

Kod szkolenia: **XP/CR**

Tytuł szkolenia: **XP: Code Review i dobre praktyki inżynieryjne**

Dni: 2

## Opis:

Szkolenie w sposób przekrojowy omawia ideę Extreme programming (XP) a w głównej mierze koncentruje się na procesie Code Review oraz dobrych praktykach inżynieryjnych.

## Umiejętności zdobywane podczas zajęć

- dostrzeganie problemów w kodzie
- skuteczna komunikacja problemów z kodem w zespole
- analiza kodu źródłowego i skupianie się na najważniejszych problemach
- interakcja z systemem Code Review
- automatyzacja wspierająca Code Review

## Wymagania

- dobra znajomość programowania w dowolnym języku programowania
- znajomość paradygmatu obiektowego
- umiejętność korzystania z systemu kontroli wersji GIT
- znajomość konceptu pisania testów automatycznych

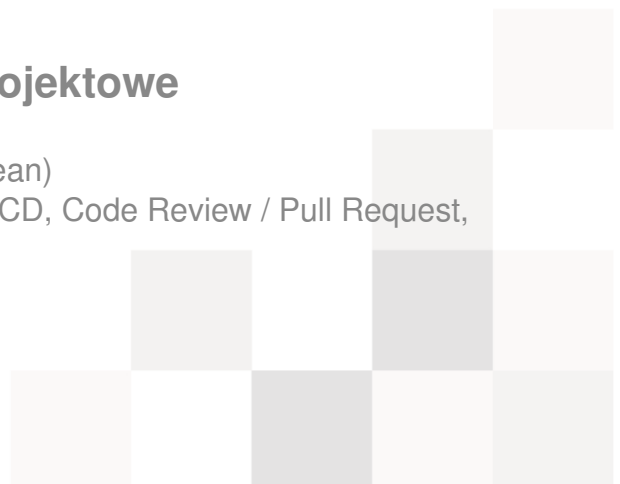
## Adresaci szkolenia

- inżynierowie oprogramowania
- inżynierowie jakości
- team-leaderzy
- architekci

## Program szkolenia:

### Praktyki developerskie a praktyki projektowe

- Praktyki projektowe (Scrum, Kanban, XP, Lean)
- Praktyki developerskie (TDD, S.O.L.I.D., CI/CD, Code Review / Pull Request, DevOps)



## Feature i bugfix branch

- GIT i GIT Flow
- Strategia budowania branchy
- Statyczna analiza kodu Pull Requestów
- Strategie repozytoriów: Fork, Central Repo, Army model

## Praktyka

- Spaghetti code
- Zależności w kodzie i między modułami
- S.O.L.I.D.
- TDD: test first vs test last development
- Czym w praktyce jest dług techniczny
- Ekosystem narzędziowy
- Refactoring legacy systemów
- Budowanie bazy wiedzy jako wynik Code Review
- Code Review jako element Collective Code Ownership

## FAQ

- Kto powinien być zaangażowany w Code Review?
- Ile czasu poświęcić na Code Review?
- Czy Code Review jest potrzebne przy TDD?
- Jak wielkość elementów w backlog wpływa na Code Review?
- Jak uniknąć wąskiego gardła w postaci Code Review?
- Czy zawsze należy robić Code Review?
- Czy można zautomatyzować proces Code Review?
- Na co warto zwrócić uwagę w Code Review?
- Jakiej wielkości zmiany poddawać Code Review?
- Jak komunikować problem z kodem?
- Jak blokować scalenie kodu, który zawiera błędy?
- Jak robić Code Review na żyjącym branchu?
- Notyfikacje i jak pozostać na bieżąco ze zmianami?
- Jak unikać silosów kompetencyjnych za pomocą Code Review?

## Code Review

- Systemy do Code Review, ich wady i zalety
- Unified diff vs side-by-side
- Komentarze i taski
- Pair programming vs Code Review
- Pull Request vs Merge Request
- Dobre praktyki Code Review



- Rozwiązywanie konfliktów w systemie kontroli wersji
- Składnia Markdown i reStructuredText

## Podstawy optymalizacji i wydajności systemów

- Złożoność obliczeniowa
- Złożoność pamięciowa
- Złożoność cyklometryczna
- Dług techniczny

## Automatyzacja Code Review

- Włączenie procesu CI/CD do Code Review
- Statyczna analiza kodu źródłowego
- Pipeline as a Code
- Quality Gates
- Triggerowanie statusów w Jirze
- Strategie mergowania (merge, rebase, squash+merge, squash+rebase)

## Jakość kodu

- SonarQube i SonarLint
- Wyciąganie wniosków z analizy statycznej
- Quality Gates
- Mierzenie pokrycia testami
- Testy mutacyjne

