

## Program szkolenia:

# Programowanie w parach - zorientowanie na cele strategiczne

## Informacje:

<b>Nazwa:</b>	<b>Programowanie w parach - zorientowanie na cele strategiczne</b>
<b>Kod:</b>	<b>Craft-practices-pair</b>
<b>Kategoria:</b>	Craftsmanship
<b>Odbiorcy:</b>	developerzy
<b>Czas trwania:</b>	1 dzień
<b>Forma:</b>	30% wykłady / 70% warsztaty

---

Programowanie w parach to nie wspólne wpatrywanie się w ekran.  
Programując w parach metodycznie możesz zwiększyć transfer wiedzy w organizacji, prowadzić szybszy onboarding nowych członków zespołu, rozwiązywać trudne problemy szybciej.

Podczas szkolenia poznasz praktyki, które działają oraz rozwiejesz popularne mity na temat programowania w parach.

## Zalety szkolenia:

- Zorientowanie na konkretny cel
- Podejście metodyczne
- Warianty dla różnych poziomów kompetencji

## Szczegółowy program:

### 1. Cele i dobranie do nich odpowiednich strategii

1.1. Transfer wiedzy

1.2. Redukcja ryzyka

1.3. Pokonanie blokad

1.4. Onboarding

### 2. Role - jak je grać dobrze i czego unikać

2.1. Driver - odpowiada na pytania: jak i co?

2.2. Navigator - odpowiada na pytania: dlaczego i co?

2.3. Podstawy neuronale

2.3.1. Rich brain vs Linear brain

2.3.2. Modele werbalne a kojarzenie wzorców

2.4. Układ - dobór taktyk w zależności od wiedzy na temat systemu lub domeny

2.4.1. Driver i Navigator są ekspertami

2.4.2. Driver posiada wiedzę a Navigator nie

2.4.3. Navigator posiada wiedzę a Driver nie

2.4.4. Ani Driver ani Navigator nie posiadają wiedzy

### 3. Kroki - model wzorcowy do dostosowania w zależności od układu

3.1. Cykl Kolba - model uczenia się ludzi dorosłych

3.2. Mapa kursu

3.2.1. Stos celów

3.2.2. Szanse na refaktoring

3.3. Korekty kursu

3.4. Jasne komunikowane intencje

**3.5. Wizualizacja kursu**

3.5.1. Pseudokod

3.5.2. Szkice

3.5.3. Nieformalny UML

3.5.4. UI

**4. Praktyki**

4.1. Rozpoczynamy od rozsądnie zdefiniowanych zdań

4.2. Umawiamy się na jeden cel w danym czasie

4.3. Trzymaj się ról

4.3.1. Navigator nie dyktuje kodu

4.3.2. Driver dąży do celu jak najszybciej

4.4. Pytania otwierające

4.5. Częsta synchronizacja z partnerem

4.6. Świątowanie małych zwycięstw

4.7. Zmiana ról co najmniej co pół godziny

**5. Zagadnienia dodatkowe**

5.1. Wątpliwości

5.2. Obiekcje

5.3. Ograniczenia

5.4. Wypaczenia

5.5. Wariacje i alternatywy

**6. Integracja z Test Driven Development**