

# VCS: narzędzia wspomagające pracę grupową w projektach programistycznych

Wojciech A. Koszek  
[wkoszek@FreeBSD.org](mailto:wkoszek@FreeBSD.org)  
25.02.2009



# Agenda

- Wprowadzenie
  - Charakterystyka pracy jednego programisty
  - Praca grupowa w projekcie programistycznym
- Omówienie zasad działania narzędzi ułatwiających pracę grupową
- Przykłady
- Podsumowanie



# Praca jednego programisty



- Jedna osoba, jedna koncepcja
- Reguła TIMTOWDI zaczyna obowiązywać
  - Nie wnikiemy całkowicie w umysł **typowego** programisty
- Programowanie to unikalna czynność intelektualna
- Problemy natury osobistej

# Dlaczego potrzebujemy wielu programistów?

- Oprogramowanie jest coraz bardziej złożone:  
Już w latach '50 powstawały programy, których stworzenie wymagało wielu osobolat''  
Frederick Brooks
- Nie możemy oprzeć działania dużej instytucji wyłącznie na wiedzy jednej osoby



# Problemy z grupą programistów

- Różne style programowania (nazewnictwo zmiennych, funkcji, formatowania)
- Odmierna „granulacja” mentalna – każdy widzi dany problem trochę inaczej:
  - 10 funkcji po 10 linii kodu <> 1 funkcja ze 100 liniami kodu
- Sporo decyzji nie związanych z kwestiami technicznymi ma duży wpływ na techniczną jakość projektu

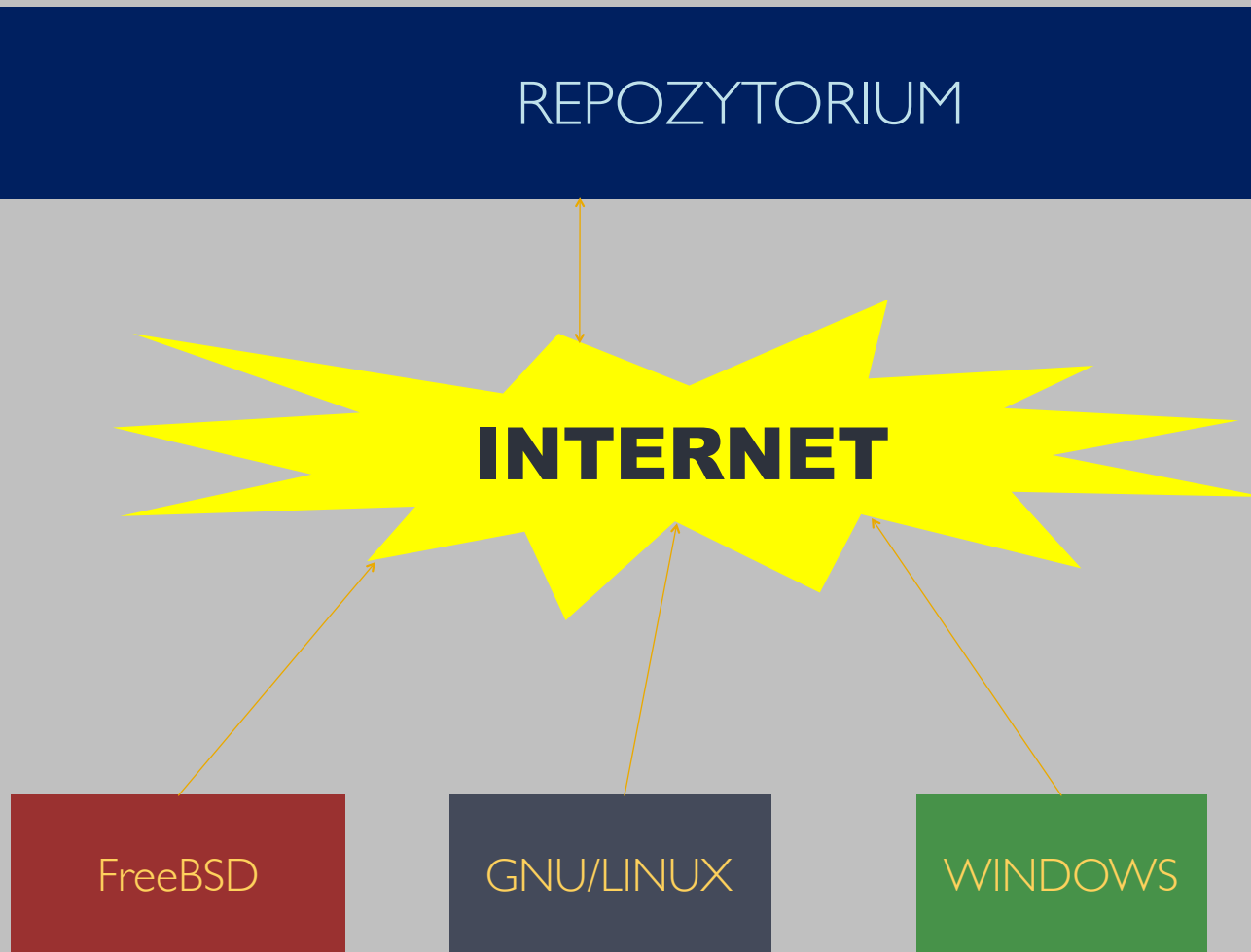


# Systemy kontroli wersji (ang.VCS): rozwiązanie wielu kłopotów

- VCS to narzędzia do polepszenia jakości pracy grupowej
- Bazują na centralnym repozytorium, w którym pliki projektu zarządzane są w spójny sposób



# Globalny punkt widzenia



# Czy to jest dla mnie? (I)

„cze nie dziaa bo tam się nie kompliuje pokazje linia 113, ale 224 w x.cpp jest dobrze, nie wiem co nie dzala nie kompiluje się dalej jak wstawie w 414 petle while poprawisz?”

■ Dariusz, lat 22

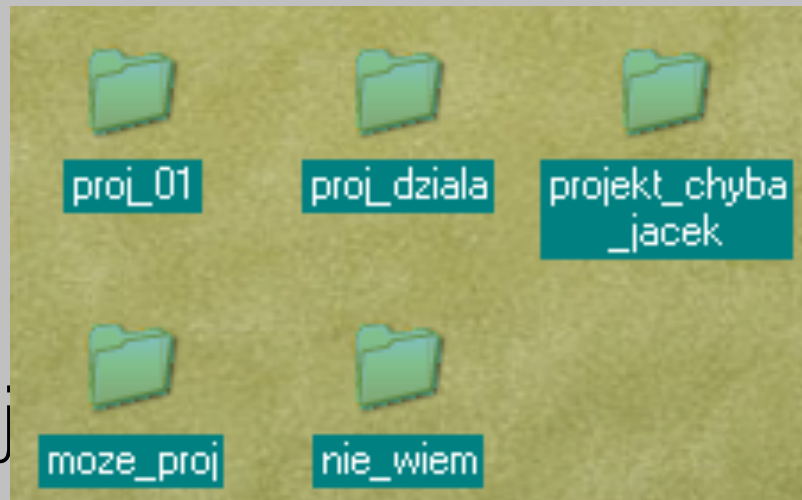


# Czy to jest dla mnie? (2)

- Czy w Twoim programie studiów pojawił się przedmiot polegający na stworzeniu projektu w 16-osobowych grupach i zastanawiasz się, jak to wszystko skoordynować?
- Dochodzisz do wniosku, że wysyłanie sobie archiwów z kolejnymi wersjami programu to może nie do końca to, czego chcesz?

# Czy to jest dla mnie (3) ?

- Jeżeli efektem większości Twoich projektów jest 10 katalogów podobnych do...



- To kolejna lista katalogów, które wydawać Ci się mogą być użyteczne.

# Wybór systemu VCS jest dość trudny



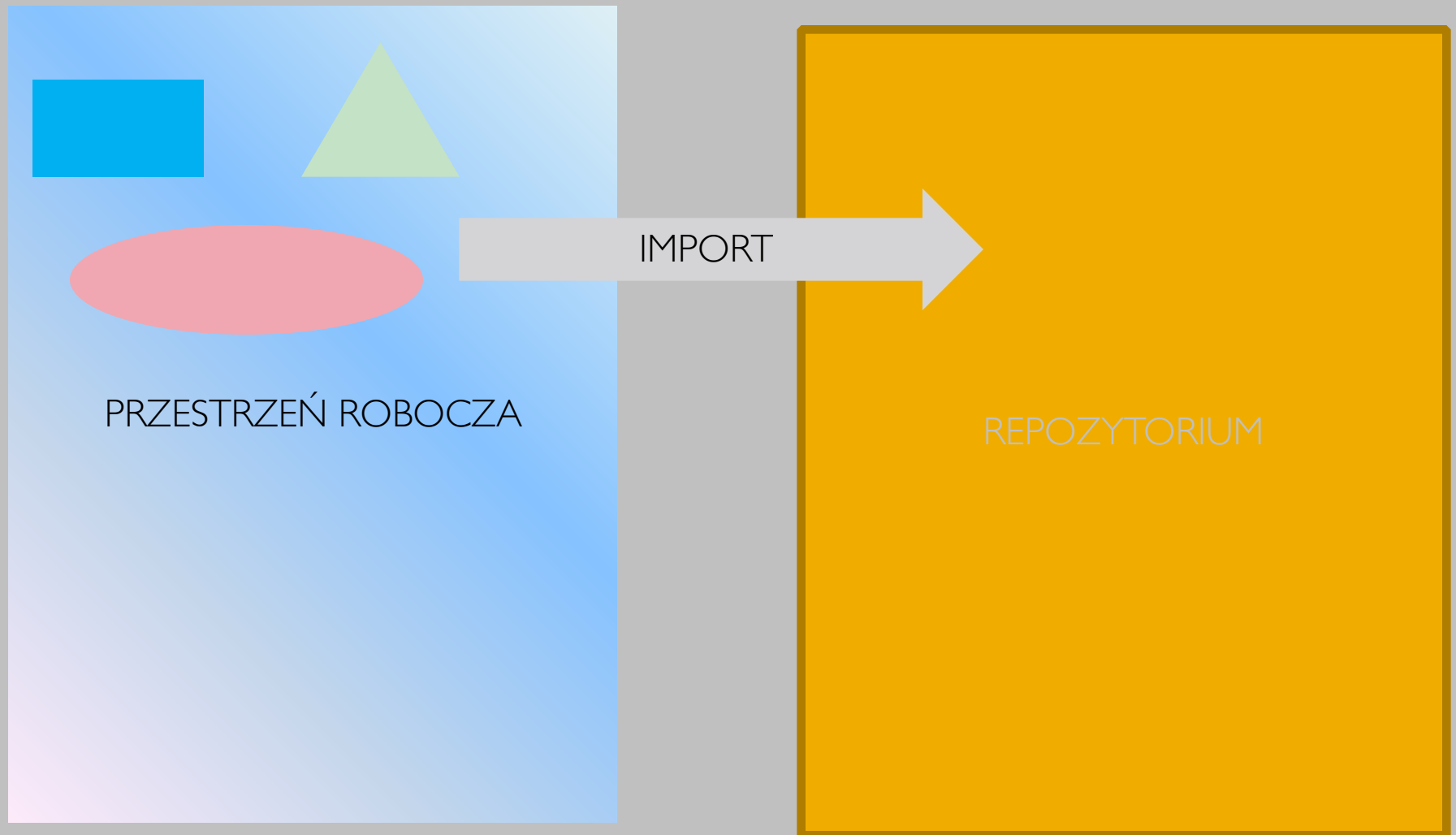
# Subversion: chyba najlepszy kompromis

- Darmowe rozwiązanie dla UNIX, GNU/Linux, Windows
- Dzięki TortoiseSVN bardzo dobra integracja z Windows (przykłady praktyczne później)
- Dobrze sprawdza się jako narzędzie do pracy dla każdego środowiska programistycznego

# Krótko o teorii (I)

- Tworzymy puste repozytorium – miejsce na archiwum plików wchodzących w skład projektu
  - Jego miejsce jest niezmiennie przez cały cykl życia projektu
- Dokonujemy inicjalizacji repozytorium plikami roboczymi (\*.cpp, \*.h, Makefile) (ang. *import*)

# Krótko o teorii (I, ciąg dalszy)

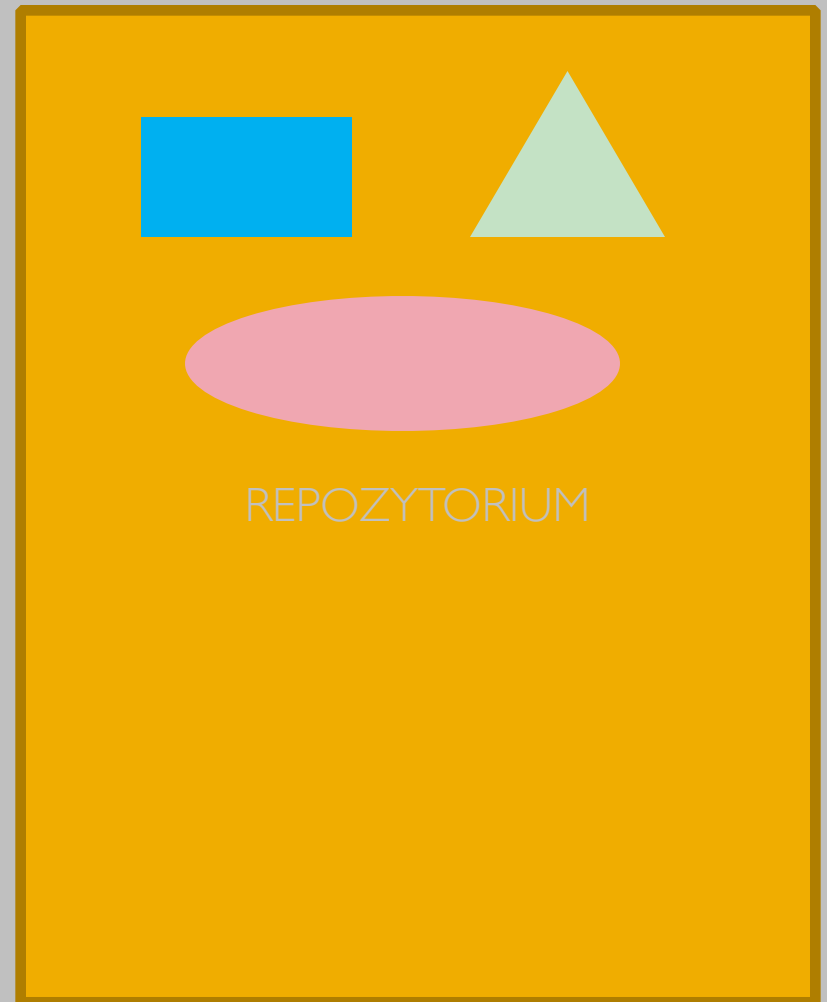


# Krótko o teorii (2)

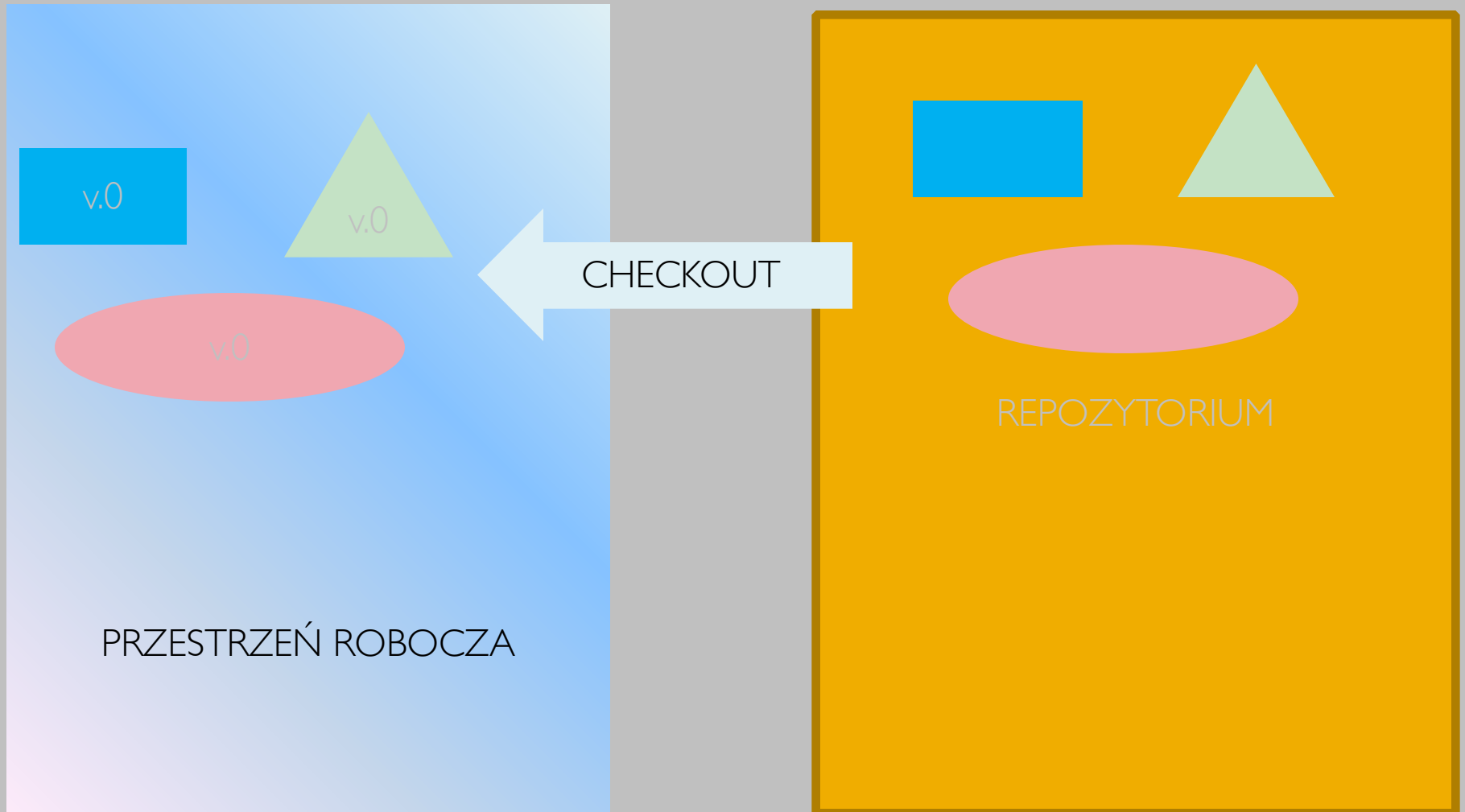
- Pliki które właśnie dodaliśmy powinniśmy zarchiwizować i przenieść w inne miejsce
  - Od tej pory naszym projektem jest to, co istnieje w repozytorium
- Aby rozpocząć pracę nad projektem, należy pobrać pliki wchodzące w jego skład z repozytorium (ang. *checkout*)



# Krótko o teorii (2, ciąg dalszy)



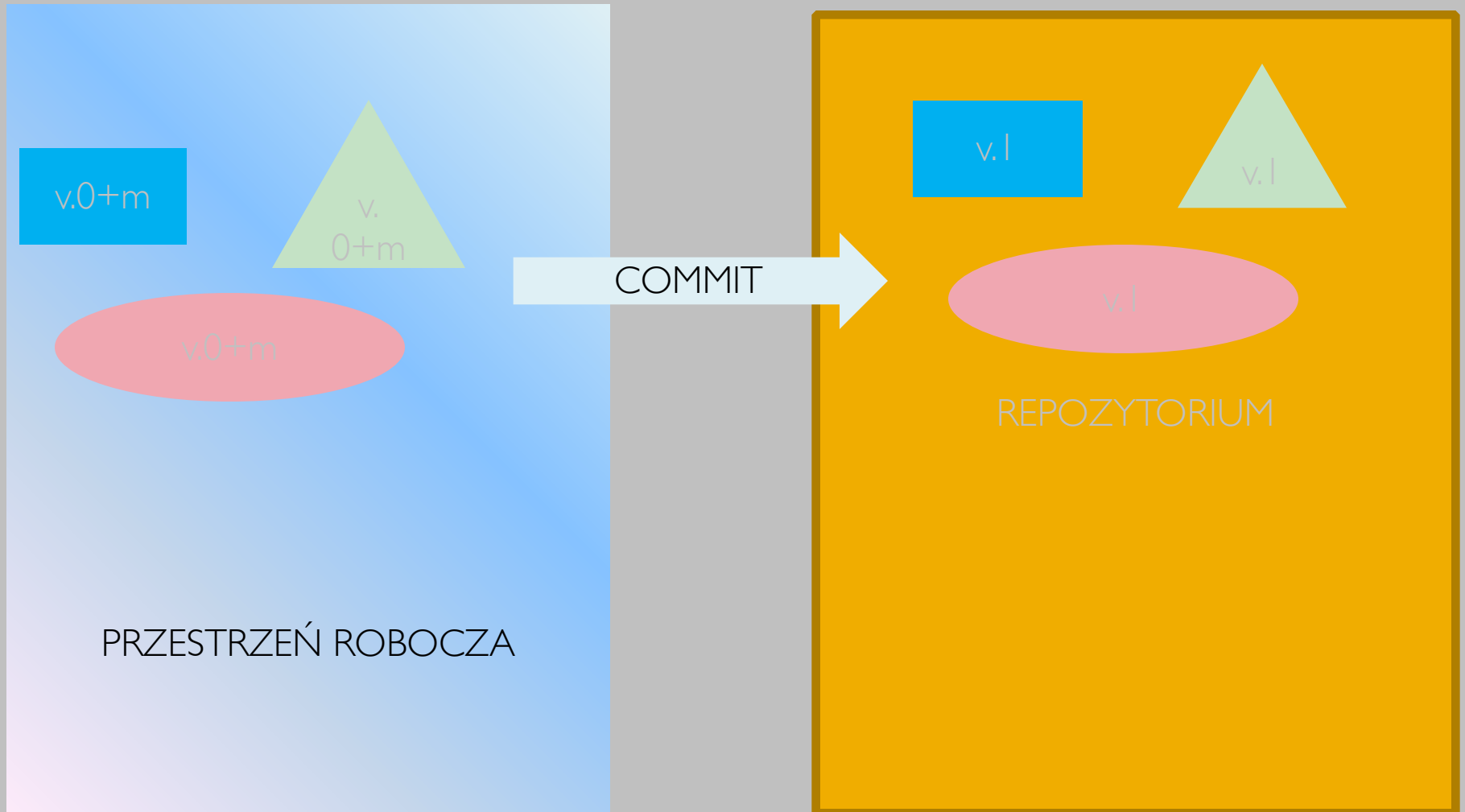
# Krótko o teorii (2, ciąg dalszy)



# Krótko o teorii (3)

- Po pobraniu możemy brać się do pracy
  - Wprowadzamy zmiany
  - Testujemy program
  - Uaktualniamy dokumentację
- Jeżeli wszystko jest w porządku, wprowadzamy nasze zmiany do repozytorium (ang. commit/submit)
  - Każdy *commit* winien posiadać opis zmian

# Krótko o teorii (3, ciąg dalszy)



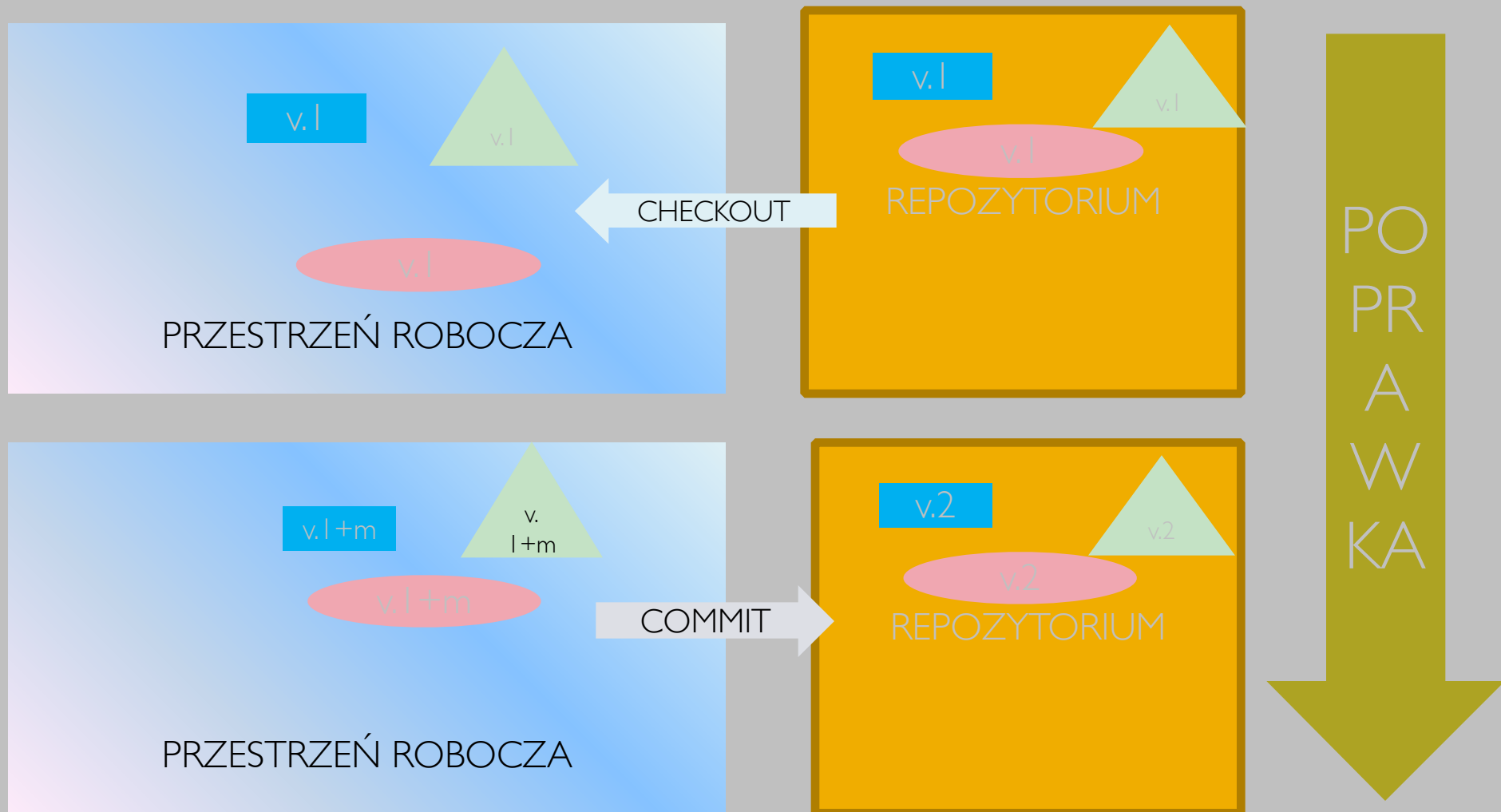
# Co to zmienia?

- Programista P2 po dokonaniu zmian przez Programistę P1 uaktualnia swoją kopię roboczą (ang. *update*)
- Jest w stanie przejrzeć historię zmian (opis zmian dokonanych przez P1)
- Wariant optymistyczny: P2 zaczyna pracować po zmianach P1
- Wariant pesymistyczny: widzi błąd P2

# Co robi P2?

- Porównuje różnice (tzw. *diff*) między wstępną a zmienioną wersją pliku
  - Widzi linie usunięte i dodane
- Dostrzega błąd P1 po czym go poprawia na swojej kopii roboczej i dokonuje commit
  - Plik uzyskuje numer wersji „2”
- Po powrocie do pracy P1 dostrzega poprawki P2 i obaj kontynuują pracę

# Co robi P2 (ciąg dalszy)





# Jak zorganizować repozytorium?

- Kod źródłowy, opis architektury, opis API, dokumentacja, testy jednostkowe, dane przykładowe, infrastruktura do testów, plik README itd..
- Cechą wspólną jest obecność tekstu w w.w plikach
  - W przypadku utrzymywania plików w formacie Microsoft Word, *'diff'* niewiele nam da

# Gdzie umieścić repozytorium?

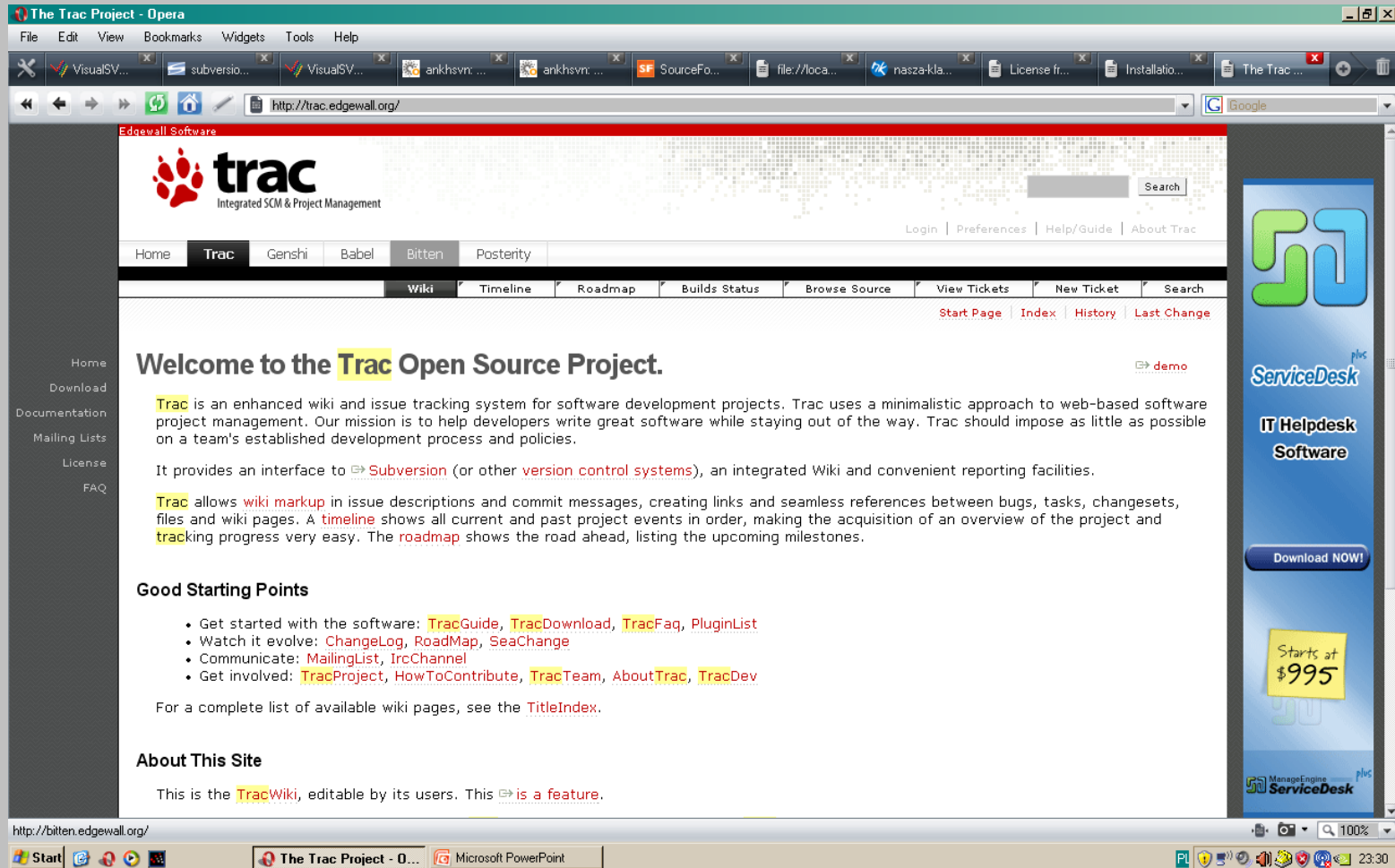
- Repozytorium ma sens, gdy jest dostępne dla wszystkich uczestników projektu
- W Internecie dostępnych jest wiele serwisów oferujących darmowy hosting repozytorium Subversion pod warunkiem publicznego rozwoju oprogramowania
  - SourceForge



# Trac: integracja Subversion z Wiki

- Trac to oprogramowanie będące swojego rodzaju nakładką na Subversion:
  - Własny serwer WWW
- WIKI w Trac oparte o pliki tekstowe
  - Łatwe do eksportu, zmiany i powrotnego importu
    - Tekst strony może być wersjonowany w Subversion

# Skrin szot



# Trac: system śledzenia błędów

- Trac posiada moduł, dzięki któremu użytkownicy mogą zgłaszać napotkane w oprogramowaniu błędy:
  - Dzięki permanentnemu wpisowi w bazie błędów posiadamy pełną historię powstałych/naprawionych usterek

# Skin szot 2 (moduł śledzenia błędów)

**{1} All Active Tickets** (912 matches)

Results (1 - 100 of 912)

| Ticket | Summary   | Component         | Version     | Milestone      | Type        | Severity | Owner       | Status   | Created  |
|--------|---|-------------------|-------------|----------------|-------------|----------|-------------|----------|----------|
| #6387  | docs about translation  | i18n              | 0.12dev     | 0.12           | task        | normal   | cboos       | new      | 11/21/07 |
| #5382  | base64 encoding/decoding breaks compatibility with python2.3  | plugin/spamfilter | devel       | not applicable | enhancement | normal   | cboos       | reopened | 05/28/07 |
| #4431  | wiki_to_wikidom   | wiki system       | 0.10.3      | 0.13           | enhancement | major    | cboos       | assigned | 12/20/06 |
| #886   | Add support for Master tickets  | ticket system     | devel       | 0.13           | enhancement | major    | nkantrowitz | reopened | 11/05/04 |
| #791   | login/logout, authentication, and authorization   | general           | devel       | 0.13           | enhancement | major    | cboos       | reopened | 09/30/04 |
| #8088  | Allow a WikiFormatted Custom Ticket Field Parse Through the Wiki Engine in Reports, without Using 'Magic' Column Name | report system     | 0.11.3      |                | enhancement | normal   |             | new      | 02/23/09 |
| #8067  | Stopped sync in 0.11.2.1  | version control   | 0.11-stable | 0.11.4         | defect      | major    | cboos       | new      | 02/18/09 |
| #8032  | IPBlacklistFilterStrategy defaults require update   | plugin/spamfilter | none        |                | task        | major    |             | new      | 02/06/09 |
| #8006  | Non-Writable Python Egg Cache Directory causes mod_fastcgi to send an error 500 with Apache 2                         | general           | 0.11.2.1    | 0.11.4         | defect      | major    |             | new      | 01/27/09 |
| #7860  | deleted account can still be seen in "assign to" list   | admin/console     | 0.11.1      | 0.11-retriage  | defect      | critical | cboos       | new      | 12/04/08 |
| #7815  | server/trac/project/login page  | general           | 0.11.2      |                | enhancement | normal   |             | reopened | 11/13/08 |

# Trac: reszta

- Wspiera wiele wtyczek
  - Powiadamianie o zmianach na pocztę
- Domyślny projekt strony jest bardzo estetyczny i nadaje się na stronę główną naszego projektu



# Praktyka...



# Podsumowanie

- Systemy VCS są nieodzownym środowiskiem pracy profesjonalnego programisty
- Nauka początkowo przychodzi wolno, jednak efekty wszystko rekompensują
- Podstawowa funkcjonalność jest prosta do opanowania
  - Bardziej zaawansowane funkcje czasami potrafią płatać figle

# Koniec

Wojciech Koszek

[wkoszek@FreeBSD.org](mailto:wkoszek@FreeBSD.org)

26.02.2009

Prezentacja i strona z opisem:  
<http://FreeBSD.czyst.pl/~wkoszek/>