

Program szkolenia:

## **Continuous Integration – ciągła integracja w projekcie**

Informacje:

<b>Nazwa:</b>	<b>Continuous Integration – ciągła integracja w projekcie</b>
<b>Kod:</b>	<b>tools-CI</b>
<b>Kategoria:</b>	Narzędzia
<b>Grupa docelowa:</b>	developerzy DevOps
<b>Czas trwania:</b>	2 dni
<b>Forma:</b>	40% wykłady / 60% warsztaty

---

**Program szkolenia jest ogólną ramą - konkretne szkolenie poprzedzamy analizą przed-szkoleniową.**

## Szczegółowy program:

### 1. Ciągła integracja - wprowadzenie

1.1. Czym jest

1.2. Co daje

1.3. Czemu przeciwdziała

1.4. CI jako składowa procesu wytwarzania oprogramowania w firmie

### 2. Automatyzacja

2.1. Dlaczego warto

2.2. Popularne problemy przy migracji z procesu ręcznego

2.2.1. Sposoby radzenia sobie z nimi

### 3. Serwer ciągłej integracji - wprowadzenie

3.1. Fnkcje i zadania

3.2. Architektura/klasy rozwiązań - wady/zalety, rekomendowane przeznaczenie

3.2.1. Proste, często hostowane zewnętrznie - jak Travis

3.2.2. Złożone, najczęściej hostowane w firmie - jak Jenkins

### 4. Automatyczne testowanie kodu - kluczowy element CI

4.1. Testowanie automatyczne - potrzeba

4.2. Testowanie jednostkowe

4.3. Testowanie integracyjne

4.4. Testowanie funkcjonalne/akceptacyjne

4.5. Testowanie wydajnościowe

### 5. Satelickie narzędzia i procesy

5.1. Repozytorium kodu i praca z kodem

5.1.1. Praca na branchach - workflow

5.1.2. Przeglądu kodu (pre- i post-commit)

5.1.3. Pull requesty

5.1.4. Wsparcie narzędzi

5.2. Repozytorium artefaktów

5.3. Mechanizm budowania projektu

5.4. Automatyczne testowanie kodu - wydzielone jako osobny punkt

5.5. Badanie jakości kodu

5.5.1. Metryki kodu

5.5.2. Statyczna analiza jakości kodu

5.5.3. SonarQube

5.6. Wersjonowanie i zarządzanie wydaniem

5.7. Automatyzacja wdrożeń

5.8. Baza wiedzy

5.9. Issue tracking

## 6. Przydatne praktyki

6.1. Wzorce i anty-wzorce

6.2. Wizualizacja procesu

6.3. Zaangażowanie całego zespołu

6.4. Rozproszone budowania

6.5. Współbieżność

6.6. Integracja z systemami (i procesami) zewnętrznymi

6.7. Główne trudności i stosowane rozwiązania

## 7. Continuous Delivery

7.1. Czym jest

7.2. Co daje

7.3. Kiedy warto stosować

7.4. Wymagania wstępne dotyczące projektu

7.5. Przydatne techniki i narzędzia