

Git - zwinny system śledzenia równoległych zmian kodu w dużym zespole

Zwinne zarządzanie kodem w dynamicznych zespołach wymaga zaawansowanych narzędzi, a Git to jedno z najpopularniejszych rozwiązań w tej dziedzinie. Dzięki szkoleniu uczestnicy opanują podstawy i zaawansowane techniki pracy z Git, co umożliwi im efektywne wersjonowanie kodu oraz zarządzanie projektami w zespołach programistycznych.



Odbiorcy szkolenia

Szkolenie jest skierowane do:

- Programistów pragnących pogłębić wiedzę o zarządzaniu wersjami kodu,
- Testerów pracujących nad dynamicznie zmieniającym się oprogramowaniem,
- Członków zespołów projektowych realizujących projekty IT,
- Osób zainteresowanych wersjonowaniem kodu źródłowego oraz innych dokumentów,
- Początkujących i średniozaawansowanych użytkowników Git, którzy chcą lepiej zrozumieć narzędzie i jego potencjał.



Korzyści

1. Zrozumienie Gita od podstaw do zaawansowanych zastosowań. Uczestnicy poznają, czym jest Git, jak działa oraz jakie są jego kluczowe funkcje, co pozwala na efektywne zarządzanie kodem w każdym projekcie.
2. Praktyczne umiejętności pracy z linią komend. Dzięki pracy z podstawowym interfejsem Git, uczestnicy zyskają dogłębne zrozumienie narzędzia, które ułatwi korzystanie z dowolnych graficznych

nakładek.

3. Efektywna współpraca w zespole. Uczestnicy nauczą się tworzyć i współdzielić repozytoria, zarządzać zmianami oraz rozwiązywać konflikty w pracy równoległej.
4. Organizacja procesu pracy w zespole. Szkolenie obejmuje strategię zarządzania dużą ilością kodu i cyklami wydawania oprogramowania, co znacząco zwiększa efektywność projektów.
5. Praktyczne doświadczenie dzięki symulacji zespołowej pracy. Realizacja projektu na platformie GitHub pozwala uczestnikom zastosować zdobytą wiedzę w realistycznych warunkach, zwiększając ich gotowość do pracy zawodowej.

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).



Program 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 plików, 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



Oczekiwane przygotowanie uczestnika

Podstawowa wiedza z zakresu obsługi komputera Założone konta w github <https://github.com/>



Czas trwania

2 dni / 14 godzin

Język

- Szkolenie: polski
- Materiały: polski