

**Metodyka zwinna** ( agile software development method) – najbardziej popularna obecnie (> metod kaskadowych i spiralnych) ze względu na mniejsze sformalizowanie. Różni się generalną ideą - sprzedają usługę, np. opracowanie oprogramowania, a nie sam produkt-oprogramowanie.

Generalne własności:

- ściśła współpraca z użytkownikiem, włączenie go w proces projektowania oprogramowania (eXtreme Programming)
- to użytkownik tak naprawdę podejmuje decyzje co i w jakiej kolejności będzie wykonywane najpierw w projekcie
- ważne jest szacowanie kosztów tak aby inwestor/ użytkownik mógł planować swoje wydatki
- wytwórca oprogramowania powinien na każdym kroku udowadniać, że dobrze wykorzystał czas i jego praca jest efektywna :D
- rezygnuje się z ponownego użycia kodu i modeli analitycznych
- w trakcie tworzenia na bieżąco zadaje się pytania użytkownikowi-inkrementalny sposób dostarczania oprogramowania
- nie specyfikuje się formalnych punktów kontrolnych w projekcie, gdyż zakończenie każdej iteracji jest końcem samym w sobie
- regułą jest przebudowa kodu – sekwencyjna realizacja wymagań użytkownika powoduje częste zmiany w architekturze systemu i konieczność przebudowy kodu

Wymagania dotyczące wprowadzania najważniejszych innowacji w SI:

- integracja systemów, danych i procesów
- unifikacja funkcji częściowych systemów
- zwiększanie dostępności do bazy danych dla wszystkich komórek organizacyjnych
- upowszechnianie nowoczesnych sposobów prezentacji danych (wizualizacji) dla celów wspomagania ich analizy
- doskonalenie procesów podejmowania decyzji i ich przekazywania
- zmierzanie do budowy modułowej i otwartości całego systemu
- zapewnienie kompleksowego charakteru funkcjonalnego całego systemu
- podnoszenie zaawansowania merytorycznego i technologicznego
- zmierzanie do elastyczności funkcjonalnej i strukturalnej
- stała zgodność ze zmieniającymi się elementami otoczenia (np. zgodność z prawem)

Bardzo ważna rzecz: poufność i integralność wszelkich danych posiadanych przez system.

Podstawowe składniki „manifestu” zwolenników „zwinnych” metod:

#### **AGILE**

- ludzie i ich kontakty, zdolność do rozwiązywania problemów są ważniejsze niż sztywne procedury i narzędzia zarządzania
- wynikiem projektu jest pracujące oprogramowanie a nie dokumentacja
- z użytkownikiem się współpracuje a nie negocjuje kontakt
- ważniejsza jest umiejętność reagowania na zmieniające się warunki otoczenia niż podążanie za opracowanym na wstępie planem
- CZŁOWIEK NADRZĘDNYM CZYNNIKIEM SUKCESU, #yolo

Metody „zwinne” można stosować głównie dla niezbyt dużych systemów.

Metody zwinne to:

**-Metodyka Crystal (Crystal family)**

- Projektowanie zorientowane na właściwości FDD (Feature Driven Development)
- Modelowanie zwinne (Agile Modeling)
- Programowanie ekstremalne (Extreme Programming)
- Adaptacyjny rozwój oprogramowania
- Metodyka SCRUM
- Prototypowanie
- Szybkie projektowanie internetowe
- Pragmatyczne programowanie

#### CRISTAL

-ma odmiany w zależności od stopnia krytyczności projektu i jego rozmiaru, mierzonego liczbą projektantów zaangażowanych w tworzenie projektu

Kategorie krytyczności to:

- C – Komfortowe
- D-Zarządzające Finansami
- E-Finansowo istotne
- L-Krytyczne dla życia

#### FDD

Ogólny model:

- inkrementacyjne opracowywanie poszczególnych funkcjonalności systemu
- prace rozpoczynają się od „ogólnego modelu systemu”
- zespół korzysta z opracowanych wcześniej wymagań systemowych i przypadków użycia
- określana jest domena projektu i iteracyjnie dzielona na coraz to mniejsze znaczeniowo obszary
- każdy niepodzielny obszar znaczeniowy jest opracowywany przez przypisaną do niego grupę projektantów, w wyniku czego powstaje model szczegółowy

Określenie listy funkcjonalności:

- listy te mają charakter hierarchiczny i zawierają funkcjonalności główne, które rozpadają się na zestawy funkcjonalności w kolejnych hierarchiach
- listy są przeglądane przez użytkowników i inwestorów w celu kontroli poprawności i kompletności

Planowanie na podstawie funkcjonalności:

- opracowanie długookresowego planu prac
- poszczególne elementy listy funkcjonalności przydzielane są zespołom a w ich ramach konkretnym programistom – opiekunom klas (w kontekście programowania obiektowego – Java, hehe)

Projektowanie na podstawie funkcjonalności i wykonanie funkcjonalności:

- Każda funkcjonalność znajduje swą realizację wewnątrz obiektu(klasy), do każdej klasy przyporządkowany jest programista dbający o nią; jest on zarządzający obiektem(klasą).Zarządzający obiektami są formowani w małe zespoły. Sposób organizacji tych zespołów odzwierciedla hierarchię funkcjonalności.

Zarządca konfiguracji - zajmuje się zagadnieniami kontroli wersji i identyfikacja każdej „historycznej” wersji kodu źródłowego

#### DSDM

**Metodyka Projektowania Systemów Zmiennych w Czasie**

- inspekcja zastosowalności:

\* wykonywana jest jednokrotnie na początku projektu po to aby potwierdzić słuszność stosowania metody DSDM; określa się ryzykowne punkty w projekcie, czasami buduje się prototypy, rozległość prac w tej fazie: do kilku tygodni.

-badania biznesowe

\*wysokopoziomowy opis systemu, specyfikacja zakresu systemu, zarys architektury systemu, plan prototypowania

-iteracyjne opracowania modelu funkcjonalnego

\* analiza i budowa prototypów, jeśli to możliwe prototypy są udoskonalane tak, aby można było je włączyć do produktu końcowego. Wynikiem tej fazy jest model funkcjonalny oraz kod prototypów. Prototypowanie jest często gęsto :D traktowane jako forma testowania modelu funkcjonalnego.

W każdej pętli iteracji tworzone są następujące dokumenty:

~lista funkcjonalności do opracowania wraz z ich priorytetami

~uwagi i komentarze użytkowników na temat prac w aktualnym cyklu

~wymagania niefunkcjonalne

~analiza ryzyka pod kątem dalszych prac

-iteracyjne projektowania i implementacje

\* Projektowanie i implementacja to właściwa faza budowania systemu. Wyniki prac są przetwarzane w kod źródłowy. Prototypy powstałe w trakcie prac nad modelem funkcjonalnym mogą być w tej fazie adaptowane do kodu aplikacji.

Wynik fazy to: przetestowany produkt zawierający uzgodniony wcześniej zestaw funkcjonalności.

-wdrożenia

Zaletą DSDM: na każdym etapie projektowania i budowy systemu produkt jest oceniany przez twórców i użytkowników, a uwagi wynikające z oceny opracowywane są w ramach kolejnych iteracji.

- Zespoły DSDM muszą być uprawnione do podejmowania decyzji;
- Naciska się na częste dostarczanie nowych wersji oprogramowania.
- Nowe wersje są oceniane pod kątem odpowiedniości zastosowań biznesowych
- Iteracyjne i inkrementalne podejście do tworzenia oprogramowania
- Kontrola wersji – tak aby każda zmiana była odwracalna
- Wymagania systemowe są określane zgrubnie i są uszczegóławiane samym procesem DSDM
- Testowanie jest integralną częścią wszystkich faz w projekcie

**Programowanie ekstremalne**

-próba zaradzenia problemom związanym z długimi cyklami dostarczania oprogramowania i spadkiem zainteresowania inwestora zadaniem.

Charakterystyka:

- ewolucyjne podejście do projektowania i programowania oraz ekstremalnie ścisła współpraca z odbiorcą

-ekstremalnie krótkie iteracje w dostarczaniu kolejnych wersji oprogramowania prowadzące do szybkich odpowiedzi użytkownika

-programowanie w parach :D, ustawiczna przebudowa kodu źródłowego, integracja i testowanie połączonych modułów

- Gra w planowanie
- Szybkie iteracje

- Metafory jako porozumiewanie się pomiędzy użytkownikiem a programistami :D
- Prostota kodu (brzytwa Ockhama)
- Kolektywne posiadanie kodu
- Unikanie nadgodzin
- Komunikacja pomiędzy programistami poprzez kod źródłowy (no co Ty:o)
- Konstrukcja kodu źródłowego wedle zasad akceptowanych i przestrzeganych w całym zespole

#### Zasady projektowania ekstremalnego (XP):

- nie ma uniwersalnej metody prowadzenia projektu informatycznego
- praktyki XP powinny być przystosowywane do aktualnych potrzeb i specyfiki projektu

#### Fazy cyklu życia projektu XP:

##### - eksploracja

\*zapoznanie się z tematem prac, pozyskanie podstawowych info od użytkownika, który poprzez opowiadanie przedstawia sposób użytkowania systemu; na tej podstawie budowany jest zarys architektury systemu (ASK!)

oraz budowana jest lista funkcjonalności. W tym czasie projektanci testują wybraną technologię tworząc niezbędne prototypy oraz zapoznają się z używanymi narzędziami.

##### -planowanie

\*analiza opowiadań użytkownika pod kątem priorytetów, zestawiana jest lista funkcjonalności, które mają być opracowane, szacuje się czas realizacji zadań przez programistów, czyli ustala się dokładny harmonogram

##### - iteracja wykonawcza

\*-1-kilkutygodniowe mini cykle

\*działania analityczne, projektowe, kodowanie i testowanie

\*na końcu każdego mini cyklu wykonywane są testy oprogramowania

##### -Przygotowanie do produkcji

\*funkcjonalności są przygotowane do wdrożenia

\* uwagi użytkownika są nadal implementowane lub przeznaczone do implementacji w następnej wersji oprogramowania

\*gruntowne testy

##### -utrzymanie w ruchu tzw. faza konserwacji

\* działające oprogramowanie dla użytkownika, które wymaga jeszcze opieki i nadzoru

\*projektanci w tym samym czasie wykonują kolejną wersję oprogramowania

\*często następują zmiany w zespołach projektantów i wzrost zatrudnienia

##### -zakończenie projektu

\*następuje gdy użytkownik nie jest już zainteresowany dodawaniem funkcjonalności do oprogramowania, tempo współpracy z użytkownikiem spada, formułowane wnioski o rozszerzenie funkcjonalności mają charakter drugorzędny i często nie są wdrażane z powodów ekonomicznych

\*opracowywanie dokumentacji systemu i projektu, instrukcje użytkownika, instrukcje konserwacji

#### **Metodyka Scrum**

-adaptacyjny, samoorganizujący się proces wytwarzania oprogramowania

- nazwa zapożyczona ze strategii gry w rugby

- podział organizacji na małe, interdyscyplinarne, samoorganizujące się zespoły

-podzielenie planowanej pracy na małe, dobrze zdefiniowane elementy

-określenie kosztów tych elementów oraz uporządkowanie ich ze względu na priorytet

Uniwersalne zasady:

- Podejście holistyczne
- zasady Manifestu Agile
- Samoorganizacja zespołu
- Wartości zespołu
- Empiryzm
- Normatywny charakter

Horyzonty planowania Scrum:

- Strategii
  - Wydania
  - Sprintu
  - \* Planowanie celu i zakresu sprintu
  - \*Planowanie zadań sprintu
- Codzienne spotkania przez 10 dni roboczych  
(jest to opisana na obrazkach – slajdy 65-76)
- Dnia
  - Ciągły

Zalecenia Scrum:

- podzielenie czasu na krótkie iteracje o stałej długości, dążenie do posiadania działającego kodu na końcu każdej iteracji
- optymalizowanie planu wytwarzania w kooperacji z klientem w oparciu o produkty wykonane w każdej iteracji
- optymalizacja procesu na podstawie każdej iteracji

**Scrum** – technika zarządzania projektem informatycznym, utworzona na podstawie doświadczeń w realnych projektach. Wolny wybór w wyborze technik prowadzenia prac programistycznych. Zakłada ewolucyjny styl tworzenia oprogramowania.

Rozwój oprogramowania zachodzi w niestabilnych warunkach, na które składają się: nieprzewidywalne zmiany w wymaganiach, terminach, zasobach, dostępnych technologiach, wobec czego sam proces tworzenia oprogramowania jest złożony.

Scrum przewiduje częste działania zarządcze skupiające się na identyfikowaniu problemów i przeszkód w pracach inżynierskich.

Etapy procesu Scrum:

- rozpoczęcie gry
- \*planowanie i opracowanie zarysu architektury systemu, opracowanie listy wymagań, która pozostaje otwarta, a zadania do realizacji dopisywane są do niej w trakcie całego procesu tworzenia

Każdy cykl to podprojekt kaskadowy składający się z opracowania wymagań, analizy, projektowania, kodowania i wdrożenia trwający do 30 dni.

-faza produkcji

\*Rozpoczęcie prac tzw. Spotkanie Planowania Cyklu

- Dwukrotne, organizowane przez zarządcę procesu
- Pierwsze spotkanie: użytkownicy, nabywcy, zarząd, zespół projektantów, ustala się cele i priorytety pierwszej iteracji. Wypisywanie wymagań w listę.

- Drugie spotkanie: jedynie wykonawcy i Zarządca, ustalają sposób przeprowadzenia prac przy implementacji wymagań.
- \* Codzienne Spotkania Zespołu projektantów i programistów
  - Krótkie, ok. 15 minut, motywowanie i śledzenie postępów prac
  - Uczestnicy to: zespół projektantów i programistów oraz Zarządca
- \*Zakończenie prac z nieformalnym Spotkaniem Przeglądowym
  - gdy lista wymagań (to do list) jest już pusta, a system przygotowany jest do instalacji -gra na zakończenie

**Estymacja pracochłonności** – w trakcie całego projektu równoległe z pracami projektowymi i implementacyjnymi trwa proces oceny pracochłonności postulatów zawartych w liście wymagań. Polega to na gromadzeniu informacji statystycznych o przebiegu projektu i wyznaczania kosztu prac. Nie bierze pod uwagę dużych zmian w architekturze systemu lub użytkowanej technologii.

W metodzie Scrum zaleca się by użytkownik wraz z zespołem projektantów spędził kilkanaście dni nad opracowaniem listy wymagań (biznesowych i technologicznych). Celem nadrzędnym pierwszej iteracji produkcyjnej jest pokazanie użytkownikowi jakiegoś fragmentu funkcjonalności systemu zaimplementowanego w ramach wybranej technologii. Pierwsza iteracja to opracowanie szkieletu systemu, zapoznanie się również z technikami Scrum, organizacje zespołów Scrum, rozdział ról w projekcie. Dalsze iteracje są prostsze i szybsze.

## Zarządzanie procesem projektowania SI

Zalecana metodologia kierowania projektem

Dla kogo?

Kadra dyrektorska

Kierownicy programu oraz przełożeni kierowników projektów

Kierownicy projektów oraz inni członkowie zespołu

Członkowie biura kierowania projektami

Klienci oraz inni interesariusze projektów

PMBok – zbiór standardów i najlepszych rozwiązań w dziedzinie zarządzania projektami zebranych i opublikowanych przez członków PMI (Project Management Institute)

To nie jest metodyka, tylko zbiór najlepszych, powszechnie uznanych praktyk znajdujących zastosowanie w zarządzaniu projektami. W USA jest to narodowy standard.

Co pokrywa PMBoK?

-umiejętności interpersonalne

-wiedza z dziedziny zarządzania ogólnego

-obszar zastosowania, wiedza, standardy, regulacje

-zrozumienie otoczenia projektu

44 procesy pogrupowane w 5 grup procesów i 9 obszarów wiedzy:

-Grupa procesów rozpoczęcia

\*Służą zdefiniowaniu i zatwierdzeniu projektu w organizacji

\*\*Opracowanie dokumentu otwarcia

\*\*Opracowanie wstępnego określenia zakresu

- Grupa procesów planowania
- jak zrealizować zamierzone cele
- w jaki sposób
- jakimi środkami
- kiedy, w jakiej kolejności itp.
- \*Opracowanie planu zarządzania projektem
- \*Planowanie zakresu
- \*Definiowanie zakresu
- \*Utworzenie pakietów roboczych, WBS
- \*Zdefiniowanie czynności
- \*Porządkowanie czynności
- \*Szacowanie zasobów czynności
- \*Szacowanie czasu trwania czynności
- \*Opracowanie harmonogramu
- \*Szacowanie kosztów
- \*budżetowanie kosztów
- \*Planowanie jakości, zasobów ludzkich, komunikacji, zarządzania ryzykiem, identyfikacja ryzyka, jakościowa i ilościowa analiza ryzyka, planowanie reakcji na ryzyko, planowanie zaopatrzenia, kontraktów
- Grupa procesów realizacji
- \*Kierowanie i zarządzanie realizacją projektu
- \*Zapewnienie jakości
- \*Przyjmowanie członków zespołu
- \*Rozwój zespołu
- \*Dystrybucja informacji
- \*Gromadzenie ofert od sprzedawców
- \*Wybór sprzedawców
- Grupa procesów monitorowania i kontroli
- \*Monitorowanie i nadzór nad pracami
- \*Zintegrowane zarządzanie zmianami
- \*Weryfikacja zakresu
- \*Sterowanie zakresem
- \*Nadzór nad harmonogramem
- \*Nadzór nad kosztami
- \*Kontrola jakości
- \*Zarządzanie zespołem
- \*Raportowanie postępu prac
- \*Zarządzanie udziałowcami
- \*monitorowanie i nadzór nad ryzykiem
- \*Administracja kontraktem
- Grupa procesów zakończenia

9 obszarów wiedzy:

- Zarządzanie integracją
    - \*Tworzenie karty projektu
    - \*Tworzenie wstępnej deklaracji zakresu projektu
    - \*Opracowanie planu kierowania projektem
    - \*Kierowanie i zarządzanie realizacją projektu
    - \*Monitorowanie i kontrola prac projektu
    - \*Zintegrowana kontrola zmian
    - \*Zamknięcie projektu
  - zarządzanie zakresem
    - \*Planowanie zakresy
    - \*Precyzowanie zakresu
    - \*Tworzenie struktury podziału pracy
    - \*Weryfikacja zakresu
    - \*Kontrola zakresu
  - zarządzanie czasem:
    - \*Okreslanie działań i ich kolejności
    - \*Szacowanie zasobów działań i czasu trwania
    - \*Tworzenie i kontrola harmonogramu
  - zarządzanie kosztem
    - \*Szacowanie, budżetowanie i kontrola kosztów
  - zarządzanie jakością
    - \*Planowanie jakości, przeprowadzanie zapewnienia i kontroli jakości
  - zarządzanie ryzykiem
    - \* Planowanie zarządzania ryzykiem
    - \*Identyfikacja ryzyk
    - \*Jakościowa i ilościowa analiza ryzyk
    - \*Planowanie reakcji na ryzyka
    - \*Monitorowanie i kontrola ryzyka
  - zarządzanie zasobami ludzkimi:
    - \*Planowanie zasobów ludzkich
    - \*Pozyskiwanie zespołu projektu
    - \*Kształtowanie i kierowanie zespołem projektu
  - Zarządzanie komunikacją:
    - \*Planowanie komunikacji
    - \*Dystrybucja informacji
    - \*Sprawozdawczość wykonania
    - \*Zarządzanie interesariuszami
  - zarządzanie zamówieniami:
    - \*Planowanie zakupów, przejęć i kontraktów
    - \*Zbieranie odpowiedzi dostawców, wybór dostawców
    - \*Administracja kontraktem i zamknięcie kontraktu
- PMBoK daje:
- ujednolicenie terminologii
  - zdefiniowanie podstawowych pojęć
  - usystematyzowanie własnej wiedzy
  - wzbogacenie narzędzi i technik



-wielkie pole do popisu...