

## **Zarządzanie wymaganiami (zakresem)**

- usystematyzowane podejście do ustalania i utrzymywania porozumienia między klientem i zespołem wykonawców
- zarządzanie procesem pozyskiwania, specyfikacji i weryfikacji wymagań
- opracowanie standardu specyfikacji wymagań
- zarządzanie specyfikacją wymagań w różnych fazach realizacji projektu (analiza, projektowanie, programowanie) – utrzymywanie zgodności
- stworzenie systemu identyfikacji i śledzenia wymagań w dokumentacji
- zarządzanie zmianami wymagań

## **Trudności w określeniu wymagań**

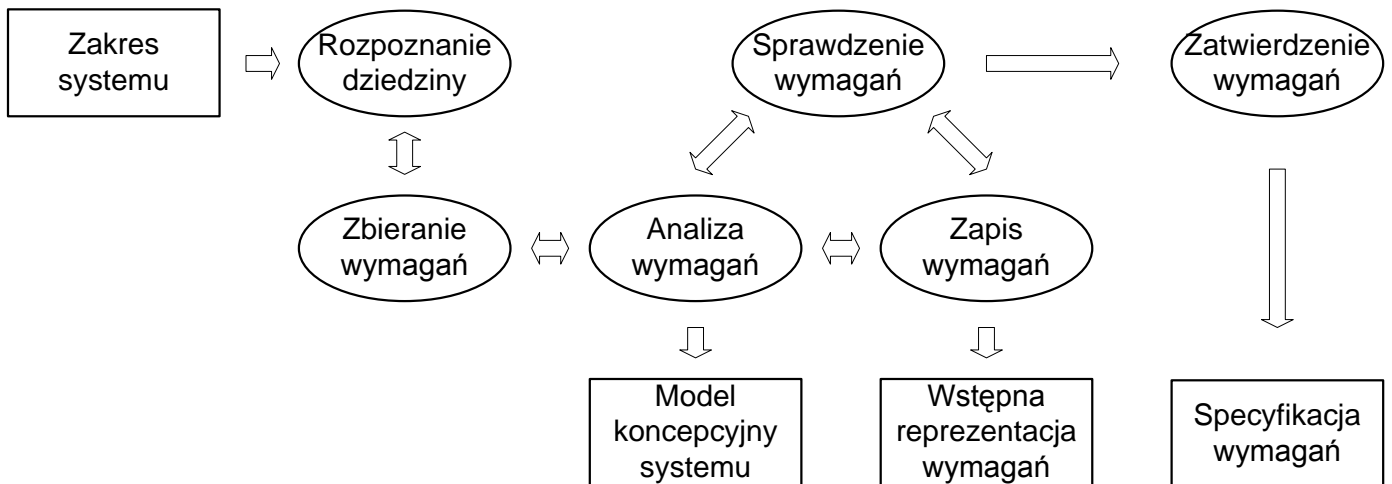
- klienci i użytkownicy często nie wiedzą dokładnie czego chcą i zaczynają rozumieć swoje potrzeby dopiero wtedy, kiedy widzą system podczas pracy
- duże systemy są wykorzystywane przez wielu użytkowników z różnymi punktami widzenia (mogą być bardzo różne nawet sprzeczne wymagania)
- klient z reguły nie wie dokładnie w jaki sposób osiągnąć założone cele – konieczność negocjacji wymagań
- różnice środowiskowe i pojęciowe – trudności we wzajemnym zrozumieniu
- klient może nie „dostrzegać” pewnych ograniczeń lub może uznawać je za oczywiste (powszechnie znane)
- konieczność wynegocjowania zmiany organizacji pracy w przedsiębiorstwie (żeby nie informatyzować bałaganu)
- niestabilne koncepcje systemu lub uwarunkowania zewnętrzne (np. wymagania prawne) – „pełzanie” wymagań

## **Pozyskiwanie wymagań (w metodach tradycyjnych)**

- ustalenie zakresu systemu
- identyfikacja źródeł pozyskiwania wymagań i różnych punktów widzenia
- wydobywanie wymagań
- analiza wymagań
- wstępna reprezentacja wymagań
- potwierdzenie właściwego zrozumienia wymagań
- konsolidacja i specyfikacja wymagań
- analiza specyfikacji: weryfikacja i walidacja
- ustalenie hierarchii wymagań
- zatwierdzenie wymagań

### **Proces iteracyjny**

# Proces określania wymagań



## **Źródła informacji**

- bezpośredni użytkownicy – funkcje systemu i sposób użytkowania
- wyższe kierownictwo – cele strategiczne, bezpieczeństwo
- administrator systemu – niezawodność, funkcje wykrywania i diagnozy błędów itp.
- kierownik marketingu – nowoczesność rozwiązań
- kierownik finansowy – koszty

## **Ponadto**

- dokumenty – instrukcje, procedury
- przykłady działania starego systemu lub systemów podobnych

# Metody zbierania wymagań

- wywiady
  - z wyższym kierownictwem odnośnie strategii firmy i celów systemu
  - z kadrami inżynierską odnośnie potrzeb i ograniczeń technicznych
  - z „bezpośrednimi” użytkownikami
- warsztaty wymagań, „burze mózgów”
- ankiety (ograniczona użyteczność)
- analiza scenariuszy zdarzeń (przypadków użycia)
- poznawanie środowiska docelowego, w którym zostanie zainstalowany system
- odgrywanie ról użytkowników
- studiowanie innych istniejących systemów
- stosowanie prototypów

## **Analiza wymagań**

- głębsze poznawanie wymagań
- kształtowanie modelu przyszłego systemu
- wykrywanie braków w zestawie wymagań
  - błędów (wymagania błędnie wyspecyfikowane)
  - sprzeczności (konfliktów)
  - nadmiarowości
  - luk (wymagania niewyspecyfikowane)
  - wymagań spoza zakresu systemu
- badanie wzajemnych związków między wymaganiami
- analiza sytuacji nietypowych

## **Pożądane cechy Specyfikacji Wymagań**

- jednoznaczność – w interpretacji wymagań
- kompletność
- poprawność
- spójność (niesprzeczność)
- weryfikowalność – istnieją procedury sprawdzenia spełnienia wymagań
- modyfikowalność – możliwość łatwego uwzględnienia zmian w specyfikacji
- jasny system identyfikacji i odwołań



# **Specyfikacja wymagań w podejściach tradycyjnych i zwinnych**

## **W podejściach tradycyjnych**

- kompletne wymagania tworzone są na samym początku
- są one bardzo szczegółowe – wszystkie wymagania muszą jednocześnie posiadać ten sam poziom uszczegółowienia
- poświęca się wiele wysiłku żeby „zrobić wszystko dobrze za pierwszym razem” i potem „za wszelką cenę” unika się zmian

## **W podejściach zwinnych**

- uważa się, że podczas tworzenia innowacyjnych produktów nie da się pozbyć niepewności i z góry stworzyć wyczerpującej listy wymagań poprzez wydłużenie czasu prac lub wzmożenie wysiłków
- nie inwestujemy dużo czasu i pieniędzy w wytworzenie szczegółowych wymagań na samym początku projektu
- na początku tworzymy jedynie wskaźniki wymagań nazywane historyjkami użytkownika
- szczegóły są precyzowane później w samą porę

## **Zapis wymagań w metodykach zwinnych**

- zbiór historyjek (opowieści) użytkownika (ang. *user stories*)
- historyjki są uporządkowane według priorytetów
- rejestr produktu jest „żywy” i podlega ciągłym zmianom – zmieniają się priorytety, modyfikowany jest zakres historyjek itd.

### **Historyjka (opowieść) użytkownika**

- prosty jedno lub dwuwierszowy opis funkcjonalności požądanej przez klienta
- historyjki zapisuje się na osobnych kartkach (kartonikach)
- od zapisu historyjki dużo ważniejsza jest rozmowa, o której papierowy kartonik ma nam przypomnieć
- historyjka powinna być napisana w języku klienta i opisywać efekt wartościowy dla klienta, a nie szczegóły techniczne
- historyjki powinny mieć jasne kryteria ukończenia (obiektywne testy akceptacyjne) – zapisywane zwykle na odwrocie kartonika

Bezpośredni kontakt i rozmowa zamiast szczegółowego dokumentu ze specyfikacją wymagań

## **Zapis wymagań w metodykach zwinnych**

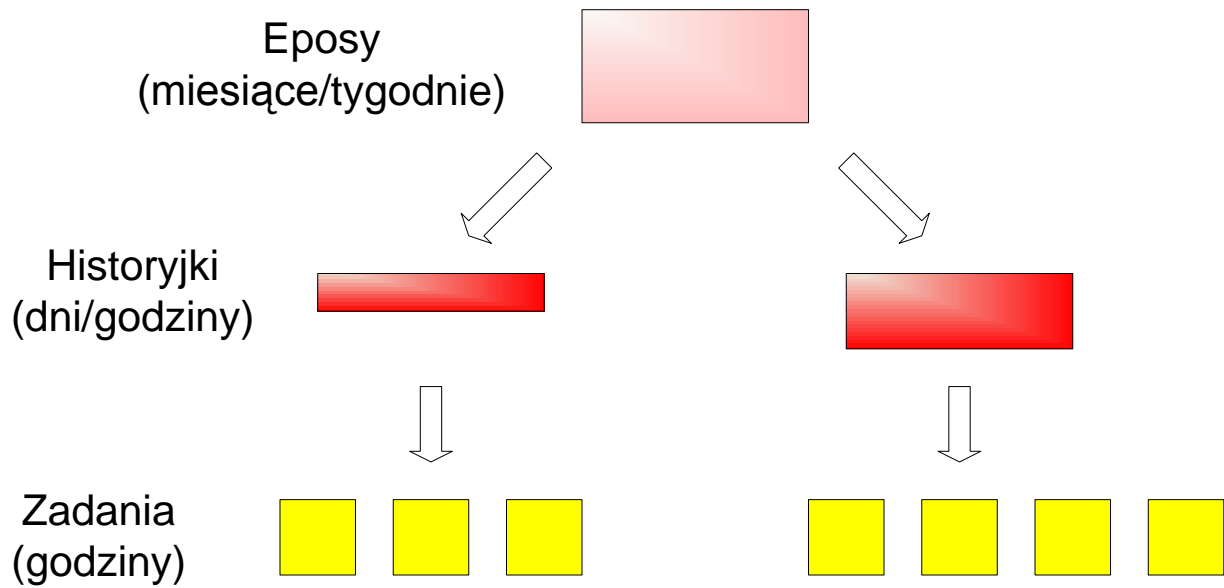
### **Atrybuty dobrze sformułowanych historyjek - INVEST**

- Niezależne (ang. *Independent*) – historyjki są niezależne jeżeli możemy implementować je w różnej kolejności
- Negocjowalne (ang. *Negotiable*) – szczegóły dopracowuje się w bezpośredniej rozmowie
- Wartościowe dla odbiorcy (ang. *Valuable for Users and Customers*) - reprezentują wartość dla klienta i są napisane w jego języku
- Możliwe do oszacowania (ang. *Estimatable*) - wymaga to możliwości zrozumienia istoty historyjki, umiarkowanego rozmiaru historyjki, doświadczenia i wiedzy zespołu
- Małe (ang. *Small*) - można je łatwo zaplanować i oszacować (np. zaimplementować w jednej iteracji)
- Testowalne (ang. *Testable*) – mają jasno określone kryteria ukończenia

## **Zapis wymagań w metodykach zwinnych**

- zbiór historyjek (opowieści) użytkownika (ang. user stories)
- historyjki są uporządkowane według priorytetów (tylko początek listy)
- historyjki w rejestrze mają różne rozmiary i są w różnym stopniu uszczegółowione
- historyjki z największym priorytetem (nad którymi planujemy pracować wkrótce) mają najmniejszy rozmiar i są bogate w detale
- stosuje się strategię stopniowego uszczegółowiania historyjek – zbliżając się do pracy nad większymi elementami, rozbijamy je na zbiory mniejszych historyjek nadających się do implementacji w jednej iteracji
- rejestr produktu jest „żywy” i podlega ciągłym zmianom – zmieniają się priorytety, modyfikowany jest zakres historyjek itd. (ciągła pielęgnacja rejestru produktu)

## Poziomy szczegółowości elementów rejestrów



# Poziomy szczegółowości elementów rejestrów

## Rejestr produktu

- **Eposy** – duże historyjki wymagające kilka miesięcy/tygodni pracy (bez szczegółów)
- (małe) **Historyjki** – kilka godzin/dni pracy (poziom szczegółowości odpowiedni do realizacji w ramach iteracji)

## Rejestr iteracji

**Historyjki z Zadaniem** (prace inżynierskie niezbędne do realizacji poszczególnych historyjek)

Zadania nie są historyjkami – specyfikują, *jak* należy coś zbudować, w przeciwieństwie do tego *co* należy zbudować (określają to eposy i historyjki)