Inżynieria oprogramowania (Software Engineering)

Wykład 2

Proces produkcji oprogramowania

Na podstawie materiałów autorstwa William Y. Arms z Cornell University

Proces produkcji oprogramowania (Software Process)

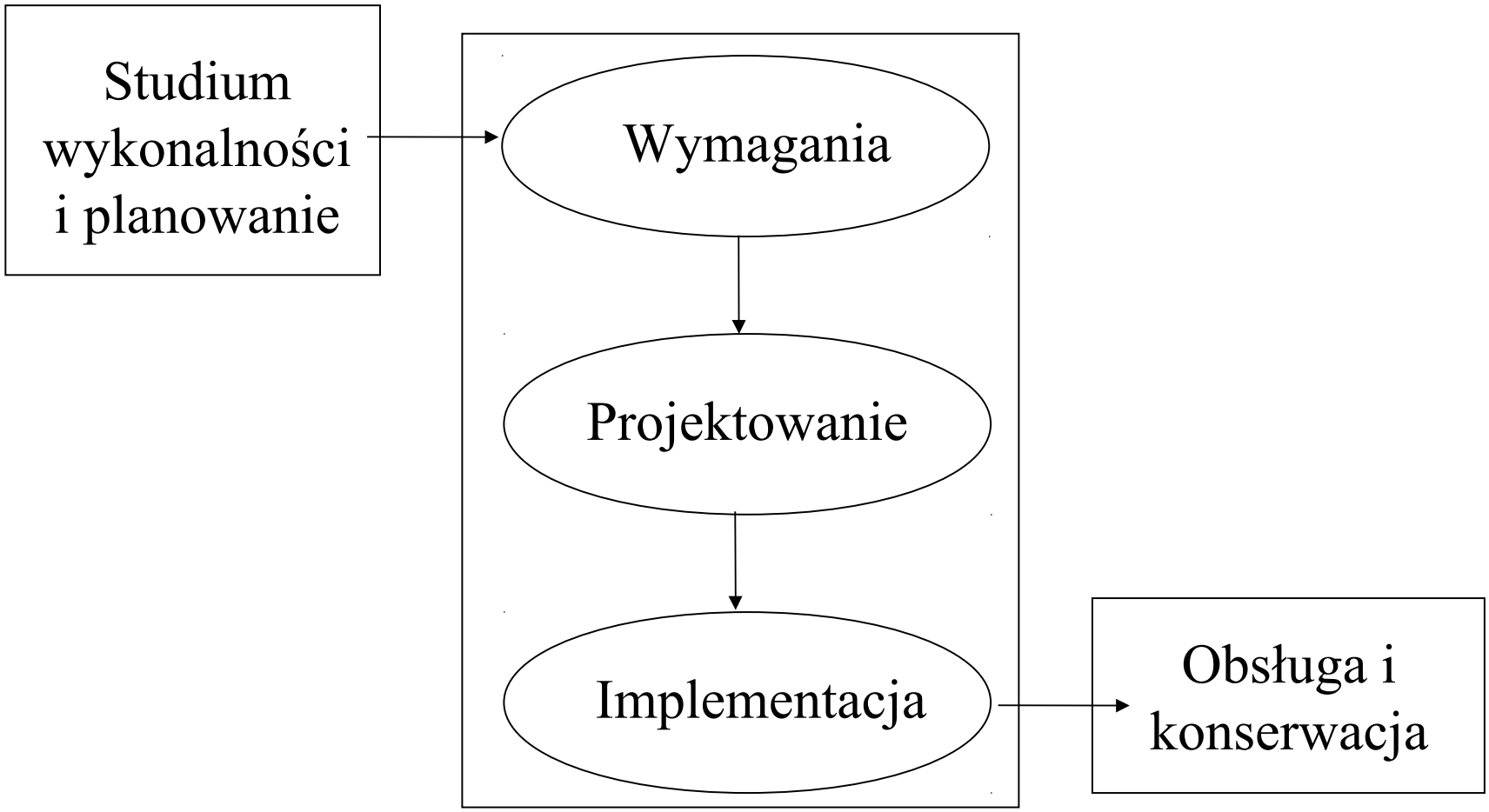
- Podstawowe założenia:
  - Dobre procesy prowadzą do dobrego oprogramowania
  - Dobre procesy zmniejszają ryzyko

Zarządzanie ryzykiem

- Co może pójść źle w projekcie oprogramowania?
- W jaki sposób można ograniczyć ryzyko?

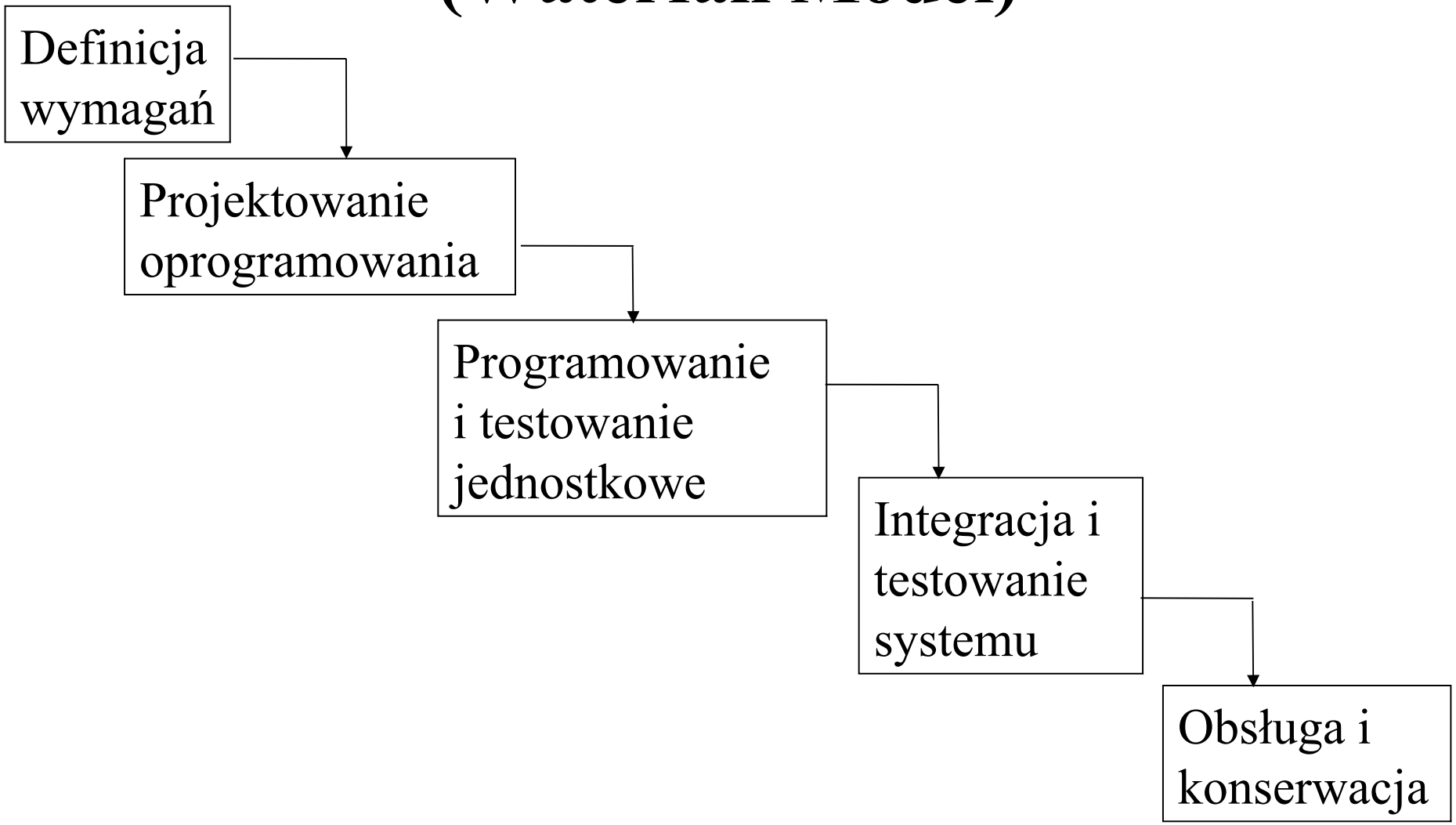
Na podstawie materiałów autorstwa William Y. Arms z Cornell University

Uproszczony proces produkcji oprogramowania



Na podstawie materiałów autorstwa William Y. Arms z Cornell University

Model kaskadowy-wodospadu (Waterfall Model)



Na podstawie materiałów autorstwa William Y. Arms z Cornell University

Analiza wymagań i definicja (Requirements Analysis and Definition)

- Systemowe usługi, ograniczenia i cele są ustalane w drodze konsultacji z użytkownikami systemu. Następnie są one określane w sposób zrozumiały przez użytkowników i pracowników rozwoju (deweloperów).
- Faza ta może być podzielona na:
  - Studium wykonalności (oddzielne przedsięwzięcie?)
  - Analiza wymagań
  - Definicja wymagań
  - Specyfikację wymagań

Na podstawie materiałów autorstwa William Y. Arms z Cornell University

Projektowanie oprogramowania (System and Software Design)

- Projekt systemu: Rozdziela wymagania do systemów sprzętowych lub softwarowych. Ustanawia ogólną architekturę systemu.
- Projekt oprogramowania: Reprezentuje funkcje systemowe oprogramowania w formie, którą można przekształcić w jeden lub więcej programów wykonywalnych.
- Zunifikowany Język Modelowania (Unified Modeling Language, UML)

Programowanie i testowanie jednostkowe (Programming and Unit Testing)

- Projekt oprogramowania realizowany jest jako zestaw programów lub jednostek (pisanych indywidualnie, nabytych z zewnątrz lub zmodyfikowanych.)
- Poszczególne elementy są testowane zgodnie ze specyfikacjami.

Na podstawie materiałów autorstwa William Y. Arms z Cornell University

Integracja i testowanie systemu (Integration and System Testing)

- Poszczególne jednostki programu są:
  - Integrowane i testowane jako kompletny system
  - Sprawdzane pod kątem określonych wymogów
  - Dostarczane do klienta

Obsługa i konserwacja (Operation and Maintenance)

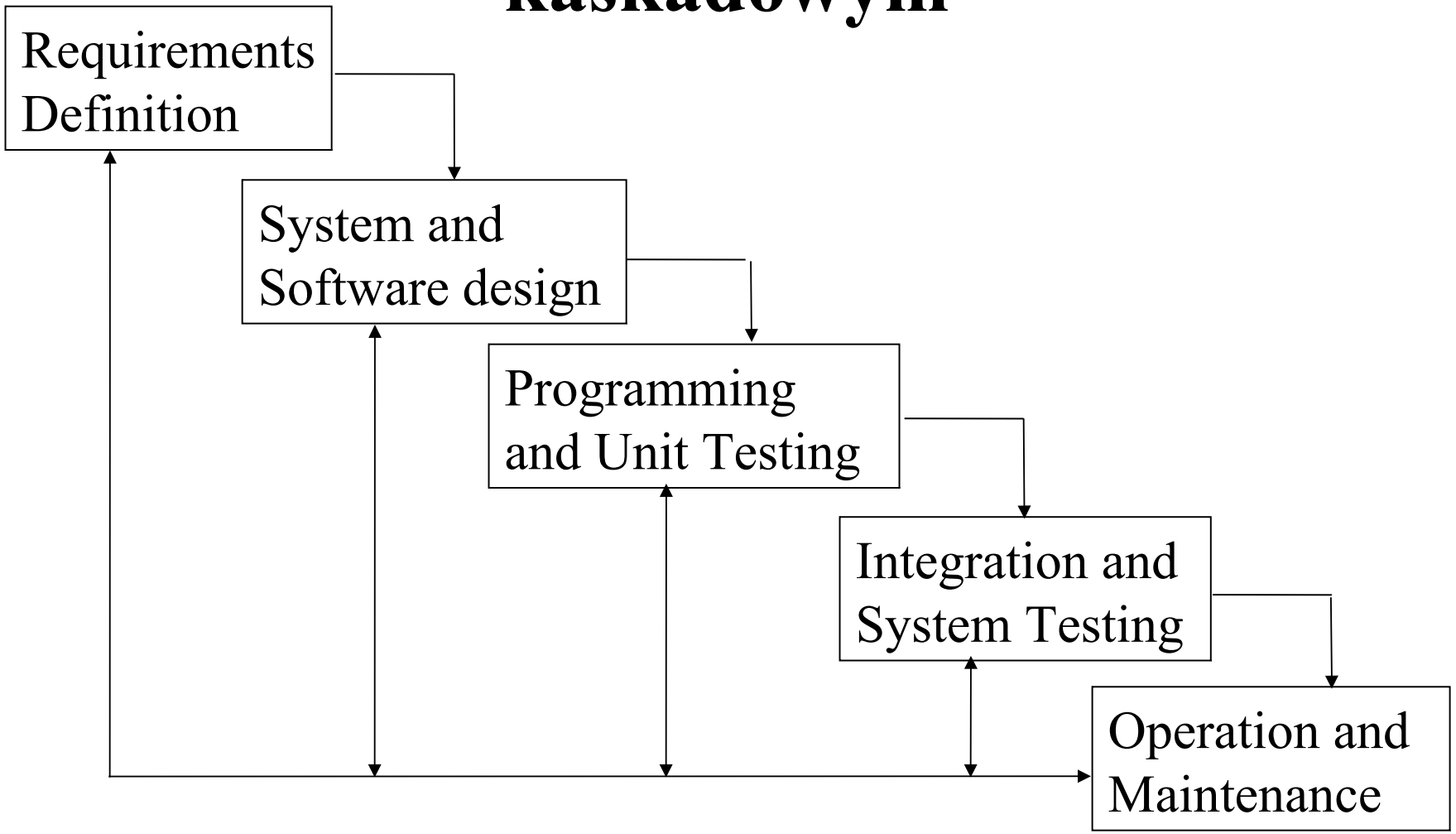
- Obsługa: System jest wprowadzany do praktycznego użycia.
- Konserwacja: Błędy i problemy zostają zidentyfikowane i naprawione.
- Ewolucja: system jest rozwijany w czasie: ze zmianą wymagań, aby dodać nowe funkcje, by dostosować do środowiska technicznego.
- Wycofywanie (Phase out): System zostaje wycofany z eksploatacji.

Rola modelu kaskadowego

- Zalety:
  - Widoczność procesu
  - Zależność od jednostek
  - Kontrola jakości
  - Kontrola kosztów
- Wady:
  - Każdy etap procesu ujawnia nowe zrozumienie poprzednich etapów, a ono wymaga zmiany wcześniejszych etapów.

Na podstawie materiałów autorstwa William Y. Arms z Cornell University

Sprzężenie zwrotne w modelu kaskadowym

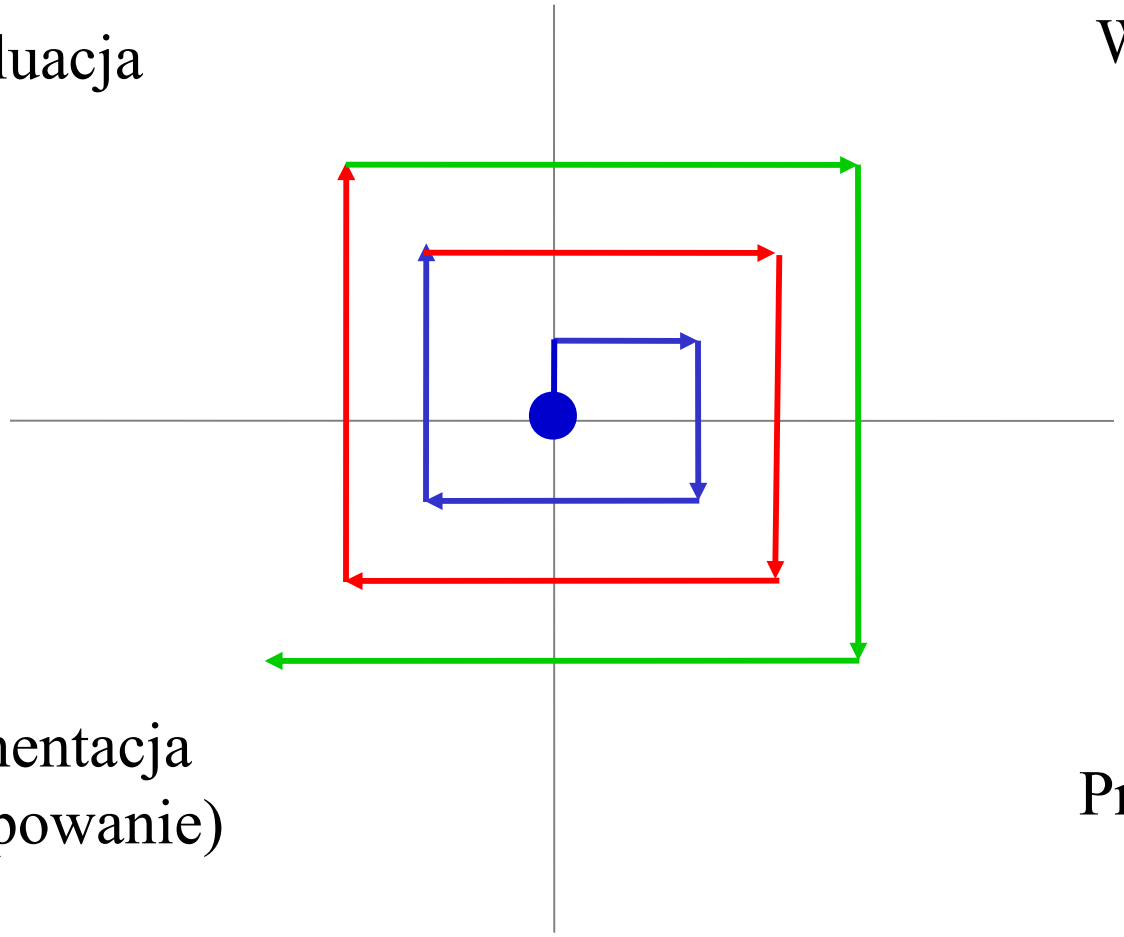


Na podstawie materiałów autorstwa William Y. Arms z Cornell University

Ewolucyjny rozwój oprogramowania

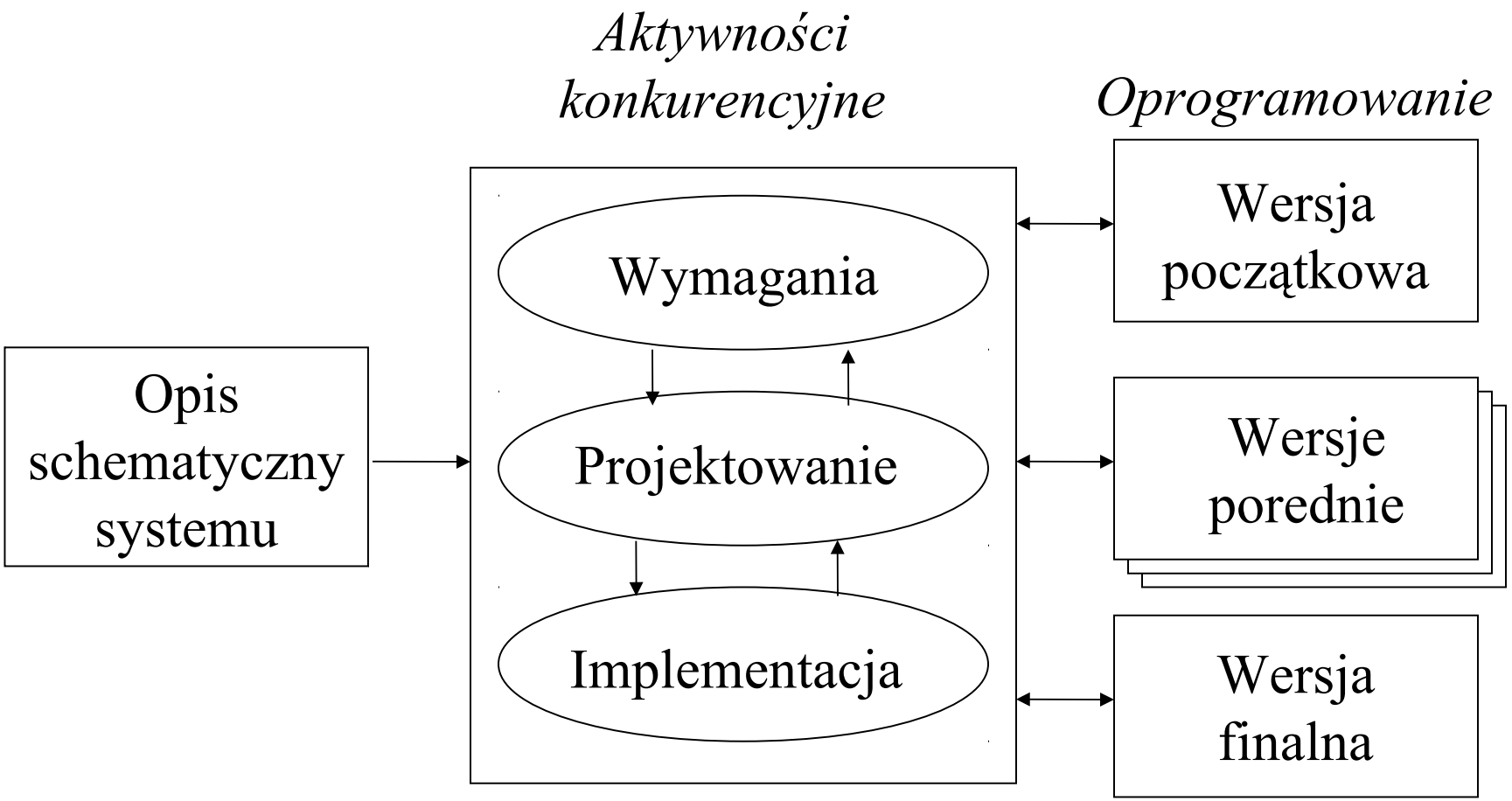
- **Koncepcja**: początkowa realizacja przeznaczona jest dla komentarza użytkownika, następne będą dopracowywane aż system stanie się kompletny.
  - Vaporware (oprogramowanie które zostało rozreklamowane, ale nie zostało ukończone), interfejs makiety
  - Porzucane składniki oprogramowania
  - Moduły-atrapy
  - Szybkie prototypowanie
  - Kolejne udoskonalenie

Idea iteracyjnego udoskonalania (Iterative Refinement)



Na podstawie materiałów autorstwa William Y. Arms z Cornell University

Proces iteracyjnego udoskonalania



Na podstawie materiałów autorstwa William Y. Arms z Cornell University

Uwagi nt. procesów wytwarzania oprogramowania

Zrealizowane projekty powinny wyglądać jak wykonane w modelu kaskadowym, ale proces rozwoju zawsze będzie częściowo ewolucyjny.

- Ryzyko przedsięwzięcia jest obniżane przez:
  - Prototypowanie elementów kluczowych
  - Podział na etapy
  - Postępowanie zgodnie z przyjętym procesem tworzenia oprogramowania
  - Wykorzystanie komponentów wielokrotnego użytku

Na podstawie materiałów autorstwa William Y. Arms z Cornell University