

# Programowanie ekstremalne (XP)

Ewa Namysł

Uniwersytet Śląski

10 marca 2022

# Spis treści

Czym jest XP?

Podstawowe wartości XP

Praktyki XP

Wady i zalety XP

Podsumowanie

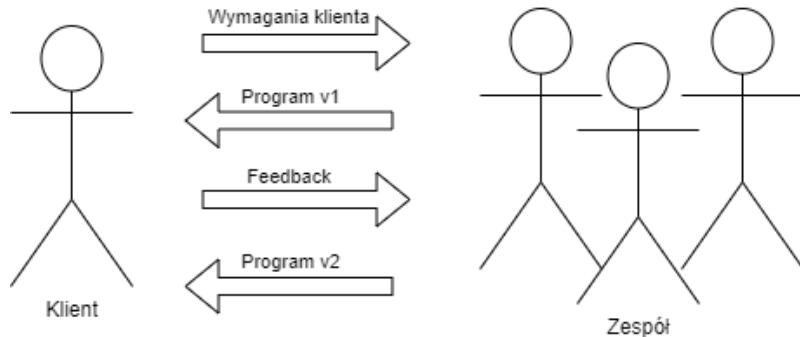
# Czym jest XP?

## **Programowanie ekstremalne (eXtreme programming):**

Metodologia tworzenia oprogramowania, kładąca nacisk na dostosowanie się do zmieniających się w czasie wymagań klienta.

Jedna z metod programowania zwinnego,  
rozwinęta w latach dziewięćdziesiątych przez Kenta Becka.  
Zbiór praktyk XP został sformalizowany w jego książce  
*Extreme Programming Explained*.

## Przykład



## Charakterystyczne cechy XP

- Klient jest częścią zespołu, obie strony pozostają w ciągłym kontakcie.
- Brak dokumentacji projektu, skupienie na informacji zwrotnej od klienta.
- Małe, ale częste wydania wersji.
- Minimalizacja złożoności oprogramowania, rozwijanie go w miarę potrzeb.
- Programiści pracują w parach.
- XP stosowane jest głównie w małych i średniej wielkości projektach.

# Podstawowe wartości programowania ekstremalnego

- Komunikacja
- Prostota
- Feedback
- Odwaga

## Komunikacja

W programowaniu ekstremalnym nie wykorzystuje się zazwyczaj dokumentacji, dlatego jednym z najważniejszych aspektów jest komunikacja nie tylko pomiędzy członkami zespołu, ale też z klientem.

Celem XP jest przedstawienie założeń projektu w jak najprostszy sposób, dzięki czemu programiści mają spójną wizję.

# Prostota

Budowanie systemów w prosty sposób.  
Unikanie złożoności, skupienie się na aktualnych potrzebach.  
Wykorzystywanie zasady KISS (*'Keep It Simple, Stupid!'*).



## Feedback

Produkt rozwijany jest w krótkich iteracjach  
i dostarczany do klienta.

Postępy pracy są oceniane przez klienta, a feedback  
wykorzystywany jest przy wprowadzaniu zmian.

Ważnym aspektem jest zadawanie pytań, stąd istotna jest bliska  
kooperacja klienta i zespołu developerskiego.

# Odwaga

Umiejętność podejmowania decyzji.  
Szczerość w kontaktach między członkami zespołu i klientem.

# Praktyki programowania ekstremalnego

W sumie dwanaście praktyk w czterech obszarach.

## Omawiane obszary:

- Planowanie
- Projektowanie
- Testowanie
- Programowanie

# Planowanie

W XP planowany jest wyłącznie krótki odcinek czasu w najbliższej przyszłości.

Ze względu na to, że projekt długofalowo może wymagać zmian lub potrzeby klienta ulegną zmianie, proces planowania dotyczy tylko najbliższej iteracji.

## Praktyki planowania

- Gra planistyczna - estymacja i określenie zakresu pracy na początku każdej iteracji.
- Metafora - przedstawienie projektu w jasny sposób, zrozumiały zarówno dla zespołu programistów, jak i klienta.
- Klient jako część zespołu - ciągły kontakt, udostępnianie klientowi najnowszej wersji tworzonego oprogramowania, feedback.

# Projektowanie

Projektowanie rozwiązań odbywa się tuż przed etapem programowania ze względu na ich ścisłe powiązanie.

Prostota projektu ma się przekładać na prostotę programowania.

Brak dokumentacji wymaga jasnego i przejrzystego kodu, który sam w sobie staje się dokumentacją.

## Praktyka projektowania

- Prostota - oprogramowanie projektowane w jak najprostszym możliwym sposobie, usuwanie złożoności.

# Testowanie

Wykorzystywanie testów jednostkowych pisanych przez programistów w celu sprawdzenia poprawności ich kodu.

Wykorzystywanie testów funkcjonalnych, sprawdzających poprawność funkcjonalności zleconych przez klienta.



## Praktyka testowania

- Testowanie przed programowaniem - zautomatyzowane testy pisane są przed implementowaniem funkcjonalności.

# Programowanie

Najważniejszy etap w XP.

Programowanie zajmuje najwięcej czasu  
i jest kluczowe dla projektu.

## Praktyki programowania

- Częste wydania - nowe wersje dla klienta w krótkich odstępach czasowych.
- Refaktoryzacja - uporządkowanie struktury kodu, zachowanie spójności i klarowności.
- Standard kodowania - wspólne dla wszystkich programistów konwencje pisania kodu.
- Ciągła integracja - nowe funkcjonalności integrowane są jak najczęściej z całością systemu.

## Praktyki programowania

- Współwłasność kodu - każdy programista może dokonywać zmian w dowolnym miejscu i na dowolnym etapie rozwijania projektu.
- Praca w parach - jedna osoba programuje, druga osoba sprawdza i sugeruje nowe rozwiązania.
- Brak nadgodzin - programiści pracują maksymalnie 40 godzin tygodniowo.

## Zalety XP:

- Priorytetyzacja zadań, prostota, rozwijanie projektu "od ogółu do szczegółu".
- Krótkie iteracje i częstsze wydania, które przekładają się na szybszy feedback i wyłapywanie nieścisłości.
- Programowanie w zespołach dwuosobowych pozwala na szybsze rozwiązywanie problemów programistycznych.
- Poleganie na komunikacji werbalnej, co może być zaletą, ale...

## Wady XP:

- ...również wadą, ponieważ nie istnieje formalna dokumentacja.
- Dalsze rozwijanie projektu bez dokumentacji może okazać się problematyczne.
- Projektowanie rozwiązań biorąc pod uwagę tylko krótką perspektywę czasową, może skutować długim technologicznym.
- Ciągła dostępność klienta w celu odpowiadania na pytania developerów bywa niemożliwa dla zleceniodawcy.
- Zespół programistów powinien wykazywać się umiejętnościami interpersonalnymi, pozwalającymi na ścisłą kooperację.

## Podsumowanie

Programowanie ekstremalne pozwala na stworzenie w krótkim czasie kolejnych wersji produktu dla klienta.

Wymaga jednak od zespołu dobrych umiejętności komunikacyjnych i technicznych, a od klienta zaangażowania w proces tworzenia.

## Bibliografia:

K. Beck, *Extreme Programming Explained: Embrace Change*, Addison-Wesley, 2018.

D. Wells, *Extreme Programming*,  
<http://www.extremeprogramming.org> (dostęp marzec 2022).

A. Kolm, *Zarządzanie projektami IT*,  
<http://zarzadzanieprojektami.it/> (dostęp marzec 2022).