RUP pro malé a velké projekty - Rozdíly

4IT421 - Zlepšování procesů budování IS

Jan Nešpor (xnesj10)
Kryštof Pacourek (xpack02)
3. 12. 2012

Vysoká škola ekonomická v Praze
ZS 2012/2013

Obsah

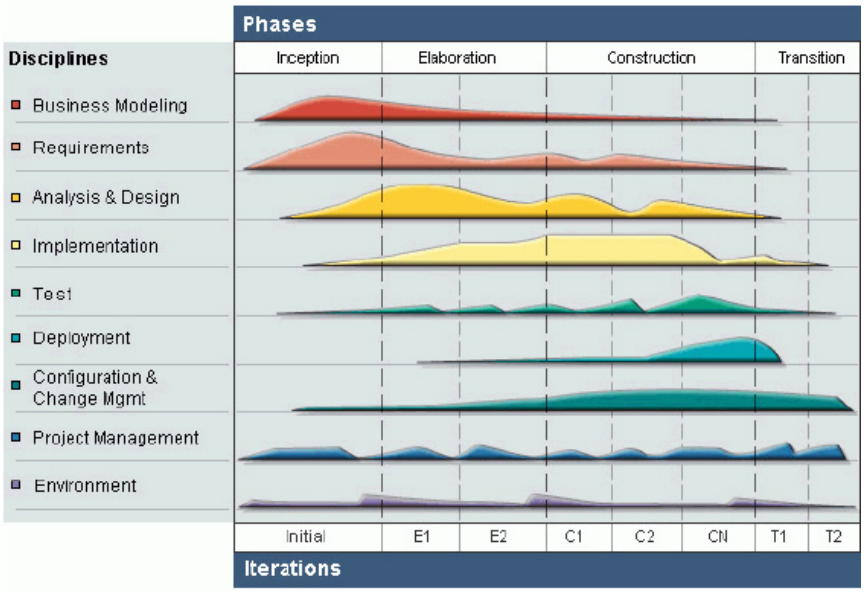
1 Úvod

Tématem této semestrální práce se stala metodika vývoje softwaru RUP (Rational Unified Process) a její dva přístupy k malým a velkým projektům. Konkrétně pak šlo o hledání rozdílů v těchto dvou verzích. Strukturu naší práce tvoří nejprve stručné představení samotné metodiky RUP, její specifika a vývoj. Dále je popsána situace na poli dnešních IT projektů, jejich velikost, komplexita a vliv na jejich úspěšnost. Také je definováno rozdělení IT projektů dle velikosti podle stanovených kritérií. To vše by mělo podtrhnout důležitost a význam realizace malých projektů a využití příslušných metodik vývoje SW. Hlavním obsahem se stává srovnání metodiky RUP pro malé a velké projekty. V případové studii poté byla přiblížena reálná situace použití v prostředí malého projektu. V závěru práce jsou získané poznatky zhodnoceny.

2 Stručně o metodice RUP

Metodika Rational Unified Process (dále jen RUP) je objektově orientovaná metodika spadající do kategorie rigorózních, neboli těžkých metodik budování IS/ICT. Metodika vznikla v roce 1995 spojením přístupu Rational a metodiky Objectory Process Ivara Jacobsona, později byla vytvořena její zobecněná verze Unified Process. (1) Jak již bylo zmíněno, RUP patří do rigorózních metodik, především díky své obsáhlosti a celkové robustnosti, avšak v současné době je vidět snaha tuto metodiku “odlehčovat” a doplňovat o agilní praktiky a tím přizpůsobit RUP jakémukoliv typu projektu.

Metodika RUP je založena na nejlepších praktikách softwarového vývoje, které představují: iterativní vývoj, řízení požadavků, použití komponentové architektury, vizuální modelování, kontrola kvality softwaru a řízení změn. (1) Proces vývoje softwaru demonstruje Obrázek 1, kde horizontální osa představuje dynamický pohled na proces, který se skládá z cyklů, fází, iterací a milníků. Vertikální osa představuje statický pohled na proces, popis činností, artefaktů, pracovníků a pracovních toků. (1)



Obrázek 1 - RUP fáze (zdroj: http://www.ibm.com/developerworks/library/ws-soa-term2/index.html)

Životní cyklus projektu v metodice RUP je rozdělen do čtyř základních fází. Každá z těchto fází je zakončena milníkem. Po dokončení každé fáze je provedeno zhodnocení, zda byly splněny veškeré požadavky, přičemž další fázi projektu je možné zahájit pouze v případě, že jsou splněny všechny požadavky. V počáteční fázi *Zahájení (Inception)* se vymezují především požadavky na vyvíjený systém, cíle projektu, vytváří se plán iterací a pro klíčové činnosti se vytvářejí modely případů užití. Na konci této fáze se rozhodne, zda je možné projekt realizovat. Cílem fáze *Příprava (Elaboration)* je vytvořit stabilní model architektury systému. Tento model vychází z počáteční fáze, přesněji z úvodních požadavků na systém. V této fázi se vytváří jeden, nebo více prototypů. Následuje fáze *Konstrukce (Construction)*, kde dojde k vytvoření a otestování první verze systému. Na rozdíl od předešlých dvou verzí, je zde hlavní náplní implementace a cílovým výstupem je první funkční verze. Poslední fází životního cyklu je *Předávání (Transition)*. Jak již vyplynulo z názvu této fáze, výsledkem by mělo být předání finální verze systému koncovému uživateli. V této fázi probíhá beta testování a drobné úpravy systému na základě připomínek uživatelů.

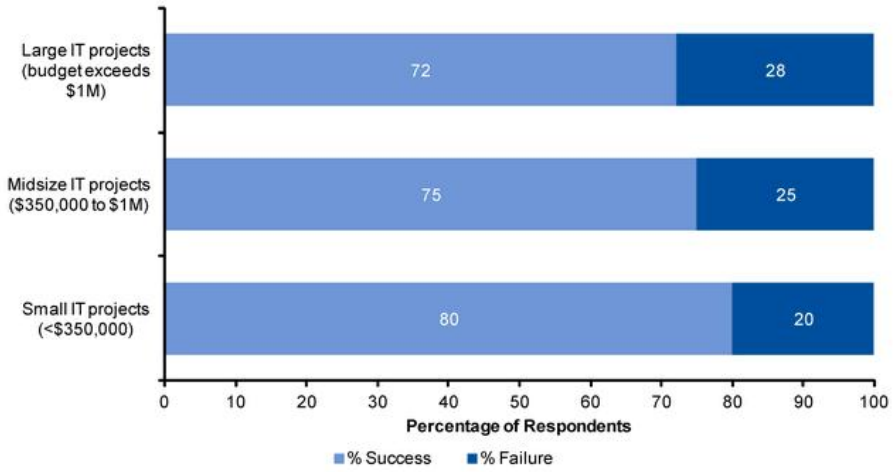
Metodika RUP obsahuje devět disciplín, které se dělí na hlavní a podpůrné. Hlavní disciplíny zajišťují práci na projektu a podpůrné disciplíny slouží k jejich řízení a koordinaci. Závěrem této obecné kapitoly o metodice RUP je nutné zmínit čtyři základní elementy metodiky RUP. Jedná se o role, aktivity, artefakty a pracovní proces. Role definují odpovědnost a práva osob do nich přidělených, role také vykonávají aktivity pomocí, kterých vytvářejí potřebné artefakty. Role tedy popisuje chování v průběhu procesu vývoje.

3 Situace na poli IT projektů

Dnešní IT projekty jsou stále složitější, což je spojeno s dlouhými vývojovými cykly, týmy s velkými počty členů a problémy s udržením konzistentní architektury napříč celým SW projektem. Na druhé straně je tu snaha minimalizovat dobu od počáteční koncepce produktu až po jeho uvedení na trh tzv. "Time-to-market" a s tím spojené, dnes velice populární, agilní přístupy. Realizují se menší projekty, které lépe reagují na měnící se podmínky. Komplexní celky se rozkládají na menší a jednodušší části, které lze rychleji vyvinout a lépe se řídí. Pro takové projekty je pak důležitá zvolená metodika vývoje SW. Klasická metodika RUP, založená na "best practicies", která komplexně pokrývá vývoj softwarového produktu, s sebou přináší i vysokou míru formality. To se projevuje ve velkém počtu artefaktů (80), rolí (40) a činností (150), které metodika definuje. Z tohoto důvodu se může stát obtížně použitelnou pro projekty menšího rozsahu. RUP tedy přichází i ve verzi pro malé projekty, ve které metodiku odlehčuje, avšak zachovává všechny její základní principy, viz předchozí kapitola.

3.1 Úspěšnost IT projektů

O tom, že velikost projektu má vliv na jeho úspěšnost svědčí následující Obrázek 2. Jedná se o graf zveřejněný ve studii společnosti Gartner, ve kterém je velikost projektu definována podle jeho rozpočtu. Ten je určen pro malé projekty do 350 000 USD, pro střední projekty od 350 000 do 1 mil. USD a pro velké projekty přesahuje 1 mil. USD. Je zde patrné, že malé projekty jsou procentuelně úspěšnější než ty velké, a to o 8 %. Průzkum byl realizován v roce 2011 a jeho respondenty se stalo 154 organizací ze Severní Ameriky, Francie a Německa. Hlavními uváděnými důvody, které vedly u firem k neúspěchu projektu, bylo překročení času a rozpočtu.



Obrázek 2 - Úspěšnost projektů (zdroj: http://thisiswhatgoodlookslike.com/2012/06/10/gartner-survey-shows-why-projects-fail/)

3.2 Členění IT projektů

Existuje mnoho kritérií, podle kterých lze členit IT projekty. Každé členění se nepatrně liší, ale ve výsledku poskytuje představu, do jaké skupiny daný projekt spadá. Pro naší semestrální práci, jsme si vybrali data, dostupná z oficiálních stránek IBM. Tento zdroj dělí IT projekty na malé, střední a velké, přičemž zohledňuje tři kritéria. Zaprvé to jsou peníze, tedy rozpočet, který je nutný vynaložit na daný projekt. Za druhé to jsou lidé, tedy velikost a složení týmu, pracující na projektu. V poslední řadě je to čas, nebo li doba trvání daného projektu. Rozdělení IT projektů dle zmíněných kritérií přehledně popisuje Tabulka 1.

			
Velké	500 000 +	13 +	12 +
Střední	100 000 – 500 000	6 - 12	6 - 12
Malé	0 – 100 000	1 - 6	1 - 6

Tabulka 1 - Rozdělení IT projektů (zdroj: http://www.ibm.com/developerworks/rational/library/jul05/kohrell/index.html)

4 Vlastní porovnání

Tato kapitola se již zaměřuje na samotné porovnání metodiky RUP, a to z pohledu malých a velkých projektů. Postup, který jsme zvolili pro porovnávání obou přístupů, spočívá v zaměření se na jednotlivé skupiny rolí obsažené v klasické metodice RUP (Classic), které jsme porovnávali s verzí pro malé projekty (Small). Hlavním srovnávacím aspektem se tedy staly role, na kterých však demonstrujeme i rozdílnost z pohledu artefaktů a činností, které jsou s rolemi bezprostředně spjaté. Naším cílem je také znázornit sdružování rolí pro účely malých projektů, redukování aktivit a s tím související úbytek artefaktů. Neklademe si za cíl podrobně popisovat každý rozdíl detailně, ale spíš nastínit způsob, kterým lze komplexní metodiku RUP využít i pro menší projekty. Výpis konkrétních rozdílů na úrovni rolí, aktivit a artefaktů se stal v tabulkové formě obsahem přílohy této práce. Vzhledem k nedostupnosti oficiálních dokumentů (white paper) k této problematice, bude využito kromě oficiálního zdroje *kitscm.vse.cz* také doplňujících dokumentů, ve kterých byla metodika RUP využita pro řízení malých projektů, avšak nejednalo se přímo o oficiálně vydanou verzi. Tato kapitola byla zpracována převážně ze zdroje *kitscm.vse.cz.*

4.1 Analyst

První skupinou rolí je analytická skupina (Analyst), jejichž hlavní činností je analyzovat a zaznamenávat požadavky kladené na vyvíjenou aplikaci. Nezbytná je častá spolupráce této skupiny se softwarovým architektem a projektovým manažerem. V případě malých projektů zde ubývají tři role. Podnikový architekt, podnikový návrhář a analytik podnikových procesů viz Tabulka 2.

Classic	Small
Business Architect	X
Business Designer	X
Business-Process Analyst	X
Requirements Specifier	Requirements Specifier
Stakeholder	Stakeholder
System Analyst	System Analyst

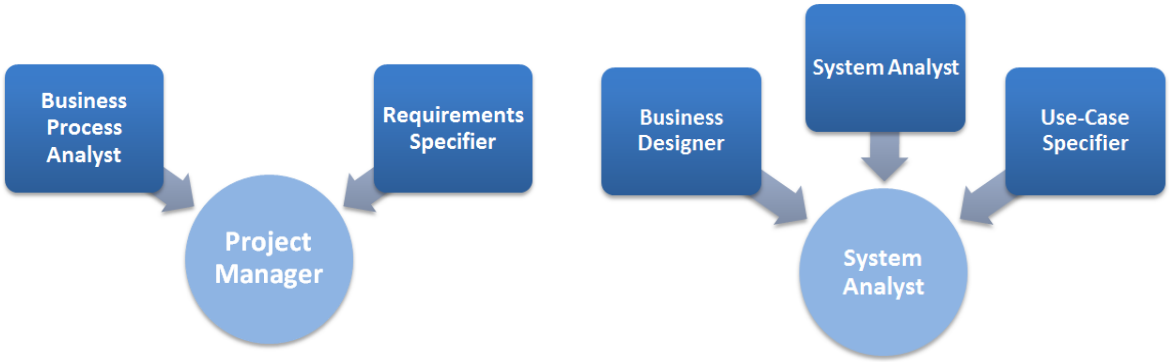
Tabulka 2 – Rozdíly Analyst

Sdružování rolí

Následuje ilustrativní ukázka, viz Obrázek 3, která znázorňuje sdružování rolí, které se může lišit v závislosti na potřebách pro daný projekt. Je zde vidět sdružení rolí *Business-Process Analyst* a *Requirements Specifier*. Obě role zde spadají pod pravomoc projektového manažera, který by tímto měl převzít odpovědnost za definování byznys architektury a získat větší kontrolu nad projektem.

Druhým příkladem je sdružení rolí *Business Designer* a *Use-Case Specifier*. Přestože je *Business Designer* zodpovědný za byznys řešení mapuje se tato role systémovému analytikovi, který je zodpovědný za požadavky projektu. *Use-Case Specifier* úzce spolupracuje se systémovým analytikem a v malých projektech se tyto role proto mohou sloučit.

Obrázek 3 - Sdružování rolí



4.2 Developers

Hlavní náplní skupiny Vývojářů je samotná implementace softwarového produktu. Lze tady zaznamenat absenci *Capsule Designera* u malých projektů, který řeší problémy souběžnosti, kdy systém musí být schopen na události správně reagovat v časové souběžnosti. Další redukovanou rolí je *Security Architect*, který volí bezpečnostní návrhové vzory a další.

Classic	Small
Capsule Designer	X
Database Designer	Database Designer
Designer	Designer
Implementer	Implementer
Integrator	Integrator
Security Architect	X
Software Architect	Software Architect
User-Interface Designer	User-Interface Designer

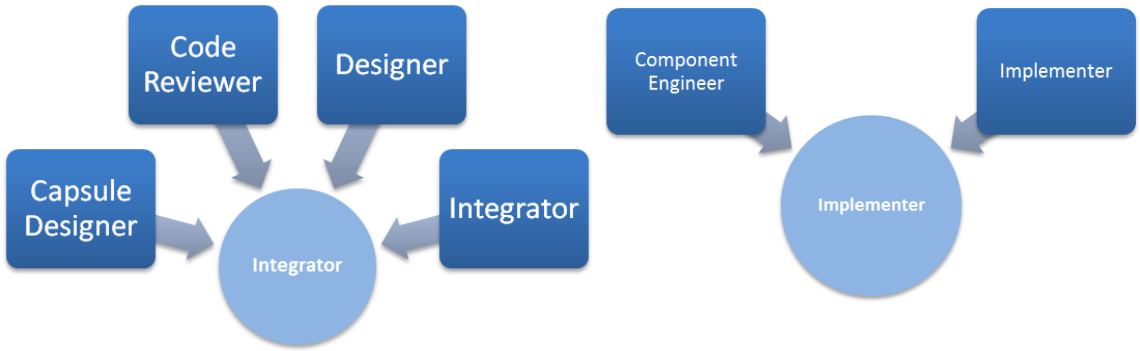
Tabulka 3 - Rozdíly Developers

Sdružování rolí

Zde můžeme vidět příklad toho, jakým způsobem se dají role mapovat a tím redukovat pro účely malých projektů. Například *Designer* má podobný profil, jako *Capsule designer*, který má však užší zaměření. Jejich hlavní náplní je převést návrh softwarového architekta do modulů a komponent. Za správné fungování těchto komponent odpovídá integrátor. Proto je možné mu tyto role namapovat. Integrátor by měl být také schopen kontrolovat, zda jsou dané komponenty správně implementovány (*Code Reviewer*).

Roli *Component Engeneera*, který je zodpovědný za vnitřní strukturu jednotlivých komponent, jejich třídy, atributy atd. je možné vzhledem k charakteru role namapovat přímo na implementátora.

Obrázek 4 – Sdružování rolí



4.3 Testers

Další skupinou jsou role, mající na starost testování, realizaci testovací strategie a návrhů testů ověřujících kvalitu vyvíjené aplikace. Jedná se o jedinou skupinu rolí, která zůstává na první pohled stejná pro velké projekty i pro malé projekty. Liší se především ve vytvářených artefaktech. Pro potřeby malých projektů ubývají některým rolím artefakty, jak je dále naznačeno.

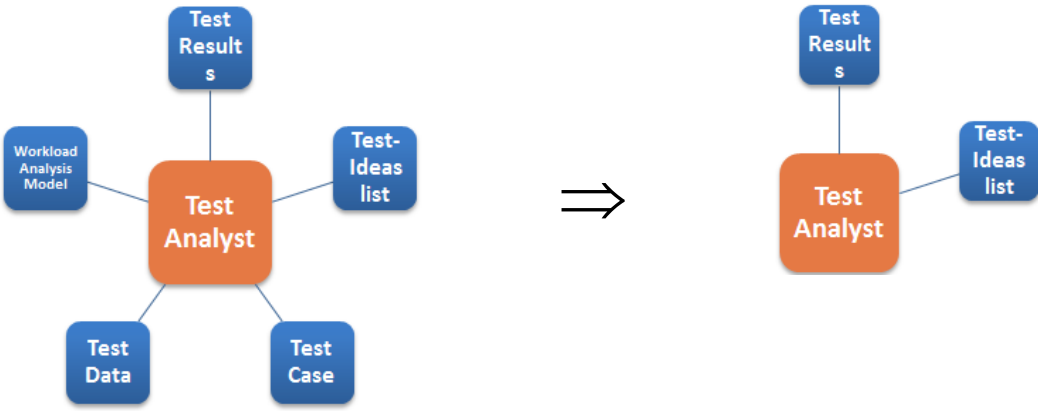
Classic	Small
Test Analyst	Test Analyst
Test Designer	Test Designer
Test Manager	Test Manager
Tester	Tester

Tabulka 4 - Rozdíly Testers

Redukce artefaktů

Pro tento příklad jsme zvolili roli *Test Analyst*, který má za úkol určit, jaká část aplikace se bude testovat a zároveň stanovuje hodnotící kritéria. Role *Test Analyst* vytváří celkem pět artefaktů v klasickém rozsahu metodiky RUP. *Test Case*, který definuje sadu testovacích vstupů a očekávaných výsledků. *Test data* a *Test Results*, *Test-Ideas list*, což je výčet testů, o kterých se uvažuje a *Workload Analysis Model*. V tomto případě se pro malé projekty sloučí tři artefakty s ostatními a ve výsledku zbývají roli *Test Analyst* pouze dva artefakty. *Test Case* může být částečně, či úplně přiložen k *Test-Ideas list*. *Test data* mohou být v některých případech sloučená s artefaktem *Test Script*, jehož zodpovědnou rolí je *Tester*. Poslední artefakt, který ubývá je *Workload Analysis Model*, který může být zapouzdřený v testovacím plánu, jehož zodpovědnou rolí je *Test Manager*. Z toho vyplývá, že některé artefakty, které se redukují u jedné z rolí, se mohou sloučit s artefakty, které mají v kompetenci jiné role. Redukce artefaktů je znázorněné na obrázku.

Obrázek 5 – Redukce artefaktů



4.4 Production & Support

Tato skupina rolí se přímo nepodílí na řízení, vývoji nebo testování softwaru, avšak zastává důležitou podpůrnou funkci vývojového procesu, a to v podobě vytváření dodatečných materiálů, důležitých pro finální produkt. Uvedená skupina rolí je redukována vcelku významně viz Tabulka 5. Jako základní a nepostradatelné role zde zůstávají *Process Engineer*, který má podpůrnou funkci, kdy zajišťuje, aby jednotlivým rolím bylo umožněno vykonávat svou práci a plnit cíle. Druhým nepostradatelným je pak *System Administrator*, který zajišťuje funkcionalitu hardwarové a softwarové infrastruktury.

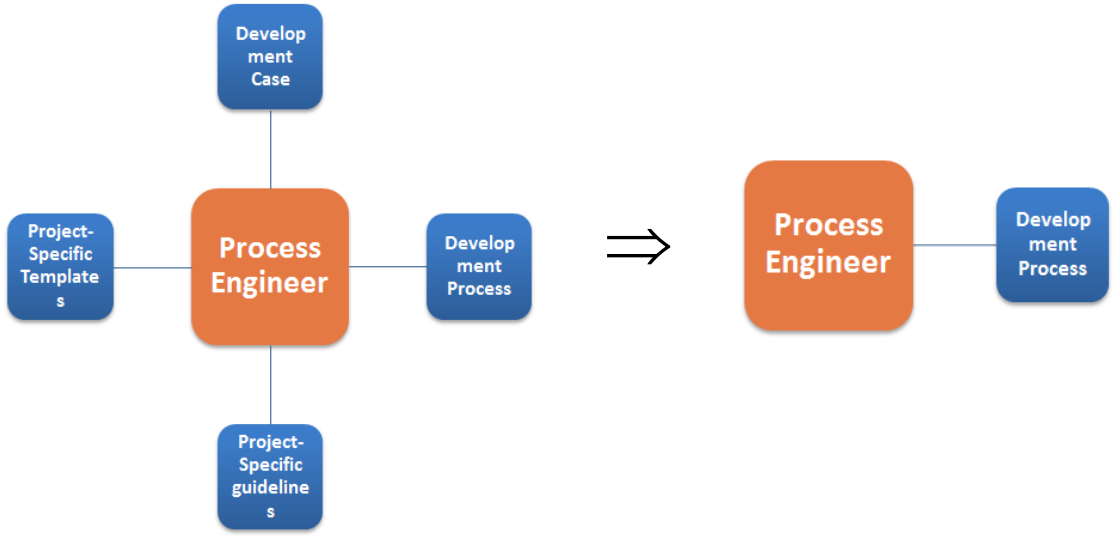
Classic	Small
Course Developer	X
Graphic Artist	X
Process Engineer	Process Engineer
System Administrator	System Administrator
Technical Writer	X
Tool Specialist	X

Tabulka 5 - Rozdíly Production & Support

Redukce artefaktů

Process Engineer v zásadě vytváří čtyři artefakty. Nejdůležitějším z nich je však *Development Process*, který představuje upravenou metodiku procesů projektu. Slouží pro podporu účastníků projektu a vytváří průvodce. Artefakt *Development Process* je velmi důležitý pro sdílení znalostí o jednotlivých procesech a umožňuje jednodušší porozumění mezi více členy projektu. Proto se také tento artefakt, jako jediný zachovává i u verze pro malé projekty.

Obrázek 6 – Redukce artefaktů



4.5 Managers

Manažeři jsou poslední skupinou rolí v metodice RUP. Jediná role, která je zde redukována, je *Deployment Manager*, který spadá v malých projektech pod projektového manažera. Důvodem pro sloučení je opět úzká spolupráce těchto dvou rolí. Na této skupině rolí ovšem chceme popsat další element metodiky RUP a to jsou aktivity, činnosti, či úkoly, které vykonávají osoby obsazené do definované role.

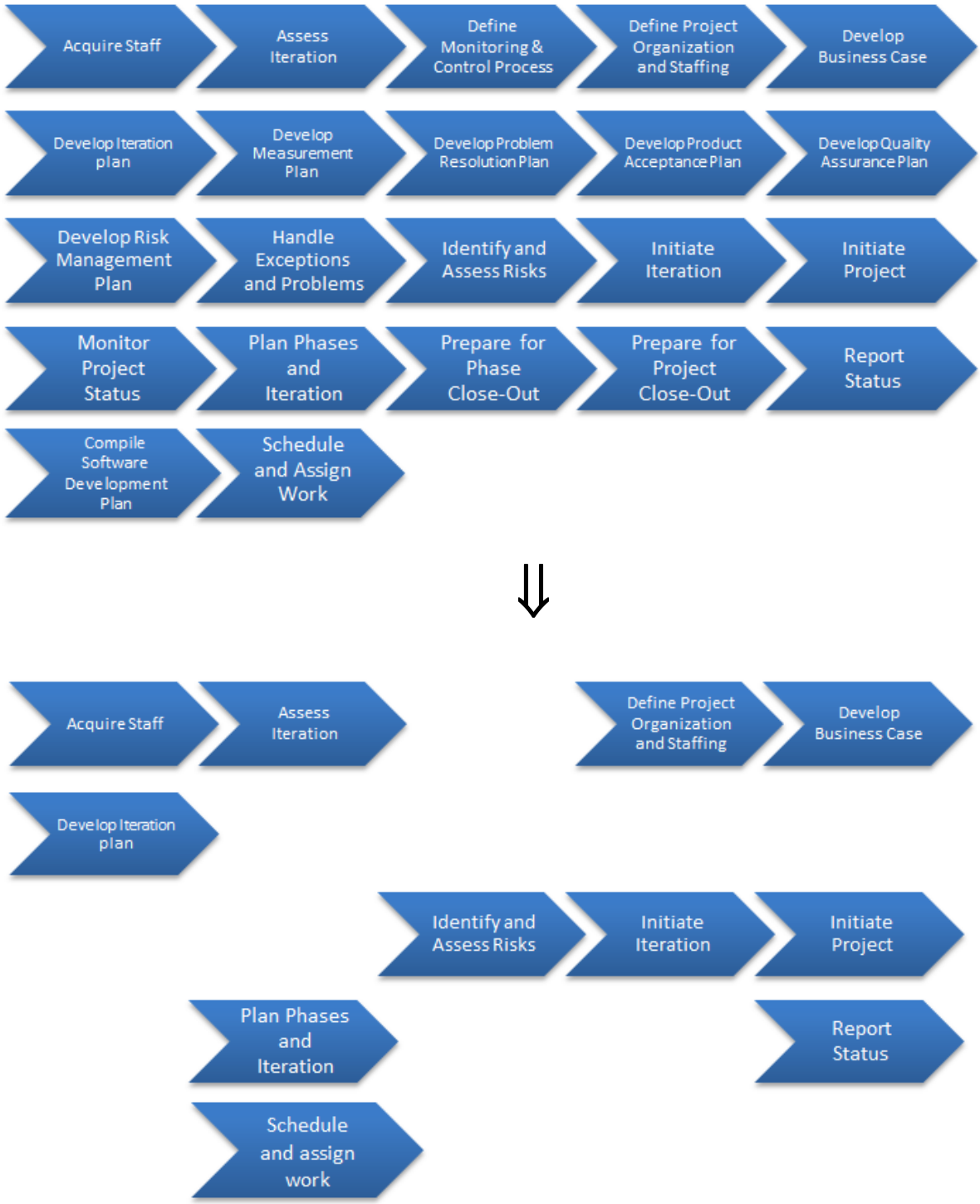
Classic	Small
Change Control Manager	Change Control Manager
Configuration Manager	Configuration Manager
Deployment Manager	X
Management Reviewer	Management Reviewer
Project Manager	Project Manager
System Administrator	System Administrator
Test Manager	Test Manager

Tabulka 6 - Rozdíly Managers

Redukce aktivit

Aktivitu lze považovat za jednotku práce, která může být přidělena dané osobě. Pro ukázku úbytku aktivit, jsme vybrali projektového manažera, který je zodpovědný za plánování, řízení, organizování a koordinaci celého projektu. Nepůsobí pouze uvnitř projektu, ale působí jako zástupce a často bývá v kontaktu se zadavatelem. Projektový manažer má 22 činností viz Obrázek 7, které se pro řízení malého projektu zredukují o polovinu. Mezi důležité aktivity, které mu zůstávají, patří: přiřazení lidí do projektu a jejich organizování v týmech, určení obchodních a finančních aspektů nového produktu, sestavování plánu iterací, identifikace rizik projektu. Další důležitou aktivitou je plánování fází projektu a iterací, jejich cíle a doba trvání.

Obrázek 7 – Redukce aktivit



5 Použití RUP pro malé projekty

V této části bychom rádi demonstrovali využití metodiky RUP na malém projektu v podobě případové studie (7), která je dostupná v podobě *white paper*. V době tvorby této semestrální práce byl tento dokument shledán, jako jediný oficiálně dostupný (od Rational

Software), který se metodikou RUP u malých projektů zabývá. Případová studie kromě použití přístupu RUP pro malé projekty dále také využívá agilní přístup, konkrétně extrémní programování (XP).

eXtreme programming je metodika vývoje softwaru, obsahující tradiční činnosti vývoje, avšak dovedené do extrému. Metodika je založena na čtyřech základních hodnotách, kterými jsou: komunikace, jednoduchost, zpětná vazba a odvaha. Je realizována kontinuální komunikace mezi vývojovým týmem a zákazníkem během vývoje softwaru, kdy zákazník určuje, co bude vyvíjeno a v jakém pořadí. Provádí se kontinuální *refactoring* kódu a vzniká pouze minimum artefaktů, které netvoří zdrojový kód SW. Jedná se o metodiku řadící se do agilních přístupů. (7)

Případová studie popisuje, jakým způsobem využít RUP na malém projektu, v podobě vymezení nejdůležitějších činností a artefaktů, které je třeba realizovat v jednotlivých fázích vývoje, podle metodiky RUP společně s použitím extrémního programování.

Začátek projektu - Inception

Klíčovým artefaktem je zde "*Vision*", který obecně popisuje systém, kdo a proč ho bude používat, co musí systém umožňovat a jaká jsou zde omezení. Dalšími očekávanými artefakty jsou: *An approved Business Case, Risk List, Preliminary Project Plan, Project Acceptance Plan, A plan for the initial Elaboration iteration.* Dále je nutné realizovat čtyři klíčové aktivity. 1.*Formulate the scope of the project* - zde je nutné zjistit klíčové požadavky, které uspokojí zainteresované strany. 2. *Plan and Prepare the business case* - Za využití artefaktu *Vision*, definujeme naší strategii pro minimalizaci rizika, náklady, projektový plán a harmonogram. 3.*Synthesize a candidate architecture* - Zde se zvažuje výběr použité architektury na základě nových přístupů a klade se otázka co vyvinout a co zakoupit. 4.*Prepare the project environment* - Příprava projektového prostředí v podobě zajištění fyzických zdrojů, software a nástrojů, které bude tým využívat. Tato fáze (Inception) může být velmi krátka v hrubém rozsahu několika dní a méně.

Elaboration

Základními aktivitami jsou: 1.*Define, validate, and baseline the architecture* - Za použití *Risk List* se stanovuje architektura z kandidátů, kteří byli vybráni v úvodní fázi. 2.*Refine the Vision* - Vize, která byla sestavená v úvodní fázi,s se zpřesňuje, upravuje a schvaluje zainteresovanými stranami. 3.*Create and baseline iteration plans for the Construction phase* - Vytváří se iterační plán pro fázi kontrukce, který se v každé jeho iteraci bude upravovat na základě možné změny prostředí. Základními artefakty jsou *Software architecture document,* který by měl na konci této fáze obsahovat detailní popis architektonicky významných případů

užití, identifikovat klíčové mechanismy a klíčové prvky. Dále pak *Iteration Plans for Construction*, ve kterém je naplánován počet konstrukčních iterací během elaborace, společně se specifickými případy užití, scénáři a pracovními položkami. U velmi malých projektů je možné fáze Inception, Elaboration a Construction sloučit, při zachování uvedených postupů a redukcí zdorjů pro plánování a recenzování.

Construction

Základními aktivitami ve fázi konstrukce jsou: 1.*Manage resources and control process* - Zde se v podstatě určuje, kdo bude na čem pracovat, tedy přiřazení kapacity zdrojů na základě harmonogramu. 2.*Develop and test components* - Provádění jednotkových a integračních testů na implementovaných komponentách. 3.*Assess the iteration* - Na konci iterace je třeba zhodnotit napnění cílů a případně upravit priority prací, aby se stihl čas dodání. Vzhledem k aplikaci metodiky XP jsou zde realizovány související praktiky jako: *Testing, Refactoring, Pair Programming, Continuous integration, Collective ownership, Simple design, Coding standards.* Co se týče artefaktů, tak jsou zde realizovány: *Components, Training materials, Deployment Plan, Transition Phase Iteration Plan.*

Transition

Základními aktivitami, realizovanými v této fázi jsou: 1.*Finalize end-user support material* - Kde se dokončují materiály pro podporu zákazníka. 2. *Test the product deliverable in a customer environment* - Produkt se testuje v prostředí, ve kterém ho zákazník bude používat. 3.*Fine-tune the product based upon customer feedback* - Pokud probíhá jedna nebo více period beta testování, je nutné uvážit a případně zapracovat drobné připomínky zákazníka. 4. *Deliver the final product to the end user* - V závislosti na typu softwaru jsou zde činnosti spojené s finální produkcí, přenos na instalační média a balení nebo poskytnutí na úložišti a další. Souvisejícími artefakty jsou: *Deployment Plan, Release Notes, Training Materials and Documentation.*

Shrnutí

Autor na závěr shrnuje, že vývoj SW je více než psaní kódu. Proces vývoje se musí zaměřit na všechny aktivity přinášející zákazníkovi kvalitu a není nutné, aby byl těžkopádný. Na závěr nabádá k využívání takového rozsahu procesů a aktivity, které jsou potřebné pro konkrétní projekt, organizaci, tým a jejich specifika.

6 Závěr

Obsah práce se zabýval hledáním rozdílů v přístupu klasické metodiky RUP a její alternativy pro malé projekty. Ty se evidentně stávají významnými na poli IT projektů, jelikož díky snazšímu řízení a schopnosti reagovat na změny dosahují větší úspěšnosti než projekty velké.

Vzhledem k rozsahu práce byla zvolena vhodná úroveň detailu a její obsah se nezabýval podrobně každým rozdílem detailně. Byla zvolena základní kostra v podobě rolí, které se jevily jako zásadní pro samotné porovnávání. Tyto role byly porovnávány v rámci obou přístupů metodiky (malé x velké projekty) a také bylo nastíněno jejich sdružování, redukce činností a artefaktů s tím spojených. Dále byl zpracován kompletní výčet rozdílů rolí, činností a artefaktů v tabulkové podobě.

Přínosy této práce jsou spatřovány spíše než v předložení podrobné dokumentace rozdílů, v nastínění možností, jakým způsobem lze metodiku, kterou je klasický RUP, využít i pro malé projekty, aniž by došlo k omezení v podobě nedodržení základních principů této metodiky.

Na závěr bychom rádi shrnuli poznatky, kterých jsme při zpracovávání této práce nabyli. Je nutné si uvědomit, že každý projekt je jiný a má svá specifika, a proto je třeba metodiku vývoje softwaru vybírat a následně upravit pro jeho konkrétní potřeby. To znamená přizpůsobit ji konkrétnímu týmu, organizaci, charakteru projektu, firemní kultuře a dalším aspektům, které určují jedinečnost daného projektu. Důležité však je si z metodiky vzít a správně použít tzv. "*best practicies*", ověřené mnoha realizovanými projekty. Na základě výše uvedených aspektů následně zvolit vhodnou úroveň formálnosti. Je důležité se zaměřit na procesy, které přináší kvalitu zákazníkovi, realizovat činnosti a vytvářet ty artefakty, které snižují rizikovost projektu.

Zdroje

(1) BUCHALCEVOVÁ, Alena. *Metodiky budování informačních systémů*. Vyd. 1. Praha: Oeconomica, 2009, 205 s. Vysokoškolská učebnice (Oeconomica). ISBN 978-80-245-1540-3.

(2) BUCHALCEVOVÁ, Alena, Jarmila PAVLÍČKOVÁ a Luboš PAVLÍČEK. *Základy softwarového inženýrství: materiály ke cvičení*. Vyd. 1. Praha: Oeconomica, 2007, 221 s. ISBN 978-80-245-1270-9.

(3) *Kitscm.vse.cz* [online]. 2012 [cit. 2012-10-21]. Dostupné z: http://kitscm.vse.cz/

(4) Gartner Survey Shows Why Projects Fail. In: *Thisiswhatgoodlookslike.com* [online]. 10. 6. 2012 [cit. 2012-12-10]. Dostupné z: http://thisiswhatgoodlookslike.com/2012/06/10/gartner-survey-shows-why-projects-fail/

(5) Using RUP to manage small projects and teams. In: *IBM.com* [online]. 15. 6. 2012 [cit. 2012-12-10]. Dostupné z: http://www.ibm.com/developerworks/rational/library/jul05/kohrell/index.html

(6) Mapping RUP Roles to Small Software Development Teams. In: *Www3.dsi.uminho.pt* [online]. 2012 [cit. 2012-12-10]. Dostupné z: http://www3.dsi.uminho.pt/rmac/privatefiles/papers/2012_SWQD_BorgesMonteiroMachado-springer.pdf

(7) Using the Rational Unified Process for Small Projects: Expanding Upon eXtreme Programming. In: *Www.cs.buap.mx* [online]. 2001 [cit. 2012-12-10]. Dostupné z: http://www.cs.buap.mx/~dpinto/semadoo/tp183.pdf

Název role	Popis	Zahrnuta ve Small Projects	Large Projects (činnosti)	Small Projects (činnosti)	Large Projects (artefakty)	Small Projects (artefakty)
Business Architect	This role is responsible for the overall Business Architecture, being involved in all the significant decisions regarding structure, main behavior and its realization, interfaces, constraints and trade-offs.	NE	Business Architectural Analysis Construct Business Architectural Proof-of-Concept Functional Area Analysis Prioritize Business Use Cases	X	Business Analysis Model Business Architectural Proof-of-Concept Business Architecture Document Business Deployment Model Business Design Model Business System	X
Business Designer	This role details the specification of a part of the organization.	NE	Business Operation Analysis Business Operation Design Business Use-Case Analysis Define Automation Requirements Detail a Business Entity Detail a Business Use Case Detail a Business Worker	X	Business Actor Business Worker Business Entity Business Event Business Operation Business Operation Realization Business Use Case Business Use-Case Realization	X
Business-Process Analyst	This role leads and coordinates business requirements elicitation by outlining and delimiting the organization being modeled.	NE	Assess Target Organization Set and Adjust Objectives Capture a Common Business Vocabulary Structure the Business Use-Case Model Define Business System Context Find Business Actors and Use Cases Identify Business Goals and KPIs Maintain Business Rules Refine a Business Use-Case	X	Business Glossary Business Goal Business Rule Business Use Case Model Business Vision Supplementary Business Specification Target-Organization Assessment	X
Requirements Specifier	This role specifies and maintains the detailed system requirements.	ANO	Detail a Use Case Detail the Software Requirements	Detail a Use Case Detail the Software Requirements	Actor Software Requirement Software Requirements Specification Use Case	Actor Software Requirement Software Requirements Specification Use Case
Stakeholder	This role represents an interest group whose needs must be satisfied by the project. It may be played by anyone who is materially affected by the outcome of the project.	ANO	X	X	X	X
System Analyst	This role leads and coordinates requirements elicitation by outlining the system's functionality and delimiting the system.	ANO	Capture a Common Vocabulary Manage Dependencies Define System Context Structure the Use-Case Model Develop Requirements Management Plan Develop Supplementary Specifications Develop Vision Elicit Stakeholder Requests Find Actors and Use Cases	Capture a Common Vocabulary Manage Dependencies Develop Supplementary Specifications Develop Vision Elicit Stakeholder Requests Find Actors and Use Cases Structure the Use-Case Model	Glossary Use-Case Package Requirements Attributes Vision Requirements Management Plan Stakeholdet Requests Storyboard Supplementary Specifications Use-Case Model	Glossary, Use-Case Package Storyboard Supplementary Specifications Use-Case Model Use-Case Package Vision
Capsule Designer	This role designs capsules, ensuring that the system can respond to events in a timely manner, in accordance with concurrency requirements.	NE	Capsule Design	X	Analysis Class Capsule Design Class Design Package Design Subsystem Message Operation Operation Realization Service Service Channel Service Component Service Contract Service Gateway Service Model Service Partion Service Provider Service Specification Testability Class Use-Case Realization	X
Database Designer	This role leads the design of the persistent data storage structure to be used by the system.	ANO	Database Design Specify Data Migration	Database Design	Data Migration Specification Data Model	Data Model
Designer	This role leads the design of a part of the system, within the constraints of the requirements, architecture, and development process for the project.	ANO	Class design Component Specification (SOA) Design Testability Elements Document Service Realization Decisitions Identify and Associate Services to Goals Identify Commonality and Variability Message Design Operation Analysis Operation Design Service Specification Subsystem Design Subsystem Design (SOA) Use-Case Analysis Use-Case Design	Class Design Subsystem Design Use-Case Analysis Use-Case Design	Analysis Class Design Class Design Package Design Subsystem Message Operation Operation Realization Service Service Channel Service Component Service Contract Service Gateway Service Model Service Partition Service Provider Service Specification Testability Class Use-Case Realization	Analysis Class Design Class Design Package Design Subsystem Use-Case Realization
Implementer	This role develops software components and performs developer testing for integration into larger subsystems, in accordance with the project's adopted standards.	ANO	Analyze Runtime Behavior Develop Installation Work Products Execute Developer Tests Implement Design Elements Implement Developer Test Implement Testability Elements	Analyze Runtime Behavior Execute Developer Tests Implement Design Elements Implement Developer Test	Developer Test Implementation Element Implementation Subsystem Installation Artifacts Test Stub Testability Element	Developer Test Implementation Element Implementation Subsystem
Integrator	This role leads the planning and execution of implementation element integration to produce builds.	ANO	Create Baselines Create Integration Workspaces Integrate Subsystem Integrate System Plan Subsystem Integration Plan System Integration Promote Baselines	Integrate Subsystem Integrate System Plan Subsystem Integration Plan System Integration	Build Integration Build Plan	Build Integration Build Plan
Security Architect	This role leads the development of the system's security architecture, which includes the selection of security patterns for system design, the documentation of technical policy concerning security implementation and supporting the key technical decisions that constrain the overall security implementation for the project.	NE	Identify Security Patterns	X	X	X
Software Architect	This role leads the development of the system's software architecture, which includes promoting and creating support for the key technical decisions that constrain the overall design and implementation for the project.	ANO	Apply Services Litmus Tests Architectural Analysis Assesss Viability of Architectural Proof-of-Concept Business Process Analysis Business Rule Analysis Business Use Case Analysis (SOA) Construct Architectural Proof-of-Concept Construct Architectural Proof-of-Concept (SOA) Data Model Analysis Describe Distribution Describe the Run-time Architecture Existing Asset Analysis Identify Design Elements Identify Design Mechanisms Incorporate Existing Design Elements Prioritize Use Cases Structure the Implementation Model	Architectural Analysis Assesss Viability of Architectural Proof-of-Concept Construct Architectural Proof-of-Concept Describe Distribution Describe the Run-time Architecture Identify Design Elements Identify Design Mechanisms Incorporate Existing Design Elements Prioritize Use Cases Structure the Implementation Model	Analysis Model Architectural Proof-of-Concept Deployment Model Design Model Event Goal-Service Model Implementation Model Interface Protocol Reference Architecture Signal Software Architecture Document	Analysis Model Architectural Proof-of-Concept Deployment Model Design Model Event Implementation Model Interface Reference Architecture Signal Software Architecture Document

User-Interface Designer	This role coordinates the design of the user interface. This includes gathering usability requirements and prototyping candidate user-interface designs to meet those requirements.	ANO	Design the User Interface Prototype the User-Interface	Design the User Interface Prototype the User-Interface	Navigation Map User-Interface Prototype	Navigation Map User-Interface Prototype
Test Analyst	This role identifies and defines the required tests, monitors detailed testing progress and results in each test cycle and evaluates the overall quality. The role also represents stakeholders who do not have direct or regular representation on the project.	ANO	Define Assessment and Tracebility Needs Define Test Details Determine Test Results Identify Targets of Test Identify Test Ideas Verify Changes in Build	Determine Test Results Identify Test Ideas	Test Case Test Data Test Results Test-Ideas List Workload Analysis Model	Test Results Test-Ideas List
Test Designer	This role leads defining the test approach and ensuring its successful implementation. This includes identifying the appropriate techniques, tools and guidelines to implement the required tests, and to provide guidance to the test effort on corresponding resources requirements.	ANO	Define Test Approach Define Test Environment Configurations Define Testability Elements Define Testability Mechanisms Structure the Test Implementation	Define Test Approach Structure the Test Implementation	Test Automation Architecture Test Design Test Environment Configuration Test Interface Specification Test Strategy Test Suite	Test Strategy Test Suite
Test Manager	This role leads the overall test effort. This includes quality and test advocacy, resource planning and management, and resolution of issues that impede the test effort	ANO	Agree on the Mission Assess and Advocate Quality Assess and Improve Test Effort Identify Test Motivators Obtain Testability Commitment	Assess and Advocate Quality	Test Evaluation Summary Test Plan	Test Evaluation Summary
Tester	This role conducts tests and logs the outcomes of his/her testing.	ANO	Analyze Test Failure Execute Test Suite Implement Test Implement Test Suite	Execute Test Suite Implement Test Suite	Test Log Test Script	Test Log
Change Control Manager	This role defines and oversees the change control process.	ANO	Confirm Duplicate or Rejected CR Establish Change Control Process Review Change Request	Confirm Duplicate or Rejected CR Review Change Request	Change Request	Change Request
Configuration Manager	This role manages the overall Configuration Management (CM) infrastructure and environment for the product development team.	ANO	Create Deployment Unit Establish Configuration Managament (CM) Perform Configuration Audit Report on Configuration Status Set up Configuration management (CM) Write Configuration Managament (CM) Plan	Set up Configuration management (CM)	Configuration Audit Findings Configuration Managament Plan Deployment Unit Project Repository	Project Repository
Deployment Manager	This role leads planning the product's transition to the user community, ensuring those plans are enacted appropriately, managing issues and monitoring progress.	NE	Define Bill Of Materials Develop Deployment Plan Manage Acceptance Test Manage Beta Test Provide Access to Download Site Release to Manufacturing Verify Manufactured Product Write Release Notes	X	Bill of Materials Deployment Plan Product Release Notes	X
Management Reviewer	This role evaluates project planning and project assessment work products at major review points in the project's lifecycle.	ANO	Iteration Acceptance Review Iteration Evaluation Criteria Review Iteration Plan Review Lifecycle Milestone Review Project Acceptance Review Project Approval Review Project Planning Review Project Review Authority (PRA) Project Review	Iteration Evaluation Criteria Review Iteration Plan Review Project Approval Review Project Planning Review	X	X
Project Manager	This role plans, manages and allocates resources, shapes priorities, coordinates interactions with customers and users, and keeps the project team focused. The Project Manager also establishes a set of practices that ensure the integrity and quality of project work products.	ANO	Acquire Staff Assess Iteration Compile Software Development Plan Define Monitoring & Control Process Define Project Organization and Staffing Develop Bussiness Case Develop Iteration Plan Develop Measurement Plan Develop Problem Resolution Plan Develop Product Acceptance Plan Develop Quality Assurance Plan Develop Risk Managament Plan Handle Exceptions and Problems Identify and Assess Risks Initiate Iteration Initiate Project Monitor Project Status Plan Phases and Iterations Prepare for Phase Close-Out Prepare for Project Close-Out Report Status Schedule and Assign Work	Acquire Staff Assess Iteration Define Project Organization and Staffing Develop Bussiness Case Develop Iteration Plan Identify and Assess Risks Initiate Iteration Monitor Project Status Plan Phases and Iterations Report Status Schedule and Assign Work	Business Case Issues List Iteration Assessment Iteration Plan Measurement Plan Problem Resolution Plan Product Acceptance Plan Project Measurements Quality Assurance Plan Risk List Risk Managament Plan Software Development Plan Status Assessment Work Order	Business Case Iteration Assessment Iteration Plan Risk List Software Development Plan Status Assessment Work Order
System Administrator	This role maintains the development infrastructure, both hardware and software. This includes installation, configuration, backup, etc.	ANO	Support Development	Support Development	Development Infrastructure	Development Infrastructure
Test Manager	This role leads the overall test effort. This includes quality and test advocacy, resource planning and management, and resolution of issues that impede the test effort.	ANO	Agree on the Mission Assess and Advocate Quality Assess and Improve Test Effort Identify Test Motivators Obtain Testability Commitment	Assess and Advocate Quality	Test Evaluation Summanry Test Plan	Test Evaluation Summanry
Course Developer	This role develops training materials for training users on how to use the product.	NE	Develop Training Materials	X	Training Materials	X
Graphic Artist	This role creates product artwork that is included as part of the product packaging.	NE	Create Product Artwork	X	Product Artwork	X
Process Engineer	This role is a support role in RUP. Its main goal is to equip the project team with an appropriate development process, and to make sure that the team members are not hindered in doing their jobs.	ANO	Develop Development Case Launch Development Process Prepare Guidelines for the Project Prepare Templates for the Project Tailor the Development Process	Launch Development Process Tailor the Development Process	Development Case Development Process Project-Specific Guidelines Project-Specific Templates	Development Process
System Administrator	This role maintains the development infrastructure, both hardware and software. This includes installation, configuration, backup, etc.	ANO	Support Development	Support Development	Development Infrastructure	Development Infrastructure
Technical Writer	This role produces end-user support material such as user guides, help texts, release notes, and so on.	NE	Develop Manual Styleguide Develop Support Materials	X	Manual Styleguide User Support Material	X
Tool Specialist	This role supports the tools used by the project. This includes selecting and acquiring tools, configuring and setting them up, and verifying that they work.	NE	Select and Acquire Tools Set up Tools Verify Tool Configuration and Installation	X	Tools	X