

Git - zwinny system śledzenia równoległych zmian kodu w dużym zespole



Przeznaczenie szkolenia

Przedmiotem szkolenia jest proces zwinnego zarządzania w czasie zmianami zbioru plików tekstowych w warunkach jednoczesnej, intensywnej edycji przez wieloosobowe zespoły, z użyciem zdecentralizowanego repozytorium Git. Potencjalnymi odbiorcami tego szkolenia są przede wszystkim programiści, testerzy, członkowie zespołów projektów informatycznych jak również inne osoby zainteresowane wersjonowaniem kodu źródłowego oraz innych artefaktów z wykorzystaniem wspomnianego oprogramowania.



Korzyści wynikające z ukończenia szkolenia

Biorąc pod uwagę fakt, że krzywa uczenia w przypadku interfejsu git jest początkowo stroma (narzędzie jest trudne do opanowania), szkolenie realizowane jest w formie cykli teoretyczno-praktycznych ze stopniowo wzrastającym poziomem trudności materiału. Każdy etap kończy się ćwiczeniami do samodzielnego wykonania (w razie potrzeby w asyście instruktora). Początkowe cykle mają za zadanie ugruntować i wyrównać podstawową wiedzę, zaś kolejne pozwalają uczestnikom osiągnąć pełną możliwość swobodnego korzystania narzędzia. Zwieńczeniem kursu jest angażująca słuchaczy symulacja pracy nad zespołowym projektem hostowanym na jednym z popularniejszych serwerów np. GitHub.

Szkolenie stawia nacisk na wykorzystanie podstawowego interfejsu oprogramowania, czyli linii komend, ponieważ z racji mnogości opcji, tylko takie podejście daje możliwość dogłębnego zrozumienia narzędzia i otwiera uczestnika na możliwość przyswojenia dowolnego GUI wspomagającego pracę (przez realizację podzbioru możliwości podstawowego interfejsu programu).

Absolwent szkolenia posiadać następującą wiedzę:

- Czym jest Git, a czym nie
- Jaka jest architektura programu Git
- Na jakie narzędzia warto zwrócić szczególną uwagę w pracy z Git, z czym się integruje to rozwiązanie
- Jaka jest rola Git w cyklach wydawania oprogramowania i jakie są związane z tym strategie oraz praktyczne umiejętności:
 - czytanie skrótów interfejsów komend (synopsis) i poruszanie się po oficjalnej dokumentacji
 - świadome przygotowanie stanowiska pracy
 - tworzenie rewizji oraz poruszanie się po grafie historii
 - tworzenie oraz współdzielenie repozytoriów z członkami zespołu
 - scalanie zmian, synchronizacja konfliktów wynikających z pracy równoległej
 - zarządzanie dużym wolumenem kodu poprzez odpowiednią organizację procesu pracy w zespole
 - zarządzanie cyklami wydawania oprogramowania
 - poruszanie się po platformie GitHub



Oczekiwane przygotowanie słuchaczy

Podstawowa wiedza z zakresu obsługi komputera.



Język szkolenia

- Szkolenie: polski
- Materiały: polski



Czas trwania

2 dni / 14 godzin

Agenda szkolenia

1. Wprowadzenie

- Zrozumienie czym jest system kontroli wersji

- Różne architektury systemów kontroli wersji
- Dlaczego Git stał się najpopularniejszym narzędziem w swojej kategorii:
 - przedstawienie cech programu
 - porównanie z innymi systemami
- Historia powstania i przeznaczenie Git
- Podstawowe programy linuksowe

2. Praca z dokumentacją

- Czytanie synopsis (skrót opisu interfejsu)
- Odnajdowanie informacji w dokumentacji

3. Git na lokalnym komputerze

- Pierwsze kroki:
 - Instalacja środowiska na platformach Windows, Mac, Linux
 - Konfiguracja środowiska pracy (*Poziomy konfiguracji, Ustawianie danych użytkownika, Konfiguracja domyślnego edytora, Podstawy VIM*)
 - Inicjalizacja repozytorium
- Operacje na plikach:
 - Katalog repozytorium (katalog roboczy, katalog git)
 - Cykl życia plików i stan plików
 - Status repozytorium
 - Dodawanie do indeksu i wycofywanie oczekujących zmian
 - Dodawanie migawek
 - Ignorowanie pików, globy
- Operacje na branchach:
 - Korzyści i koszty pracy z gałęziami
 - Tworzenie gałęzi
 - Zmienianie nazwy gałęzi
 - Usuwanie gałęzi lokalnie
 - Przełączanie się między gałęziami
 - Scalanie gałęzi
- Poruszanie się po acyklicznym grafie skierowanym historii:
 - Wskazywanie przodków
 - Wskazywanie zbiorów commitów (notacje double, triple dots)
 - Porównywanie rewizji
 - Historia zmiany (*Całego repozytorium, linijki w pliku, pliku*)
 - Filtrowanie commitów
 - Nadpisywanie historii
 - Usuwanie commitów
- Jak git przechowuje dane
 - Anatomia folderu .git
 - Czym jest funkcja haszująca SHA-1 i jaka jest jej rola

- Anatomia obiektów Git (*Wskaźniki zarządzane, Wskaźniki niezarządzane, Podstawowe obiekty w bazie key-val*)

4. Git, a praca zespołowa w sieci:

- Synchronizacja pracy
 - Pobieranie zdalnego projektu
 - Opcje autentykacji użytkownika (*HTTPS, SSH, GPG*)
 - Wskaźniki typu remote
 - Synchronizacja ze wskaźnikami remote
 - Wypychanie zmian
 - Pull Request
- Oprogramowanie serwerowe Git - wprowadzenie:
 - GitHub
 - BitBucket
 - GitLab
- Strategie efektywnej organizacji pracy:
 - Tagowanie
 - Dobre praktyki sprzyjające przepływowi informacji
 - Dobór strategii ze względu na charakterystykę projektu
 - Przydatne konwencje w pracy zespołowej
 - Łączenie metodyk zwinnych z funkcjonalnościami Git

5. Dodatkowe narzędzia:

- Mechanizm schowka do szybkiego przełączania kontekstu pracy
- Wykorzystanie komendy rebase
- Cherry pick, Aliasy, Reflog

6. Projekt zespołowy symulujący rzeczywistą pracę:

- Kilka wariantów projektów
- Przedstawienie problemu i konwencji pracy
- Podział zadań
- Praca równoległa
- Scalenie zmian
- Wydanie efektu pracy