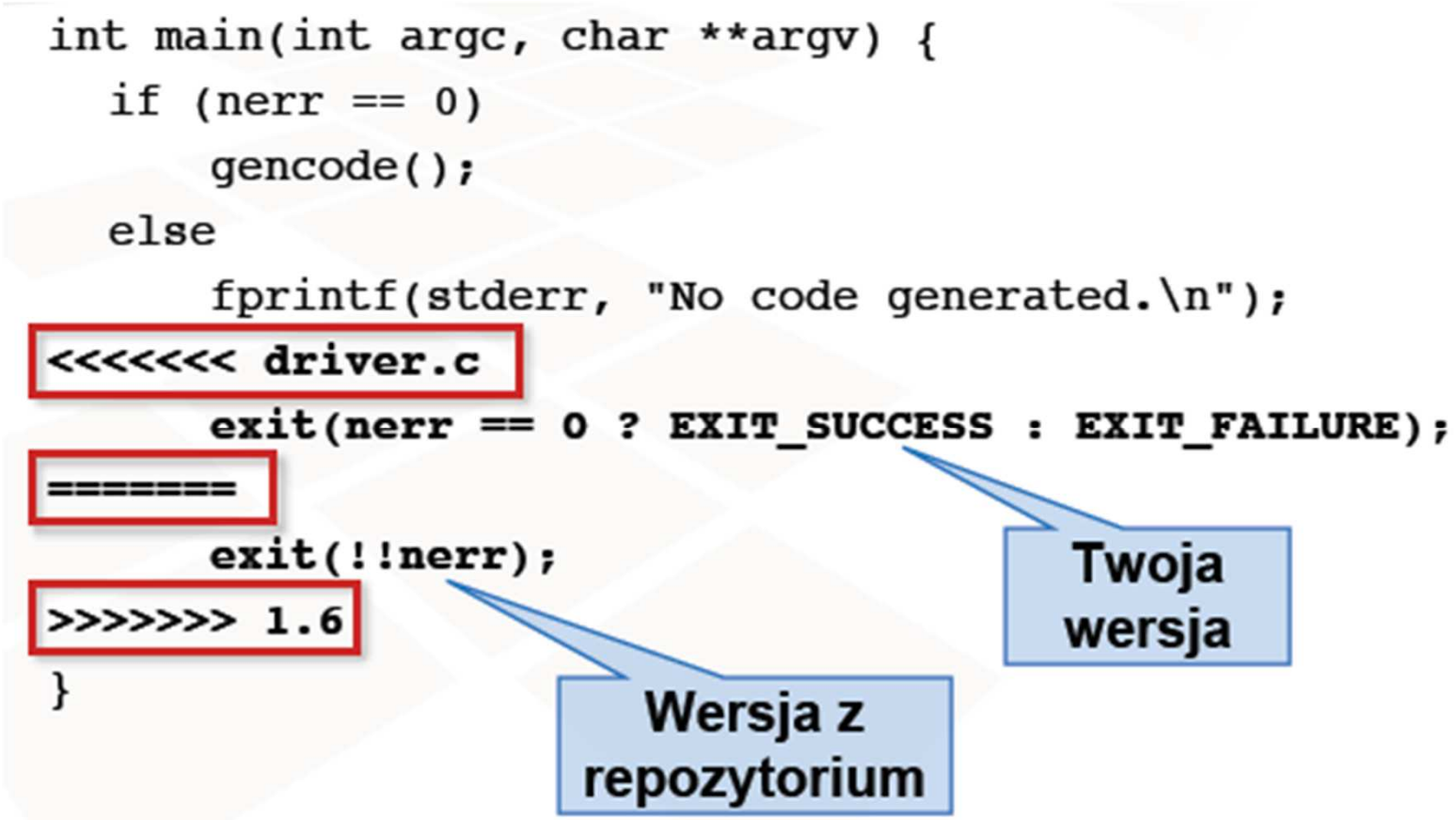
Kontrola wersji – próba scalania zmian



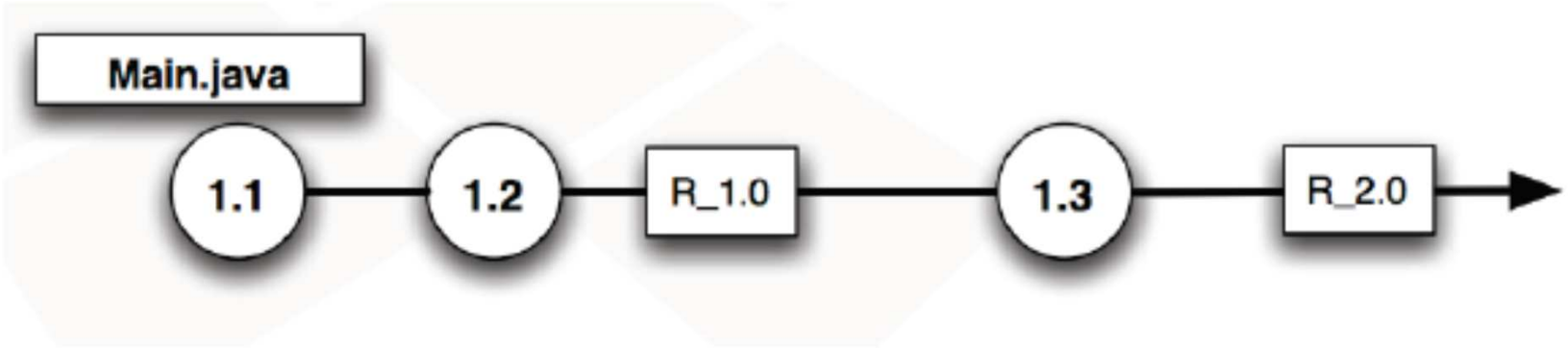
Kontrola wersji – próba scalenia zmian



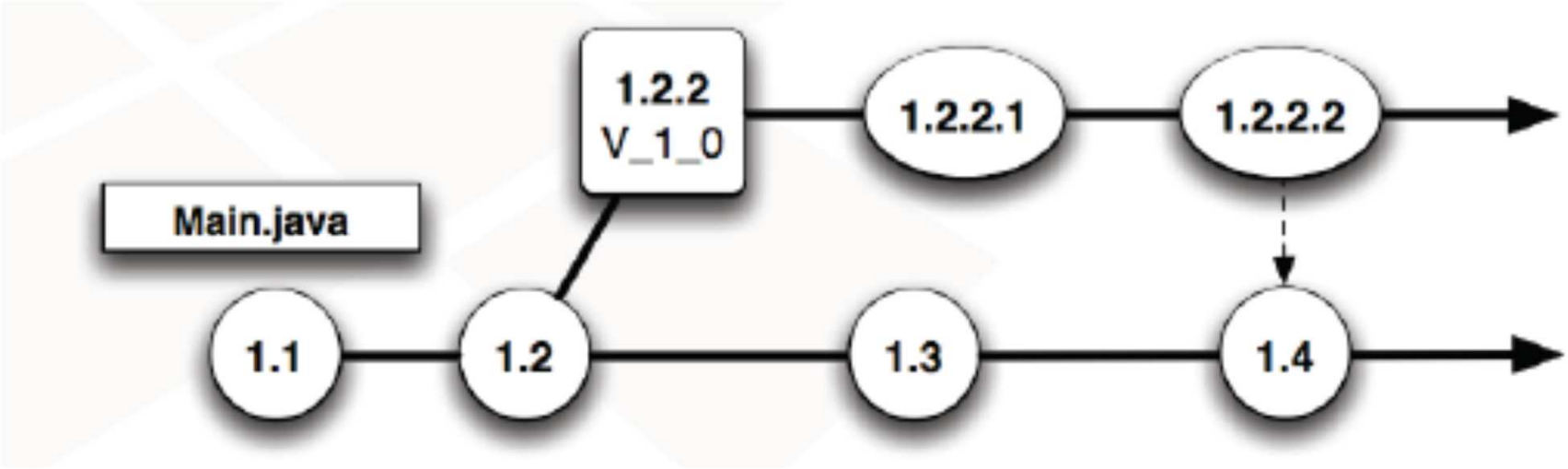
Kontrola wersji – próba scalenia zmian



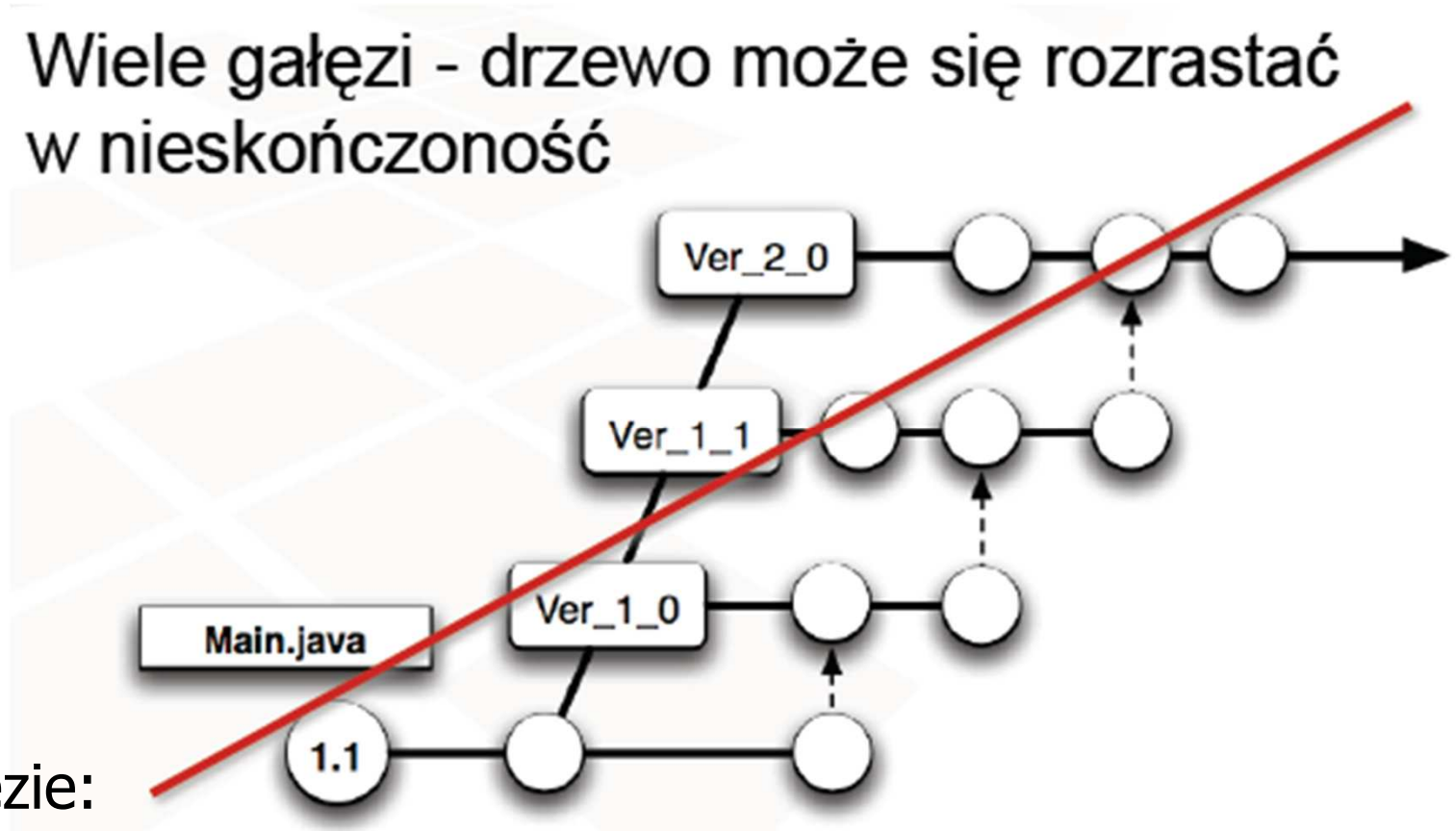
Kontrola wersji – nadawanie etykiet



Kontrola wersji – rozdzielanie i łączenie gałęzi

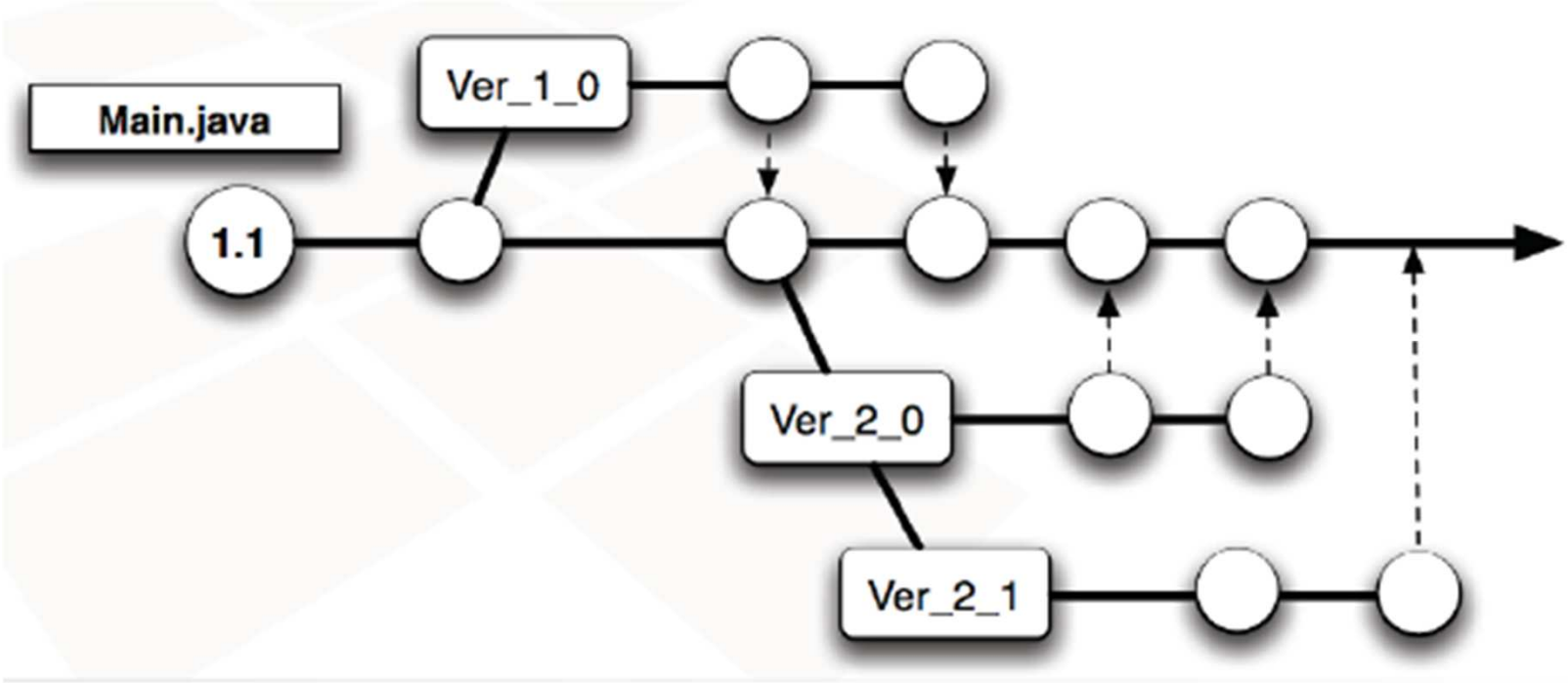


Dobre praktyki

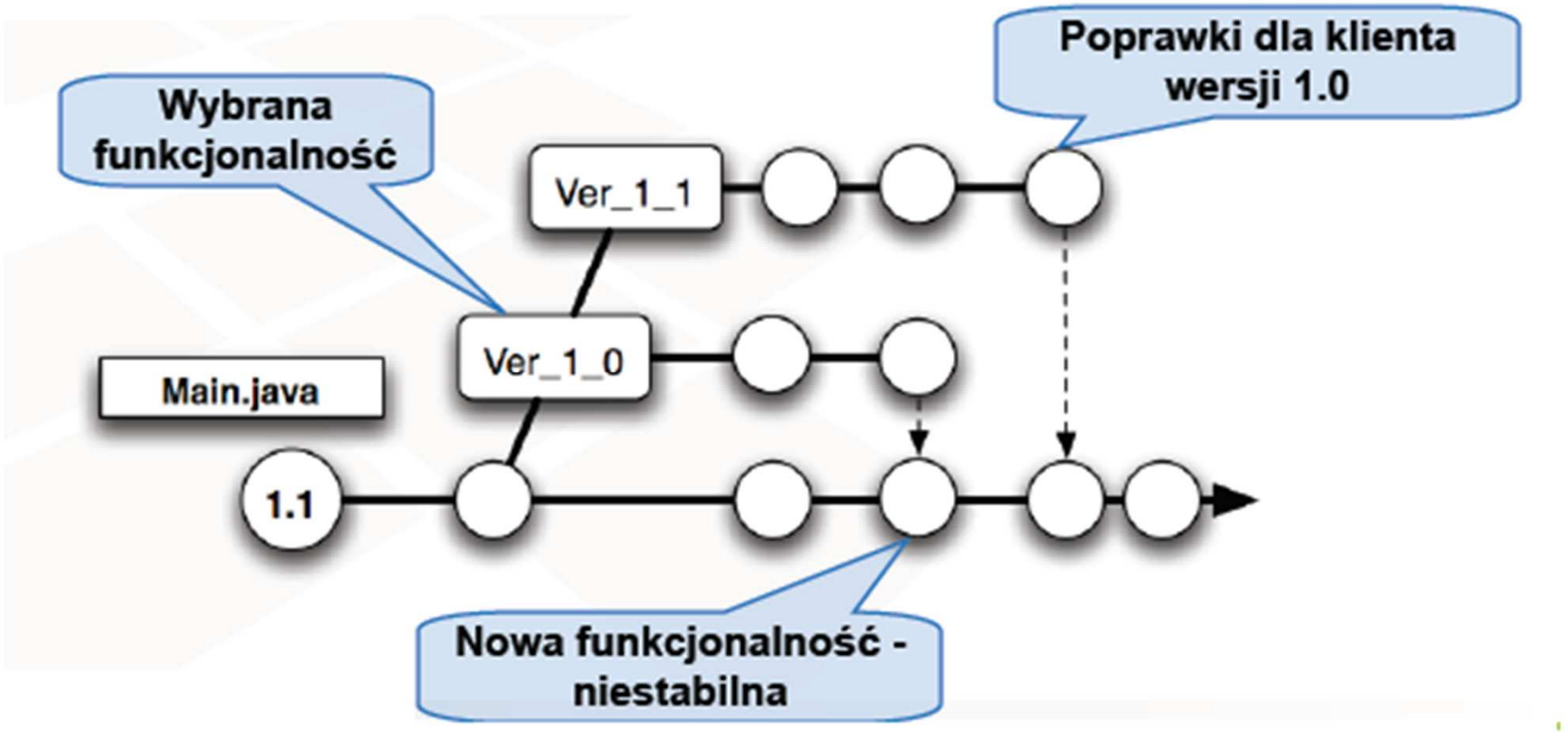


- Gałezie:
  - bazowa
  - przed wydaniem
  - dla zadań

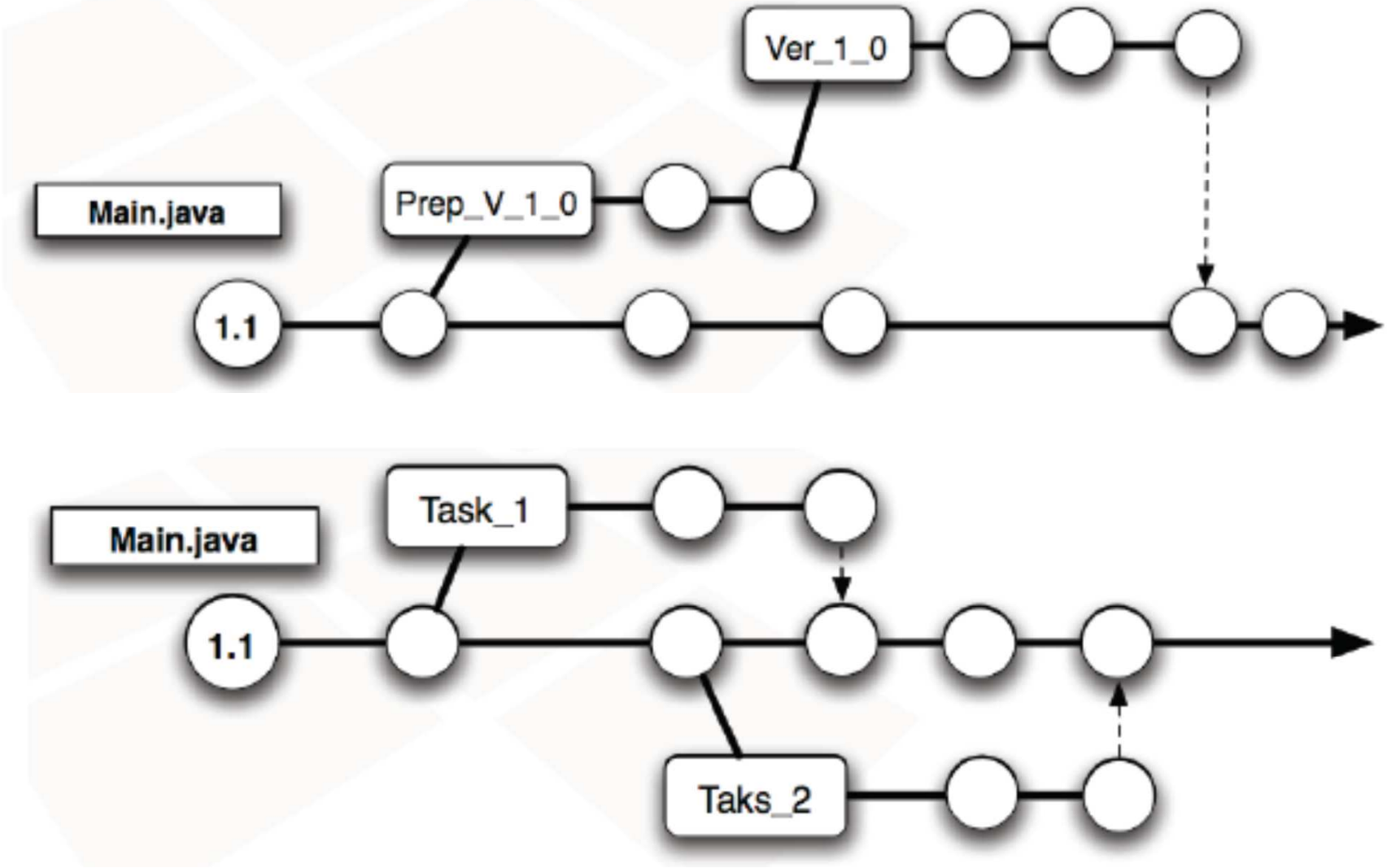
Gałąź bazowa



Gałąź wydania



Gałąź przed wydaniem, gałąź dla zadań



Kontrola wersji to nie wszystko!

- Zarządzanie konfiguracją =
  - Kontrola wersji
  - Zarządzanie zmianą
  - Budowanie wydań

Przykłady pytań do „systemu zarządzania konfiguracją”

- Jaka platforma jest wymagana dla danej wersji systemu?
- Które wersje systemu są zależne od zmiany danego komponentu?
- Ile błędów zgłoszono do danej wersji?
- Jakie komponenty zmodyfikowano przy realizacji danej zmiany?

Zarządzanie zmianą

- żądanie zmiany w postaci wniosku zmiany
- analiza wniosku zmiany
- **jeżeli** (wniosek zasadny) **wtedy**
  - analiza sposobu implementacji
  - analiza wpływu na harmonogram i kosztorys
  - przygotowanie propozycji zmiany
  - **jeżeli** (propozycja zmiany przyjęta) **wtedy**
    - **powtarzaj**
      - wprowadź zmiany
      - przedstaw wynik do weryfikacji
    - **dopóki** (jakość wyniku jest adekwatna)
    - utwórz nowe wydanie systemu
  - **w przeciwnym przypadku**
    - odrzuć wniosek zmiany
- **w przeciwnym przypadku**
  - odrzuć wniosek zmiany

Budowanie wydań

- Prosty schemat identyfikowania wydań
  - v1, v1.1, v1.2, v2.1, v2.2
  - ale:
    - faktyczna struktura nazywania to raczej drzewo niż lista
    - takie nazwy nie są znaczące
- Dodatkowe atrybuty
  - identyfikator (np. R3)
  - status (np. 'beta test')
  - wersja kompilatora
  - parametry narzędzi wspierających budowę systemu

Budowanie wydań a zarządzanie zmianami

- Wydania muszą uwzględniać zmiany
  - wymuszone w systemie w związku z wykryciem przez użytkowników błędów
  - związane z nową funkcjonalnością
- Planowanie wydań jest związane z
  - terminem udostępnienia nowej wersji
  - zakresem zmian włączanych do nowej wersji
- Dobre praktyki
  - Przyrostowa zmiana systemu jaka może być włączona do nowego wydania jest w przybliżeniu stałego rozmiaru
  - Jeśli zbyt wiele nowych cech zostanie włączonych razem z poprawkami błędów, wówczas koszt przygotowania nowego wydania znacząco wzrasta
  - Jeśli do wydania włączono wiele zmian, musi nastąpić po nim wydanie poprawiające błędy wynikające ze zmian wprowadzonych w pierwszym wydaniu

Problemy z nowymi wydaniami

- Klienci mogą nie chcieć nowego wydania systemu
  - Są zadowoleni z obecnego
  - Nowa wersja może wprowadzać niepotrzebne funkcjonalności (np. Word 6)
- Zarządzanie wydaniami nie może zakładać, że wszystkie poprzednie wydania zostały przez użytkowników wdrożone
  - Wszystkie wymagane dla wydania pliki muszą być odtworzone podczas instalowania nowego wydania