

# Programowanie Zespołowe

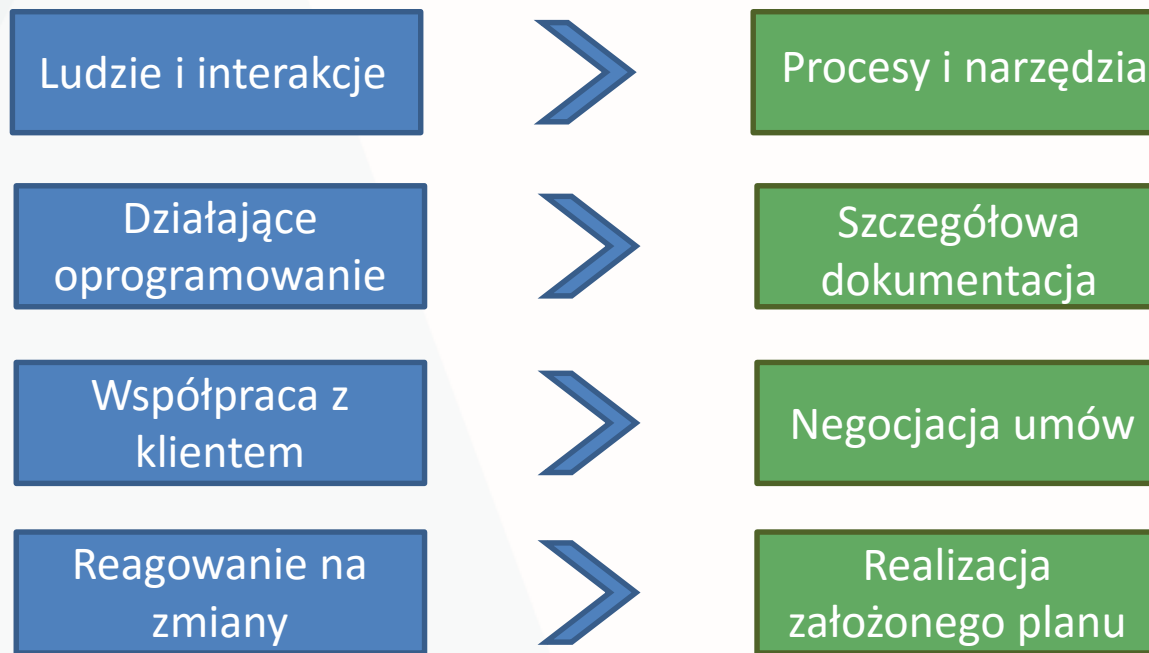
Programowanie zwinne

dr Rafał Skinderowicz  
mgr inż. Michał Maliszewski

## Programowanie zwinne

- Grupa metodyk wytwarzania oprogramowania oparta na modelu iteracyjno-objektowym
- Powstała w odpowiedzi na wysoki wskaźnik niepowodzeń w projektach prowadzonych metodami klasycznymi
- Wymagania klienta ewoluują w czasie działania projektu
- Zdyscyplinowane zarządzanie projektem
- Samoorganizujący się zespół
- Bezpośrednia, częsta komunikacja
- Szybkie wytwarzanie oprogramowania wysokiej jakości

# Manifest Agile



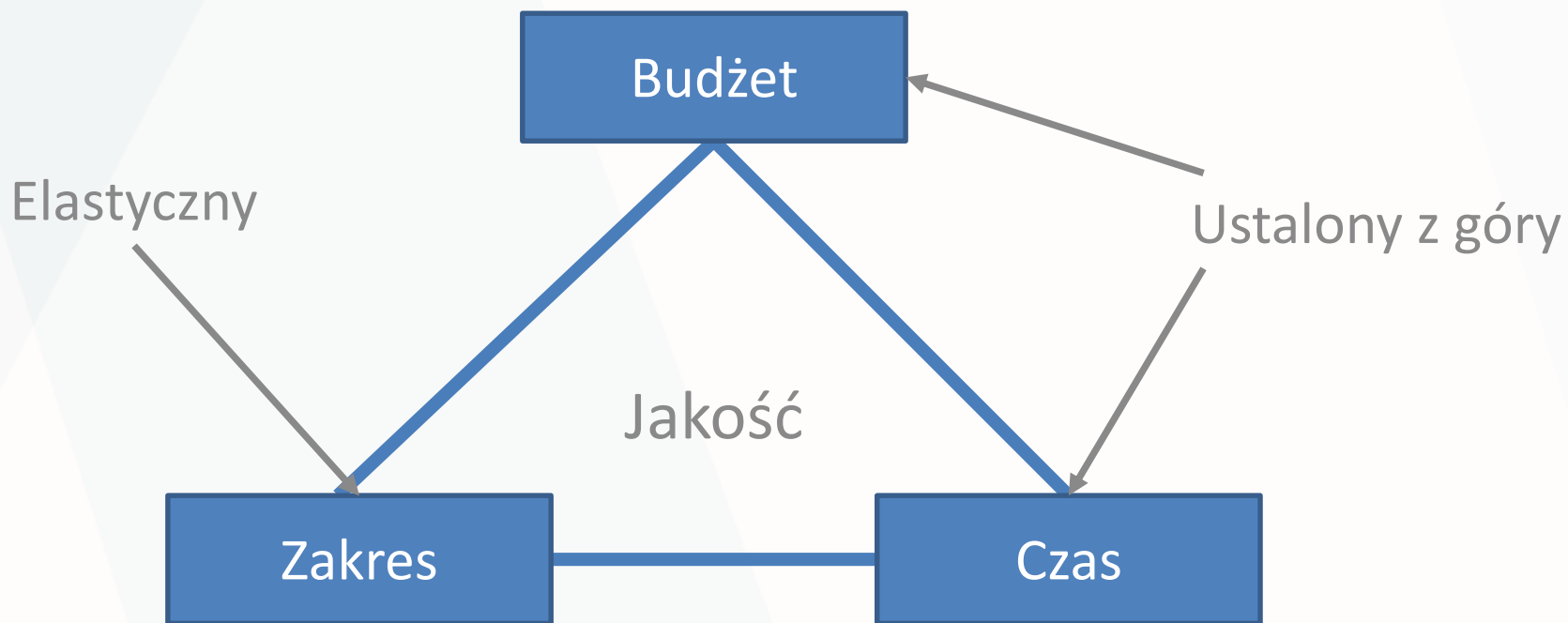
## 12 Zasad Agile

- Zadowolenie klienta dzięki szybkim dostawom użytecznego oprogramowania
- Zmiany wymagań nawet na późnym etapie rozwoju projektu
- Działające oprogramowanie jest często dostarczane klientowi
- Bliska współpraca biznesu z zespołem projektowym
- Projekt jest realizowany przez zmotywowanych ludzi, którzy mają pełne zaufanie
- Rozmowy bezpośrednie są najlepszą formą komunikacji
- Działające oprogramowanie jest podstawową miarą postępu prac

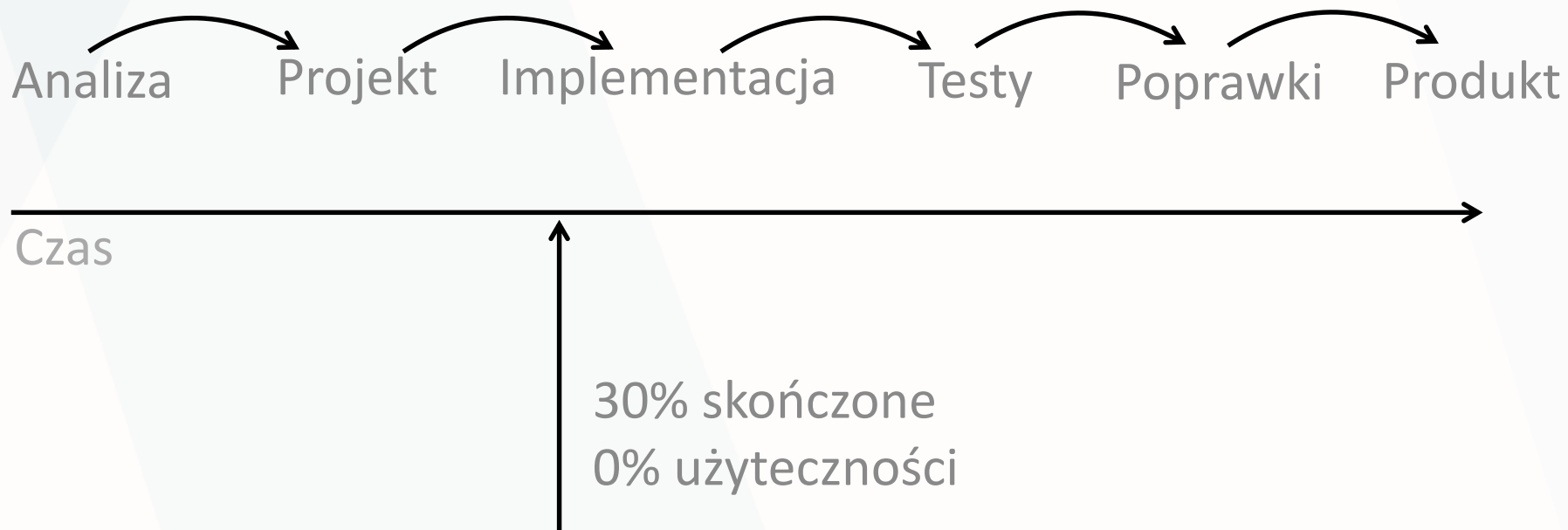
## 12 Zasad Agile c.d.

- Zrównoważony tryb prac, które zespół jest w stanie utrzymywać cały czas
- Ciągła dbałość o jakość techniczną i wysoki poziom architektury projektowanego systemu
- Kluczowa jest prostota, minimalizacja ilości koniecznej pracy
- Samoorganizujące się zespoły
- Regularna adaptacja zespołu do zachodzących zmian

## Trójkąt zależności

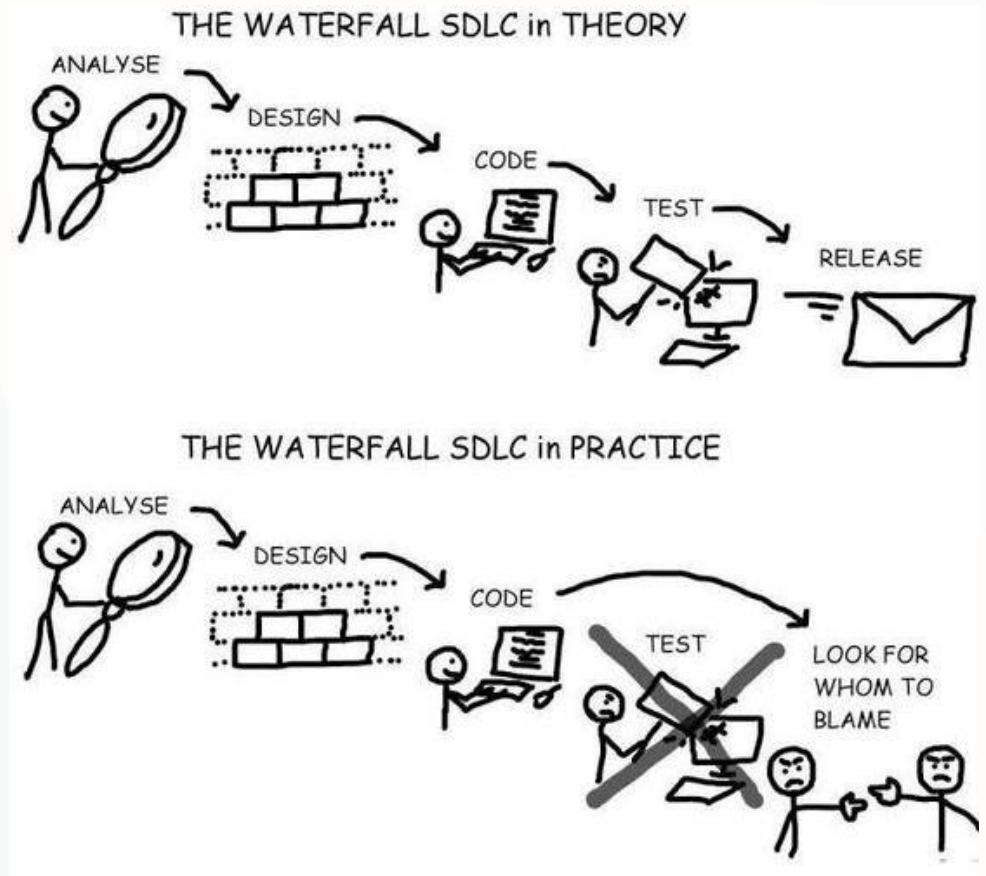


# Model kaskadowy



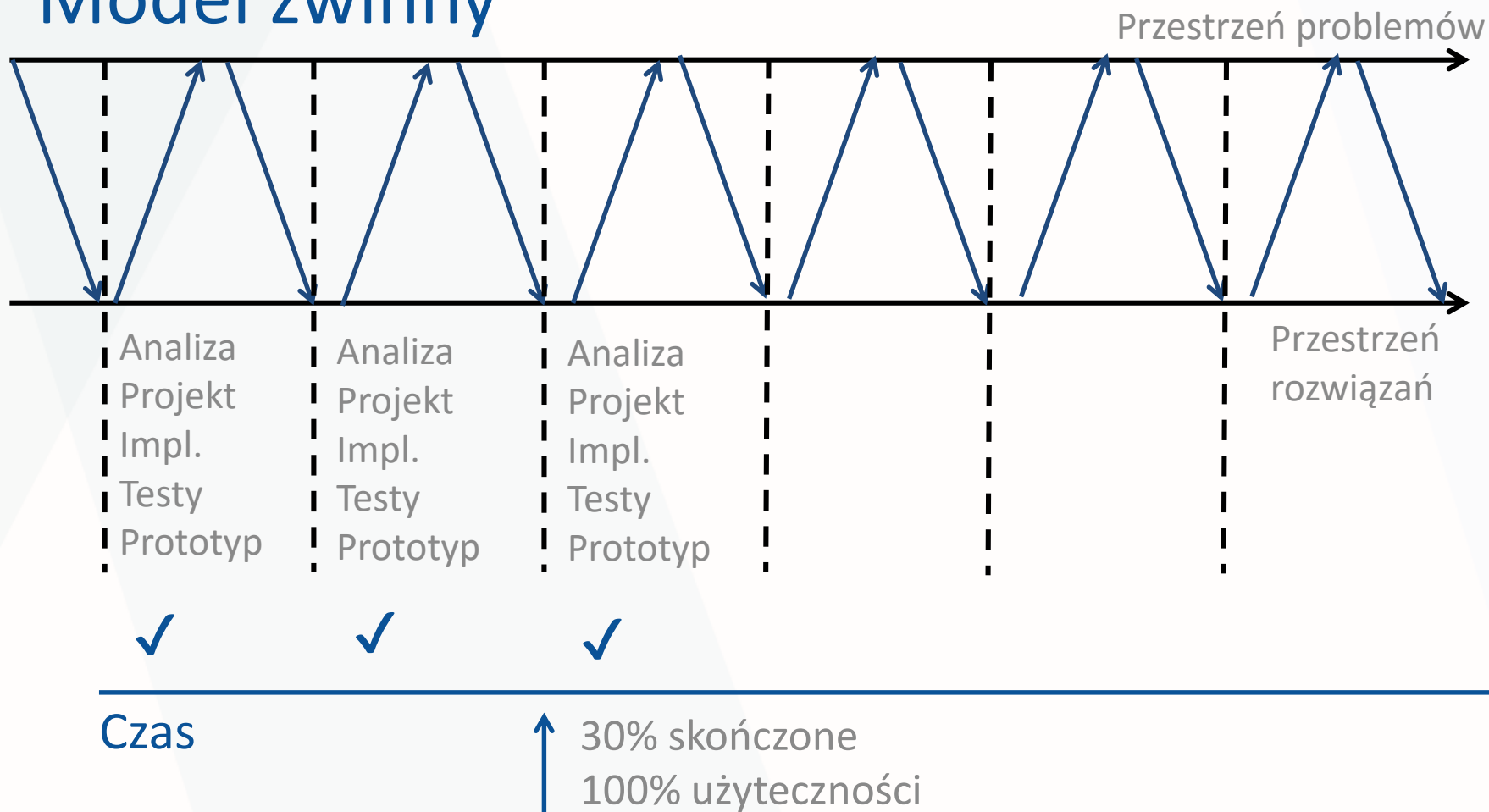
# Model kaskadowy

## Teoria vs Praktyka

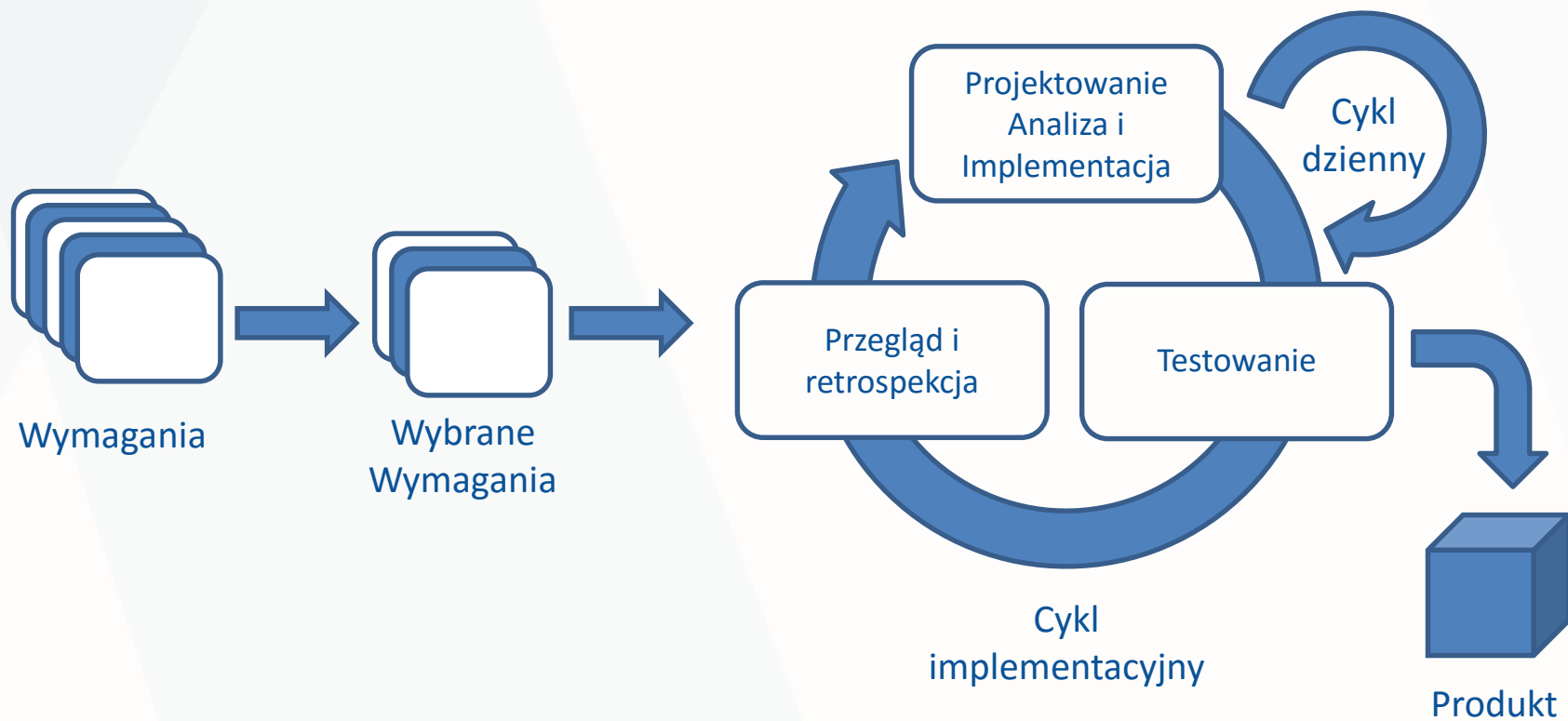




# Model zwinny

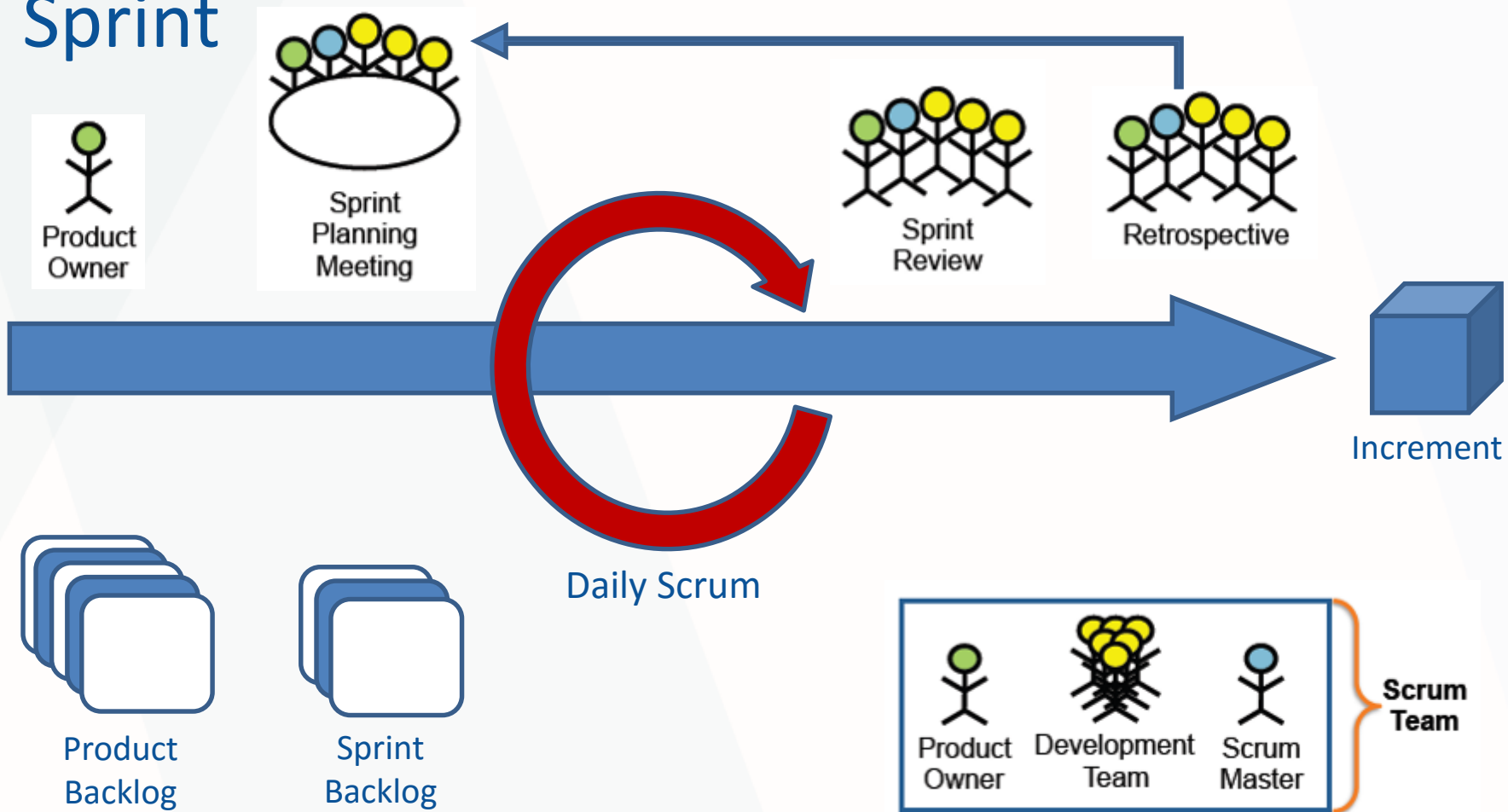


# Scrum





# Sprint



## Product Owner

- Zajmuje się stroną biznesową
- Definiuje wymagania
- Tworzy i dba o wymagania projektowe (Product Backlog)
- Jest odpowiedzialny za wartość produktu

## Scrum Master

- Dbą o prawidłowy przebieg procesu powstawania projektu
- Wdraża wartości, praktyki i zasady Scrum w organizacji
- Chroni zespół
- Rozwiązuje problemy z którymi zespół nie może sobie sam poradzić
- Motywuje zespół
- Szkoli i wspiera zespół



## Zespół

- Organizuje siebie jako zespół oraz swoją pracę
- Dostarcza produkt wysokiej jakości w każdej iteracji
- Jest uniwersalny (wielofunkcyjny)
- Współpracuje z PO (Product Owner) w celu zwiększenia wartości produktu
- Zazwyczaj składa się z 3-9 osób

## Zespół c.d.

- Zespół dostarcza i decyduje o rozwiązaniach zastosowanych w projekcie
- Jest odpowiedzialny za jakość
- Zespół jest motywowany i wspierany przez menadżerów





## Product Backlog

- Uporządkowana lista wymagań
- Jedyne źródło informacji o planowanych zmianach w produkcie
- Jest ogólnodostępny



## Sprint Backlog

- Wymagania wybrane z Product Backlogu
- Stworzone przez zespół w trakcie planowania
- Zawiera funkcjonalności które zespół planuje dostarczyć w danej iteracji

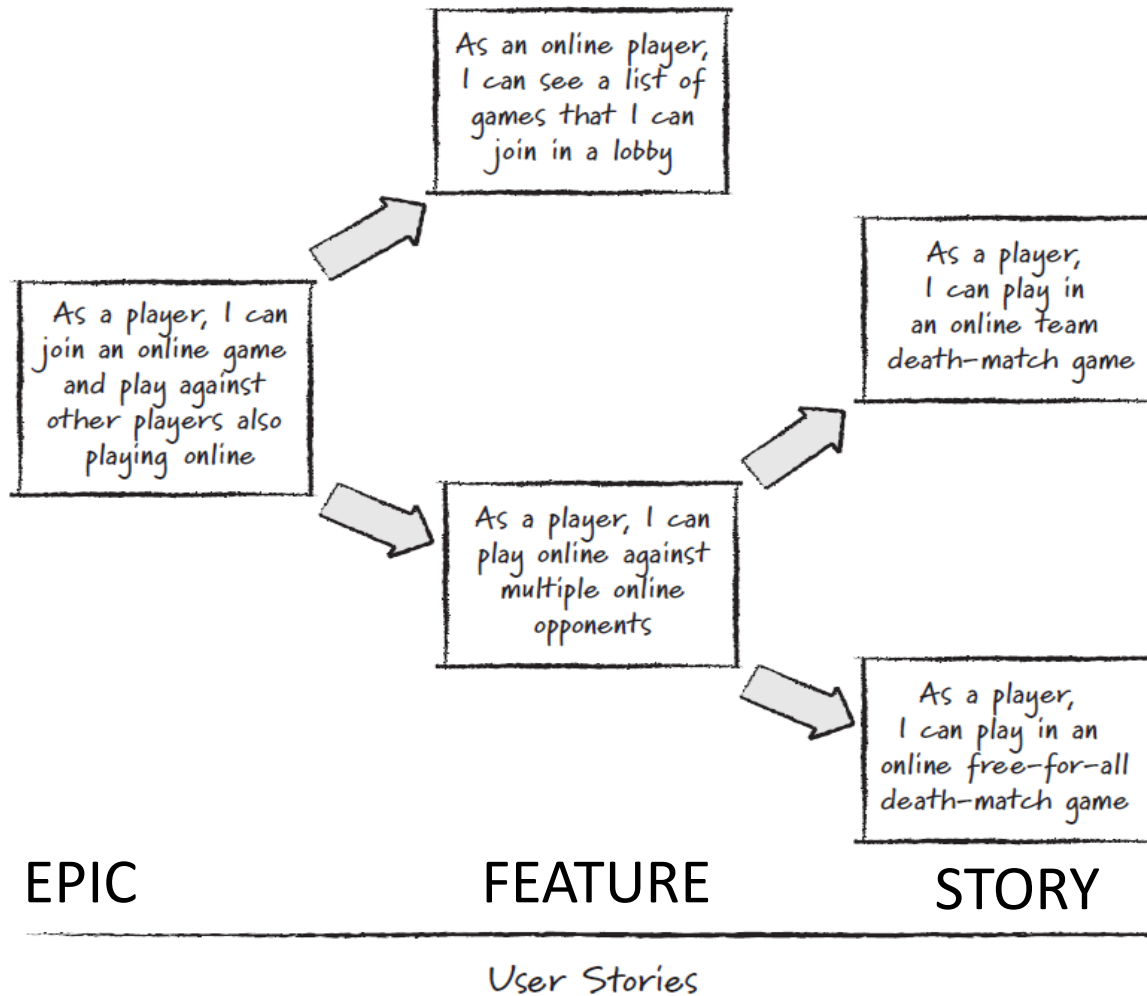
## User Stories

- Krótki opis wyjaśniający o co chodzi w zadaniu
- User Stories mogą być rozbijane na pojedyncze zadania
- Standardowo schemat każdej historii wygląda następująco:

Jako <rola użytkownika>  
Chciałbym <cel>  
[żeby <powód>]

## User Stories

- **<rola użytkownika>** - użytkownik, klient lub osoba z zespołu, która czerpie korzyści z danej historii
- **<cel>** - cel US, zazwyczaj funkcjonalność lub narzędzie
- **<powód>**[opcjonalnie] – zysk płynący z danej funkcjonalności



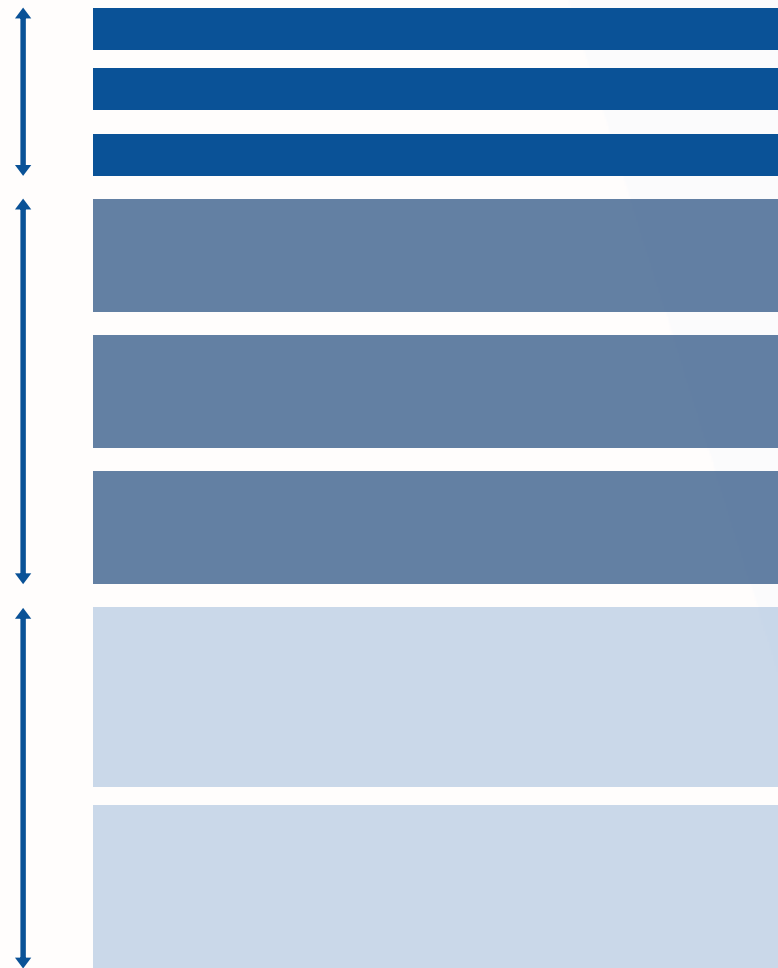
# User Stories



Wysoki priorytet, jasne  
wymagania, krótkie zadania

Średni priorytet, wymagania  
nie do końca sprecyzowane lub  
stosunkowo długie

Niski priorytet, brak  
dokładnych wymagań, mogą  
być bardzo złożone lub  
czasochłonne





## Increment

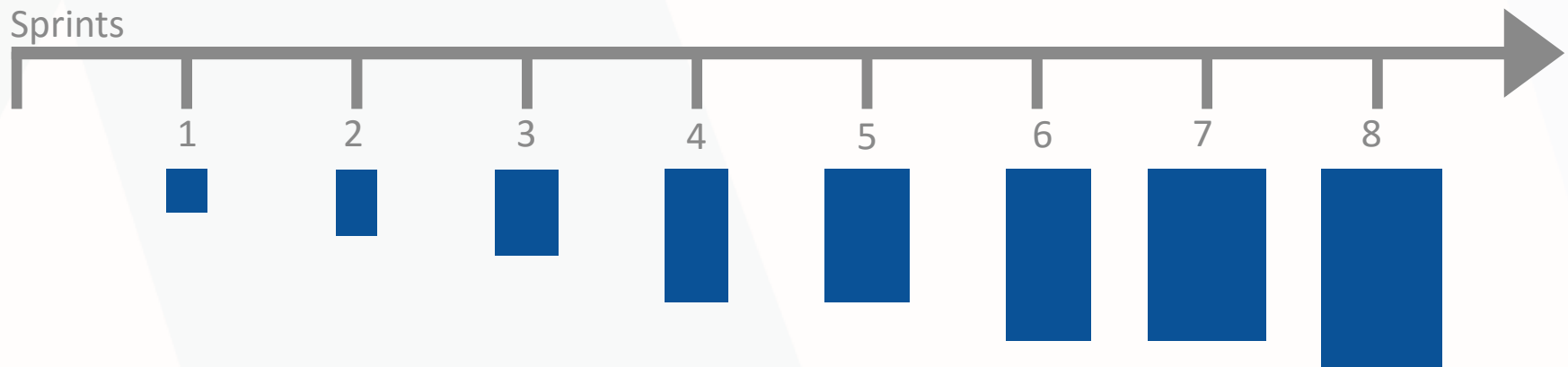
- Suma wszystkich ukończonych elementów Product Backlog'u
- Jest funkcjonalne i działa
- Musi zostać ukończone bez tzw. „długu technicznego”

# Sprint

- Zbiór wszystkich aktywności
- Zawiera implementację, testy oraz wszystkie pozostałe elementy Scrum
- Zaczyna się od planowania
- Kończy się retrospekcją
- Trwa nie dłużej niż 30 dni (zazwyczaj 2-4 tygodnie)
- Kolejne Sprints następują bezpośrednio po sobie
- Zakres prac jest ciągle kontrolowany

## Sprint c.d.

- Każdy Sprint to mały projekt o zdefiniowanym początku i końcu
- Stała długość iteracji
- Potencjalna wersja produkcyjna oprogramowania na końcu każdego Sprintu





# Sprint Planning

- Decyzja co będzie głównym zadaniem iteracji
- Powstaje Sprint Backlog
- Zespół wybiera zadania z puli produktu (Product Backlog)
- W planowaniu uczestniczy cały zespół
- Zapadają decyzje, co będzie skończone w tym Sprincie

## Daily Scrum

- Zawsze w tym samym miejscu i o tej samej porze
- Scrum Master może, ale nie musi w nim uczestniczyć
- Spotkanie organizowane aby:
  - Zsynchronizować pracę zespołu
  - Stworzyć plan na następne 24h
  - Sprawdzić postępy

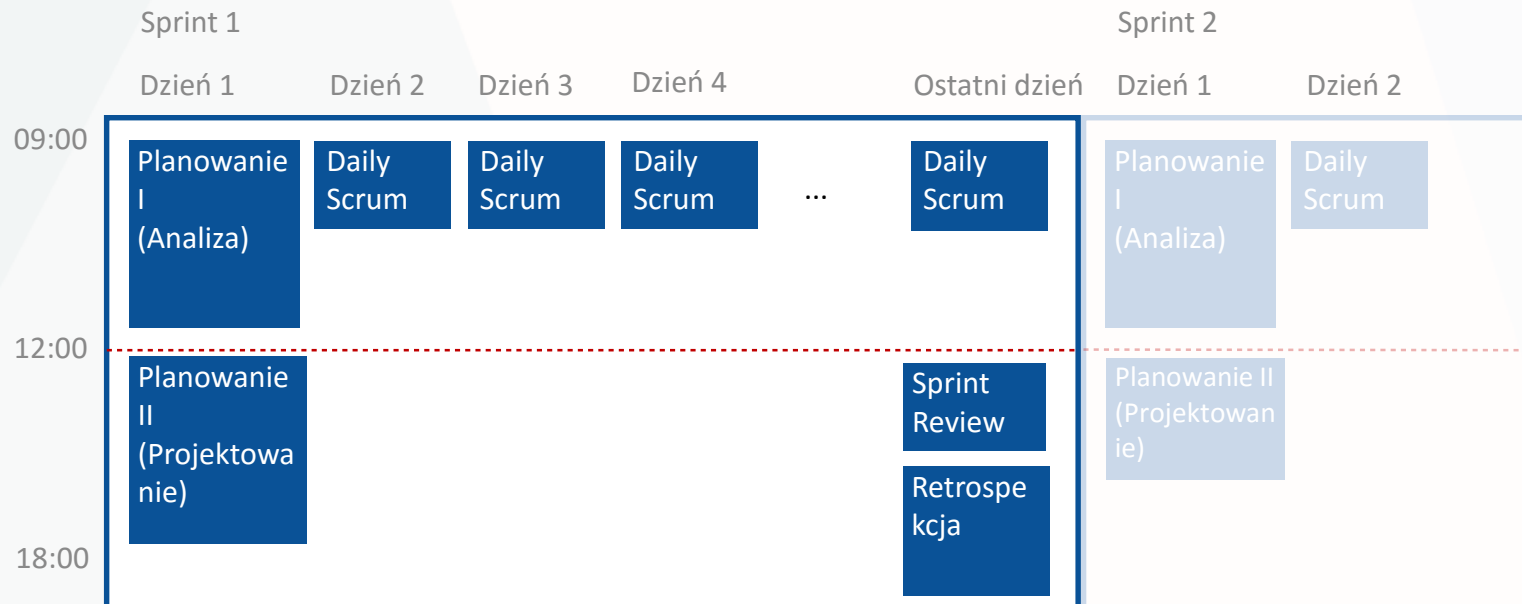
## Sprint Review

- Zespół prezentuje postępy prac (jako działający produkt)
- Inspekcja i ocena postępów (feedback)
- Tylko ukończone zadania są prezentowane (DoD)
- Product Owner akceptuje lub odrzuca User Stories

## Retrospective

- Ocena postępów prac i monitorowanie sposobu pracy zespołu
- Zespół omawia następujące tematy:
  - Co się udało w tym Sprincie?
  - Co można usprawnić?
  - Co chcielibyśmy zmienić w kolejnej iteracji?

# Organizacja pracy



# Ćwiczenie 1

Skutki braku dokładnych wymagań.

- Zespoły 4-5 osobowe
- Każdy wybiera 1 państwo i zapisuje je na karteczce Post-It
- Państwa nie mogą się powtarzać (wewnątrz zespołu)

## Ćwiczenie 2

### Praca sterowana vs samoorganizacja

- Zespoły 2 osobowe
- Jedna osoba jest szefem, druga pracownikiem
- Szef może mówić jedynie: idź, stój, lewo, prawo, szybciej, wolniej
- Pracownik musi wykonywać polecenia szefa
- Szef kieruje pracownikiem przez 2 minuty
- Pracownik głośno liczy kroki
- Szef nie może dotykać pracownika
- Nie można opuszczać wyznaczonej przestrzeni
- Chodzenie w miejscu nie jest liczone jako krok

## Ćwiczenie 3

### Symulacja Scrum

- Praca w zespole
- Każdy musi dotknąć każdej piłki
- Piłka przy każdym przekazaniu musi zależeć się w powietrzu
- Nie można podawać piłki do sąsiadów (ramię w ramię)
- Punkt zostaje zaliczony kiedy piłka wróci do osoby która zaczynała
- Upuszczone piłki są traktowane jako odrzucone (nie liczą się)



## Ćwiczenie 3 c.d.

### Proces

- 2 minuty – zapoznanie się z zadaniem, rozplanowanie i estymacja
- 2 minuty – gra
- 2 minuty – retrospekcja, zmiany i nowa estymacja

Powtórzyć 3-4 razy (iteracje)



Dziękuję za uwagę