

# RUP Metodický rámec pre rozvoj elektronizácie verejnej správy na báze SOA architektúry

Pri realizácii projektov v rámci Informatizácie verejnej správy a rozvoj elektronických služieb operačného programu Informatizácia spoločnosti je veľmi dôležitým predpokladom úspešnosti je dodržiavať metodiku vývoja nových systémov. Metodika by mala umožniť zladiť prácu rôznych tímov na viacerých projektoch, tak aby bola dodržaná koncepcia a architektúra informačných systémov vo verejnej správe. Pre projekty budovania nových systémov softvérovým vývojom je takouto metodikou Rational Unified Process (RUP), ktorá zdôrazňuje dôležitosť dodržiavanie modelovania a zvolenej architektúry. RUP obsahuje informácie o tom ako vykonávať úlohy v rámci softvérového vývoja, aké sú zodpovednosti členov tímu, aké produkty majú byť výstupom ich práce a akým spôsobom je práca vykonávaná.

## ***Základné princípy***

RUP metodika dodržiava počas vývoja projektu šesť základných princípov:

1. Adaptovanie procesu
2. Dodržiavanie priorít
3. Spolupráca v rámci tímu
4. Doručovanie hodnoty v iteráciách
5. Klásť dôraz na abstraktnosť
6. Kontinuálne sa sústreďovať na kvalitu

**Adaptovanie procesu** znamená, že správny rozsah procesov je aplikovaný na príslušný projekt. Podľa typu a veľkosti projektu sa nastavujú očakávané výstupy v podobe produktov a vykonávaných úloh.

Počas projektu je dôležité neustále vyvažovať **priority projektu**. Zostavenie jasnej množiny požiadaviek sa uskutočňuje na základe reálnej potreby a očakávaní zadávateľov projektu. V rámci prioritizácie sa skúmajú vzťahy medzi časovým harmonogramom, funkcionalitou a nákladmi na projekt.

Je potrebné motivovať jednotlivcov v rámci tímu odstraňovať bariéry medzi rôznymi tímami a zvyšovať mieru **spolupráce** počas zberu požiadaviek, samotného vývoja a fáze prevádzky riešenia.

Pre dosiahnutie možnosti získania objektívneho stavu vývoja riešenia je potrebným aspektom doručovať výsledný kód produktu v pravidelných **iteráciách**. Iterácie projektu je potrebné adaptovať vzhľadom na vývoj projektu. Je potrebné zapracovávať a riadiť zmeny priorít a očakávaní.

Zjednodušenie práce a komunikácie sa dosahuje využívaním modelovacích nástrojov pre vytváranie architektúry riešenia a dosiahnutie úrovne **abstrakcie**. Pri riešeníach na báze SOA architektúry sa

požiadavky riešenia pretvárajú do služieb a riešenie je založené na ich modelovaní, transformácii a znovupoužívaní.

Počas celého životného cyklu projektu je **kvalita** medzi prioritami projektu. V každej iterácii riešenia dochádza k testovaniu a monitorovaniu kvality.

## ***Projektový plán***

RUP metodika rozlišuje medzi plánovaním projektu a plánovaním špecifickej iterácie. V prípravnej fáze projektu sa vyhodnocuje význam projektu z pohľadu vlastníkov projektu a vypracováva sa biznis case, ktorý hodnotí ekonomické prínosy projektu. V ďalšej fáze sa pristupuje k identifikovaniu a vyhodnoteniu rizika projektu. Po schválení biznis case sa štartuje samotný projekt definovaním zodpovedností, plánovaním tímu a zostavením akceptačných kritérií. Plánovanie projektu pokračuje definovaním organizačnej štruktúry tímu a definovaním štruktúry projektu.

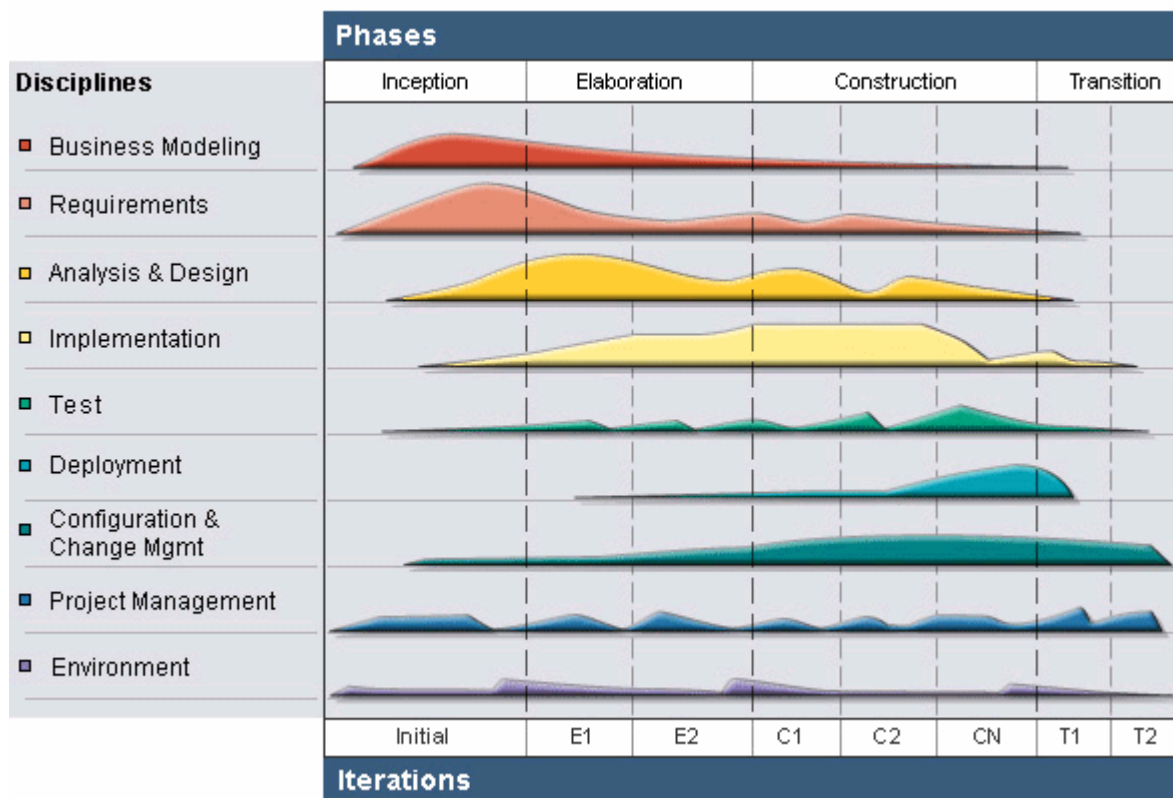
Projektový plán musí obsahovať:

- identifikáciu produktov a fáz ich dodania
- zloženie tímu a zodpovednosti členov tímu
- plán aktivít a dodania artefaktov
- časový harmonogram projektu

## ***Etapy realizácie***

RUP identifikuje štyri fázy vývoja projektov. Každá fáza zameriava projektový tím na iný aspekt projektu, ktorý je asociovaný s niekoľkými míľnikmi. Míľniky pomáhajú projektovým manažérom sledovať vývoj projektu a zabezpečujú dodanie funkcionality v očakávanom čase a kvalite.

Toto je celkový pohľad na koncept RUP projektov. Každý z projektov je rozdelený do 4 fáz. Každá fáza projektu je rozdelená na niekoľko iterácií. Každá iterácia je vertikálne rozdelená na disciplíny, ktoré obsahujú potrebné aktivity v danej fáze projektu.



Každý projekt založený na metodike RUP je rozdelený na cykly. Predmetom každého cyklu je nová verzia produktu. Jeden vývojový cyklus je v RUP rozdelený do 4 po sebe idúcich fáz:

1. Přípravná fáza (Inception),
2. Inicializačná fáza (Elaboration),
3. Realizačná fáza (Construction),
4. Dokončovacia fáza (Transition).

V prípravnej fáze sa definuje rozsah projektu. Počas tejto fázy sa definujú ciele projektu, use case diagramy, kľúčové parametre, indikátory a metriky vývoja projektu.

Inicíalizácia je fáza, kde sa automatizujú všetky pracovné postupy a uskutočňujú sa príslušné úpravy v dokumentoch vzniknutých v predchádzajúcej fáze.

Vo fáze realizácie sa kompletizuje vytváranie zdrojového kódu a jeho testovania. Na konci etapy je vzniknutý softvér pripravený k uvedeniu do prevádzky.

Vo fáze dokončovania vzniká konečná verzia softvérového produktu, dokumentácia, školiace materiály, inštalčné média a dochádza k odstráneniu nájdených chýb. Ak je táto fáza ukončená, implementovanie ďalších zmien vyžaduje novú iteráciu projektu.

## Disciplíny

Defect/enhancement rámci životného cyklu projektu sú aktivity logicky organizované v rámci disciplín. V každej iterácii projektu musia byť splnené aktivity pre každú z disciplín.

Technické disciplíny:

- Biznis modelovanie
- Požiadavky
- Analýza a návrh
- Implementácia
- Testovanie
- Nasadenie

Podporné disciplíny:

- Projektové riadenie
- Riadenie konfigurácie a zmien
- Prostredie

Cieľom **business modelovania** (Business Modeling) je pochopiť biznis organizácie, ktorý je relevantný zo systémom, ktorý sa ide vyvíjať. Dochádza k zmapovaniu rolí, zodpovedností a procesov.

V disciplíne **požiadavky** (Requirements) dochádza k ujasneniu rozsahu funkcionality systému. V tejto disciplíne je dôležité identifikovanie a prioritizácia funkčných a nefunkčných požiadaviek na systém.

Pri **návrhu riešenia** (Analysis and Design) dochádza k zohľadneniu požiadaviek na systém, obmedzení a aplikovateľných štandardov a predpisov. V tejto fáze vzniká architektúra riešenia, návrh komponentov, služieb a modulov.

**Implementácia** (Implementation) riešenia transformuje návrh riešenia do vykonateľného kódu a vykonávajú sa základné jednotkové testy a integrácia kódov jednotlivých subsystémov.

Objektívne vyhodnotenie funkcionality a kvality sa realizuje prostredníctvom **testovania** (Test). Dochádza k odstraňovaniu nájdených chýb a overeniu implementácie požiadaviek riešenia.

Systém sa stáva dostupný pre užívateľov prostredníctvom **nasadenia systému** (Deployment), kde sa zostavuje plán nasadenia a organizuje sa alfa, beta, pilotné a akceptačné testovanie.

**Riadenie konfigurácie a zmien** (Configuration Management) sa zaoberá riadením verzií konfigurácii a manažovaním požiadaviek na zmeny. Dochádza k nasadzovaniu prostredia a manažovaniu konfiguračných položiek.

Manažovaním rizika, zdrojov a koordinovanie projektového tímu je súčasť **projektového riadenia** (Projekt Management).

Zabezpečenie správnych procesov, štandardov a pravidiel, nástrojov (softvérových a hardvérových) pre tím je riešené v rámci disciplíny **prostredie** (Environment)

## ***Dodanie výsledkov projektu***

Aktivity počas jednotlivých fáz vývoja projektu majú svoje vstupy a výstupy, ktoré sa nazývajú artefakty. Artefakt obsahuje informácie, ktoré boli vytvorené, modifikované alebo použité počas procesu vývoja. Artefakty sú produkty, ktoré vznikajú v rámci projektu.

Popri porozumení a vyjasnení cieľov projektu je potrebné dodať v každej iterácii požadované výstupy - artefakty, ktoré dokumentujú uskutočnenie úloh v príslušných fázach. Každý výsledný produkt súvisí s identifikovanými aktivitami, zdrojmi alebo výsledkami práce v iných fázach projektu. Na konci každej fázy projektu je nevyhnutné skontrolovať doručenie definovaných výstupov:

Tieto artefakty sa očakávajú ako štandardné výstupy RUP projektu:

**Biznis architektúra** (Business architecture) - opisuje procesy a ich realizáciu v rámci projektu

**Biznis case** (Business case)- informácie o nákladoch a benefitoch, ktoré projekt prináša

**Vízia** (Vision)- definuje hlavné ciele projektu

**Use Case model** (Use case model) - definuje funkčné požiadavky na systém

**Vedľajšie požiadavky** (Supplementary requirements) - definovanie vedľajších a nefunkčných požiadaviek na riešenie

**Prototyp užívateľského rozhrania** (User interface prototype) - definovanie a simulácia užívateľského rozhrania s dôrazom na nefunkčné požiadavky dostupnosti rozhrania

**Návrh subsystémov** (Subsystem design) - návrh subsystémov v rámci analytického modelu, pozostávajúceho z komponent

**Zoznam rizík** (Risk list) - zoznam rizík ktoré sa zohľadňujú počas projektu

**Komponenty** (Component) - zoznam jednotiek kódu, ktorý sa bude nasadzovať v rámci systému

**Konfigurácia systému** (Build) – zoskupenie komponent, z ktorých systém pozostáva

**Verzie** (Release) - súbor aktivít, ktoré súvisia s príslušným vývojom verzie

**Funkčné testy** (Functional test) - testy funkcionality pre príslušné požiadavky

**Výkonnostné testy** (Performance test) - testy výkonnosti systému

**Testovacie prostredie** (Test environment) - nastavenie testovacieho prostredia v príslušnej fáze

**Identifikovanie chýb** (Defect/enhancement)- riešenie nájdených chýb užívateľmi

**Vývojové prostredie** (Development environment) - nastavenie vývojového prostredia v vykonávanie jeho zmien

**Zdrojový kód** (Source code) – súhrn všetkých zdrojových kódov a výkonateľných súborov v rámci projektu spolu s potrebnými dátovými súborami

**Inštalčný manuál** (Installation material ) – postup inštalácie produktu pri jeho implementácii

**Školiaci materiál** (Training material) – materiál potrebný pre zaškolenie užívateľov

**Dokumentácia** (User documentation) – kompletný popis systému s pohľady používania systému v prevádzke