BO·TT·EGA
IT minds

Program szkolenia:

Jenkins - Continuous Integration

Informacje:

Nazwa:	**Jenkins - Continuous Integration**
Kod:	**tools-Jenkins-CI**
Kategoria:	Narzędzia
Grupa docelowa:	developerzy testerzy
Czas trwania:	2 dni
Forma:	50% wykłady / 50% warsztaty

Podczas szkolenia uczestnicy poznają informacje oraz praktyczne techniki pozwalające na samodzielne projektowanie oraz implementowanie procesu ciągłej integracji z wykorzystaniem Jenkins.

W pierwszej kolejności zaprezentowane zostaną zagadnienia podstawowe takie jak: instalacja i konfiguracja narzędzi, automatyzacja kompilacji, różnych rodzajów testowania oraz statycznej analizy kodu.

Następnie poruszone zostaną zaawansowane rozwiązania, takie jak: integracja z systemami zewnętrznymi, współbieżne wykonywanie planów oraz wykorzystanie procesu ciągłej integracji podczas wdrażania koncepcji „Continous Delivery”.

Na zakończenie uczestnicy poznają polecane metody projektowania i optymalizacji oraz wskazówki pozwalające wybrać narzędzia najbardziej odpowiadające ich potrzebom.

Zalety szkolenia:

- Poprawienie jakości oprogramowania
- Automatyzacja zadań
- Continuous Delivery
- Zalecane wzorce i praktyki

BOTTEGA IT minds

Szczegółowy program:

1. Przedstawienie koncepcji Continuous Integration (CI).

- 1.1. Co to jest ciągła integracja: Cechy i Korzyści
- 1.2. Ciągła integracja jako proces
  - 1.2.1. Prezentacja praktyk programistycznych wymaganych do poprawnej implementacji CI, wykazanie, że CI to proces a nie tylko narzędzie do automatycznej kompilacji

2. Automatyzacja

- 2.1. Dlaczego warto automatyzować
  - 2.1.1. Przedstawienie korzyści płynących z automatyzacji zadań
- 2.2. Narzędzia
  - 2.2.1. Przedstawienie najpopularniejszych narzędzi umożliwiających automatyzację (Ant, Maven, Gradle i inne), ich zalet, wad oraz potencjalnych zastosowań.
- 2.3. Serwer ciągłej integracji

3. Opis dostępnych narzędzi dla CI (przegląd konkurencyjnych rozwiązań do Jenkinsa).

4. Jenkins overview:

- 4.1. Jenkins a Hudson
- 4.2. Instalacja
- 4.3. Wymagania
- 4.4. Konfiguracja
- 4.5. Dostęp i uprawnienia
- 4.6. Dobre praktyki, ciekawostki
- 4.7. Backupy
- 4.8. Narzędzia towarzyszące

5. Pluginy:

- 5.1. Zestaw absolutnie podstawowy

BOTTEGA IT minds

5.2. Pluginy rozwiązujące konkretne problemy

6. Analiza jakości kodu:

6.1. Findbugs,

6.2. Checkstyle/pmd

6.3. Pokrycie kodu testami i inne metryki

6.4. Integracja z narzędziem udostępniającym (a, b, c)

7. Testowanie

7.1. Wprowadzenie

7.1.1. Ogólne przedstawienie możliwości testowania za pomocą serwera CI

7.2. Korzyści

7.3. Testy jednostkowe

7.3.1. Przedstawienie możliwości wykonywania testów jednostkowych oraz sposobu pracy z wynikami

7.4. Testy funkcjonalne

7.4.1. Przedstawienie możliwości automatyzacji testowania funkcjonalnego za pomocą Selenium oraz RobotFramework'a, opis przebiegu i możliwości integracji z serwerem CI,

7.4.2. Przykłady użytkowania oraz pracy z wynikami

7.5. Testy wydajnościowe

7.5.1. Przedstawienie najpopularniejszych narzędzi umożliwiających przeprowadzanie testów wydajnościowych oraz jakie są możliwości ich integracji z serwerem CI

7.6. Użyteczne opcje

7.6.1. Opis użytecznych opcji wspomagających testowanie, oraz jak wygląda praca z wynikami testów(raportowanie błedów, opcje powiadamiania)

7.7. Optymalizacja

7.7.1. Przedstawienie praktyk i rozwiązań umożliwiających optymalizację wykonywania testów, aby wyniki były bardziej wiarygodne i szybciej dostępne

8. Statyczna analiza kodu

BO·TT·EGA
IT minds

- 8.1. Narzędzia
  - 8.1.1. Przegląd dostępnych narzędzi(PMD, Sonar, CheckStyle, FindBugs), jakie są ich zalety, wady oraz potencjalne zastosowania
- 8.2. Integracja
  - 8.2.1. Opis procesu integracji z serwerem CI, przykłady użytkowania oraz pracy z wynikami
- 8.3. Korzyści

9. "Distributed Builds" z użyciem Jenkinsa.

- 9.1. kiedy używać
- 9.2. jak skonfigurować

10. Automatyczny deployment z użyciem Jenkinsa

11. Uruchamianie testów automatycznych przykładowo: funkcjonalne lub integracyjne oraz wydajnościowe - np z wykorzystaniem frameworków do testów Python lub Java.

- 11.1. Cucumber
- 11.2. Capybara

12. Przykład procesu CI z użyciem Jenkinsa - czyli case-study od commit'a do release'a zawierające kroki:

- 12.1. Commit to dev branch,
- 12.2. Run unit test,
- 12.3. Run code analysis plugins,
- 12.4. Deploy to test environment,
- 12.5. Run integration test/performance test,
- 12.6. Build documentation,
- 12.7. Promote package to stable when all test passed.

13. Zalecane praktyki

- 13.1. Przykłady optymalizacji
  - 13.1.1. Przedstawienie przykładów oraz kluczowych praktyk i elementów które pozwolą uczestnikom na dalszą samodzielną optymalizację

BOTTEGA IT minds

13.2. Wzorce i anty-wzroce

13.2.1. Zaprezentowanie najczęściej występujących błędów projektowych oraz ogólnie sprawdzonych i rekomendowanych praktyk.

13.3. Repozytorium Maven'a

13.3.1. Przedstawienie możliwości wykorzystania repozytorium Maven'a w środowisku CI, jak również korzyści i wyzwań z tego płynących

13.4. Trudne problemy

13.4.1. Przedstawienie podstawowych i najczęściej występujących problemów oraz ich potencjalnych rozwiązania.

14. Zaawansowane praktyki

14.1. Rozproszone budowanie

14.1.1. Zaprezentowanie możliwości wykonywania planów w środowisku rozproszonym, jakie płyną z tego korzyści oraz jakie niesie to wyzwania

14.2. Integracja z systemami zewnętrznymi

14.2.1. Zaprezentowanie konkretnych przykładów użytkowania CI w środowisku Cloud czy też integracji z narzędziami zewnętrznymi, np.: Atlassian JIRA

14.3. Rozszerzanie funkcjonalności

14.3.1. Opis możliwości rozszerzania funkcjonalności popularnych serwerów CI za pomocą pluginów, tworzenie i konfiguracja

14.4. Współbieżność

14.4.1. Przedstawienie funkcjonalności współbieżnego wykonywania planów, przykłady zastosowań, korzyści i kosztów

14.5. Wspieranie repozytoriów

14.5.1. Omówienie wspieranych repozytoriów kodu źródłowego przez serwery CI, jakie oferują one funkcje pomocnicze, dodatkowo opis wsparcia dla DVCS(jak Git czy Mercurial), przykłady użytkowania

14.6. Automatyczna integracja baz danych

14.6.1. Przedstawienie koncepcji oraz korzyści z niej płynących, operacji bazodanowych podlegających automatyzacji oraz przykładowych rozwiązań /strategii

15. Nowe trendy i rozwiązania w CI

15.1. Wprowadzenie do idei continuous delivery

BOTTEGA IT minds

15.1.1. Wprowadzenie

15.1.1.1. Korzyści

15.1.1.2. Czego potrzebujemy

15.1.1.3. Zalecane praktyki

15.1.2. Przedstawienie zalecanych rozwiązań i pomysłów wspomagających stosowania Continuous Delivery

15.2. Case study istniejących implementacji

15.3. Pre-tested commit na przykładzie jenkinsa i gerrita