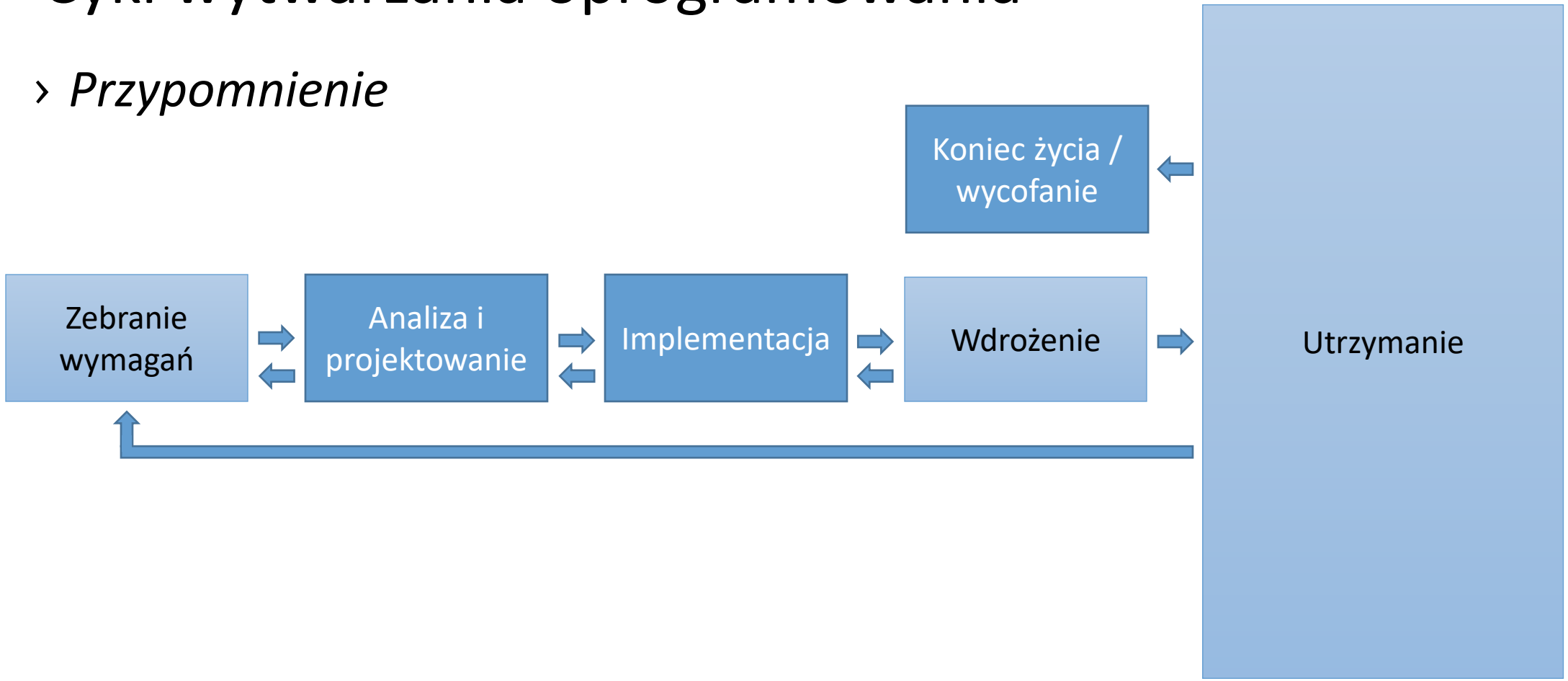
Cykl wytwarzania (życia) oprogramowania c.d.

Sztuka Wytwarzania Oprogramowania, w. 4 cz. 1

Konrad Grochowski

Cykl wytwarzania oprogramowania

› *Przypomnienie*



Wymagania

- Sformułowana potrzeba cechy produktu / usługi
  (bądź sposobu działania / realizacji usługi)
- Skąd:
  - Wewnętrznie – analiza rynku, poprzednich wersji, strategia firmy,
  - Zewnętrznie – dostarczone w całości przez klienta (przetargi),
  - Mieszane – ogólna potrzeba klienta jest uszczegóławiana w ramach usługi
- Klasyczny podział:
  - Wymagania funkcjonalne (*co ma robić?*)
  - Wymaganie niefunkcjonalne („wszystko inne”- wydajność, niezawodność, jakość etc., *jak ma robić?*)

Wymagania do wymagań

- Bez dobrych wymagań (dobrze uzgodnionych) *nie da się* zrealizować projektu (w zaplanowanym czasie i budżecie)
- Problem – programiści (twórcy oprogramowania) często nie znają dziedziny (i nie są docelowymi użytkownikami systemów, które tworzą)
- Z drugiej strony – poświęcanie czasu na „dogranie” wymagań może być kosztowne – metodyki *agile* (zwinne) sugerują robić to razem z prototypowaniem / dostarczaniem wersji pośrednich
- Ale nawet „w *agile*” obowiązują cechy „dobrego wymagania”
- Czyli inżynier *musi* umieć ocenić i *wymagać dobrych wymagań*

Dobre wymaganie

- **Jasne** (*Clear*) – zwarty, jasny opis, bez „lania wody” etc.
  - Często sformalizowany/uproszczony język
  - „Ustandaryzowane” konstrukcje gramatyczne (stały schemat zdań)
  - *Koszmarek: Celem biznesowym systemu jest zwiększenie satysfakcji klienta przez udostępnienie mu możliwości zaspokojenia potrzeby wygodnej realizacji obsługi ...*
  - *Lepiej: System ma obsłużyć ...*
- **Kompletne** (*Complete*) – wszystkie informacje potrzebne do zrozumienia (i sprawdzenia) wymagania są w nim zawarte
  - *Źle: System ma być wydajny*
  - *Lepiej: System ma obsłużyć do 1000 zapytań HTTP na sekundę*

Dobre wymaganie

- **Zgodne** (*Consistent*) – nie może być sprzeczne z innymi
  - w zbiorze wymagań nie może być wewnętrznych sprzeczności
  - nie może być też sprzeczności z dokumentami zewnętrznymi
- **Spójne** (*Cohesive*) – w opisie systemu nie powinno być braków
  - ani „nachodzących” na siebie wymagań
  - jedno i drugie otwiera „pole do interpretacji”, czemu wymagania mają zapobiegać
  - dla pojedynczego wymagania:
    Wymaganie odnosi się do jednej i tylko jednej sprawy.

Dobre wymaganie

- **Atomowe** (*Atomic*) – niepodzielne, bo ciężko weryfikować coś, co ma w sobie „i” (a jeszcze gorzej „lub”)
  - *Źle: System będzie wspierał formaty danych: JSON, XML*
  - *Lepiej:*
    System pozwoli na zapis danych do pliku w formacie JSON
    System pozwoli na zapis danych do pliku w formacie XML

- **Jednoznaczne** (*Unambiguous*) – wszystkie pojęcia użyte w definicji wymagania powinny być zrozumiałe dla każdego odbiorcy (skróty i żargon etc. ograniczone do ustalonego – najlepiej sformalizowanego – słownika pojęć)
  - Żargon klienta i programisty może być *drastycznie* różny

Dobre wymaganie

- **Weryfikowalne** (*Verifiable*) – zrealizowanie wymagania powinno dać się *jednoznacznie* i możliwie *obiektywnie* ocenić
  - Ważne i dla klienta i dla producenta (wypłata)
  - Najlepiej, jeśli wymaganie jest ***testowalne***
  - Metoda weryfikacji powinna być zdefiniowana razem z wymaganiem, np.:
    - Test
    - Analiza dokumentacji
    - Niezależna inspekcja

Dobre wymaganie

- **Może mieć priorytet** – w jakiej kolejności / czy musi być zrealizowane
  - *shall/must, should, could*
- **Może mieć uzasadnienie** – przydatne, jeśli to *my* je tworzymy
  - warto jakoś zanotować, skąd dana rzecz się wzięła
- **Może mieć możliwość *śledzenia* (traceability)** – posiada powiązanie z wymaganiami wyższego rzędu / standardami
  - często najlepsza forma uzasadnienia
  - ważne w sformalizowanych procesach (ale nie tylko)

Zbieranie wymagań / analiza wymagań

- Etap rozwoju oprogramowania niekiedy traktowany pobieżnie
  - „są wymagania jakieś to robimy, zobaczymy co wyjdzie”
    (to ani agile, ani extreeme programming)
- Faza kluczowa dla ustalenia celu projektu i zdefiniowania co usatysfakcjonuje klienta
- Dobre wykonanie analizy często wymaga zestawienia *sprawnego* kanału komunikacji od klienta aż do „technicznych”

Dziękuję za uwagę

Konrad.Grochowski@pw.edu.pl