

RATIONAL UNIFIED PROCESS AND SMALL PROJECTS

Vladimír Rytíř

Master Degree Programme (2), FIT BUT

E-mail: xrytir01@stud.fit.vutbr.cz

Supervised by: Jitka Kreslíková

E-mail: kreslika@fit.vutbr.cz

ABSTRACT

The purpose of this work is to give the best answer to the question whether the Rational Unified Process (RUP) can be applied to small projects and projects of a one-man team. RUP methodology has primarily been designed for larger projects and big development teams.

1. ÚVOD

Rational Unified Process je platforma obsahující model a nástroje pro řízení a vývoj softwarových projektů. Její nasazení do společnosti není jednoduché, protože metodika je velmi rozsáhlá a obsahuje spousty návodů a dokumentů, které mohou být pro firmu prosperující, ale také jí mohou uškodit. Je důležité si předem určit, jaké dokumenty jsou pro projekt důležité a jaké je lepší vynechat. A co teprve, pokud vyvíjíme produkt menšího rozsahu? Je použití metodiky RUP výhodné? Na tuto otázku se zde budu snažit odpovědět.

2. ROZBOR

V dalších podkapitolách si postupně ukážeme, z jakých částí se skládá metodika RUP, jaké jsou její hlavní praktiky a co vše je nutné ke tvorbě menších systémů, kde tým pracovníků tvoří pouze jediný člověk.

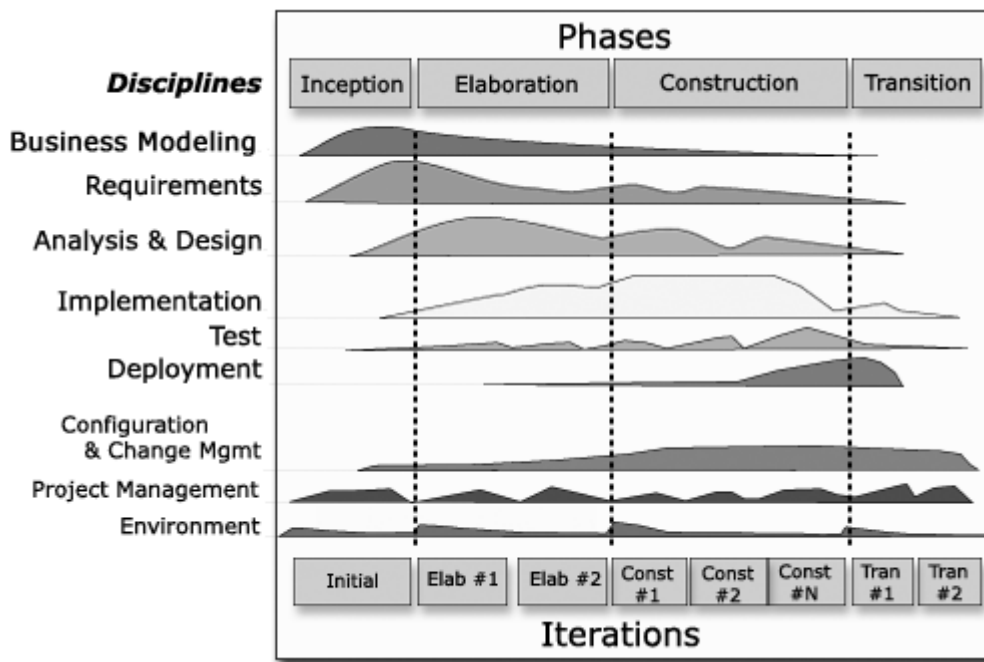
Tématem této práce není podrobné vysvětlení metodiky RUP. Další užitečné informace můžete nalézt v [1].

2.1. RUP A JEHO ČÁSTI

Metodika RUP je založena na dvou-dimenzionálním přístupu vývoje softwaru, který je zobrazen na obrázku 1.

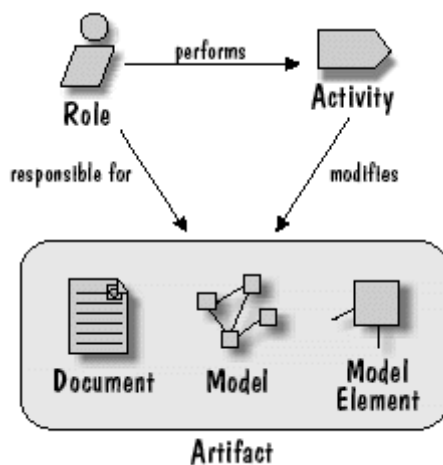
Horizontální dimenzi tvoří čtyři fáze (zahájení, rozpracování, konstrukce a zavedení). Každou fází je možné provádět v několika iteracích. Počet iterací závisí na rozsahu projektu a může být u každé fáze jiný. U menších projektů postačí pouze jedna iterace.

Vertikální dimenzi tvoří aktivity seskupené do několika disciplín. Tyto disciplíny se dále dělí do dvou částí. Prvních šest disciplín se zabývá vlastním vývojem softwaru, zbylé tři pak jeho řízením. Disciplíny určené pro řízení projektů můžeme u malých projektů úplně vynechat.



Obrázek 1: Dvou-dimenzionální pohled na metodiku RUP [2].

Vstupem a výstupem aktivit jsou artefakty. Jedná se o různé dokumenty, které vznikají při tvorbě systémů. Artefaktem mohou být zprávy, posudky, zápisy atd. Jednotlivé role, podílející se na vývoji softwaru, artefakty používají, modifikují a vytvářejí. Vztah mezi rolí, aktivitou a artefaktem zobrazuje obrázek 2.



Obrázek 2: Role, aktivity a artefakty [2].

2.2. NEJLEPŠÍ PRAKTIKY

I u malých projektů je dobré dodržovat šest nejlepších praktik metodiky. Některé praktiky, jako iterativní vývoj, správa požadavků, řízení změn a ověřování kvality, jsou již zaneseny ve dvou-dimenzionálním přístupu k metodice RUP. Vizuální modelování pomůže zachytit modelovaný systém z několika pohledů a komponentovou architekturu je dobré využít např. pro opakované podobné projekty.

2.3. KLÍČOVÝ KONCEPT

Jaké artefakty se budou během projektu tvořit a jaké aktivity se budou provádět, stanoví tzv. development case. Ten určuje, které disciplíny jsou pro projekt důležité.

Prvním důležitým artefaktem je vize projektu, kde jsou sepsány veškeré požadavky na systém a jsou stanoveni aktéři, kteří se systémem komunikují. Use-case diagram modeluje funkční požadavky systému popsané ve vizi. Tento diagram se tvoří, aby nedocházelo k nedorozumění mezi zákazníkem a vývojářem. Vývojář tak přesně ví, jaké má uživatel požadavky.

Samozřejmostí je glosář, který slouží jako slovník pro lepší komunikaci mezi vývojářem a zákazníkem.

Dalšími artefakty, které by měly být zhotoveny i pro malé projekty jsou seznam rizik, dokument popisující architekturu systému a softwarové požadavky. V softwarových požadavcích jsou zaneseny vlastnosti systému. Při tvorbě tohoto dokumentu musí vývojář brát v potaz všechny skupiny uživatelů, kteří jsou zainteresováni do vývoje a budoucího použití. Zde je důležité stanovit, která „schopnost“ systému je důležitější, protože požadavky si mohou odporovat. V neposlední řadě musí vývojář formálně specifikovat veškeré vlastnosti systému. Tyto vlastnosti určují use-case model a dokument s doplňkovými požadavky.

Pokud se jedná již o trochu složitější systém nebo systém, který tvoří pouze část softwaru, který společnost (zákazník) využívá, je dobré vytvořit objektový podnikový model (business object model). Tento podnikový model se skládá z kontextového diagramu a use-case modelu.

Jakou mají mít výše popsané artefakty strukturu a v jaké fázi vývoje se aktivity provádějí, přesně definuje RUP.

3. ZÁVĚR

Implementace velmi obsáhlé metodiky RUP nemá pro menší firmu s několika málo vývojáři užitečný přínos, spíše naopak. Některé postupy a metody se ale dají při tvorbě systémů využít. Díky tomu vývojář předchází rizikům, která vývoj systému doprovázejí a nedorozuměním, které vznikají při komunikaci mezi vývojářem a zákazníkem.

Development case by měl vždy obsahovat aktivity a artefakty, které jsou důležité pro pochopení požadavků a vlastností systému. Jedná se o vizi a softwarové požadavky určené use-case modelem a doplňkovými požadavky na systém. Pro lepší komunikaci mezi všemi zainteresovanými osobami a vývojáři slouží glosář s use-case modelem. Seznamem rizik a popisem architektury se zvyšuje šance na úspěšné dokončení celého projektu. V této oblasti hraje také důležitou roli dodržení šesti nejlepších praktik metodiky RUP.

Pro složitější činnosti prováděné systémem je dobré vytvořit workflow diagramy. Tato schémata rozdělují činnosti na jednodušší části a přehledně zobrazují jejich vazby.

LITERATURA

- [1] Kruchten, P.: Rational Unified Process An Introduction Second edition, Addison-Wesley, 1999, ISBN 0-201-70710-1
- [2] Eeles, P., Houston, K., aj.: Building J2EE™ Applications with the Rational Unified Process, I. vydání Boston, Addison-Wesley, 2002, ISBN 0-201-79166-8