

CONTINUOUS INTEGRATION – CIĄGŁA INTEGRACJA W PROJEKCIE INFORMATYCZNYM

KOD: CI

Profil uczestnika

Uczestnik:

- członek zespołu, w którym stosuje się lub będzie stosować się technikę continuous integration;
- ma doświadczenie w realizacji projektów na stanowisku kierownika projektu, team leadera, release managera, programisty lub osoby odpowiedzialnej za zarządzanie konfiguracją w projekcie;
- chce poznać podstawowe założenia stojące za continuous integration.

Korzyści ze szkolenia

1. **Zmniejsza się ryzyko w projekcie** – dzięki wdrożeniu continuous integration projekt jest pod ciągłą kontrolą i monitoringiem.
2. **Minimalizuje się ilość czynności, które należy wykonać w ramach procesu** – dzięki automatyzacji budowania oprogramowania i czynności wdrożeniowych.
3. **W projekcie zawsze mamy do czynienia z oprogramowaniem gotowym do wdrożenia** – dzięki zastosowaniu continuous integration, zespół jest zobligowany do tworzenia projektu, który umożliwia jego wdrożenie w dowolnym momencie.
4. **Interesariusze projektu mogą samodzielnie zbadać postęp prac w projekcie** – wymiernym wskaźnikiem, który określa postęp projektu, jest stan oprogramowania, które można wdrożyć.
5. **Większa pewność w odniesieniu do produktu** – continuous integration wymusza, aby wytwarzane oprogramowanie weryfikować pod kątem zaimplementowanych funkcji.

Parametry szkolenia

CZAS TRWANIA: 2 dni - 16 godzin.

FORMA ZAJĘĆ: Laboratorium CI - 30%, wykład – 70%.

WIELKOŚĆ GRUPY: do 10 osób.

Szczegółowy program

Moduły szkoleniowe	Nabyte wiedza i umiejętności, poruszane zagadnienia
Wprowadzenie do continuous integration	<ul style="list-style-type: none">• Czym jest ciągła integracja?• Nie tylko automatyczne budowanie projektu• Redukcja ryzyka• Automatyzacja powtarzalnych czynności• Tworzenie oprogramowania gotowego do wdrożenia• Ciągła integracja a inne praktyki wytwarzania oprogramowania• Częste zatwierdzenia kodu• Testy a ciągła integracja
Minimalizacja ryzyka w projekcie	<ul style="list-style-type: none">• Antywzorce i ich konsekwencje:<ul style="list-style-type: none">○ brak kodu gotowego do wdrożenia,○ późne wykrywanie błędów,○ utrudniona komunikacja w projekcie,○ niska jakość tworzonego oprogramowania.
Automatyczne budowanie	<ul style="list-style-type: none">• Dlaczego warto automatyzować proces budowania?• Izolowanie skryptów budujących od IDE• Spójna struktura katalogów w projekcie• Szybkie wykrywanie błędów• Serwery ciągłej integracji• Manualne skrypty budujące oprogramowanie• Metryki budowanego kodu

Bazy danych w continuous integration	<ul style="list-style-type: none">• Automatyzacja integracji z bazą danych• Lokalna baza danych dla programistów• Przechowywanie artefaktów bazodanowych w repozytorium plików• Administrator bazy danych w procesie ciągłej integracji
Ciągłe testowanie (ang. continuous testing)	<ul style="list-style-type: none">• Automatyzacja testów jednostkowych• Automatyzacja testów integracyjnych• Automatyzacja testów funkcjonalnych, akceptacyjnych• Strategie uruchamiania testów• Powtarzalność i niezależność testów
Ciągła weryfikacja (ang. continuous inspection)	<ul style="list-style-type: none">• Metryki kodu• Minimalizacja złożoności kodu• Ciągłe przeglądy kodu• Pokrycie kodu• Narzędzia weryfikacji jakości kodu
Ciągłe wdrażanie (ang. continuous deployment)	<ul style="list-style-type: none">• Co oznacza oprogramowanie gotowe do wdrożenia?• Oznaczanie wersji• Uruchamianie testów• Raporty powdrożeniowe• Wykorzystanie informacji zwrotnej• Wdrożenie procesu ciągłej integracji