

SZKOLENIE PODSTAWOWE

Kontrola wersji z Git

GIT

Czas trwania: 2 dni

Efektywne wykorzystanie systemu kontroli wersji Git w codziennej pracy

Cele szkolenia

- Praktyczne zapoznanie się z działaniem rozproszonego systemu kontroli wersji Git
- Nauka stylu pracy z Gitem uwzględniająca dobre praktyki, polecenia dostępne w tym narzędziu oraz modele współpracy zespołowej
- Uczestnicy użyją w praktycznych zastosowaniach konsolowych narzędzi Git wraz z graficzną nakładką Git Extensions

Zalety

- Nacisk na najbardziej problematyczne sytuacje dla początkujących, takie jak rozwiązywanie konfliktów, wycofywanie zmian, praca ze zdalnymi gałęziami czy efektywna współpraca zespołowa
- Uczestnicy szkolenia będą na bieżąco wykonywali różnorodne ćwiczenia, które pozwolą im zapoznać się z typowymi scenariuszami pracy z Gitem
- Szkolenie Git - kontrola wersji może być dopasowane do wymagań konkretnej grupy, na przykład poprzez skupienie się na bardziej zaawansowanym i specyficznym wykorzystaniu Gita przez dany zespół lub pracę z wybranymi narzędziami, takimi jak konkretne środowisko deweloperskie czy wybrany graficzny klient Git
- Praktyka przed teorią - wszystkie szkolenia technologiczne prowadzone są w formie warsztatowej. Konieczna teoria jest wyjaśniana na przykładzie praktycznych zadań
- Konkretność umiejętności - w ramach każdego szkolenia rozwijamy praktyczne umiejętności związane z daną technologią i tematyką
- Nauka z praktykami - wszyscy trenerzy na co dzień pracują w projektach, gwarantuje to dostęp do eksperckiej wiedzy i praktycznego know-how

Dla kogo?

- Szkolenie Git - kontrola wersji adresowane jest do programistów, testerów, wdrożeniowców oraz administratorów, którzy chcą efektywnie korzystać z systemu kontroli wersji Git podczas codziennej pracy



Wymagania

- Od uczestników szkolenia wymagana jest znajomość obsługi komputera oraz pracy z wierszem poleceń



Program

1. Wprowadzenie
 - a. Jak działają systemy kontroli wersji
 - b. Charakterystyka pracy z rozproszonym systemem kontroli wersji
 - c. Instalacja i podstawowa konfiguracja Gita
 - d. Narzędzia dostarczane z Gitem
2. Lokalna praca z Gitem
 - a. Inicjalizacja i konfiguracja repozytorium
 - b. Katalog .git
 - c. Zatwierdzanie zmian (commit), narzędzie Git GUI
 - d. Dobre praktyki tworzenia opisów zmian
 - e. Przechowywanie zmian (stash)
 - f. Ignorowanie i modyfikacja zmian
 - g. Przeglądanie zmian (diff), narzędzia do przeglądania zmian
 - h. Przegląd historii (log), narzędzie gitk
 - i. Tworzenie etykiet, rodzaje etykiet (tag)
 - j. Cofanie zmian (reset, reflog)
 - k. Praca z gałęziami (branch)
 - l. Scalanie zmian i zmiana bazy (merge, rebase, cherry pick)
 - m. Rozwiązywanie konfliktów
 - n. Tworzenie aliasów
3. Zdalne repozytoria
 - a. Metody dostępu do repozytoriów, używane protokoły (SSH, HTTP)
 - b. Pobieranie repozytorium (clone)
 - c. Wypychanie i pobieranie zmian (push, fetch, pull)
 - d. Powiadamianie o zmianach (pull request)
 - e. Praca ze zdalnymi gałęziami
 - f. Praca z podprojektami (submodule, subtree)
4. Praca grupowa z Git
 - a. Metody współdzielenia zmian
 - b. Praca w środowisku użytkowników o różnych rolach (programista, tester, wdrożeniowiec)
 - c. Zasady zarządzania wydaniem i historią projektu
 - d. Rozszerzenie GitFlow
5. Inne możliwości Gita
 - a. Zaawansowane opcje konfiguracji
 - b. Architektura Gita, obiekty i system ich nazywania
 - c. Automatyzacja zadań Gita, skrypty dla zdarzeń (hooks)
 - d. Konserwacja i optymalizacja repozytorium (fsck, gc)
 - e. Wyszukiwanie błędów za pomocą Gita (bisect)
 - f. Wyszukiwanie winnych (blame)
 - g. Przepisywanie historii (filter-branch)
 - h. Przechowywanie dużych plików



6. Git i dodatkowe narzędzia

- a. Graficzne nakładki na Git na przykładzie Git Extensions
- b. Przegląd kodu, działanie Gita z Gerritem
- c. Praca z GitHub i GitLab

