



Popis konceptu
Unified Method Architecture (UMA)
a jeho použití v RUPv.7

Jiří Kůfner (xkufj01), Ondřej Andr (xando03)

Zimní semestr 2012/2013

4IT421 - Zlepšování procesů budování IS

1 Obsah

2	Úvod.....	4
3	Vznik UMA.....	5
4	Obecný koncept	6
4.1	Základní obsahové elementy	6
4.1.1	Role.....	7
4.1.2	Work Product (Pracovní produkt)	8
4.1.3	Task (Úkoly)	8
4.2	Element na pomezí Obsahu metod a Procesů.....	9
4.2.1	Guidance („Řízení“)	9
4.3	Základní elementy procesů.....	10
4.3.1	Activities (Aktivity).....	10
4.3.2	Capability Pattern	11
4.3.3	Delivery Process („Doručovací proces“)	11
4.4	Descriptors (Deskriptory).....	12
4.5	Procesní diagramy.....	12
4.5.1	Workflow Diagram	13
4.5.2	Activity Detail Diagram	14
4.5.3	Work-Product Dependency Diagram	15
5	Klíčové vlastnosti UMA	16
5.1	Oddělení obsahu metod od procesů (Separation of Method Content and Process).....	16
5.2	Znovupoužitelný obsah (Content Reuse).....	16
5.3	Rodina procesů (Process families)	16
5.4	Podpora více životních cyklů (Multiple Lifecycles)	16
5.5	Flexibilní rozšiřitelnost a plug-in mechanismus (Flexible Extensibility and Plug-in Mechanisms).....	17

5.6	Víceprocesní pohled (Multiple Process 'Views').....	17
5.7	Znovupoužitelné procesní vzory (Reusable proces patterns).....	17
6	Porovnání RUP 2003 a RUP 7.0.....	18
6.1	Přehled terminologických rozdílů	18
6.2	Komentář k terminologickým rozdílům	18
6.3	Přehled nových pojmů v RUP 7.0.....	19
6.4	Další novinky v RUP 7.0.....	19
7	Závěr	20
8	Zdroje	21
9	Seznam obrázků	22

2 Úvod

Před samotným obsahem práce si dovolíme jen krátký úvod s popisem toho, co v této práci najdete.

Nejprve přijde řada na pár slov o vzniku samotného konceptu UMA následované jeho krátkým představením. Poté už naleznete slovní popis jednotlivých elementů, které tento koncept definuje. Nejprve z oblasti obsahu metod a poté také z procesní části. Přičemž u některých elementů budete mít možnost najít ilustrovaný příklad převzatý přímo z metodiky RUP.

Po popisu jednotlivých elementů už přijde řeč na několik klíčových vlastností tohoto konceptu.

Poslední obsahovou částí této práce je krátké porovnání RUPv7, který je postavený na konceptu UMA s předchozí verzí RUPv2003, který vycházel z tehdejšího metamodelu SPEM 1.0.

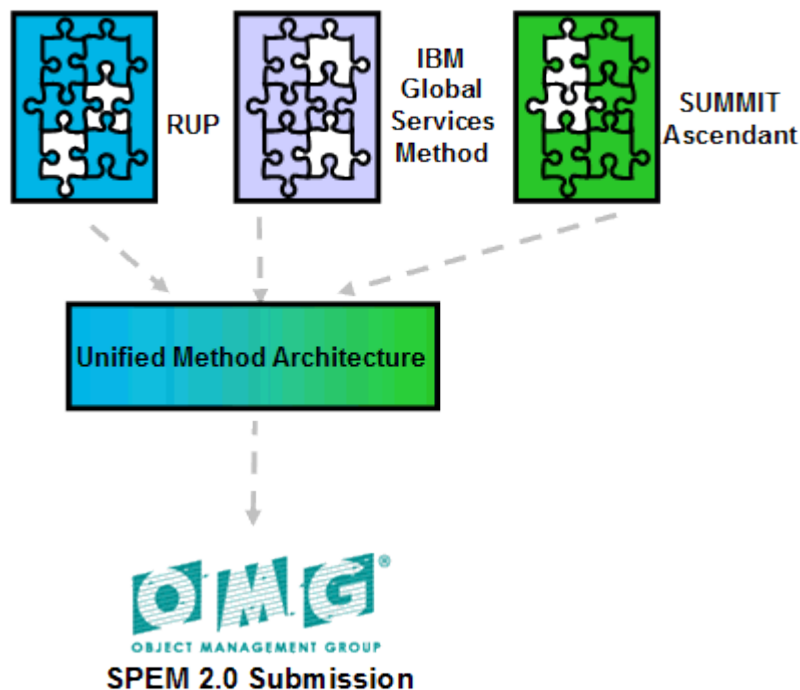
V závěru se dočtete shrnutí dříve napsaného a budou zopakovány věci, které byste si z této práce určitě měli odnést.

Na úplný konec práce jsou pak zařazeny dvě kapitoly, z nichž jedna obsahuje použité zdroje a druhá seznam použitých obrázků.

3 Vznik UMA

V roce 2003 byly ve společnosti IBM používány 3 metodiky, a sice IBM RUP, IBM Global Services Method a IBM Rational SUMMIT Ascendant. UMA proto byla vyvinuta s cílem sjednotit schémata a terminologii všech metod a přístupů procesního inženýrství ve společnosti IBM, zároveň se snažila podpořit nejdůležitější průmyslové normy. Z tohoto důvodu byli sjednoceni architekti předchozích tří metodik, kteří společnými silami poskládali jediný koncept nazvaný Unified Method Architecture. Toto sjednocení znázorňuje Obrázek 1.

Po svém dokončení byla UMA předložena společnosti OMG, která návrh tohoto konceptu přijala a později na jeho základě postavila SPEM 2.0. Proto, pokud si budete procházet specifikaci konceptu SPEM 2.0, budete moci nalézt některé prvky, které jsou z UMA patrně převzaté, pro příklad se jedná o elementy typu Role, Task, či Work Product.

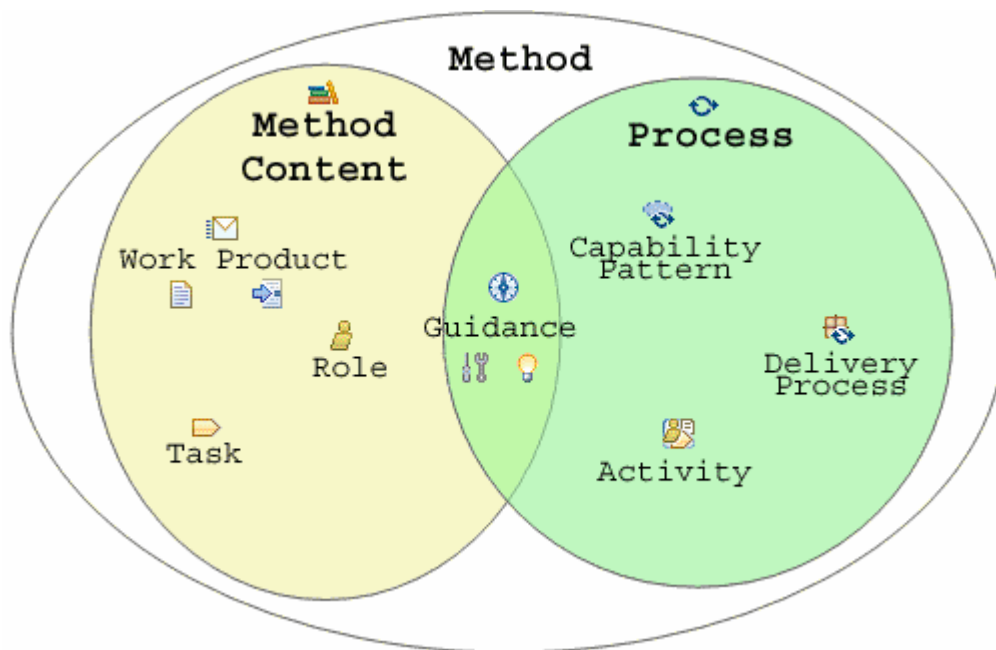


Obrázek 1 - Znázornění vzniku UMA

4 Obecný koncept

Před popisem samotného konceptu UMA, je zajisté namísto začít definováním toho, co vlastně UMA znamená. Nejprve tedy otázka, co se za těmito třemi písmeny skrývá. Asi nikoho nepřekvapí, že zkratka samotná pochází z angličtiny, ve které UMA znamená Unified Method Architecture. Samozřejmě toto asi není odpověď, která by byla dostatečná a už vůbec nevyjasňuje, o co přesně se jedná. Z toho důvodu je v záloze také bližší vysvětlení. Podle něho se v případě Unified Method Architecture (dále už uváděnou pouze jako UMA) jedná o procesně inženýrský meta-model definující schéma a terminologii pro metody, které se skládají z obsahu metod (Method Content) a procesů (Process). UMA je architektura určující, jakým způsobem se mají ukládat metody a procesní metadata. Pokud to mám říct zjednodušeně, pod konceptem UMA si můžete představit jakýsi manuál, podle kterého je koncipována metodika RUPv.7 do jednotlivých elementů. UMA dále popisuje, co by tyto elementy měli obsahovat.

Tento koncept odděluje definici obsahu metod od jejich aplikace v procesech. UMA tímto vytváří definici znovupoužitelného obsahu metod ve formě popisu hlavních částí metod. Naproti tomu jsou definovány aplikace specifické pro jednotlivé projekty, ty jsou definovány ve formě procesního popisu. V další části dokumentu budou popsány základní elementy, které jsou ukázány na Obrázku 2.



Obrázek 2 - Diagram základních elementů

4.1 Základní obsahové elementy

V následujících podkapitolách bude řeč o třech základních elementech, těmi jsou Role, Work Product a Task. Tyto elementy jsou jakýmsi stavebními kameny pro všechny ostatní části UMA. Opět trochu zjednodušeně v praxi tyto elementy odpovídají na otázky. Kdo, co? Pomocí čeho? Vytváří Co? Při jaké činnosti?

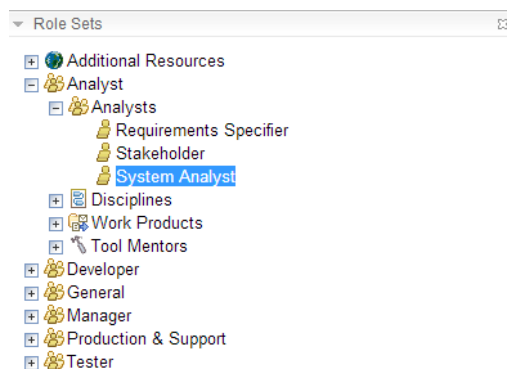
4.1.1 Role

Role je jednou ze základních součástí obsahu metod. Role jsou přirovnávány k rolím herců na divadelním jevišti, stejně jako oni v dané hře hrají členové IT týmu svou roli v rámci daného projektu. Někdy se může stát, že jedna osoba hraje více rolí, případně více osob hraje jednu roli. Zde může sloužit jako příklad opět divadelní hry, které se hrají celosvětově avšak postavy jako takové (role) zůstávají ve své podstatě všude stejné. Stejně je to mezi pracovníky v rámci projektů. Ve více projektech mohou být hrány stejné role. Příkladem rolí zahrnutých v metodice RUPv7 mohou být například System Analyst nebo Project Manager.

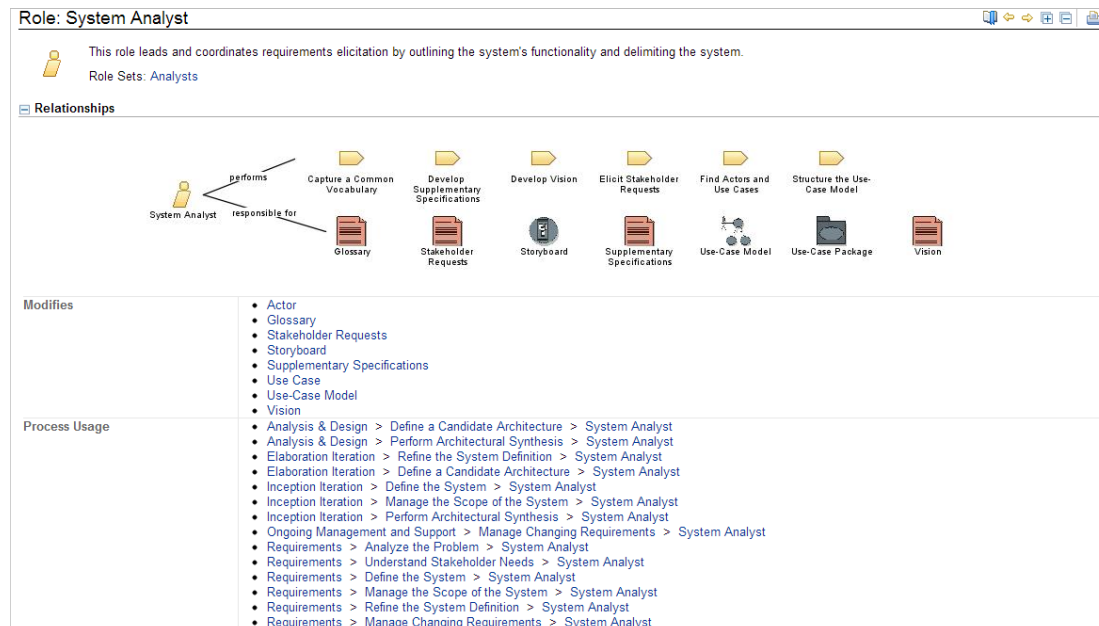
Z výše zmíněného vyplývá, že role mohou být stanoveny individuálně pro jednotlivce, ale také pro více osob pracujících v jednom týmu. Každý člen projektového týmu může vystupovat v různých situacích v rozdílných rolích. V souvislosti s tímto je nutné si uvědomit, že není možné roli zaměňovat za pracovní pozici.

Každá role by měla být definována jako sada dovedností (skills), kompetencí (competencies) a odpovědností (responsibilities). Lidé, nebo nástroje v daných rolích vytváří prostřednictvím plnění zadaných úkolů (Task) výsledné produkty, které jsou v rámci UMA koncepce označovány jako Work Product.

Následují dva obrázky, oba jsou příklady přímo z metodiky RUP. Na prvním z nich, kterým je Obrázek 3, je znázorněna struktura Rolí v metodice RUP. Druhý, Obrázek 4, pak ukazuje náhled na detail Role System Analyst. Na druhém z obrázků je zřejmé, za jaké úkoly (Tasks) a pracovní produkty (Work Product) je daná role zodpovědná. Je zde rozepsáno, s jakými dalšími elementy je svázána a v jakých procesech je využívána.



Obrázek 3 - Ukázka struktury Rolí v RUPu



Obrázek 4 - Detailní popis role System Analyst v RUP

4.1.2 Work Product (Pracovní produkt)

Je abstraktním konceptem poskytujícím zobecnění konkrétních produktů. Těch jsou v UMA stanoveny 3 typy a sice Artifact, Outcome a Deliverable.

Opět s provázáním na ostatní elementy obsahu metody je možné říci, že pracovní produkt je vytvářen nebo modifikován v průběhu úkolů. Naproti tomu pak role pracovní produkt využívají pro plnění úkolů a produkování nových produktů. Důležité je upozornit, že za každý pracovní produkt je zodpovědná vždy pouze jedna role.

Jednotlivé typy Work Product

- 1) Artifact („Artefakt“)** - jedná se o hmatatelné produkty, které mohou být použity v jiných artefaktech. Příkladem může být softwarový vývojový plán obsahující mimo jiné třeba seznam rizik.
- 2) Outcome („Výsledek/Výstup“)** – obvykle je nehmatatelný a znovu nevyužitelný, může se jednat například o stav nebo výsledek. Příkladem může být třeba nainstalovaný server, nebo optimalizovaná síť.
- 3) Deliverable („K dodání“)** – tento typ pracovního produktu má přinášet zúčastněným stranám nějakou hodnotu. Jedná se o balíček ostatních pracovních produktů, složených z předchozích dvou typů, tedy z artefaktů a výstupů.

4.1.3 Task (Úkoly)

Pokud vyjdeme z definice úkolu přímo v metodice RUP, dozvídáme se, že úkol je popis jednotky práce a vždy je vykonáván stanovenou rolí. Může trvat několik hodin, ale také několik dní. Úkoly jako takové mají za cíl vytvářet a upravovat pracovní produkty, jako například modely, třídy, nebo plány.

Úkoly se dále rozpadají do malých částí, takzvaných kroků (Step).

4.1.3.1 Step (Krok)

Krok reprezentuje jednotku s největší mírou granularity, která může být při práci vykonána. Má svoje jméno a textový popis. Kroky jsou výsledkem rozpadu úkolu. Přičemž ne všechny kroky musejí být v rámci úkolu vykonávány v sekvenci, některé mohou probíhat také paralelně.

Také kroky mohou být tři typů.

- 1) **Thinking Steps** – jedná se o kroky, které slouží roli k pochopení úkolu, seznámení se vstupními pracovními produkty a formulování výsledků
- 2) **Performing Steps** – v kroce tohoto typu Role vytváří nebo upravuje pracovní produkty
- 3) **Reviewing Steps** – v tomto typu kroků provádí role prověření výsledků proti zadaným kritériím

Následuje Obrázek 5, na kterém je vidět detail úkolu Identifikace a posouzení rizik. Je zde znázorněn popis úkolu. Je vidět, že je zpracováván rolí Project Managera. Volitelným vstupem je Vize a výstupem úkolu je seznam rizik. V další části jsou pak rozepsané jednotlivé kroky, ze kterých se úkol skládá, pro příklad jsou to Identifikace potenciálních rizik, Analýza a setřídění rizik dle priorit atd.

RUP Small Project Lifecycle > Inception > Inception Iteration [n] > Conceive New Project > Identify and Assess Risks

Task: Identify and Assess Risks

This task describes how to identify, analyze and prioritize risks to the project, determine appropriate risk management strategies, and reflect these in the Risk List for the project.

Purpose

- To identify, analyze and prioritize risks to the project
- To determine appropriate risk management strategies
- To update the Risk List to reflect the current project status

Relationships

Roles	Primary: • Project Manager	Additional:	Assisting:
Inputs	Mandatory: • None	Optional: • Vision	External: • None
Outputs	• Risk List		

Steps

- Identify Potential Risks
- Analyze and Prioritize Risks
- Identify Risk Avoidance Strategies
- Identify Risk Mitigation Strategies
- Identify Contingency Strategies
- Revisit Risks During the Iteration
- Revisit Risks at the End of Iteration

Properties

Multiple Occurrences	
Event Driven	
Ongoing	
Optional	
Planned	
Repeatable	

Obrázek 5 - Detail úkolu Identifikace a posouzení rizik

4.2 Element na pomezí Obsahu metod a Procesů

Sem spadá jediný element, který se může být spojován jak s elementy z části obsahu metod, tak také s procesy. Tímto elementem je Guidance.

4.2.1 Guidance („Řízení“)

Guidance obsahuje několik volitelných elementů, které se mohou pojit s úkoly, pracovními produkty nebo procesními elementy. Záměrem jejich použití je přidat jednotlivým elementům pro konkrétní situace více detailů. Typ guidance určuje obsah elementu. Počet guidance ke každému

elementu není nijak omezen. Guidance elementy mohou být mezi sebou propojovány. Následuje abecední seznam některých typů guidance.

- **Checklist** – Je možné jej přirovnat k nákupnímu seznamu. Checklist umožňuje specifikovat sadu příkazů, které by měli být provedeny ke zkompletování určité sady aktivit. Checklist může být přiložený k pracovnímu produktu. Využívá se především při hodnocení a kontrole stavu.
- **Concept** – Poskytuje širší kontext klíčových myšlenek k odkazovanému elementu. Koncept pro disciplínu by například mohl popisovat základní principy, motivace a výhody skupinových elementů dané disciplíny.
- **Example („Příklad“)** – Obsahuje praktickou ukázkou, jak je možné aplikovat pracovní produkt, či určitý úkol.
- **Guideline („Směrnice“)** – Informace o každém elementu jdou v případě guideline více do hloubky. Například může být stanoveno, jak spustit určité kroky, případně jak dokončit pracovní produkt. Využívají se zejména při nástupu nových uživatelů, kteří potřebují
- **Supporting Material** – Tento typ guidance popisuje vše, co není zachyceno v jiném typu guidance.
- **Template** – Template specifikují strukturu pracovních produktů, poskytují například předdefinované tabulky, povinné sekce dokumentů, hlavičky apod.
- **Tool Mentor** – Tool Mentor popisuje specifické nástroje, které slouží k vytváření částí pracovních produktů.
- **Whitepaper** - Whitepapers jsou dokumenty publikované mimo společnost a jsou přikládány jako příloha.

Koncept UMA nabízí ještě několik dalších typů Guidance, ovšem v metodice RUP jsou blíže popsány pouze výše zmíněné.

4.3 Základní elementy procesů

Pokud si vzpomenete na obrázek 2 a na krátký popis UMA, jistě si také vzpomenete na charakteristiku UMA. Ta říká, že UMA odděluje obsah metod, který už byl zmíněn v části 3.1 od Procesů, jejichž elementy přichází na řadu právě teď.

4.3.1 Activities (Aktivity)

Aktivity jsou základním elementem pro definici procesů. Definují rozpad pracovního toku. Aktivity obsahují deskriptory (o deskriptorech bude řeč později) jednotlivých obsahových elementů, tedy rolí, úkolů a pracovních produktů.

Následující dvě položky, tedy Iterace a Fáze, jsou zvláštním typem aktivity, u které obsahují specifickou hodnotu atributu. Další dva elementy, Capability Pattern a Delivery Process, jsou speciálním typem aktivit a obsahují dodatečnou dokumentaci, která určuje, kdy a z jakého důvodu mají být využity.

4.3.1.1 Phase (Fáze)

Fáze je aktivita, která je specifická tím, že se neočekává její opakovatelnost.

4.3.1.2 Iteration (Iterace)

Iterace jsou plánovány a spouštěny napříč všemi fázemi. Elementy procesu iterace povolují procením inženýrům seskupovat aktivity které jsou plánovány více než jednou. Iterace reprezentují speciální formy aktivit. Očekává se, že tento typ aktivit se bude v projektu opakovat.

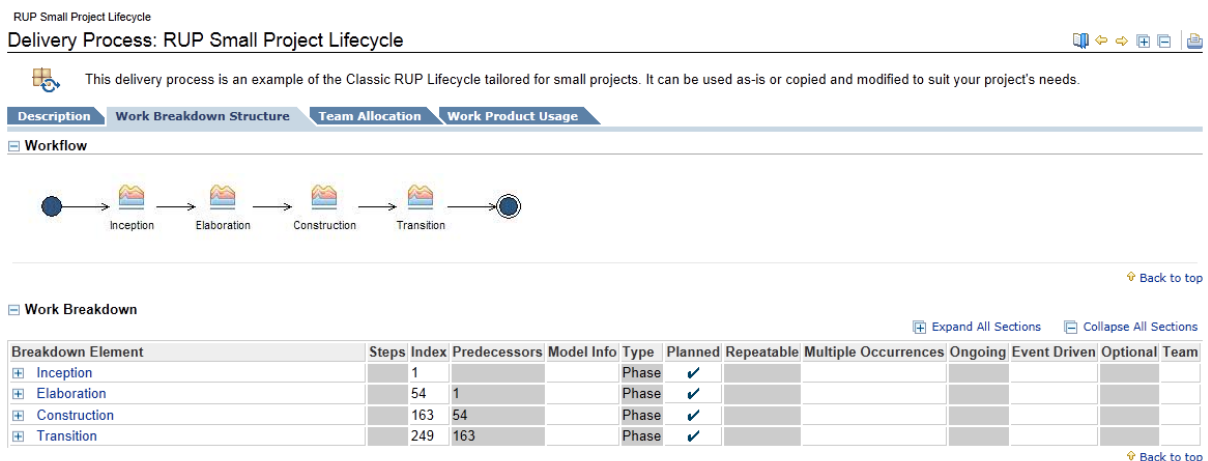
4.3.2 Capability Pattern

Capability Pattern je speciální proces, který popisuje znovupoužitelné části aktivit v oblastech, ve kterých jsou pozorovatelné přínosy. Tyto části jsou pak používány k sestavení Delivery Process nebo se skládají do větších Capability Pattern.

Příkladem takového Capability Patternu může být například proces jednotkového testování, či use case analýzy. Obvykle Capability Pattern obsahují rozpad znovupoužitelných aktivit. Obsahuje vztahy, mezi rolemi a úkoly, které vykonávají v těchto aktivitách. Stejně tak v daných aktivitách popisují pracovní produkty, které jsou využívány a produkovány touto aktivitou.

4.3.3 Delivery Process („Doručovací proces“)

Zde se jedná o proces, který zahrnuje popis celého projektu od jeho zahájení až k jeho dokončení. Jelikož UMA podporuje několik typů životního cyklu projektu, je Delivery Process vždy upraven na míru konkrétnímu životnímu cyklu. Na Obrázku 6 je zobrazený Delivery Process metodiky RUP, kde můžete vidět průchod všemi fázemi RUPu od Inception, přes Elaboration a Construction až k finálnímu Transition. Takto vypadá celý proces vcelku, ovšem je možné jej rozpadnout do větších detailů, až k jednotlivým úkolům. Příklad takového rozpadu je znázorněn na Obrázku 7, kde je vidět rozpad fáze Inception, jejíž jednou z aktivit je Inception Iteration. Ta je opět složena z několika aktivit například z rozpadlé Conceive New Project, která už obsahuje konkrétní úkoly. V případě, že byste rozkleply daný úkol, dostanete klasický obrázek podobný obrázku 5, který byl znázorněný u popisu elementu Task.



Obrázek 6 - Příklad Delivery Process v metodice RUP

Breakdown Element	Steps	Index	Predecessors	Model Info	Type
[-] Inception		1			Phase
[-] Inception Iteration [n]		2			Activity
[-] Conceive New Project		3			Activity
Identify and Assess Risks	●●●●●	4			Task
Develop Business Case	●●●●●	5			Task
Initiate Project	●●●●●	6			Task
Project Approval Review	●●●●●	7			Task
[+] Prepare Project Environments		8			Activity

Obrázek 7 - Znázornění rozpadu fáze Inception v RUP

4.4 Descriptors (Deskriptory)

K využití stejného obsahu elementů v různých aspektech prvků procesu slouží koncept nazývaný descriptor. Tyto deskriptory nahrazují aktuální obsah daného elementu (role, úkolu a pracovního produktu) v kontextu aktivity, ke které element přistupuje. Proto jsou v rámci rozpadu činností využívány role descriptor, task descriptor a work-product descriptor.

Díky využívání descriptorů, namísto obsahových elementů, můžete upravovat obsah používaných elementů (role, úkol, pracovní produkt) podle aktuální situace bez ztráty obsahu zdrojového elementu. Například pomocí task descriptoru je možné smazat určitý krok z daného úkolu a sestavit tak univerzálnější proces. Přičemž ale uživatelé se budou moci kdykoliv vrátit k původnímu stavu, na který descriptor odkazuje. Z tohoto důvodu jsou ve všech výše popsaných diagramech používány deskriptory reprezentující dané elementy, namísto zdrojových elementů samotných.

Existují tři druhy descriptorů: Work Breakdown descriptor, Role descriptor a Task descriptor. Každý z nich sdílí následující atributy:

- **isPlanned** – indikuje, zda je daný prvek zařazený do vytvořeného projektového plánu
- **hasMultipleOccurrences** – označuje, zda je element generován vícekrát
- **isOptional** – značí, zda je daný element volitelný, nebo povinný během průběhu projektu

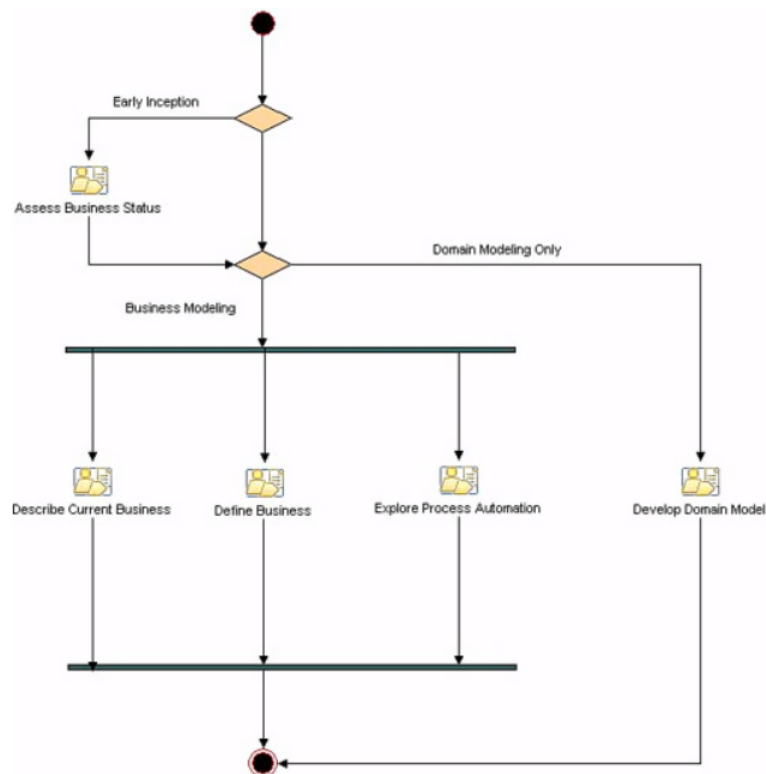
Mimo výše zmíněné obsahují Task descriptor ještě dodatečné tři atributy:

- **isRepeatable** – indikuje, zda se očekává, že daný element bude sekvenčně opakovatelný v rámci jedné sady work-productu.
- **isEventDriven** – určuje, která událost má nastat předtím, než bude naplánovaný element vykonán. Určuje, bez které události nemůže být tento úkol dokončen.
- **isOngoing** – značí, zda má prvek přesně určenou dobu trvání, nebo zda musí být doba trvání plánována průběžně. Například, zda má projektový manager opakovaně kontrolovat dodržování standardní doby trvání projektu.

4.5 Procesní diagramy

Následuje seznam 3 typů diagramů, které jsou v konceptu UMA definovány. Pomocí těchto diagramů jsou znázorněny procesy, ale také odpovědnosti rolí, či fáze jakými prochází pracovní produkty v průběhu projektu zpracovávaného danou metodikou.

4.5.1 Workflow Diagram

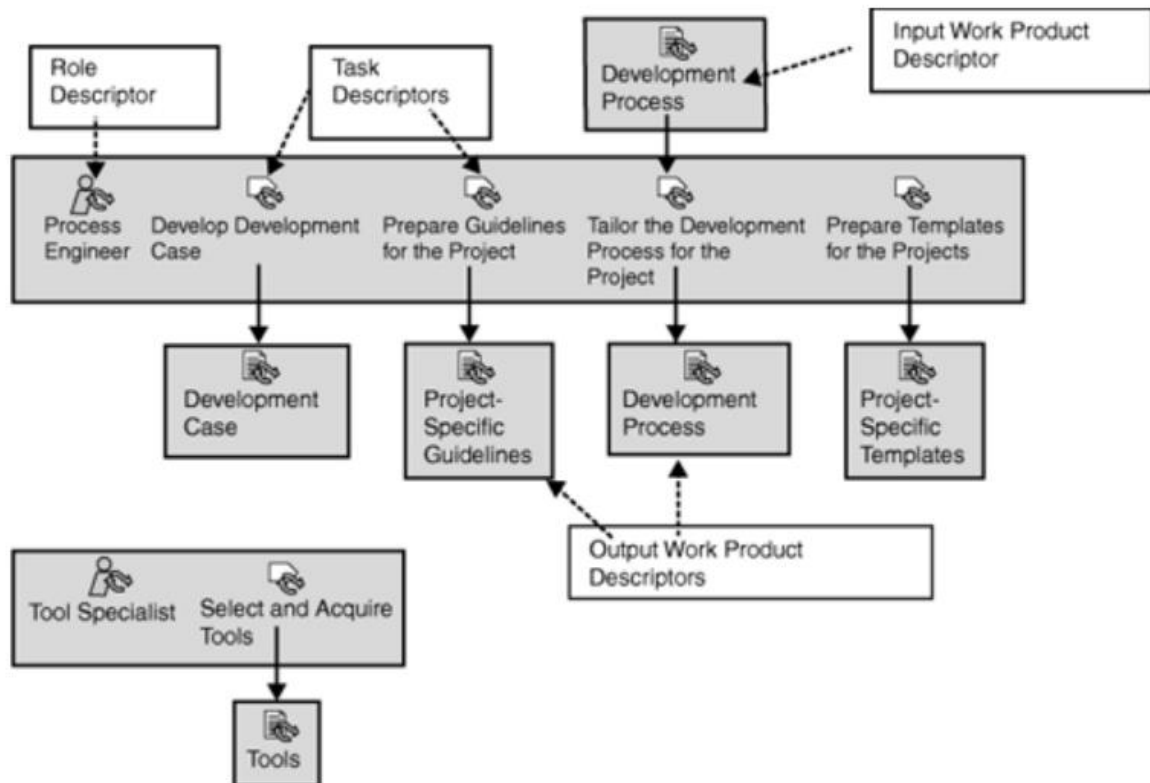


Obrázek 8 - Workflow Diagram

Obrázek 8 zobrazuje první typ využívaného procesního diagramu, konkrétně se jedná o Workflow diagram, neboli volně přeloženo o diagram pracovních toků. Jedná se o jeden z diagramů, který je na míru šitý jazyku UML.

Jeho prvky jsou počáteční a koncový bod, rozhodovací uzly, linky (odkazy), aktivity, synchronizační lišty, úkoly jako popisy fází a milníků. Workflow diagram poskytuje přehled o aktivitách zařazených v určité sekvenci. V případě, kdy byste chtěli znázornit, že některé aktivity probíhají paralelně, vložíte do diagramu synchronizační lištu (synchronization bar). Na Obrázku 8 jsou takto namodelované aktivity Describe Current Business, Define Business a Explore Process Automation. Dané aktivity jsou vždy v diagramu aktivní a kliknutím na ně jste vždy přesměrováni na detailní popis dané aktivity.

4.5.2 Activity Detail Diagram

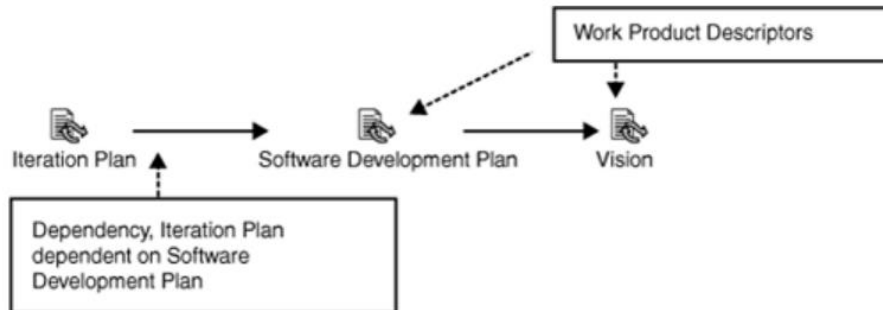


Obrázek 9 - Activity Detail Diagram

Na obrázku 9 jsou příklady dvou Activity Detail Diagramů. Tento typ diagramu zobrazuje vždy danou roli a jí přiřazené úkoly. Zároveň je vždy znázorněn výstupní a případně také vstupní pracovní produkt. Jak si můžete všimnout, jsou na tomto obrázku také označeny popisy jednotlivých elementů a pokud se podíváte pozorně, zjistíte, že u popisů je naznačeno, že v tomto diagramu jsou používány descriptor. Příklad na obrázku znázorňuje, že role Process Engineer nese odpovědnost například za úkol Develop Development Case, jehož produktem má být Development Case.

4.5.3 Work-Product Dependency Diagram

Work-Product Dependency diagram, jehož příklad je na obrázku 10, ilustruje vztahy a závislosti mezi variantami pracovního produktu. Opět na obrázku je například vidět, že Software Development Plan je závislý na Iteration Planu.



Obrázek 10 - Work-Product Dependency Diagram

5 Klíčové vlastnosti UMA

Koncept UMA obsahuje několik klíčových vlastností, které jej charakterizují. Ty hlavní jsou blíže rozebrány v následujících odstavcích.

5.1 Oddělení obsahu metod od procesů (Separation of Method Content and Process)

Jednou z hlavních vlastností konceptu UMA je oddělení definice obsahu metod (Method Content) od jejich aplikací v procesech. Toho je docíleno odděleným definováním následujícího

- Znovupoužitelné jádro obsahu metod. Tohoto je docíleno zobecněním popisu obsahu s využitím rolí, pracovních produktů, úkolů a guidance
- Specifických aplikací typových projektů ve formě popisu procesů, jejichž části odkazují na obsah metody

Obsah metod poskytuje popis jednotlivých elementů, které jsou využívány v procesech. Popis je zde bez ohledu na to, v jakých fázích životního cyklu projektu se právě nachází. Procesy využívají tyto elementy obsahu metod a organizují je do posloupností, jejich pořadí je pak vždy upraveno dle potřeb konkrétního typu projektu.

5.2 Znovupoužitelný obsah (Content Reuse)

UMA dovoluje každému procesu odkazovat se na společné prvky obsahu metod. Z důvodu toho, že se jedná pouze o odkazy na obsah metod, dojde vždy při změně určité části obsahu k jeho automatické úpravě ve všech procesech, které jej využívají. Avšak UMA stále dovoluje úpravu některých částí obsahu používaných v procesech a to jejich individuální modifikací pro konkrétní specifický vztah s každým procesním elementem. (Příkladem může být přidání vstupního pracovního produktu do úkolu, přejmenování role, nebo například smazání kroku, který není pro úkol užitečný)

5.3 Rodina procesů (Process families)

Cílem UMA není vytvářet specifikaci jednoho konkrétního vývojového procesu, nebo udržovat několik nesouvislých procesů, ale poskytnout procesním inženýrům sadu nástrojů jak konsistentně a efektivně navrhovat celou „rodinu“ provázaných procesů. UMA realizuje tuto definici pomocí Capability Pattern a Delivery Processes. Tento koncept dovoluje procesním inženýrům dodržet konsistenci mezi více typy Delivery Process, které jsou specifické pro různé typy. Výsledkem jsou různé varianty specifických procesů postavených na dynamickém znovupoužitelném obsahu metod a vzorů, které je možné aplikovat znovu aplikovat v jiných fázích projektu.

5.4 Podpora více životních cyklů (Multiple Lifecycles)

UMA podporuje různé typy modelů životního cyklu procesního modelování. Jako bonus dokonce nabízí možnost jejich kombinace. Pro příklad je možné uvést, že UMA počítá s vodopádovým (Waterfall), iterativním (Iterative), přírůstkovým (Incremental), vývojovým (Evolutionary) a některými dalšími životními cykly. Celý model byl navržen tak, aby se dokázal přizpůsobit rozdílným přístupům. Poskytuje tak bohatou sadu konceptů a customizačních atributů pro specifikaci dočasné sémantiky pro procesní elementy jako jsou fáze, iterace apod.

5.5 Flexibilní rozšiřitelnost a plug-in mechanismus (Flexible Extensibility and Plug-in Mechanisms)

UMA nabízí unikátní cestu, jak pomocí plug-inů customizovat obsah metod a procesů, bez přímého vlivu na původní obsah. Namísto toho jsou vždy jen popisovány rozdíly oproti originálnímu elementu. Tento systém plug-inů dovoluje uživatelům jednoduchý přechod na novější verzi obsahu metod bez ztráty jejich vlastní customizace předchozí verze.

5.6 Víceprocesní pohled (Multiple Process 'Views')

UMA podporuje nahlížení na procesy z několika různých pohledů při zachování jejich konzistence. Tento přístup pomáhá procesním inženýrům přistupovat k tvorbě procesů dle svých vlastních preferencí. Procesní inženýři si mohou vybrat, jakými způsoby chtějí procesy definovat. Na výběr mají z následujících možností

- Work Breakdown (Structure) – tento pohled, známý také jako WBS, klade důraz na rozpad etap projektu na jednotlivé dílčí úkoly
- Work Product Usage – v tomto pohledu jsou definovány stavy určitých dokumentů a artefaktů v jednotlivých fázích procesu
- Team Allocation – v tomto pohledu jsou definovány potřebné Role, kterým je přiřazena zodpovědnost za daný Work Product

UMA poskytuje konzistenci mezi všemi výše zmíněnými pohledy, je to tak díky tomu, že všechny jsou založeny na jedné objektové struktuře. Díky tomu se změna v jednom pohledu se okamžitě projeví také v ostatních.

5.7 Znovupoužitelné procesní vzory (Reusable proces patterns)

Funkční vzory v UMA jsou složené ze znovupoužitelných bloků připravených k vytváření nových vývojářských procesů. Výběr a aplikace vzorů může probíhat dvěma způsoby

- Může být aplikována pomocí kopírování a modifikace operací, které dovolí procesním inženýrům individuální customizaci obsahových vzorů
- Vzor může být aplikován pomocí dynamických vazeb. Dynamické vazby jsou unikátní metoda, která dovoje skládat znovupoužitelné Aktiviny do vzorů, které pak mohou být použity v dalších procesech. Díky dynamice vazby pak při revizi, či aktualizaci dané aktivity dojde k její aktualizaci ve všech procesech, kde je použita.

6 Porovnání RUP 2003 a RUP 7.0

V této kapitole představíme základní mezi verzemi 2003 a 7.0 metodiky RUP. Zásadním rozdílem obou zmíněných verzí je samozřejmě fakt, že 7.0 se řídí konceptem UMA, který byl detailně popsán v předchozích kapitolách. Pojďme se proto podívat na terminologické rozdíly a některé další novinky, které RUP 7.0 přináší.

6.1 Přehled terminologických rozdílů

RUP 2003	RUP 7.0
activity	task
Artifact	Work Product/Artifact
Workflow detail	Activity
Lifecycle/Configuration	Delivery Process
Artifact set	Domain
Process Model Plug-in	Method Plug-in
Process Component	Method Package
Templates	Template
Glossary	Term Definition

6.2 Komentář k terminologickým rozdílům

V RUP 7.0 došlo k přejmenování devíti pojmů z předchozí verze metodiky. Některé termíny zůstaly podobné a jde pouze o lehkou korekci. Jinde je termín nahrazen úplně jiným. Activity (Aktivita), tedy základní element pro definici procesů, byl přejmenován na „Task“, tedy úloha nebo úkol. Pro nejmenší programovatelnou jednotku je „Task“ obecně používanějším pojmem. To je také hlavní důvod přejmenování. Dalším důvodem byl požadavek větší provázanosti s projektovým managementem.

Artifact byl zařazen pod základní element Method Content – Work Product. To je popsáno v podkapitole 3.1.2. Z důvodu, aby tento vztah byl patrný, došlo k přejmenování na Work Product/Artifact.

Workflow Detail byl přejmenován na Activity. K tomu vedl fakt, že pracovní toky jsou znázorněny v diagramu aktivit. Termín "Activity" většinou odkazoval právě k elementům v diagramu aktivit, někdy vyjadřoval i samotný diagram.

Lifecycle/Configuration byl přejmenován na „Delivery Process“. Ten popisuje proces od jeho počátku do konce, jak prochází jednotlivými fázemi projektu.

Artifact set byl přejmenován na „Domain“. Představuje seskupení artefaktů různých projektových oblastí. Typicky se jedná o Business modeling (Business designer, Business rules, Business Use-case model), testování (test designer, test manager), analýza a design (software architect, design package, data model).

Pojem „komponenta“ je chápána standardně jako černá skříňka, která se použije v konkrétním dobře definovaném rozhraní. V metodice RUP jsou však komponenty chápány jako aktivity, disciplíny, artefakty, role.. Proto bylo důležité tyto dva výklady stejného slova od sebe oddělit. Protože termín „komponenta“ nevystihovala význam komponent v metodice RUP, byl přejmenován na „Method Package“.

6.3 Přehled nových pojmů v RUP 7.0

Zde uvádím výčet nových pojmů, které se objevují v konceptu UMA. Všechny jsou popsány v předchozích kapitolách.

- Work Product
- Work Product/Outcome
- Work Product/Deliverable
- Capability Pattern
- Method Package/Content Package
- Method Package/Process Package
- Process Package/Process Component
- Estimate
- Practise

6.4 Další novinky v RUP 7.0

V konceptu UMA se nově objevuje Task s názvem „Develop Supplementary Specifications“. Úloha slouží k zachycení požadavků, které nelze snadno popsat pomocí Případů užití. Úloha, dříve aktivita, umožňuje být zastávána více rolemi. Dříve byla každá aktivita přiřazena určité konkrétní roli. Tento nový přístup se více blíží realitě. Podporuje takto kooperaci více rolí lidských zdrojů na konkrétní úloze. V RUP 2003 byly artefakty a role tříděny podle disciplín. Avšak někdy nelze artefakt přiřadit pouze jedné disciplíně, protože se používá napříč více disciplínami. Pro tyto účely byly vytvořeny tři kategorie disciplín: pracovní produkty, pracovní definice, role. Tzv. Best practices, které představují šest osvědčených postupů pro vývoj systémů a řízení projektů, byly přeformulovány jako "Key Principles for Business-Driven Development".

7 Závěr

V předchozích kapitolách byly popsány základní elementy, které koncept Unified Method Architecture rozlišuje. Pokud se s každým z nich chcete seznámit blíže, doporučuji projít si metodiku RUP dostupnou ze stránek kitscm.vse.cz

Pokud mám říct, co byste si určitě měli odnést z této práce, vyznačil bych asi následující.

Měli byste si pamatovat, že UMA je manuál, který určuje, jakým způsobem má být metodika sepsána. Hlavním rozlišovacím vzorem tohoto konceptu je oddělení obsahových elementů a procesů. V rámci obsahu metod by metodika podle UMA měla definovat role jednotlivých pracovníků v projektu, dále vstupní a výstupní produkty. Poslední částí obsahu metod jsou pak úkoly, ve kterých jsou produkty produkovány a zpracovávány stanovenými rolemi. Elementy z obsahu metod jsou pak využívány jako základní stavební kameny pro část procesů.

Druhou oddělenou součástí jsou procesy. V těch jsou základními elementy aktivity, capability pattern a delivery process. Přičemž aktivity jsou základními prvky při sestavování procesů. Capability pattern jsou takové části procesů, které jsou znovupoužitelné v jiných procesech. Delivery proces je naproti tomu popis procesu celého projektu od jeho zahájení až do úplného konce.

8 Zdroje

Shuja, K. Ahmad a Krebs, Jochen. 2008. *IBM Rational Unified Process Reference and Certification Guide: Solution Designer.* místo neznámé : IBM Press, 2008. 9780131562929.

VŠE. RUP pro malé projekty. *KITSCM.VSE.CZ.* [Online] <http://kitscm.vse.cz/>.

—. RUP pro velké projekty. *KITSCM.VSE.CZ.* [Online] <http://kitscm.vse.cz/>.

9 Seznam obrázků

Obrázek 1 - Znázornění vzniku UMA.....	5
Obrázek 2 - Diagram základních elementů	6
Obrázek 3 - Ukázka struktury Rolí v RUPu	7
Obrázek 4 - Detailní popis role System Analyst v RUP	8
Obrázek 5 - Detail úkolu Identifikace a posouzení rizik	9
Obrázek 6 - Příklad Delivery Process v metodice RUP	11
Obrázek 7 - Znázornění rozpadu fáze Inception v RUP.....	12
Obrázek 8 - Workflow Diagram.....	13
Obrázek 9 - Activity Detail Diagram	14
Obrázek 10 - Work-Product Dependency Diagram.....	15