

---

**Blok I: Agile Software Development – Zamysły (2.0h)**

Celem modułu jest przedstawienie dwóch przeciwstawnych sposobów realizacji projektów wraz z omówieniem ich specyfiki i przeznaczenia. Uczestnik na podstawie własnych doświadczeń i ich konfrontacji z obserwacjami reszty grupy będzie mógł dokonać oceny, który z przedstawionych sposobów pełniej adresuje problemy występujące w jego organizacji (zespole, departamencie, firmie). W ramach tego modułu naszkicowana zostanie historia lekkich metod produkcji oprogramowania oraz zaprezentowany zostanie krótki ich przegląd.

1. Nieudane projekty - fakty i przyczyny - ujęcie ogólne
    - a. *ćwiczenie - zebranie i grupowanie najczęściej obserwowanych przez uczestników problemów związanych z realizacją projektów*
    - b. *ćwiczenie - czy Scrum jest w stanie pomóc w obserwowanych przez uczestników problemach w realizacji projektów? [ćwiczenie wykonywane na zakończenie szkolenia]*
  2. Zarządzanie niezarządzalnym
    - a. Ekonomiczna potrzeba planowania adaptacyjnego
    - b. Procesy preskryptywne a empiryczne (sposoby realizacji projektów: plan-wykonanie, wizja-eksploracja)
    - c. Planowanie adaptacyjne a podejście tradycyjne
    - d. Potrójne ograniczenie, miara sukcesu projektu
  3. Ujęcie historyczne: Toyota Production System, Lean Development
  4. Agile Software Development
    - a. manifest
    - b. pryncypia
  5. Przegląd lekkich metod, wzajemne ich umiejscowienie w ramach ruchu Agile
- 

**Blok II: Scrum – Skrócone wprowadzenie do metody (2.0h)**

Celem modułu jest zapoznanie słuchacza z podstawowymi założeniami metody zarządzania projektami Scrum, zarysowanie tła i okoliczności jej powstania. Moduł wprowadza uczestnika w najważniejsze elementy struktury Scrum, definiuje role projektowe i związane z nimi zakresy odpowiedzialności. Przedstawione zostaną także kwestie związane z planowaniem i monitorowaniem postępów prac w ramach pojedynczego cyklu produkcyjnego.

1. Wstęp do Scrum-a  
(definicja, rys historyczny, oczekiwane korzyści, najczęstsze przyczyny niepowodzeń podczas wdrażania)
  2. Role i zakresy odpowiedzialności
  3. Minimalistyczne narzędzia (Rejestr Produktowy, Rejestr Zadaniowy, Rejestr Blokad)
  4. Przebieg pojedynczego cyklu produkcyjnego (Sprintu)
    - a. ceremonia – zebrania zespołowe ich cele i przebieg
    - b. monitorowanie stanu zaawansowania prac w oparciu o wykres spalania
  5. Monitorowanie przebiegu projektu, definicja prędkości zespołu, rozszerzony wykres spalania
-

### Blok III: Scrum – Kompleksowe przedstawienie metody (8.0h)

Celem modułu jest kompleksowe zapoznanie uczestników z metodą realizacji projektów Scrum oraz udostępnienie wszystkich narzędzi i praktyk niezbędnych do wdrożenia tej metody w zespole projektowym. W trakcie wykładu przeplatane ćwiczeniami praktycznymi, uczestnicy nabędą umiejętności stosowania Scruma w realizacji własnych projektów software-owych.

1. Wstęp do Scrum-a  
(definicja, rys historyczny, oczekiwane korzyści)
2. Podstawowe elementy procesu i praktyki scrumowe  
(omówienie cyklu produkcyjnego, wprowadzenie do ceremonii)
3. Role i zakresy odpowiedzialności; minimalistyczne narzędzia  
(omówienie rejestrów stosowanych w Scrum-ie, przyporządkowanie tych rejestrów do ról; omówienie zakresów odpowiedzialności i kompetencji; definicja zespołu i otoczenia projektowego)
  - a. *ćwiczenie - identyfikowanie otoczenia projektu, identyfikowanie potencjalnych kandydatów do wypełniania ról scrumowych*
4. Przygotowanie do realizacji projektu, Rejestr Produktowy
5. Przygotowanie do realizacji Sprintu, definiowanie Celu Sprintu, praca z Rejestrem Zadaniowym  
(planowanie iteracji, przebieg spotkań i ich cele, definiowanie listy zadań)
6. Realizacja Sprintu  
(omówienie ceremonii przestrzeganej podczas trwania iteracji, omówienie zakresów kompetencji i obowiązków członków zespołu; osiąganie optymalnego rytmu pracy)
  - a. *ćwiczenie - codzienny scrum*
  - b. *ćwiczenie - samoorganizacja pracy*
7. Blokady, praca z Rejestrem Blokad
8. Monitorowanie przebiegu iteracji, wykres spalania  
(narzędzia stosowane do monitorowania i analizy stanu zaawansowania projektu)
9. Zamknięcie Sprintu i odbiór produktu  
(omówienie spotkań zamykających iterację, ocena kompletności wykonania planu, kolaudacja - odbiór produktu)
  - a. *ćwiczenie - definicja gotowości produkcyjnej (done-done)*
10. Analiza i adaptacja – Retrospekcja  
(organizacje uczące się a organizacje bez pamięci, samoświadomość i doskonalenie się zespołu)
  - a. *ćwiczenie – retrospekcja*
  - b. *ćwiczenie – przygotowanie Rejestru Zespołowego*
11. Definicja prędkości zespołu, dobór czasu trwania iteracji  
(czynniki mające wpływ na czas trwania iteracji)
12. Monitorowanie przebiegu projektu i raportowanie stanu prac, rozszerzony wykres spalania

---

### Blok IV: Techniki gromadzenia wymagań, szacowania i planowania pracy w metodach lekkich (4.0h)

Celem modułu jest zapoznanie słuchacza z technikami zarządzania wymaganiami, szacowania i planowania pracy stosowanymi w lekkich metodach produkcji oprogramowania. W trakcie ćwiczeń praktycznych uczestnicy będą samodzielnie konstruować Rejestr Produktowy oraz szacować (estymować) pracę.

1. Zastosowanie historii użytkownika, opowieści i tematów  
(definicja historii użytkownika jako alternatywy dla książki wymagań, omówienie roli użytkownika w systemie, omówienie kryteriów/testów akceptacyjnych jako dopełnienia historii użytkownika)

2. Kryteria oceny kompletności historii użytkownika
  3. Techniki pozyskiwania historii użytkownika
    - a. *ćwiczenie - definiowanie historii użytkownika*
  4. Szacowanie, jednostki estymacyjne, techniki szacowania w metodach lekkich  
(definicja i natura procesu szacowania w projektach software-owych, szacowanie czasu a szacowanie rozmiaru, techniki szacowania, stosowane jednostki i zakresy)
    - a. *ćwiczenie - Planning Poker*
- 

## **Blok V: Planowanie przedsięwzięcia, zarządzanie projektem i produktem w metodzie Scrum (4.0h)**

Celem modułu jest wprowadzenie uczestnika w rolę i obowiązki właściciela produktu. W trakcie wykładu, ćwiczeń praktycznych oraz studiów przypadku uczestnicy dowiadują się jak planować cykle produkcyjne, jak tworzyć i efektywnie zarządzać wymaganiami, jak monitorować postępy prac, analizować dostarczone rozwiązania i definiować nowe cele, jak współpracować z interesariuszami projektu i zespołem projektowym tak, aby wytwarzane oprogramowanie reprezentowało najwyższą możliwą wartość.

1. Product Owner – Rola i oczekiwania
  2. Przygotowanie do realizacji projektu
    - a. *ćwiczenie – przygotowanie Project Charter-a – jednostronicowego dokumentu opisującego przedsięwzięcie*
  3. Planowanie produkcji – planowanie w podejściu agile, horyzonty planistyczne  
(poziomy i szczegółowość planowania, planowanie metoda *just-in-time*, linia czasu, definicja i zastosowanie idei *release-u*)
  4. Zarządzanie wymaganiami i budżetowanie projektu  
(planowanie dla wartości / *business value*, kryteria i techniki nadawania i zarządzania priorytetami)
    - a. *ćwiczenie - określanie i analiza wartości biznesowej historii użytkownika*
  5. Scrum w przedsięwzięciach wieloprojektowych i wielozespołowych
    - a. Planowanie i organizacja pracy w przedsięwzięciach wieloprojektowych i utrzymaniowych
    - b. Planowanie i synchronizacja pracy wielu zespołów projektowych, zespoły rozproszone
    - c. Scrum w procesach kaskadowych
  7. Monitorowanie przebiegu prac i raportowanie stanu prac, rozszerzony wykres spalania, iteracyjny odbiór produktu, adaptacja planu
  8. Kredyt techniczny
  9. Odbiór produktu i zamknięcie projektu
  10. Studium przypadku
- 

## **Blok VI: Praca z samzarządzającym się zespołem (4.0h)**

Celem modułu jest wprowadzenie uczestnika w rolę i obowiązki lidera zespołu projektowego stosującego metodę Scrum. W trakcie wykładu oraz ćwiczeń praktycznych uczestnicy dowiadują się na jakie aspekty pracy grupy należy zwracać uwagę, w jaki sposób prowadzić zebrania zespołowe aby zapewnić ich maksymalną efektywność, w jaki sposób analizować i optymalizować procesy i procedury którymi grupa posługuje się w codziennej pracy.

1. Scrum Master – Rola i oczekiwania
2. Przeszkody, blokady, praca z Rejestrem Blokad
3. Myślenie systemowe

4. Definicja i przykłady nieefektywności procesów i organizacji pracy
    - a. *ćwiczenie - analiza zależności wewnętrzzespołowych i przepływu produktu przez zespół (value stream map)*
  5. Charakterystyka samzarządzającego się zespołu, cykl rozwoju zespołu
  6. Jednostka w samzarządzającym się zespole, motywacja i rozwój
  7. Facylitacja
    - a. Prowadzenie spotkań scrumowych
    - b. Współpraca w zespole i zespołowe podejmowanie decyzji – techniki i narzędzia
    - c. *ćwiczenie – przygotowanie Team Charter-a*
  8. Praca w środowiskach wielozespołowych
    - a. Praca z zespołem rozproszonym
  9. Przestrzeń zespołu - aranżacja środowiska pracy
  10. Studium przypadku
- 

### **Blok VII: Wdrażanie metodyki Scrum w organizacji (4.0h)**

Celem modułu jest omówienie aspektów związanych z wdrażaniem metody Scrum w organizacji (zespole, departamencie, firmie), przedstawienie kolejnych etapów wdrożenia oraz uczulenie uczestników na najczęściej występujące problemy. Dodatkowo przedstawione zostaną najpopularniejsze narzędzia software-owe wspomagające organizację pracy w projektach realizowanych lekkimi metodami.

1. Złamane paradygmaty i zmiana kultury organizacyjnej
  2. Podejścia do wdrażania lekkich metod, motywatory
    - a. podejście top-down a podejście bottom-up
  3. Etapy wdrażania  
(ewaluacja procesu wdrażania, najczęstsze przyczyny niepowodzeń podczas wdrażania)
  4. Rola konsultantów w procesie wdrażania
  5. Skalowalność Scruma, warianty
  6. Przegląd narzędzi wspomagających wdrożenie
- 

### **Blok VIII: Przegląd technik Extreme Programming (1.0h)**

Celem modułu jest przedstawienie słuchaczowi przeglądu praktyk eXtreme Programming najczęściej adaptowanych przez zespoły stosujące metodę Scrum.

1. Ciągła integracja
  2. Test-Driven Development
  3. Refaktoryzacja
  4. Programowanie w parach
- 

### **Blok IX: Warsztat - Iteracyjna realizacja projektu, symulacja (1.0h)**

(ćwiczenie praktyczne - uczestnicy wezmą udział w czterech symulowanych cyklach produkcyjnych, w praktyczny sposób stosując większość zagadnień omówionych podczas szkolenia)

---

### **Blok XI: eXtreme Programming – Wprowadzenie (8.0h)**

Celem modułu jest zapoznanie uczestników z podstawowymi technikami inżynierskimi eXtreme Programming. Szczególny nacisk położony jest na Test-Driven Development jako optymalne podejście do realizacji projektów w duchu ruchu Agile.

1. Manifest Agile
  - a. Znaczenie postulatów
  - b. Priorytety dla zespołu
  - c. Agile a eXtreme Programming
  - d. Konsekwencje dla projektów informatycznych
2. Wymagania użytkownika w eXtreme Programming
  - a. Budowanie wizji projektu
  - b. User Stories – historie użytkownika
  - c. Testy akceptacyjne
3. Inżynieria oprogramowania w eXtreme Programming
  - a. Continuous Integration – ciągła integracja
  - b. Programowanie w parach
4. Test-Driven Development
  - a. Cykl wytwarzania oprogramowania w TDD
  - b. Wzorce i zasady TDD
5. Rola testów
  - a. Automatyzacja testów
  - b. Typy testów i wzorce
  - c. Testowanie zależności
  - d. Testy jednostkowe i integracyjne
6. Projektowanie oprogramowania – Simple Design
  - a. Simple Design a Big Design Up Front
  - b. Programowanie obiektowe w kontekście TDD
  - c. Projektowanie klas – zasady SOLID
7. Refaktoryzacja
  - a. Refaktoryzacja kodu
  - b. Refaktoryzacja testów
8. Uwagi praktyczne

---

## **Blok XII: eXtreme Programming – Warsztaty (16.0h – 24.0h)**

Celem modułu jest zapoznanie uczestników z podstawowymi technikami inżynierskimi eXtreme Programming. Szczególny nacisk położony jest na Test-Driven Development jako optymalne podejście do realizacji projektów w duchu ruchu Agile. W trakcie ćwiczeń uczestnicy będą mogli na przykładach zapoznać się z poszczególnymi praktykami oraz odnieść się do ich zastosowania w swoich projektach.

1. Manifest Agile
  - a. Historia
  - b. Znaczenie postulatów
  - c. Agile a eXtreme Programming
  - d. Priorytety dla zespołu
  - e. Konsekwencje dla projektów informatycznych

- f. Korzyści wynikające z podejścia Agile
  - g. Role w zespole
2. Wymagania użytkownika w eXtreme Programming
- a. Budowanie wizji projektu
  - b. User Stories – historie użytkownika
  - c. Wskazówki dotyczące tworzenia historii użytkownika
  - d. *ćwiczenie 1 – gromadzenie wymagań użytkownika*
  - e. Testy akceptacyjne
  - f. Metryki jakości kodu
3. Inżynieria oprogramowania w eXtreme Programming
- a. Continuous Integration – ciągła integracja
  - b. Automatyzacja testów akceptacyjnych
  - c. Programowanie w parach
4. Test-Driven Development
- a. Cykl wytwarzania oprogramowania w TDD
  - b. Wzorce i zasady TDD
  - c. *ćwiczenie 2 – cykl wytwarzania oprogramowania w TDD*
  - d. Wyzwania dla TDD
5. Rola testów
- a. Automatyzacja testów
  - b. Typy testów i wzorce
  - c. Testowanie zależności
  - d. Testowanie zachowań i testowanie stanu
  - e. *ćwiczenie 3 – testowanie zależności*
  - f. Testy jednostkowe i integracyjne
6. Projektowanie oprogramowania - Simple Design
- a. Simple Design a Big Design Up Front
  - b. Programowanie obiektowe w kontekście TDD
  - c. Projektowanie klas – zasady SOLID
  - d. Zastosowanie UML w ewolucyjnym wytwarzaniu oprogramowania
7. Refaktoryzacja
- d. Refaktoryzacja kodu
    - a. Techniki
    - b. Sygnały ostrzegawcze
  - e. Refaktoryzacja testów
    - a. Techniki
    - b. Sygnały ostrzegawcze
  - f. Refaktoryzacja do wzorców projektowych
  - g. *ćwiczenie 4 – refaktoryzacja kodu produkcyjnego i testów*
8. Uwagi praktyczne