

# *Metodyki wdrażania systemów informatycznych*

Beata Karasek

Ewa Sitarek

# Metodyki wdrażania systemów

W procesie wytwórczym oprogramowania oraz przy wdrożeniach systemów informatycznych projektów często wykorzystuje się następujące metodyki:

- PRINCE 2,
- RUP (Rational Unified Process),
- MSF (Microsoft Solution Framework),
- PMBoK (Project Management Body of Knowledge),
- EUP (Enterprise Unified Process)

# *Metodyka RUP*

# Wprowadzenie

Język UML jest niezależny od przyjętych procedur projektowych, co oznacza, że nie jest związany z jakimkolwiek szczególnym cyklem tworzenia oprogramowania. RUP to jeden z cykli, które wyjątkowo pasują do UML. Ma na celu wspomaganie opracowywania, w ramach przewidywalnych terminarzy i budżetów, oprogramowania wysokiej jakości, które spełnia oczekiwania użytkowników. Obejmuje wiele sprawdzonych technik, które można łatwo dostosować do wymagań różnych przedsięwzięć programistycznych i przedsiębiorstw. Z punktu widzenia kadry zarządzającej RUP ułatwia przydzielanie zadań i zobowiązań w ramach firmy produkującej oprogramowanie.

# Metodyka RUP

RUP jest to iteracyjna i przyrostowa metodyka przedstawiona przez firmę Rational. (Teraz IBM). Pierwsza jej wersja powstała w 1998 roku.

RUP, czyli Rational Unified Process jest konfigurowalną platformą do obsługi procesu tworzenia oprogramowania, która zapewnia najlepsze praktyki i jest oparta na konfigurowalnej architekturze, dzięki której można wybierać i wdrażać tylko te składniki procesu, które są potrzebne na danym etapie projektu.

# Metodyka RUP cd.

Metodyka RUP jest oparta na pomysłach i doświadczeniu najlepszych firm w branży informatycznej, partnerów i tysięcy rzeczywistych projektów, doskonale połączonych w efektywny zestaw najlepszych praktyk, schematów przepływu pracy i artefaktów umożliwiających iteracyjne tworzenie oprogramowania.

Do praktyk tych należą:

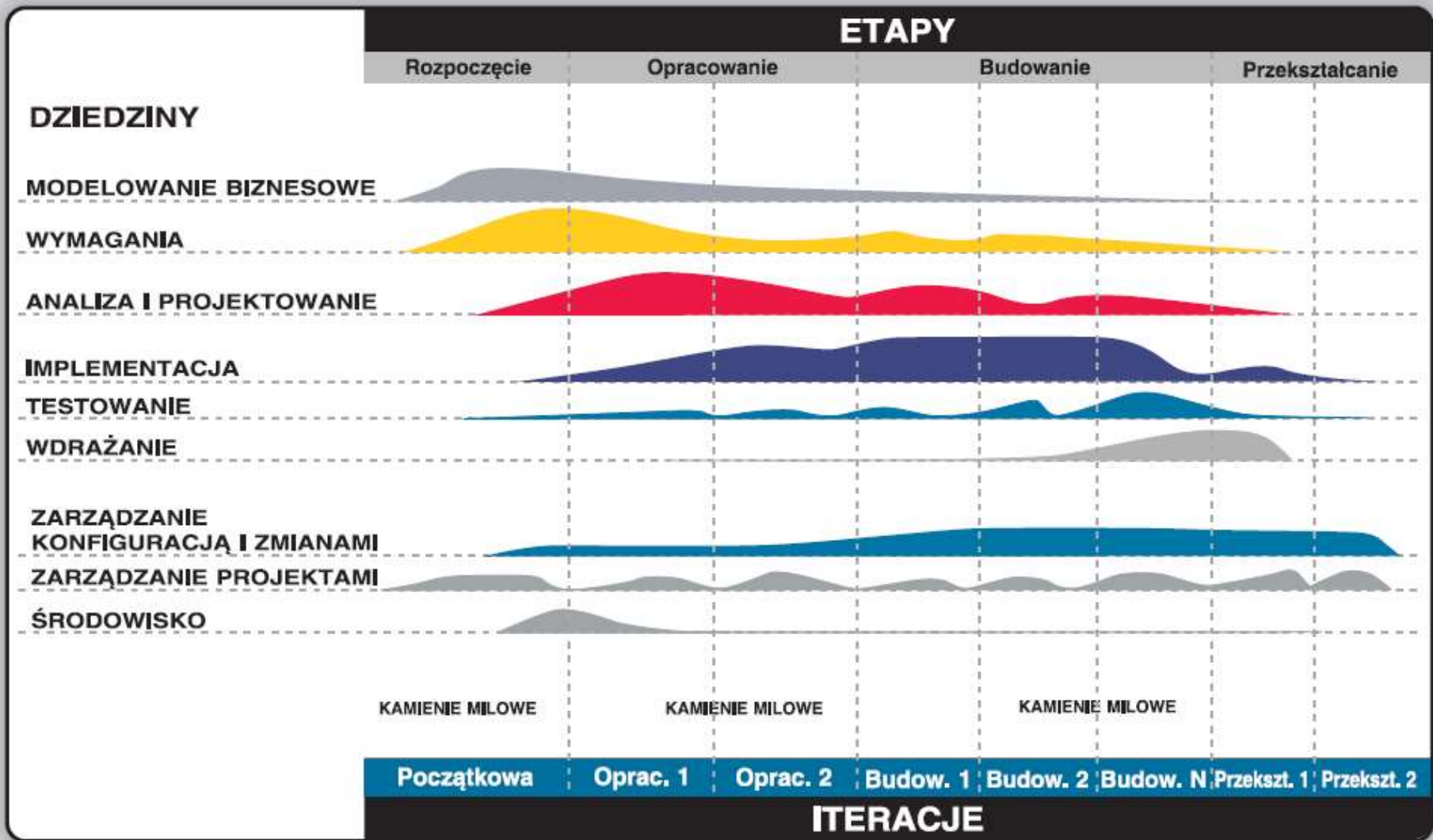
- Iteracyjny rozwój,
- Zarządzanie wymaganiami,
- Architektura oparta o komponenty,
- Wizualne modelowanie (graficzne modelowanie oprogramowania),
- Systematyczna weryfikacja jakości (kontrola jakości),
- Zarządzanie zmianami,

# Cykle metodyki RUP

Metodyka RUP zakłada, że oprogramowanie będzie tworzone w cyklach. Każdy z tych cykli dostarcza kolejną wersję oprogramowania. Pierwszy cykl to cykl początkowy, kolejne zaś noszą nazwę ewolucyjnych. Cykl życia oprogramowania można podzielić na następujące etapy:

- Rozpoczęcie (Inception)
- Opracowanie (Elaboration)
- Konstruowanie (Constuction)
- Przekazanie (Transition)

Każdy z etapów kończy się wytworzeniem odpowiedniego artefaktu





# Faza rozpoczęcia

- Ma na celu ustaleniu zakresu projektu i warunków granicznych.
- Podczas fazy rozpoczęcia zakładamy biznesowy przypadek dla systemu i ustalamy zakres projektowy. Aby tego dokonać trzeba zidentyfikować wszystkie zewnętrzne byty, z którymi system będzie współpracować (aktorzy) i zdefiniować naturę tego współdziałania na wysokim poziomie. To wymaga identyfikacji wszystkich przypadków użycia i opisanie ich w szczegółowy sposób. Biznesowy przypadek zawiera kryteria sukcesu, ocenę ryzyka oraz ocenę potrzebnych zasobów. Faza planowania pokazuje daty głównych kamieni milowych.

# Faza rozpoczęcia cd.

Wynikiem fazy rozpoczęcia są:

- Dokument wizji: ogólna wizja wymagań rdzenia projektu, kluczowych cech i głównych ograniczeń
- Model przypadków użycia (10% -20% całości).
- Początkowy zestaw definicji.
- Przypadek biznesu, który zawiera kontekst biznesu, kryteria sukcesu (dochód, rozpoznanie rynku, itd.) i finansowe prognozy.
- Ocena ryzyka.
- Plan projektowy prezentujący fazy i iteracje.
- Biznesowy model (o ile jest wymagany).
- Jeden albo kilka prototypów.

# Faza opracowania

Celem tej fazy jest bardziej szczegółowa analiza domeny problemu, oraz wypracowanie fundamentów dla architektury. Należy wyeliminować te elementy, które są obciążone nieakceptowanym poziomem ryzyka, jak również zająć się rozwinięciem szczegółowego planu projektowego. Decyzje architektoniczne muszą bazować na zrozumieniu całości systemu: jego funkcjonalności i ograniczeń. Faza opracowania jest jedną z najbardziej krytycznych faz i cechuje się wysokimi kosztami i ryzykiem. Wymagania, architektura i plany powinny osiągnąć stabilną postać, tak aby ryzyko zostało zminimalizowane, co pozwoli na określenie. Bada się tutaj różne rozwiązania, budując prototypy - w jednej lub kilku iteracjach. Budowa prototypów ułatwia eliminację ryzyka oraz ustanawianie kompromisów między wymaganiami a możliwościami środowiska implementacji.

# Faza opracowania cd.

Wynikiem fazy opracowania jest:

- Kompletny model przypadków użycia ( w 80%), który powinien zawierać wstępne opracowanie wszystkich przypadków użycia wraz z zidentyfikowanymi aktorami, oraz szczegółowymi specyfikacjami dla większości z tych przypadków.
- Dodatkowe wymagania (funkcjonalne i нефunkcjonalne nie ujęte w żadnym z przypadków użycia)
- Opis architektury
- Prototyp
- Plan dla całości projektu (z uwzględnieniem iteracji i kryteriów przechodzenia między nimi)
- Specyfikacje procesów, które będą wykorzystywane

# Faza konstruowania

Podstawowym zadaniem tej fazy jest budowa i rozwój komponentów oraz ich integracja i testowanie. Faza konstrukcji jest poświęcona procesom wytwarzania. Kładzie ona duży nacisk na zarządzanie zasobami, optymalizację kosztów i planu oraz poprawę jakości. Dla dużych projektów, można zrównoleglić procesy, co przyspiesza realizację, aczkolwiek takie rozwiązanie komplikuje zarządzanie poprzez konieczność synchronizacji przepływów prac. Stabilna architektura i dobry plan znacząco wspierają zadania fazy konstrukcji. Dlatego tak ważne jest to, co wypracowano w fazach poprzedzających.

# Faza konstruowania cd.

Wynikiem fazy konstruowania jest:

- Produkt zintegrowany z docelową platformą
- Podręcznik użytkownika
- Opis bieżącego wydania

# Faza wdrażania

Podstawowym zadaniem tej fazy jest przekazanie produktu do użytkowników końcowych. Z reguły pociąga to za sobą korektę błędów, dokończenie elementów odłożonych, ulepszanie niektórych własności, itp. W praktyce oznacza to z reguły, iż pewien użyteczny (o akceptowalnej jakości) podzbiór własności systemu został ukończony. Wynikiem Fazy Wdrażania jest działający system

# Grupy czynności

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

- specyfikacja wymagań (ang. requirements capture),
- analiza wymagań (ang. requirements analysis),
- projektowanie (ang. design),
- implementacja (ang. implementation),
- testowanie (ang. test),
- konserwacja (ang. deployment).

Natężenie prac związane z różnymi czynnościami zależy od fazy, w której znajduje się rozwój projektu. W początkowych fazach istotne są czynności związane ze zbieraniem wymagań i analizą, natomiast w końcowych istotniejsza staje się implementacja i konserwacja wytworzonego systemu.



# Dziedziny wytwarzania oprogramowania

Metodyka RUP wyróżnia następujące dziedziny wytwarzania oprogramowania:

- Modelowanie biznesowe
- Wymagania
- Analiza i projektowanie
- Implementacja
- Testowanie
- Wdrażanie
- Konfiguracja i zarządzanie zmianami
- Zarządzanie projektem
- Określenie środowiska

# Modelowanie biznesowe

Modelowanie biznesowe jest pierwszym z przepływów prac i powinno poprzedzać proces specyfikowania wymagań na oprogramowanie. W tym modelu, między innymi zajmujemy się:

- analizowaniem struktury i dynamiki organizacji: oprogramowanie ma być wdrażane (tzw. “organizacji docelowej”).
- zrozumieniem aktualnych problemów organizacji, które identyfikują miejsca dla potencjalnych ulepszeń.
- obszernym opisem procesów biznesowych zachodzących u klienta, aby był jednakowo zrozumiały przez wszystkich uczestników projektu (klienta i zespół projektowy).

# Wymagania

Wymagania są drugim etapem prac i z reguły poprzedzone są procesem modelowania biznesowego. Specyfikują, jak zbudować wizję systemu oraz jak przenieść wizję na model przypadków użycia, który (wraz ze specyfikacją uzupełniającą) szczegółowo zdefiniuje zbiór wymagań. Celem tego działania jest:

- Osiągnięcie konsensusu wśród uczestników projektu: „co i dlaczego powinien robić projektowany system”.
- Uzyskanie lepszego zrozumienia wymagań dla systemu przez członków zespołu projektowego.
- Określenie granic systemu.
- Ustanowienie bazy dla planowania iteracji przy pracach projektowych.
- Ustanowienia bazy dla szacowania kosztów i czasu niezbędnego dla realizacji projektu.
- Zdefiniowanie interfejsu użytkownika w oparciu o cele i potrzeby użytkowników.

# Analiza i projektowanie

Podstawowym celem analizy i projektowania jest zamiana wymagań w specyfikację sposobu implementowania systemu: „określenia najlepszej strategii dla jego implementacji”. Osiąga się to poprzez:

- ustanowienie stabilnej architektury - bazy dla budowy systemu łatwego do zrozumienia i rozwijania,
- przystosowanie projektu do środowiska implementacji,
- uwzględnienie własności systemu, takich jak: wydajność, skalowalność, itp.

# Analiza

Zadaniem analizy jest transformacja wymagań na system w postać, która jest dobrze mapowana do obszaru zainteresowań projektantów, czyli do zbioru klas i podsystemów. Ta transformacja oparta jest w RUP o przypadki użycia i wymagania funkcjonalne. Analiza skupia się na funkcjonalności systemu i ignoruje zarówno wymagania niefunkcjonalne, jak i ograniczenia środowiska implementacji. Można powiedzieć, że analiza prowadzi do uzyskania „prawie idealnego” obrazu systemu.

# Projektowanie

Zadaniem projektowania jest przystosowanie wyników analizy do wymagań niefunkcjonalnych i ograniczeń środowiska implementacji. Projektowanie jest uszczegółowieniem analizy. Aktywności skupione są na optymalizacji projektu systemu, w konkretnym środowisku implementacji, z jednoczesnym zapewnieniem pełnego pokrycia dla funkcjonalności.

# Artefakty

Artefakty to informacje i elementy użyte lub wytworzone w czasie produkcji oprogramowania.

Główne artefakty wytworzone podczas fazy analizy i projektowania to:

- *Model projektowy*, który składa się z klas, pakietów i podsystemów. Pakiety i podsystemy traktowane są jako rodzaj środka organizacyjnego pozwalającego na zmniejszenie złożoności modelu wynikowego.
- *Model analityczny* jest to artefakt będący efektem aktywności związanych z fazą analizy. Będąc rodzajem abstrakcji (generalizacji modelu projektowego), skupia się wyłącznie na funkcjonalności systemu. Dalsze uszczegóławianie modelu jest przeprowadzane w trakcie projektowania.
- *Interfejsy*, które specyfikują zbiór operacji możliwych do wykonania na elemencie modelu. Interfejsy specyfikowane są poprzez sygnatury operacji dostępnych dla danego elementu modelu. Sygnatura każdej operacji jest opisywana przez liczbę i rodzaj argumentów oraz typ wartości zwracanej.

# Implementacja

Implementacja jest definicją sposobu tworzenia lub wyliczania czegoś.

Celem tej fazy jest wytworzenie działającej aplikacji. Opiera się na modelu stworzonym podczas fazy projektowania.



# Testy

Podczas testów sprawdzana jest zarówno zgodność z wymaganiami przygotowanymi podczas wcześniejszych etapów jak również stabilność działania aplikacji.

Na testowanie składają się następujące aktywności:

- przygotowanie planu testów
- projektowanie i implementacja testów
- wykonanie testów integracyjnych oraz testów systemowych
- analiza wyników

# Wdrożenie

Wdrożenie polega na ustaleniu konfiguracji gotowego systemu.

Celem wdrażania jest udane wytworzenie końcowego produktu i dostarczenie oprogramowania dla końcowych użytkowników. To wiąże się z szerokim zakresem dodatkowej działalności, m. in.:

- fizyczne wytworzenie wersji instalacyjnej oprogramowania.
- opakowania oprogramowania
- dystrybucja oprogramowania.
- instalacja oprogramowania.
- utworzenie dokumentacji i pomocy dla użytkowników.

# Wdrożenie cd.

Faza wdrożenia w wielu przypadkach zawiera też działalności takie jak:

- planowanie i przeprowadzenie testów beta
- migracja istniejącego oprogramowania albo danych.
- ogólna akceptacja.

# Zarządzanie konfiguracjami

Jest to panowanie nad zmianami i dbanie o spójność produktów pracy. W tej fazie opisujemy jak skontrolować liczne artefakty produkowane przez dużą ilość osób pracujących nad wspólnym projektem.

Kontrola pomaga uniknąć nadmiernych kosztów i zapewnia, że wynikłe artefakty nie są w konflikcie z powodu wymienionych niżej problemów:

- **Symultaniczne uaktualnienia** - kiedy dwóch albo więcej pracowników pracuje oddzielnie na tym samym artefakcie, ostatni dokonuje zmiany i niszczy pracę poprzedniego.
- **Limitowane zawiadomienia** - kiedy problem w artefaktach jest podzielony na wielu producentów i kilku z nich nie informuje o zmianach.
- **Wiele wersji** - niektóre największe programy są rozwinięte w ewolucyjnych wersjach. Jedna wersja mogłaby być w użyciu klienta, kiedy inna jest w próbie a trzecia jest nadal w rozwoju. Jeśli problemy są znalezione w każdej wersji, zmiany muszą zostać rozpropagowane pomiędzy nimi.

# Zarządzanie przedsięwzięciem

Opisanie rozmaitych strategii prowadzenia procesu iteracyjnego.

W fazie tej określamy jak można zarządzać równoległym rozwojem oraz jak zautomatyzować proces tworzenia. Jest to szczególnie ważne w iteracyjnym procesie. Określamy tu również jak można prowadzić dziennik kontroli.

# Zarządzanie projektem

Zarządzanie projektem wymaga umiejętności lawirowania pomiędzy konkurującymi często wymaganiami, ograniczeniami i ryzykami, po to by zaspokoić potrzeby klienta i użytkowników końcowych.

Trzy główne cele zarządzania projektem to:

- Dostarczenie praktycznych wskazówek wspomagających planowanie prac, organizowanie zespołu (przydział zadań do grup, osób i ustalanie odpowiedzialności), kontrola realizacji projektu i monitorowanie postępów
- Dostarczenie szablonów dla zarządzania ryzykiem.
- Dostarczenie szablonów dla zarządzania procesem budowy złożonego oprogramowania.

# Zarządzanie projektem cd.

W RUP położono nacisk głównie na:

- Planowanie obejmujące szerszy zakres projektu (z uwzględnieniem cykli, faz i iteracji) oraz planowanie szczegółowe dla poszczególnych iteracji.
- Zarządzanie ryzykiem (wykrywanie potencjalnych problemów).
- Pomiar (metryki) i monitorowanie postępów (odpowiednio do założonych planów).

# Określenie środowiska

Określenie środowiska to opisanie infrastruktury niezbędnej do opracowania systemu.

Celem etapu „Określenie środowiska” jest wybranie i dostarczenie narzędzi, które będą użyte w procesie tworzenia aplikacji. Podczas tego etapu następuje również określenie środowiska systemowego.



# *Metodyka MSF*

# Metodyka MSF

**Microsoft Solution Framework (MSF)** został opracowany przez Microsoft Consulting Services. MSF stanowi renomowany standard i jest stosowany na całym świecie przy realizacji wszelkich dużych projektów prowadzonych z udziałem firmy Microsoft i innych partnerów. Metodologia MSF jest zgodna ze standardami największych światowych firm analitycznych. Możliwość ruchu zawartej w projekcie wiedzy do formatów wykorzystywanych w konkretnych narzędziach analitycznych pozwala na pełną kontrolę spójności i niesprzeczności modelu, tak przez jego autora, jak i dowolne firmy zewnętrzne

# Metodyka MSF cd.

MSF jest zbiorem zasad projektowych i najlepszych praktyk wypracowanych w trakcie realizacji setek projektów przez firmę Microsoft. Przesłanką tak zbudowanego procesu Wytwarzania Systemów Informatycznych jest skrócenie czasu powstania Systemów Informatycznych, redukcja kosztów procesów ryzyka przy odpowiednio wysokim poziomie jakości.

MSF jest elastycznym, zintegrowanym zbiorem modeli, które pomagają organizacjom przy gromadzeniu zasobów, ludzi i metod potrzebnych do przystosowania infrastruktury technicznej do własnych celów. Przy planowaniu i zarządzaniu projektem migracji organizacja może korzystać z MSF wraz ze swoimi własnymi narzędziami i metodami.

# Etapy MSF

- Tworzenie wizji
- Planowanie
- Projektowanie
- Wdrażanie

# Tworzenie wizji (Definicja celów i ograniczeń).

Etap ten pozwala zespołowi skupić się na utworzeniu solidnego projektu spełniającego potrzeby przedsiębiorstwa. Należy unikać szczegółowego rozważania kwestii podrzędnych i zamiast ulepszania złych procesów należy utworzyć dobre. Najlepsze wyniki daje myślenie nie tylko o najbardziej widocznych aktualnych potrzebach, lecz także o zasadniczych powodach tych potrzeb. Ważne jest także rozważenie podobnych sytuacji w innych działach oraz potrzeb, które mogą zaistnieć w przyszłości.

# Planowanie (Napisanie Parametrów funkcyjnych i Planu projektu).

Proces ten obejmuje zebranie informacji o istniejących usługach, definicję nowej usługi oraz ocenę potrzebnych zasobów i czasu. W ten sposób uzyskać można informacje potrzebne do określenia parametrów funkcyjnych oraz opracowania macierzystego planu projektu i harmonogramu. Etap planowania zakończy się akceptacją planu przez zespół projektowy.

# Projektowanie (projektowanie, testowanie i budowanie systemu).

Zaakceptowane parametry funkcyjne i plan projektu stanowią podstawę rozpoczęcia intensywnej pracy nad projektowaniem. Podczas tego etapu zespół projektancki tworzy i sprawdza szczegółowy projekt, przeprowadzając testy i usuwając błędy. System zostaje zbudowany, rozpoczyna się szkolenie administratorów i najważniejszych użytkowników, a na końcu przeprowadzany jest pilotaż. Na etapie projektowania kluczowym wydarzeniem jest wydanie projektu. W tym momencie funkcje produktu oraz plany wdrażania i wsparcia technicznego powinny być sprawdzone. Etap projektowania jest zakończony, a funkcje niezrealizowane powinny być udokumentowane w celu włączenia ich w następny projekt.

# Wdrażanie (udostępnianie nowej usługi użytkownikom).

Ostatnim etapem procesu migracji, według modelu MSF, jest wdrażanie. Na tym etapie zespół musi przejść od projektowania eleganckich rozwiązań do spełniania surowych wymogów pełnego i wszechstronnego testowania. Celem tego etapu jest udostępnienie nowych usług użytkownikom. Wymaga to zakończenia szkolenia użytkowników i administratorów, wprowadzenia systemu do użytku, kontrolowania go i usuwania błędów. Po zakończeniu tego etapu system może zostać formalnie przekazany zespołom Operacji i Wsparcia technicznego, które będą go obsługiwać i konserwować.



# Podstawowe zasady MSF:

Zbiór zawiera 8 podstawowych zasad:

1. rozwijaj komunikację w zespole oraz poza nim
2. podążaj za wcześniej określonym i znanym przez wszystkich celem
3. "uprawniaj" członków grupy
4. wprowadź wspólną i jasną odpowiedzialność
5. skup się na dostarczaniu wartości biznesowych
6. bądź przygotowany, spodziewaj się zmian
7. inwestuj w jakość
8. ucz się na doświadczeniach

Razem zasady te wyrażają filozofię MSF.

# Rozwijaj komunikacje w zespole oraz poza nim

Komunikacja pomiędzy ludźmi jest niezwykle ważnym elementem. Każda osoba tworząca projekt posiada pewne umiejętności. Aby maksymalizować ich wykorzystanie informacje muszą być łatwo dostępne i współdzielone. Bez tego, członkowie zespołu nie będą w stanie wykorzystać efektywnie swoich umiejętności, co znacznie spowolni pracę całego zespołu.

Ponadto brak komunikacji może wprowadzić nieporozumienia oraz niedomówienia, które doprowadza do powstawania błędnych produktów.

# Podążaj za wcześniej określonym i znanym przez wszystkich celem

Dobrze określony cel jest kluczem do sukcesu. Wspólna wizja systemu pomaga zawsze widzieć ten cel i koncentrować się na dążeniu do niego. Brak takiego celu budzi strach i niepewność wśród pracowników. Bez wspólnej wizji różni ludzie mogą różnie postrzegać cel aplikacji i sposób dążenia do niego, co znacznie pogarsza współpracę w grupie.

Tak więc, niezwykle ważne jest, by wszyscy pracownicy byli w pełni świadomi zadań i celów jakie przed nimi stoją.

# „Uprawniaj” członków drużyny

Strukturą zespołu jest sieć, nie hierarchia. Model zespołu MSF zakłada, że w zespole panuje równość. Każdy z członków równomiernie odpowiada za dążenie zespołu do celu. Takie zespoły akceptują odpowiedzialność nakładaną na nich przez kierownictwo i znacznie efektywniej dążą do celu czując się odpowiedzialnym za część projektu i zwiększając prawdopodobieństwo jego sukcesu.

Pracownicy zobowiązują się do wykonania określonej porcji pracy w pewnym terminie (pracownicy ustalają termin), rozkład taki jest następnie przekazywany kierownictwu. Członkowie zespołu w pełni akceptują taki plan i dążą do jego zrealizowania, gdyż sami go ułożyli.

# Wprowadź wspólną i jasną odpowiedzialność

Każdy członek zespołu odpowiada przed zespołem oraz przed udziałowcami. Każda osoba odpowiada za część rozwiązania. Całkowita odpowiedzialność rozkłada się na zespół, gdyż każda osoba może spowodować porażkę projektu.

Każdy członek zespołu pracuje znacznie efektywniej, jeśli jest świadomy odpowiedzialności drużyny. Ponadto prowokuje to członków zespołu do dbania "nie tylko o swoje obowiązki".

# Skup się na dostarczaniu wartości biznesowych

Skuteczne rozwiązania muszą spełniać podstawową zasadę: przynosić zysk osobie zamawiającej. Zespół musi jasno rozumieć, dlaczego projekt istnieje i jakie korzyści sukces projektu przyniesie dla przedsiębiorstwa oraz dla klienta. Czynny udział klienta w projekcie znacznie ułatwia zrealizować ten cel.

Projekt nie posiada żadnej wartości biznesowej dopóki nie jest w pełni skończony, wdrożony i efektywnie używany.

# Bądź przygotowany, spodziewaj się zmian

MSF potwierdza dualną naturę projektów: panuje w nich zarówno uporządkowanie jak i chaos. MSF zakłada, że należy ciągle oczekiwać na zmiany i nigdy nie można wyizolować stabilnego projektu niepodlegającego tym zmianom.

Zmian należy oczekiwać zarówno od klienta jak i od zespołu projektowego.

MSF opracowało 2 metody, aby przewidywać i radzić sobie ze zmianami:

- Metoda zespołowa
- Metoda procesowa

# Inwestuj w jakość

MSF uważa, że dążenie do uzyskania jak najlepszej jakości jest ciągłe i nigdy nie ma końca. Nie istnieje pojęcie najwyższej jakości. Jakość jest zawsze względna. Model zespołu MSF obarcza każdego członka zespołu odpowiedzialnością za jakość projektu. Członkowie zespołu muszą stale zapewniać, by jakość projektu odpowiadała wymaganiom klienta.

Testowanie ma miejsce przez cały cykl życia projektu. Definiowane są kamienie milowe, które są następnie weryfikowane pod względem kryteriów jakości.



# Ucz się na doświadczeniach

MSF zakłada, że ciągła nauka i bazowanie na zdobytych doświadczeniach znacznie przyczyniają się do osiągnięcia sukcesu. Wiedza na temat projektu przekazana członkom innych zespołów znacznie zwiększy ich kompetencje i usprawni podejmowanie decyzji w przyszłości.

Przekazywanie wiedzy zarówno przyczyni się do zmniejszenia ilości popełnianych błędów jak i do wykorzystania trafnych, sprawdzonych już pomysłów i rozwiązań...

# *Metodyka PRINCE2*

# Metodyka PRINCE2

Metodyka PRINCE (Project IN Controlled Environment) została stworzona i była modyfikowana w Wielkiej Brytanii.

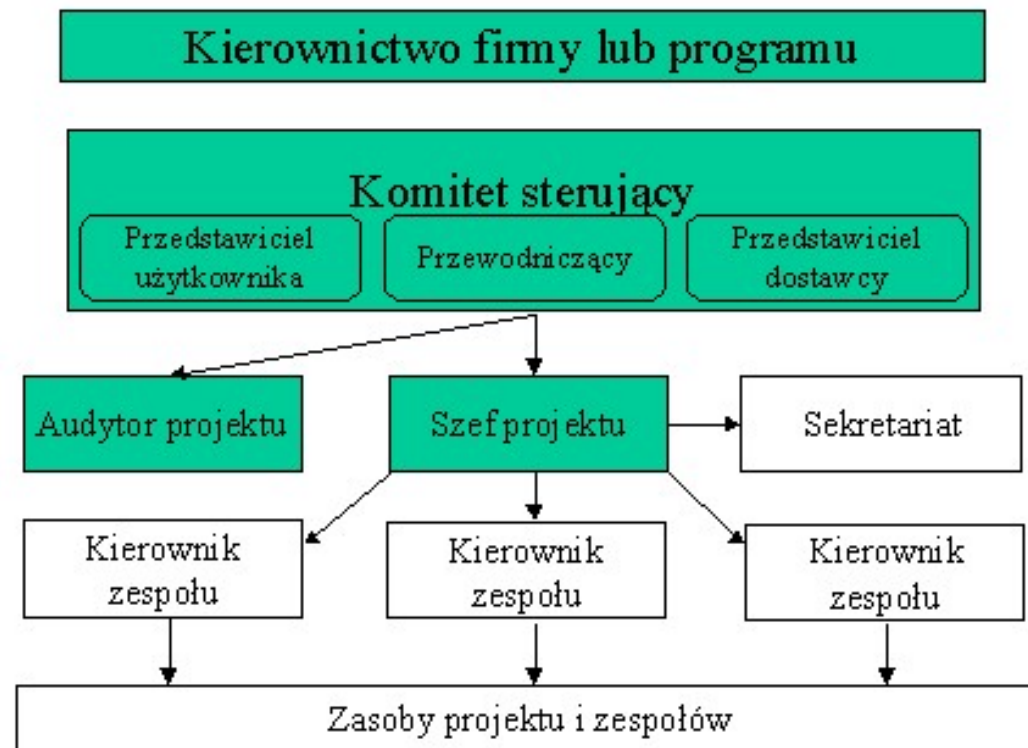
Powstała w roku 1989 dzięki Central Computer and Telecommunications Agency (CCTA). Nie została stworzona jednak od podstaw przez CCTA, lecz powstała na bazie metodyki PROMPT opracowanej przez firmę Simpack System w 1975 roku.

# Metodyka PRINCE2 cd.

Kolejny etap rozwoju opisywanej metodyki przypada na 1996 rok., kiedy to powstała unowocześniona metodyka nazwana PRINCE2. Nowa wersja tej metodyki stała się dużo bardziej uniwersalna i można ją stosować już nie tylko do projektów informatycznych. Brytyjska firma SPOCE Project Management upowszechniła tę metodę czyniąc ją łatwiejszą i bardziej dostępną dla ogółu użytkowników począwszy od małych firm usługowych, a na wielkich korporacjach finansowych kończąc.

W Polsce metodyka PRINCE2 jest od kilku lat dość powszechnie stosowana. Firma CRM S.A w roku 1999 przetłumaczyła ją na język polski oraz uzupełniła tę metodykę dostosowując ją do wymagań rynku polskiego. Ten rozszerzony produkt pod nazwą PRINCE2™ - SPOCE-CRM na mocy zawartych porozumień jest wyłącznie sprzedawany przez firmę CRM S.A.

# Struktura organizacyjna w PRINCE 2



# Zasady metodyki PRINCE 2

- projekt jest skończonym zbiorem aktywności, mającym swój początek i koniec
- projekt musi być zarządzany tak, aby skończył się sukcesem;
- właściwe zaangażowanie zainteresowanych stron, wszyscy muszą mieć pełną jasność co do tego
  - po co dany projekt jest realizowany,
  - co zamierzamy osiągnąć,
  - jak zamierzamy to osiągnąć
  - w jakim stopniu strony ponoszą odpowiedzialność za powodzenie projektu

# Podstawą PRINCE 2 jest osiem procesów:

1. Przygotowanie założeń projektu
2. Planowanie
3. Konstruowanie projektu
4. Strategiczne decyzje projektu
5. Sterowanie etapem
6. Zarządzanie wytwarzaniem produktów
7. Zarządzanie zakresem etapu
8. Zamykanie projektu

# Przygotowanie założeń projektu:

- Określenie zespołu zarządzania projektem i jego funkcji
- Określenie formuły organizacyjnej projektu
- Rejestr ryzyka
- Wstępny plan konstruowania projektu.



# Podstawowe założenia projektu

- o Cel dokumentu
- o Przesłanki projektu
- o Cele projektu
- o Zakres projektu i jego styk z innymi podmiotami
- o Najważniejsze produkty projektu
- o Ograniczenia
- o Założenia, przy jakich projekt ma być realizowany
- o Ogólne korzyści biznesowe i ich uzasadnienie
- o Wstępna ocenę ryzyka
- o Oczekiwania klienta co do jakości
- o Kryteria akceptacji
- o Ogólny plan projektu
- o Ocenianie stanu realizacji i składanie sprawozdań
- o Wymagania finansowe i budżetowe
- o Czas realizacji
- o Wszystkie inne informacje dodatkowe istotne dla powodzenia projektu

# Podprocesy Strategicznych decyzji projektu:

- Decyzja o rozpoczęciu konstruowania projektu
- Decyzja o realizacji projektu
- Decyzja o wprowadzeniu w życie planu kolejnego etapu lub planu awaryjnego
- Doraźne decyzje strategiczne
- Zatwierdzenie zamknięcia projektu

# System dokumentacji

Metodyka PRINCE 2 wprowadza jednolity system dokumentacji projektów:

- Teczka projektu
- Teczka etapu (teczki etapów)
- Teczka jakości
- Teczka merytoryczna

# Teczka projektu:

- Organizacja projektu
- Plany
- Uzasadnienie biznesowe
- Rejestr ryzyka
- Dokumenty dotyczące sterowania
- Lista produktów
- Sprawozdanie z nabytych doświadczeń

# Teczka etapu:

- Organizacja
- Plany
- Dziennik etapu
- Korespondencja
- Lista produktów

# Teczka jakości:

- Charakterystyka produktów
- Oceny jakości
- Zagadnienia realizacyjne (problemy)
- Rejestr problemów realizacyjnych
- Rejestr jakości

# Teczka merytoryczna:

- Elementy konfiguracji
- Rejestr konfiguracji
- Lokalizacja elementów
- Poprawki specyfikacji

# *Metodyka PMBoK*



# Grupy procesów

Metodyka PMBoK zawiera pięć podstawowych grup procesów:

1. Procesy rozpoczęcia.
2. Procesy planowania.
3. Procesy realizacji.
4. Procesy kontroli.
5. Procesy zakończenia.

Procesy mogą zachodzić na siebie w czasie realizacji projektu lub jego fazy. Obowiązkiem kierownika ewentualnie zespołu kierowniczego projektu jest wybranie tych procesów, które mają zastosowanie dla konkretnego projektu.

# Procesy rozpoczęcia

To procesy, które służą zdefiniowaniu i zatwierdzeniu projektu w organizacji:

1. Opracowanie dokumentu otwarcia
2. Opracowanie wstępnego zakresu projektu

# Procesy planowania

Procesy te mają na celu odpowiedzenie na pytanie: jak, w jaki sposób zrealizować zamierzone cele, jakimi środkami, kiedy, w jakiej kolejności itp.,

1. Opracowanie planu zarządzania projektem
2. Planowanie zarządzania zakresem projektu
3. Definiowanie zakresu projektu
4. Utworzenie pakietów roboczych, WBS (ang. Work Breakdown Structure)
5. Zdefiniowanie czynności
6. Porządkowanie czynności
7. Szacowanie zasobów czynności
8. Szacowanie czasu trwania czynności
9. Opracowanie harmonogramu
10. Szacowanie kosztów
11. Budżetowanie kosztów
12. Planowanie jakości
13. Planowanie zasobów ludzkich
14. Planowanie komunikacji
15. Planowanie zarządzania ryzykiem
16. Identyfikacja ryzyka
17. Jakościowa analiza ryzyka
18. Ilościowa analiza ryzyka
19. Planowanie reakcji na ryzyko
20. Planowanie zaopatrzenia
21. Planowanie kontraktów

# Procesy realizacji

Grupują i koordynują wykorzystanie zasobów i ludzi w projekcie w celu wykonania założonego planu,

1. Kierowanie i zarządzanie realizacją projektu
2. Zapewnienie jakości
3. Przyjmowanie członków zespołu
4. Rozwój zespołu
5. Dystrybucja informacji
6. Gromadzenie ofert od sprzedawców
7. Wybór sprzedawców

# Procesy kontroli

Monitorują postępy prac w projekcie, badają ewentualne odchylenia, aby w razie konieczności uruchomić odpowiednie działania zapobiegawcze lub/i korygujące,

1. Monitorowanie i nadzór nad pracami projektu
2. Zintegrowane zarządzanie zmianami
3. Weryfikacja zakresu
4. Sterowanie zakresem
5. Nadzór nad harmonogramem
6. Nadzór nad kosztami
7. Kontrola jakości
8. Zarządzanie zespołem
9. Raportowanie postępu prac
10. Zarządzanie udziałowcami (interesariuszami)
11. Monitorowanie i nadzór nad ryzykiem
12. Administrowanie kontraktem

# Procesy zakończenia

Przygotowanie formalnej akceptacji produktu, finalnego projektu lub jego fazy.

1. Zamknięcie projektu
2. Zamknięcie kontraktu

*KONIEC*