

## Program szkolenia:

# Agile Software Engineering

### Informacje:

<b>Nazwa:</b>	<b>Agile Software Engineering</b>
<b>Kod:</b>	<b>Agile-engine</b>
<b>Kategoria:</b>	Agile Software Development
<b>Odbiorcy:</b>	Scrum Masters, developerzy, Product Owners, analitycy, management, architekci
<b>Czas trwania:</b>	3 dni
<b>Forma:</b>	50% wykłady / 50% warsztaty

Dyscyplina inżynierii oprogramowania ewoluowała wraz z wprowadzeniem zasad i praktyk Lean-Agile i DevOps. Nowe umiejętności i podejścia pomagają organizacjom dostarczać rozwiązania skoncentrowane na oprogramowaniu szybciej, bardziej przewidywalnie i o wyższej jakości.

Podczas tego trzydniowego, zorientowanego na warsztaty szkolenia, uczestnicy poznają podstawowe zasady i praktyki, które składają się na dyscyplinę Agile Software Engineering. Uczestnicy dowiadują się, w jaki sposób ciągły przepływ dostarczania wartości i wbudowana jakość są możliwe dzięki nowoczesnym praktykom, w tym praktykom technicznym XP (eXtreme Programming), procesowi i praktykom wytwarzania oprogramowania BDD (ang. Behavioural Driven Development, BDD) i rozwojowi opartemu na testach (ang. Test Driven Development, TDD).

Uczestnicy zapoznają się ze sprawdzonymi praktykami w zakresie uszczegółowienia, modelowania, projektowania, wdrażania, weryfikowania i sprawdzania poprawności historyjek użytkownika w ramach ciągłego dostarczania wartości w SAFe. Poznają praktyki zapewniające jakość i czystość kodu (między innymi: abstrakcja, enkapsulacja, programowanie intencyjne) i wzorce projektowe (SOLID, wzorce projektowe). Uczestnicy rozumieją także, w jaki sposób inżynieria oprogramowania pasuje do szerszego kontekstu rozwiązania i poznają ich rolę we współpracy przy kształtowaniu architektury rozwiązania i DevOps.

**Szkolenie umożliwia podejście do egzaminu Certified SAFe® Agile Software Engineer.**

### Zalety szkolenia:

- Przygotowanie do egzaminu: Certified SAFe® Agile Software Engineer.
- Poznanie znaczenia zwinnej inżynierii oprogramowania oraz leżące u jej podstaw wartości, zasady i praktyki.
- Stosowanie zasad Test-First w celu uzyskania zgodności między testami a wymaganiami.
- Zapewnienie wspólnego zrozumienia, dzięki rozwojowi opartemu na zachowaniu (ang. Behavioural Driven Development, BDD).
- Nauka stosowania modelowania zwinnego przy komunikacji.
- Nauka projektowania z uwzględnieniem kontekstu testowalności.
- Nauka tworzenia aplikacji o wysokiej jakości kodu i projekcie.
- Wykorzystanie infrastruktury testowej do automatycznego testowania.
- Nauka współpracy przy zamierzonej architekturze i powstającym projekcie rozwiązania.
- Nauka stosowania zasad Lean-Agile w celu optymalizacji przepływu wartości.
- Nauka budowy planu zwinnej inżynierii oprogramowania.

## Szczegółowy program:

1. Wprowadzenie do zwinnej inżynierii oprogramowania.
2. Łączenie zasad i praktyk z wbudowaną jakością.
3. Przyspieszenie przepływu.
4. Stosowanie intencjonalnej architektury.
5. Najpierw myślenie testowe.
6. Odkrywanie szczegółów historii użytkownika.
7. Tworzenie wspólnego zrozumienia z rozwojem opartym na zachowaniu (ang. Behavioural Driven Development, BDD).
8. Komunikacja z użyciem zwinnych modeli.
9. Systemy budowlane z jakością kodu.
10. Systemy budowlane z jakością projektowania.
11. Wdrażanie z jakością.