Rational Unified Process

Opis metodyki i procesu produkcji oprogramowania

Rational Unified Process

- Rational Unified Process (RUP) – to iteracyjny proces wytwarzania oprogramowania opracowany przez firmę Rational Software, a obecnie rozwijany przez IBM.
- Proces RUP nie jest pojedynczym, ściśle określonym procesem, ale raczej szablonem procesu. Został on zaprojektowany w celu przystosowania do charakteru konkretnej organizacji (przedsiębiorstwa), konkretnego zespołu projektowego lub nawet charakteru konkretnego projektu. Z szablonu RUP można wybrać elementy w zależności od konkretnych potrzeb.

Analiza procesu wytwórczego

- Autorzy procesu skupili się na diagnozowaniu charakterystyk projektów, które zakończyły się fiaskiem. Postępując w ten sposób, próbowali poznać przyczyny owych niepowodzeń. Przyglądali się również ówcześnie istniejącym procesom inżynierii oprogramowania i sposobom, w jaki rozwiązywały one problemy.

Analiza procesu wytwórczego

Najczęstsze błędy napotykane w procesie realizacji projektów:

- Zarządzanie wymaganiami (najczęściej brak zarządzania nimi)
- Niejednoznaczna, nieprecyzyjna komunikacja
- Architektura oprogramowania trudna do utrzymania lub rozwijania
- Zbytnia, niepotrzebna złożoność oprogramowania
- Niewykryte niespójności w wymaganiach, projekcie, oraz implementacji
- Brak lub niewystarczające testowanie
- Subiektywna ocena postępu projektu
- Brak zarządzania ryzykiem
- Brak automatyzacji prowadzenia projektu

Analiza procesu wytwórczego

- Na podstawie obserwacji błędów jakie najczęściej są popełniane przez zespoły programistów oraz dzięki analizie sposobów ich rozwiązywania zawartych w już istniejących metodologiach inżynierii oprogramowania opracowano zbiór dobrych praktyk, które nazwano Rational Unified Process.

Podstawy i najlepsze praktyki

RUP bazuje na zbiorze zasad inżynierii programowania oraz najlepszych praktykach, takich jak:

- Iteracyjne wytwarzanie oprogramowania (Iterative Development)
- Zarządzanie wymaganiami (Requirement Management)
- Używanie architektury bazującej na komponentach (Component-based architecture)
- Graficzne projektowanie oprogramowania
- Kontrola jakości oprogramowania (Quality Assurance)
- Proces kontroli zmian w oprogramowaniu (Change

Cykl życia projektu

Cykl życia w RUP bazuje na modelu spiralnym. RUP jest dostępny jako struktura prowadzenia projektu, która może być personalizowana w celu przystosowania do specyficznych potrzeb projektowych. Cykl życia w RUP układa zadania w fazy i iteracje.

Projekt został podzielony na cztery fazy:

- Faza rozpoczęcia (Inception phase)
- Faza opracowywania (Elaboration phase)
- Faza konstrukcji (Construction phase)
- Faza przekazania systemu (Transition phase)

Faza rozpoczęcia

- W fazie tej formułowany jest problem - zagadnienie biznesowe (business case). Przy opracowaniu tego zagadnienia określa się jego kontekst (business context); czynniki wpływające na jego powodzenie (success factors) - na przykład spodziewany zwrot z inwestycji, zwiększenie udziału w rynku; oraz prognozę finansową. Dodatkowo uzupełnia się go o prosty model przypadków użycia, plan projektu, wstępną analizę ryzyka oraz opis projektu (główne wymagania, ograniczenia, główna funkcjonalność).

Faza rozpoczęcia

Wynikiem tej fazy są:

- Dokument wizji (Vision)
- Model przypadków użycia (10%-20%)
- Początkowy zestaw definicji
- Przypadek Biznesowy

  (kontekst biznesu, kryteria sukcesu, prognozy finansowe)
- Dokument podsumowujący studium osiągalności
- Plan projektowy (fazy i iteracje)
- Model Biznesowy (jeśli jest wymagany)
- Prototyp

Faza opracowania

- W tej fazie projekt systemu nabiera kształtów. Przeprowadzona jest analiza dziedziny zagadnienia i budowana podstawowa architektura systemu.
- Jeżeli projekt nie może przejść tej fazy, ciągle mamy czas na jego zaniechanie, lub ponowne opracowanie. Przechodząc do następnej fazy przechodzimy w obszar większego ryzyka, w którym zmiana (np. wymagań) jest dużo trudniejsza i znacząca.

Faza opracowania

Wynikiem tej fazy są:

- Kompletny model przypadków użycia (min. 80%)
- Dodatkowe wymagania
- Opis architektury
- Prototyp
- Końcowy plan projektu
- Specyfikacja procesów
- Wstępna wersja podręcznika użytkownika (opcjonalnie)

Faza konstrukcji

- W fazie tej główny nacisk położony jest na budowę komponentów i innych funkcjonalności opracowywanego systemu. W tej fazie odbywa się większość prac programistycznych. W większych projektach może być wiele iteracji konstrukcji, w celu podzielenia dziedziny przypadków użycia na mniejsze, zarządzalne poddziedziny. Pozwala to także na szybsze przekazywanie części prac (lub prototypów).

Faza konstrukcji

Wynikiem tej fazy są:

- Działająca wersja oprogramowania zintegrowana z platformą docelową
- Podręcznik użytkownika
- Opis wydania

Faza przekazania

- W tej fazie produkt przekazywany jest od zespołu programistycznego do użytkowników końcowych (potocznie mówiąc: do produkcji). W tej fazie znajdują się takie czynności jak: trening użytkowników końcowych i administratorów, testy akceptacyjne (testy beta). Sprawdzana jest zgodność produktu z miarami jakości określonymi w pierwszej fazie.

Faza przekazania

- Wynikiem tej fazy jest działający system posiadający funkcjonalności określone w fazie pierwszej.

Dyscypliny

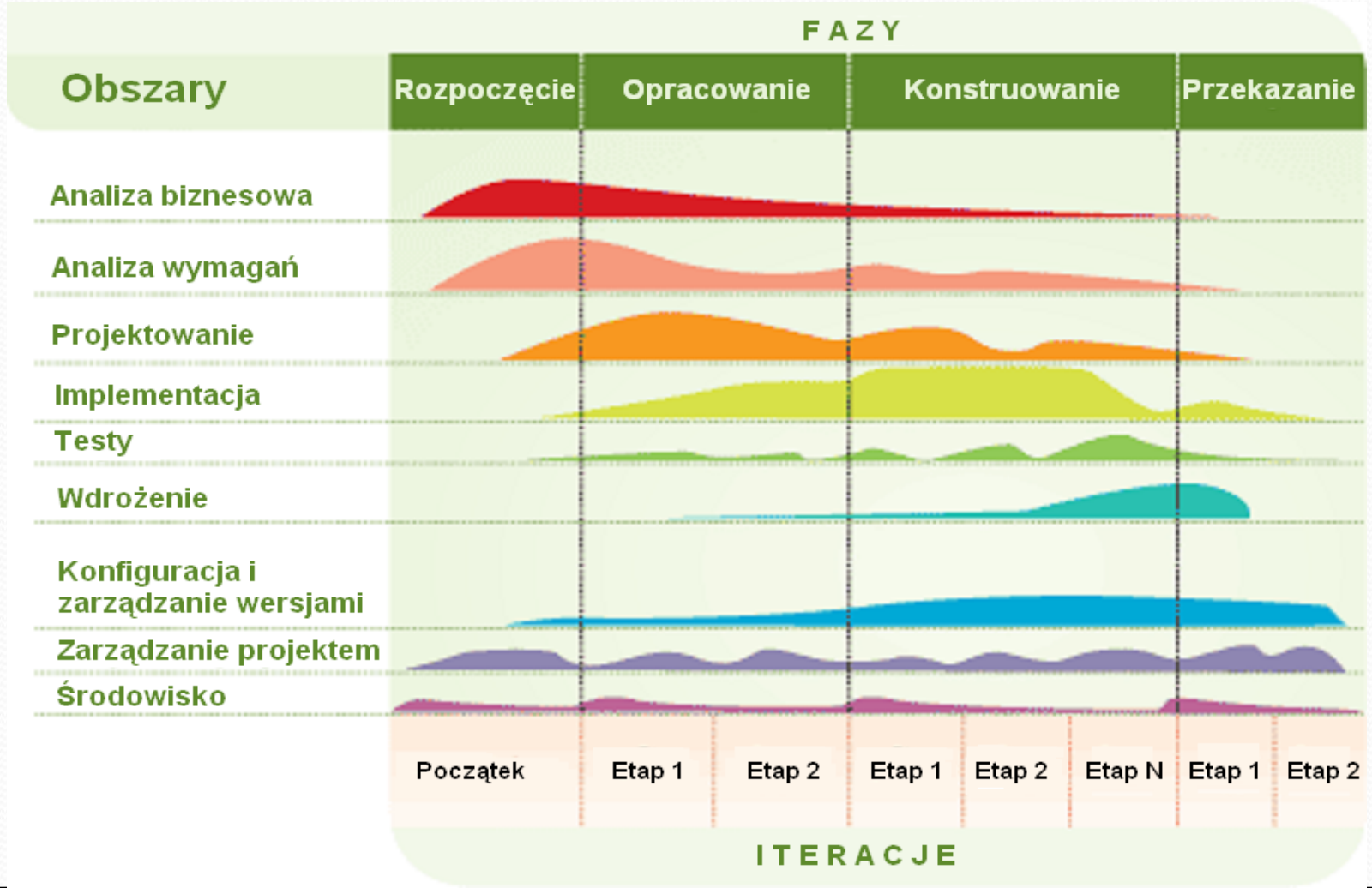
Z każdą z powyżej opisanych faz związane są grupy czynności (dyscypliny), które wykonuje się podczas każdej z iteracji. Są to:

- Dyscypliny inżynierskie (Engineering Disciplines):
  - Modelowanie biznesowe (Business modeling)
  - Wymagania (Requirements)
  - Analiza i projektowanie (Analysis and design)
  - Implementacja (Implementation)
  - Testowanie (Test)
  - Wdrożenie (Deployment)

Dyscypliny

- Dyscypliny pomocnicze (Supporting Disciplines):
  - Zarządzanie zmianami oraz konfiguracją (Configuration and change management)
  - Zarządzanie projektem (Project management)
  - Środowisko (Environment)

Dyscypliny


FAZY
Obszary
Rozpoczęcie
Opracowanie
Konstruowanie
Przekazanie
Analiza biznesowa
Analiza wymagań
Projektowanie
Implementacja
Testy
Wdrożenie
Konfiguracja i zarządzanie wersjami
Zarządzanie projektem
Środowisko
Początek
Etap 1
Etap 2
Etap 1
Etap 2
Etap N
Etap 1
Etap 2
ITERACJE

Modelowanie biznesowe

- Celem modelowania biznesowego jest przede wszystkim zapewnienie komunikacji i lepsze zrozumienie pomiędzy biznesem (inżynieria biznesowa) a IT (inżynieria oprogramowania). Zrozumienie biznesu oznacza, że inżynierowie oprogramowania muszą zrozumieć strukturę i dynamikę organizacji swojego klienta, jego bieżące problemy i możliwe usprawnienia. Muszą także zapewnić wspólne zrozumienie celów pomiędzy klientami, użytkownikami końcowymi a programistami.

Modelowanie biznesowe

- Modelowanie biznesowe tłumaczy w jaki sposób opisać wizję organizacji, w której będzie wdrożony system i jak później użyć jej do opisania procesów, ról i odpowiedzialności w organizacji.

Wymagania

- Celem Wymagań jest opisanie tego, co system powinien robić. Wymagania zbierane są przez analityków, którzy odkrywają je, klasyfikują i dokumentują. Proces zbierania wymagań polega na dyskusji i uzgodnieniach pomiędzy tworzącymi system a klientem.

Analiza i projektowanie

Celem Analizy i projektowania jest zobrazowanie sposobu w jaki będzie tworzony system w fazie implementacji

- Zadaniem analizy jest transformacja wymagań do postaci klas i podsystemów w oparciu o przypadki użycia i wymagania funkcjonalne
- Na etapie projektowania wyniki analizy są przystosowywane do wymagań niefunkcjonalnych oraz ograniczeń środowiska implementacji

Analiza i projektowanie

Na tym etapie są tworzone:

- Model analityczny
- Model projektowy
- Interfejsy

Implementacja

Polega na wytworzeniu działającej aplikacji na podstawie modelu stworzonego w fazie projektowania

- Opracowanie planu scalania systemu
- Implementacja komponentów
- Scalanie podsystemów i całego systemu

Testowanie

Celami dyscypliny testowania są:

- Weryfikacja interakcji pomiędzy obiektami.
- Weryfikacja poprawnej integracji komponentów.
- Sprawdzenie, czy wszystkie wymagania zostały zaimplementowane w sposób poprawny.
- Identyfikacja i sprawdzenie, że defekty zostały usunięte przed wdrożeniem oprogramowania.

Wdrożenie

Celem wdrożenia jest dostarczenie oprogramowania do użytkowników końcowych. Na ten cel mogą składać się:

- Fizyczne wytworzenie wersji instalacyjnej oprogramowania.
- Opakowania oprogramowania
- Dystrybucja oprogramowania
- Instalacja oprogramowania
- Utworzenie dokumentacji i pomocy dla użytkowników
- Zapewnienie pomocy i wsparcia technicznego

Zarządzanie zmianami oraz konfiguracją

- Dyscyplina zarządzania zmiami w RUP dotyka trzech obszarów:
- Zarządzanie konfiguracją - wersjonowanie artefaktów, utrzymywanie rejestru zależności pomiędzy artefaktami
- Zarządzanie zleceniami zmian - utworzenie rejestru propozycji i zleceń zmian
- Zarządzanie stanem i miarami - określenie i przechowywanie informacji na temat zleceń zmian takich jak: stan, przyczyna, natura, priorytet

Zarządzanie projektem

RUP na tym etapie skupia się na:

- Planowaniu projektu iteracyjnego w ramach całego cyklu i pojedynczych iteracji
- Zarządzaniu ryzykiem
- Kontroli realizacji projektu i monitorowaniu postępów

Nie próbuje natomiast objąć wszystkich aspektów zarządzania projektem, takich jak:

- Organizacja zespołów
- Zarządzanie budżetem
- Zarządzanie umowami ze sprzedawcami i klientami

Środowisko

- Wybór i określenie narzędzi, które będą użyte w procesie wytwórczym
- Identyfikacja środowiska systemowego

Koniec