

Metodyka prowadzenia PI – zbiór reguł i zaleceń określających sposób organizacji procesu wytwarzania oprogramowania, powstające artefakty i zakres zaangażowania uczestników procesu

Metodyki klasyczne – kosztowne, duży narzut kosztów związany z analizą, projektowaniem dokumentowaniem

Metodyki lekkie – koncentrują się na wytwarzaniu kosztem redukcji artefaktów innych niż ostateczny produkt

Cechy wspólne:

- Dzielią proces na fazy wykonywane (częściowo) w określonym porządku
- Czynności przypisane są rolom,
- Wytwarzają artefakty,
- Narzucają lub proponują zbiory odpowiednich praktyk.

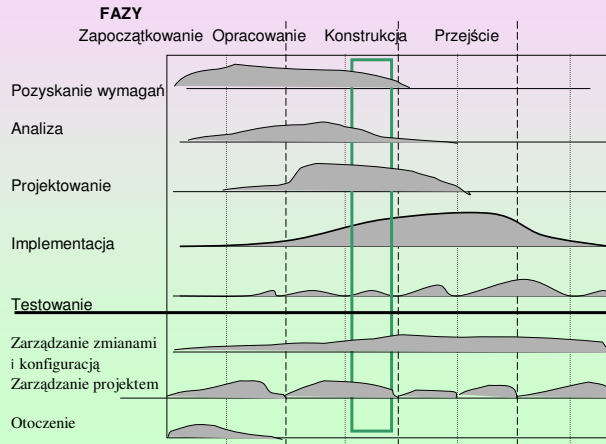
Przykład metodyki klasycznej – USDP (Unified Software Development Process)

Elementy opisu:

- Model rozwoju produktu,
- Fazy rozwoju (zapoczątkowanie opracowanie, konstrukcja przejście)
- Aktywności i dyscypliny
- Role
- Produkty (artefakty)
- Przepływy (częściowo uporządkowane sekwencje aktywności w ramach dyscyplin)

Dyscypliny w USDP:

- Pozyskanie wymagań
- Analiza
- Projekt
- Implementacja
- Testowanie

**Pozyskanie wymagań:**

Produkty (artefakty): przypadki użycia, aktorzy

Role: analityk, ekspert, projektant UI, architekt

Aktywności:

- Wydzielenie przypadków użycia i aktorów - analityk
- Określenie priorytetów PU - architekt
- Uszczegółowienie PU - ekspert
- Prototypowanie UI – projektant UI
- Ustrukturalizowanie PU - analityk

Analiza wymagań:

Produkty (artefakty): model, analiza realizacji PU, struktura modelu, opis architektury modelu

Role: architekt, inżynier PU, inżynier komponentów

Aktywności:

- Analiza architektury - architekt
- Analiza przypadków użycia – inżynier PU,
- Analiza klas – inżynier komponentów
- Analiza pakietów – inżynier komponentów

Projekt:

Produkty (artefakty): model projektowy, klasy projektowe, podział na podsystemy, interfejs, opis architektury, model rozmieszczenia

Role: architekt, inżynier PU, inżynier komponentów

Aktywności:

- Projekt architektury - architekt
- Projekt przypadków użycia – inżynier PU,
- Projekt klasy – inżynier komponentów
- Projekt podsystemu – inżynier komponentów

Implementacja:

Produkty (artefakty): kod wynikowy, procedury instalacyjne, dokumentacja testów jednostkowych

Role: architekt, inżynier komponentów, integrator

Aktywności:

- Implementacja architektury - architekt
- Implementacja klas i podsystemów – inżynier komponentów,
- Integracja systemu – integrator
- Testowanie komponentów – inżynier komponentów

Testowanie:

Produkty (artefakty): model testowania (metodyka), przypadki testowe, procedura testowa, plan testowania, programy i narzędzia pomocnicze, ocena testu, defekty

Role: inżynier testów, integrator, tester, inżynier komponentów

Aktywności:

- Planowanie testów – inżynier testów
- Projektowanie testów – inżynier testów,
- Zaimplementowanie narzędzi pomocniczych – inż.. komp.
- Wykonanie testów integracyjnych i systemowych – inżynier komponentów
- Ocena testów – inżynier testów

Podstawowe założenia metodyki XP:

- Metodyka dostosowana do niewielkich projektów
- Prostota,
- Zdolność do elastycznego planowania funkcjonalności,
- Skrócenie cyklu wytwarzania wersji (stopniowe zmiany),
- Skrócenia fazy analizy i projektowania,
- Ograniczenie produkcji dokumentacji,
- Opieranie się na komunikacji słownej i tekstach źródłowych
- Nieustanne testowanie,
- Zatrudnianie rozsądnych i odpowiedzialnych ludzi,
- Praca dobrej jakości.

Fazy życia projektu w XP

*Idealny projekt XP przechodzi przez krótką **początkową fazę programowania**, potem jest poddawany latom doskonalenia a na koniec podlega poważnej przebudowie, kiedy dotychczasowa postać projektu jest już bezużyteczna...*

- **Badanie – 1 tydzień**
- **Zaangażowanie (uzgodnienie z klientem zakresu projektu, opracowanie terminarza prac – 1-2 dni**
- **Iteracje do pierwszej wersji – 1-4 tygodnie * ilość iteracji (2-6 miesięcy)**
- **Wdrożenie**
- **Utrzymanie – kolejne wersje co 2 miesiące**

Cechy:

- Bardzo skrócona faza rozpoczęcia (inception W RUP)
- Praktycznie brak projektu
- Brak fazy opracowania (elaboration)
- Krótkie cykle pomiędzy wydaniem

Planowanie w XP**Faza badania**

- Pisanie opisu zadania,
- Ocena opisu,
- Podział opisu

Faza zaangażowania

- Sortowanie wg wartości (konieczne, niezbędne, przydatne)
- Sortowanie wg ryzyka (precyzyjnie wycenialne, przybliżone, niewycenialne)
- Ustalenie zakresu wersji

Faza sterowania

- Okresowe uaktualnianie planów

Projektowanie w XP - podstawowe zasady

- Möglichie najmniejsza inwestycja początkowa w projekt,
- Wykorzystanie informacji zwrotnej o projekcie,
- Założenie prostoty ,
- Przyrostowe zmiany projektu,
- Unikanie obciążeń nadmiarową funkcjonalnością.
- Projektowanie przez refaktoring

Artefakty (30)

Stories	Story estimates
Constraints	Task estimates
Tasks	Defects (and associated data)
Technical tasks	Additional documentation from conversations
Acceptance tests	Supporting documentation
Software—code	Test data
Releases	Testing framework tools
Metaphors	Code management tools
The design—UML sketch	Test results
Design documents—produced at the end of the project	Spikes (solutions)
Coding standards	Task work time records
Unit tests	Metrics data
Workspace (development and other facilities)	o Resources
Release plan	o Scope
Iteration plan	o Quality
Reports and notes on any meetings	o Time
Overall plan—budget	Other metrics
Reports on progress	Tracking results

Aktywności W XP:

- Kodowanie,
- Testowanie
- Słuchanie,
- Projektowanie

Odpowiednik w USDP - dyscypliny

- Modelowanie biznesowe
- Analiza wymagań,
- Analiza i projekt,
- Implementacja,
- Testowanie
- Dostarczanie

Podstawowe praktyki XP:

- Krótkie wersje,
- Metafora,
- Prosty projekt,
- Ciągłe testowanie,
- Refaktoring,
- Programowanie parami
- Kolektywne prawa
- 40-godzinny tydzień pracy
- Klient na miejscu
- Standardy kodowania
- Ciągła integracja

Podstawowe różnice w stosunku do innych zaakceptowanych metodologii:

Brak precyzji opisu, brak opisu procesu,

Brak zaleceń co do zakresu i postaci artefaktów,

Skrajne zredukowanie faz analizy i projektu,

Negowanie zasady lawinowego wzrostu kosztów korekt w funkcji wykonanych faz.

Dyskusyjne strony XP:

Skrajne zredukowanie faz analizy i projektu,

Refaktoryzacja

Wspólnota kodu

Programowanie parami

Częsta integracja

Krótkie wersje

XP vs USDP – porównanie ilościowe

RUP	XP
Czas trwania iteracji: 2-24 tygodni Ilość iteracji: 3-9 Łączny czas trwania projektu: 1.5 - 54 miesiące	2 tygodnie ? 2 - 6 miesięcy
Liczba członków zespołu: ?	<=10
Ilość artefaktów: 100	30
Ilość ról: 30	7

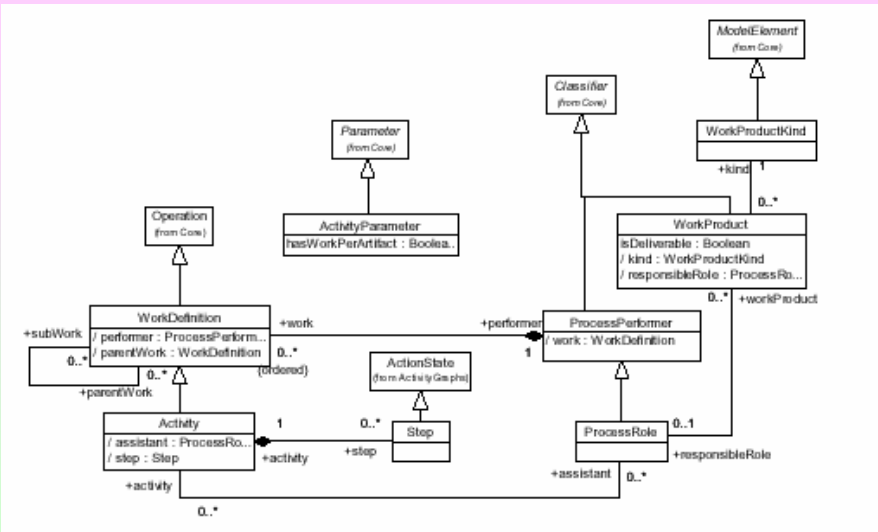
Software Process Engineering Metamodel SPEM

zestaw pojęć (i proponowana dla nich notacja w j.UML), spotykanych i niezbędnych do specyfikacji dowolnego procesu inżynierii oprogramowania.

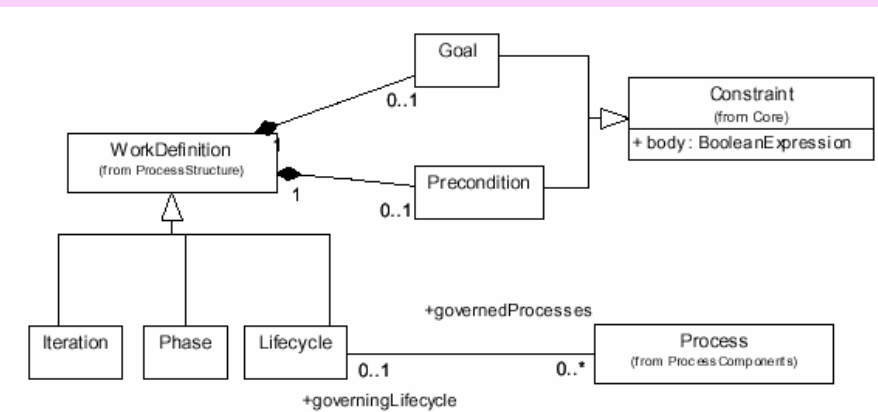
Termin polski	Termin angielski	Znaczenie
Aktywność	Activity	Specjalizacja definicji_pracy, opisującej fragment pracy wykonywanej przez jedną rolę procesu; może składać się z atomowych elementów zwanych krokami.
Cel	Goal	Ograniczenie związane z definicją_pracy, wyrażone w terminach stanów produktów pracy.
Cykl życia	Life cycle	Definicja_pracy, której elementami składowymi są fazy; definiuje zachowanie kompletnego procesu, stosowanego w danym projekcie lub programie.
DefinicjaPracy	WorkDefinition	Rodzaj operacji, która opisuje prace wykonywane w procesie; reprezentuje złożone fragmenty prac, które podlegają dekompozycji
Dyscyplina	Discipline	Specjalizacja pakietu, która grupuje aktywności w obrębie procesu związane z tym samym tematem. Każda aktywność może być powiązana tylko z 1 dyscypliną.

Wykład 3 (26)		INZYNIERIA OPROGRAMOWANIA Metodyki zarządzania projektem - SPEM
Termin polski	Termin angielski	Znaczenie
Faza	Phase	Definicja_pracy, w której warunek wstępny określa kryteria wejścia do definicji pracy, natomiast jej cel, zwany kamieniem milowym definiuje kryteria zakończenia fazy.
Iteracja	Iteration	Definicja_pracy reprezentująca zbiór <u>aktywności</u> , związanych z pewnym fragmentem wytwarzania oprogramowania, którego wynikiem jest wewnętrzna lub zewnętrzna edycja produktu oprogramowania.
KomponentProcesu	ProcessComponent	Wewnętrznie spójny pakiet reprezentujący zbiór elementów, pozwalających opisać fragment procesu.
Krok	Step	Atomowa czynność obliczeniowa.
Opis zewnętrzny	External Description	Opis elementu modelu przeznaczony dla zewnętrznego odbiorcy.
Pakiet	Package	Kontener, który może zarówno posiadać, jak i importować elementy modelu.
Proces	Process	Pakiet grupujący elementy pozwalające opisać proces wytwarzania oprogramowania w terminach wykonawców procesu, ról procesu, definicji_prac, produktów prac oraz powiązanych z nimi przewodników.
ProduktPracy	WorkProduct	Cokolwiek, co jest produkowane, konsumowane lub modyfikowane przez proces.

Wykład 3 (27)		INZYNIERIA OPROGRAMOWANIA Metodyki zarządzania projektem - SPEM
Termin polski	Termin angielski	Znaczenie
Przewodnik	Guidance	zczegółowe informacje na temat elementu modelu. SPEM wyróżniono następujące rodzaje przewodników: technika (<i>technique</i>) profil UML (<i>UML Profile</i>)
RodzajProduktuPracy	WorkProductKind	lista kontrolna (<i>check list</i>) Kategoria produktu pracy, np. dokument tekstowy, dokument wykonywalny itp. opis narzędzi (<i>tool mentor</i>)
Rola procesu	Process role	linia porad (<i>guide/line</i>) Wykonawca procesu, odpowiedzialny za lub asystujący w wykonaniu przypisanych mu produktów pracy. szablon (<i>template</i>)
Warunek wstępny	Precondition	szacowanie (<i>estimate</i>) Ograniczenie związane z definicją_pracy, wyrażone w terminach stanów produktów pracy.
Wykonawca procesu	Proces performer	Właściciel zbioru definicji_prac w procesie.



Elementy pakietu Process Structure



Elementy pakietu *Process Lifecycle*