

RUP disciplína Configuration & Change Management

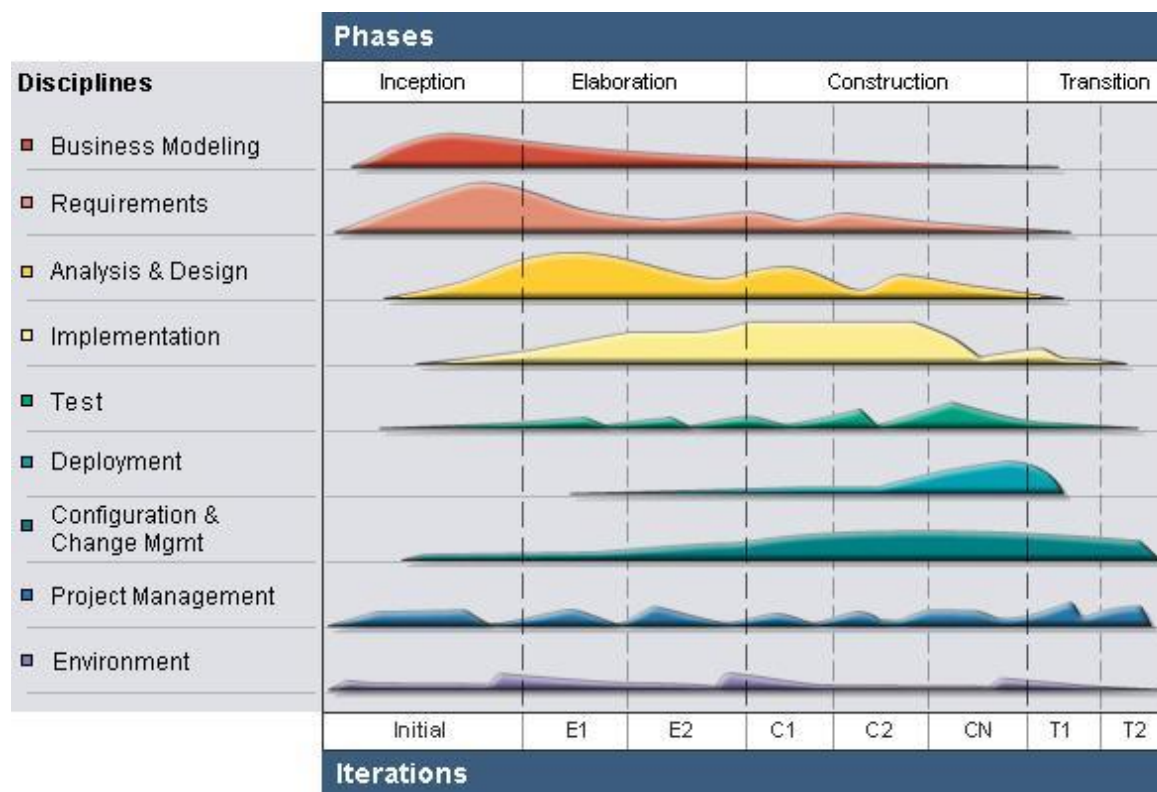
charakteristika, role, činnosti, produkty, návody, vlastní zhodnocení

Obsah

1	Charakteristika.....	3
2	Role	6
2.1	Any Role.....	6
2.2	Change Control Manager	8
2.3	Configuration Manager	10
2.4	Integrator	13
2.5	Test Analyst	14
3	Činnosti	16
3.1	Manage Change Requests.....	17
3.2	Plan Project CM	17
3.3	Create Project CM Environments.....	17
3.4	Manage Baselines & Releases	18
3.5	Monitor & Report Configuration Status.....	18
4	Produkty.....	19
4.1	Change Request.....	19
4.2	Configuration Audit Findings.....	21
4.3	Configuration Management Plan	22
4.4	Project Repository	22
4.5	Workspace.....	23
5	Návody	24
5.1	Configuration and Change Request Management.....	24
5.2	Baselining	24
5.3	Unified Change Management (UCM)	25
5.4	Important Decisions in Configuration & Change Management.....	25
6	Vlastní zhodnocení.....	27
7	Zdroje	28

1 Charakteristika

Configuration & Change Management, který je v češtině nazývaný - Správa konfigurací a změn nebo někdy také Správa požadavků na změnu, je jednou z 9 disciplín, které obsahuje metodika Rational Unified Process. Tyto disciplíny můžeme rozdělit do 2 skupin. První skupina obsahuje 6 hlavních “inženýrských” disciplín a druhá skupina obsahuje 3 podpůrné disciplíny, jejíž součástí je kromě Project Management a Environment také Configuration & Change Management. V jiných metodikách se můžeme setkat s rozdělením Správy konfigurací a Správy změn, ale v RUP je to bráno jako jedna disciplína. RUP je rozdělen do 4 fází - Zahájení, Projektování, Realizace a Nasazení. Správa konfigurací a změn se promítá ve všech těchto fázích, ale nejvíce je využívána v posledních 2 fázích. Viz obrázek níže:



Tato disciplína vysvětluje, jakým způsobem řídit a synchronizovat dílčí produkty, které tvoří celek, kterým je výsledný softwarový produkt. CM je nezbytný pro řízení a kontrolu velkého množství dílčích produktů, které jsou tvořeny velkým množstvím lidí pracujících v rámci společného projektu. Hlavním cílem je zamezit zmatkům, problémům a nekonzistenci celého systému.

Mezi tyto problémy mohou patřit například: Simultánní aktualizace, omezená notifikace nebo více verzí.

Simultánní aktualizace

Dochází k ní v případě, kdy na jednom dílčím produktu pracují 2 nebo více pracovníků. To samotné však nemusí způsobit problém. Ten vznikne, pakliže na produktu pracují odděleně, což způsobí, že poslední verze bude ta, kterou uložil jakýkoliv pracovník jako poslední. Úpravy ostatních pracovníků v tomto případě budou ztraceny. Základní problém tedy vzniká, když systém nepodporuje simultánní aktualizace, což vede k sériovým změnám, které značně zpomalují proces vývoje. Nicméně při simultánních aktualizacích je úkolem zjistit, zda byly provedeny nějaké úpravy a na základě nich provést změny, které nebudu bránit výsledné integraci.

Omezená notifikace

Může nastat, pokud dojde k nějaké změně, která je provedena více vývojáři, ale ne všichni jsou informováni o provedených změnách.

Více verzí

V současné době je většina velkých programů vyvíjena formou evolučních verzí. Tím ve firmě vzniká více verzí, které je nutné dostatečně monitorovat a kontrolovat, protože mohou zapříčinit celkovou nekonzistenci mezi verzemi nebo dokonce záměnu verzí. Tyto situace je nutné později nákladně napravovat. Pro příklad je možné uvést, že ve firmě kolují 3 verze. Jedna je připravena na vydání, další může být právě ve fázi testování a další může být aktuální verze, do které je přidáván fix.

Tyto problémy se snaží ve velké míře řešit právě Správa konfigurací a změn, která je užitečná právě pro správu více verzí a sledování, které verze jsou využívány a za jakým účelem. Slouží pro sledování daných buildů, vytváření buildů pro jednotlivé programy nebo celých verzí. To všechno na základě uživatelem definovaných specifikací.

Seznam prvků poskytovaných Správou konfigurací a změn:

- správa požadavků na změnu
- vykazování stavu konfigurací
- správa konfigurací
- sledování změn
- výběr verzí
- výroba softwaru

Jako vedlejší funkce může být považováno, že jsou ukládána účetní a manažerská data v rámci celého vývojového procesu. Jako například kdo, kdy a proč vytvořil danou verzi a jaké buildy jsou její součástí a další užitečná data.

Pro Správu konfigurací a změn pro malé a velké projekty existují malé odlišnosti.

2 Role

2.1 Any Role

Vzhledem k provázanosti Správy konfigurací a změn se všemi fázemi projektu se zde vyskytuje větší množství rolí. Jednou z nich je “role”, která má název - Jakákoliv role (Any role).

Součástí této role může být jakýkoliv pracovník, který má určenou alespoň jednu specifickou roli v rámci RUP. To znamená, že jediným předpokladem je schopnosti orientovat se a pracovat v rámci projektu. Jakmile bude pracovníkovi přiřazena jakákoliv role, stává se součástí role “Jakákoliv role”. Pod tuto roli spadají úkoly, může realizovat jakýkoliv člen týmu. Pracovník této role je zodpovědný za pracovní prostor.

V rámci malých projektů existuje jen úkol “Submit Change Requests”. Pro velké projekty je však tato role obsáhlejší



2.1.1 Create Development Workspace

Jak již z názvu vyplývá, jedná se o vytváření pracovního prostoru, který bude sloužit jakémukoliv členu týmu pro tvorbu produktu. Na tomto produktu mohou být prováděny změny, které nebudou ihned viditelné pro ostatní členy týmu. Součástí tohoto úkolu je zajistit, že je možné se “dostat” k jakémukoliv produktu, na kterém je potřeba právě pracovat. Tento úkol musí respektovat firemní práva přístupu.

Výstupy: Workspace

2.1.2 Deliver changes

Tento úkol definuje, jakým způsobem mají být data převedena z Developer Workspace do Project Repository. Jedná se o integraci provedených změn v rámci projektu.

Vstup: Workspace (volitelný)

Výstupy: Workspace, Project Repository

Jednotlivé kroky:

- Příprava k přenesení
- Přenést změny
- Aktualizovat Status

2.1.3 Make Changes

Úkol popisuje, jaké jsou hlavní operace pro vytváření a úpravu dílčích produktů a jak je provádět. Členové týmu musí nejprve vytvořit vlastní to-do seznam, který bude respektovat Work Order. Další krok je vytvoření nebo získání dílčího produktu pro práci. Další operace jsou následující:

- Check Out
- Check In
- Add to Source Control - Vytvoření počáteční verze a zařídit kontrolu verzí
- Deliver - Odeslat změny integrátorovi
- Rebase - Zpřístupnit změny provedené ostatními pracovníky

Vstupy: Work Order (volitelný), Workspace (volitelný)

Výstupy: Workspace

2.1.4 Submit Change Request

Popis jak vytvořit požadavek na změnu. Slouží k logování požadované změny. Change Request může obsahovat požadavek na novou funkcionalitu, opravy, požadavky, atp.

Vstupy: Iteration plan (volitelný), Project-Specific Guidelines (volitelný)

Výstupy: Change Request

Kroky:

- Zpracovat formulář pro požadavek na změnu
- Odeslat požadavek na změnu

2.1.5 Update Change Request

Definuje, kdo a jak může provádět požadavky na změny.

Vstupy: Change Request (volitelný)

Výstupy: Change Request

Kroky:

- Obdržet požadavek na změnu
- Aktualizovat a znovu odeslat požadavek na změnu

2.1.6 Update Workspace

Tento úkol zdůrazňuje nutnost aktualizovat pracovní prostory. Účel je zajistit, aby členové pracovali na nejnovějších verzích produktu. Hlavní myšlenka je aktualizovat soubory v jejich pracovním prostoru těmi, které jsou doporučeny jako výchozí

Vstupy: Project Repository (volitelný), Workspace (volitelný)

Výstupy: Workspace

2.2 Change Control Manager

Definuje a dohlíží na proces řízení změn. Osoba vykonávající tuto roli by měla rozumět principům správy konfigurací, zručná při odhadování nákladů a harmonogramu, schopná efektivně komunikovat a určovat jak a kým bude s požadavky manipulováno. Tato role je často sílena více lidmi v rámci Configuration (nebo Change) Control Board. Skládá se ze zástupců všech zúčastněných stran, včetně zákazníků, vývojářů a uživatelů.



V rámci malých projektů, zde chybí úkol “Establish Change Control Process”

2.2.1 Confirm Duplicate or Rejected CR

Tato úloha popisuje, jak ověřit, že je požadavek na změnu třeba zamítnout nebo označit jako duplicitní. K danému podezření by měl být přiřazen CCB správce, který případně potvrdí podezření.

Vstupy: Change Request

Výstupy: Change Request

Kroky:

- Obdržet formulář pro požadavek na změnu
- Potvrdit duplicitu nebo platnost

- Aktualizovat stav

2.2.2 Establish Change Control Process

Úkolem je definovat jak vytvořit Change Control Process. Účelem standardních a řádně zdokumentovaných procesů při řízení změn je, aby změny v rámci projektu probíhaly konzistentním způsobem a příslušné zúčastněné strany byly informovány o stavu výrobků, úprav a nákladů a dopadu těchto změn.

Vstupy: Software Development Plan, Configuration Management Plan (volitelný), Development Infrastructure (volitelný), Development Process (volitelný)

Výstupy: Configuration Management Plan

Kroky:

- Zřídit Change request Process
 - Vypracovat Change Request Form
 - Analyzovat Change Request
 - Odhadnout cenu Change Request
 - Použít Change Request
 - Udržovat Change History
- Zřídit Change Control Board
 - Vybrat členy
 - Jmenovat předsedu z řad PM
 - Setkat se za účelem vytvoření návrhu změn
- Definovat Change Review Notification Protocols - jakým způsobem budou členové informováni o změnách

2.2.3 Review Change Requests

Tento úkol určuje jak provádět posuzování požadavků na změny. Smyslem tohoto úkolu je zjistit, zda žádosti o změnu by měl být přijat nebo označen k zamítnutí. Pro přijaté požadavky tento úkol přiřazuje prioritu, námahu, plán a také zjištění, zda je změna v rámci aktuální verze.

Vstupy: Change Request (CR), Iteration Plan, Problem Resolution Plan (volitelný), Product Acceptance Plan (volitelný), Project-Specific Guidelines (volitelný), Quality Assurance Plan (volitelný)

Výstupy: Change Request (CR)

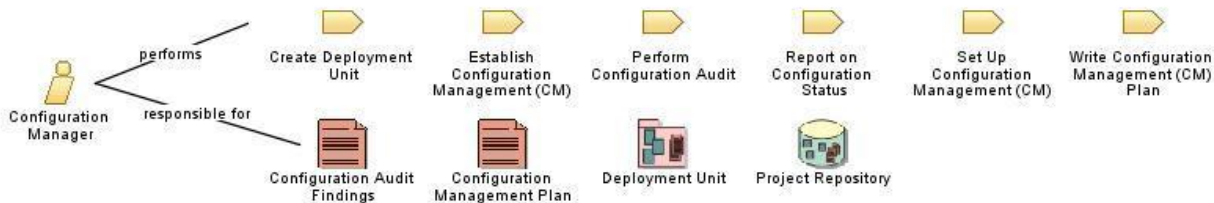
Kroky

- Uspořádat CCB Review Meeting - jednou za týden; Test Manager, Development Manager a člen Marketing Managementu; Probíhá přezkoumání žádostí o CR

- Přijmout CR k přezkoumání
- Přezkoumat předložené CR

2.3 Configuration Manager

Tato role řídí celkovou infrastrukturu a prostředí pro vývojový tým v rámci CM. Podporuje vývoj produktu, aby vývojáři a integrátoři měli vhodné pracovní prostory k vytváření a testování jejich práce. Dále musí zajistit, že prostředí CM usnadňuje hodnocení, změny a chyby produktů. Také je odpovědný za psaní CM Plan a podávání zpráv o statistikách celkového postupu založeného na požadavcích na změnu.



V malých projektech je pouze zodpovědný za vytvoření CM prostředí (Set Up CM Environment)

2.3.1 Create Deployment Unit

Tato úloha popisuje vytvoření média, které bude možné použít v cílovém prostředí. Jedná se o vytvoření kopie dodávaných položek, které jsou uloženy v Project Repository a zároveň jsou pod kontrolou verzí, na cílové médium, kterým může být CD-ROM nebo zazipovaná kopie přístupná ke stažení (V případě online produktu)

Vstupy: Build, Deployment Plan, Bill of Materials (volitelný), Deployment (volitelný), Instalation Artifacts (volitelný), Project Repository (volitelný), Release Notes (volitelný), Training Materials (volitelný), User Support Material (volitelný)

Výstupy: Deployment Unit

2.3.2 Establish CM Policies

Popis, jak vytvořit zásady CM. To obsahuje konvence pro identifikaci konfigurace, praktiky specifikací, praktiky archivování a požadavky na konfigurační reporty.

Vstupy: Software Development Plan, Configuration Management Plan, Development Process

Výstupy: Configuration Management Plan

Kroky:

- Definovat konvence pro identifikaci konfigurace

<SYSTEM>[<A>]_[<SUBSYSTEM>]_[<A>]_[R|A|B]<X>[.<Y>.<Z>][.BL<#>]

<SYSTEM> Identifikuje systém

<A> 3 písmena určující druh dílčího produktu v rámci systému

PLN	Project Plans
REQ	Requirements Files
USC	Use Cases
MOD	Model Files
SRC	Source Code Files
INT	Public Interfaces
TST	Test Scripts and Results
DOC	Documentation (User, Release Notes)
BIN	Executables

<SUBSYSTEM> Identifikuje podsystém

<A> Určuje druh dílčího produktu v rámci podsystému

R A B	Stand for release, alpha, or beta
<X>	Integer, stands for a major release (e.g. 1)
<Y>	Integer (optional), stands for a minor release
<Z>	Integer (optional), stands for an alternative release (patches, ports, etc.)
BL	Stands for base level (an internal release)
#	Integer, for internal releases

- Definovat praktiky specifikací
- Definovat praktiky archivování
- Definovat požadavky na konfigurační reporty
 - Výběr reportů týkajících se požadavků na změnu
 - Definovat frekvenci reportingu

2.3.3 Perform Configuration Audit

Z názvu jednoznačně vyplývá, čeho se týká tento úkol. Jedná se o provedení auditu konfigurace za účelem zjištění, zda specifikace obsahuje všechny potřebné dílčí produkty a zda splňuje všechny požadavky.

Vstupy: Project repository, Configuration Management Plan (volitelný)

Výstupy: Configuration Audit Findings

Kroky:

- Provést fyzický audit
 - Identifikovat specifikaci, která má být nasazena
 - Ujistit se, zda všechny požadované dílčí produkty (definované vývojem) jsou součástí specifikace
 - Sepsat seznam chybějících produktů do Configuration Audit Findings
- Provést audit funkcí
 - Připravit report, který obsahuje všechny požadavky a výsledky testů související se specifikací
 - Ujistit se, zda existuje alespoň jeden test nebo zda nějaké nechybí nebo neprošli testem. Nesrovnalosti zapsat do Configuration Audit Findings
 - Vygenerovat seznam požadavků na změnu, které souvisí se specifikací. Nesrovnalosti zapsat do Configuration Audit Findings
- Uvést závěry
 - Určit nápravná opatření
 - Pro každou opravu přiřadit odpovědnost a datum dokončení

2.3.4 Report on Configuration Status

Popis, jak podávat zprávy o stavu navrhovaných změn a jejich implementaci. Účelem je podpořit úkoly Configuration Status Accounting, usnadnit hodnocení produktů pomocí sledování chyb a podávání zpráv o úkolech.

Vstupy: Configuration Management Plan, Project Repository

Výstupy: Project Measurements

Kroky: Podávat zprávy o stavu konfigurace a chybách

2.3.5 Set Up CM Environment

Úkolem je popsat jak vytvořit prostředí, kde může být produkt vyvíjen a sestavován. To se provádí ve 2 etapách. Nejprve vytvoření hardwarového prostředí a poté vývojového.

Vstupy: Software Development Plan, Configuration Management Plan (volitelný), Development Infrastructure (volitelný), Development Process (volitelný), Implementation Model (volitelný), Iteration Plan (volitelný)

Výstupy: Project Repository

Kroky:

- Vytvořit HW prostředí pro CM
- Vytvořit strukturu do Repository
- Vytvořit základní sadu verzovaných součástí - převést existující materiál do nové struktury
- Definovat postup mezi úrovněmi specifikace

2.3.6 Write CM Plan

Definice jak psát CM Plan. Účelem je popsat všechny úkoly související s CM, které mají být prováděny v průběhu životního cyklu produktu/projektu. Sepsat jakým způsobem mají být úkoly CM plánovány, prováděny, řízeny a organizovány.

Vstupy: Software Development Plan, Development Process

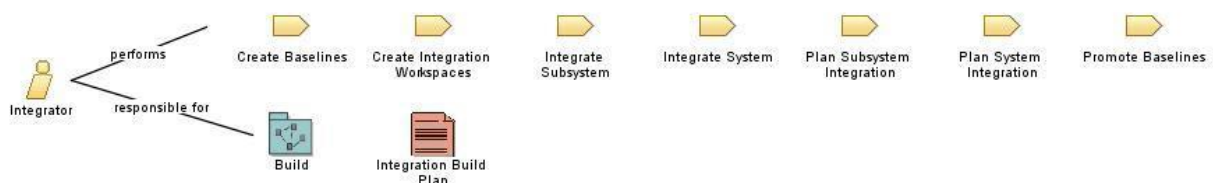
Výstupy: Configuration Management Plan

Kroky:

- Napsat CM Plan
- Posoudit a schválit CM Plan
- Udržovat CM Plan

2.4 Integrator

Hlavní náplň práce integrátora je jednoznačná. Shromažďuje součásti od vývojářů a kombinuje je za účelem vytvoření buildu. Dále je odpovědný za plánování integrace. Mezi jeho schopnosti musí patřit znalost systému nebo části, která musí být integrována, a jejich závislostí. A samozřejmě se musí znát s vývojáři.



2.4.1 Create Baselines

Popis co je to specifikace a kdy by měla být vytvořena. Specifikace označuje jednu jedinou verzi prvku, který se skládá z množiny verzovaných souborů a složek. Specifikace by měly být vydávány na konci každé iterace.

Vstupy: Project Repository

Výstupy: Project-Specific Guidelines, Software Development Plan (volitelný), Work Order (volitelný), Workspace (volitelný)

2.4.2 Create Integration Workspaces

Úkol popisuje, jak vytvořit Integration Workspace, které je potřebné k integraci. To je místo, kde se systémoví a subsystémoví integrátoři ubezpečují, že odděleně vyvinuté komponenty mohou společně tvořit produkt. Existují 2 druhy pohledů. První je dynamický, který poskytuje okamžitý, transparentní přístup k soborům a složkám v Project Repository. Druhý je tzv. Snapshot View, kde se vytvoří kopie a vloží se na lokální počítač. Při tvorbě nové specifikace je třeba toto prostředí uzamknout, aby byla jistota, že se pracuje na statické množině souborů.

Vstupy: Project Repository, Development Process (volitelný), Project-Specific Guidelines (volitelný)

Výstupy: Workspace

2.4.3 Promote Baselines

Definuje, kdy je proces tvorby specifikace hotový a jak má být specifikace označena. Účelem tohoto úkolu je zajistit, aby specifikace byly označeny tak, aby odrážely úroveň zralosti softwaru, kvalitu a stabilitu, které může být dosaženo. Vhodně označená a specifikovaná verze je pak k dispozici pro vydání nebo další vývoj v následujících iteracích.

Vstupy: Project Repository, Development Process (volitelný), Project-Specific Guidelines (volitelný), Workspace (volitelný)

Výstupy: Project Repository, Workspace

Kroky:

Rozhodnout o vhodném označení specifikací - Může být pojmenována po fázi a iteraci, ve které byla vytvořena, například BL-Product-X-c2 (vytvořena na konci 2. iterace fáze Construction) nebo označena nevyšším stádiem, kterého dosáhla, například Integration Tested

2.5 Test Analyst

2.5.1 Verify Changes in Build

Prověření změn v buildu definuje, jak prověřit zda došlo k požadovaným změnám. Kontrole předchází dokončení požadavků, označí se stavem „resolved“ a zařadí se do aktuálního buildu. Po té se build testuje, kdy test ověří, jestli je změnový požadavek v pořádku „verify“ nebo „failed“. Pak se kontroluje repase a pokud je to vše ok, tak je požadavek „closed“. [KUC06]

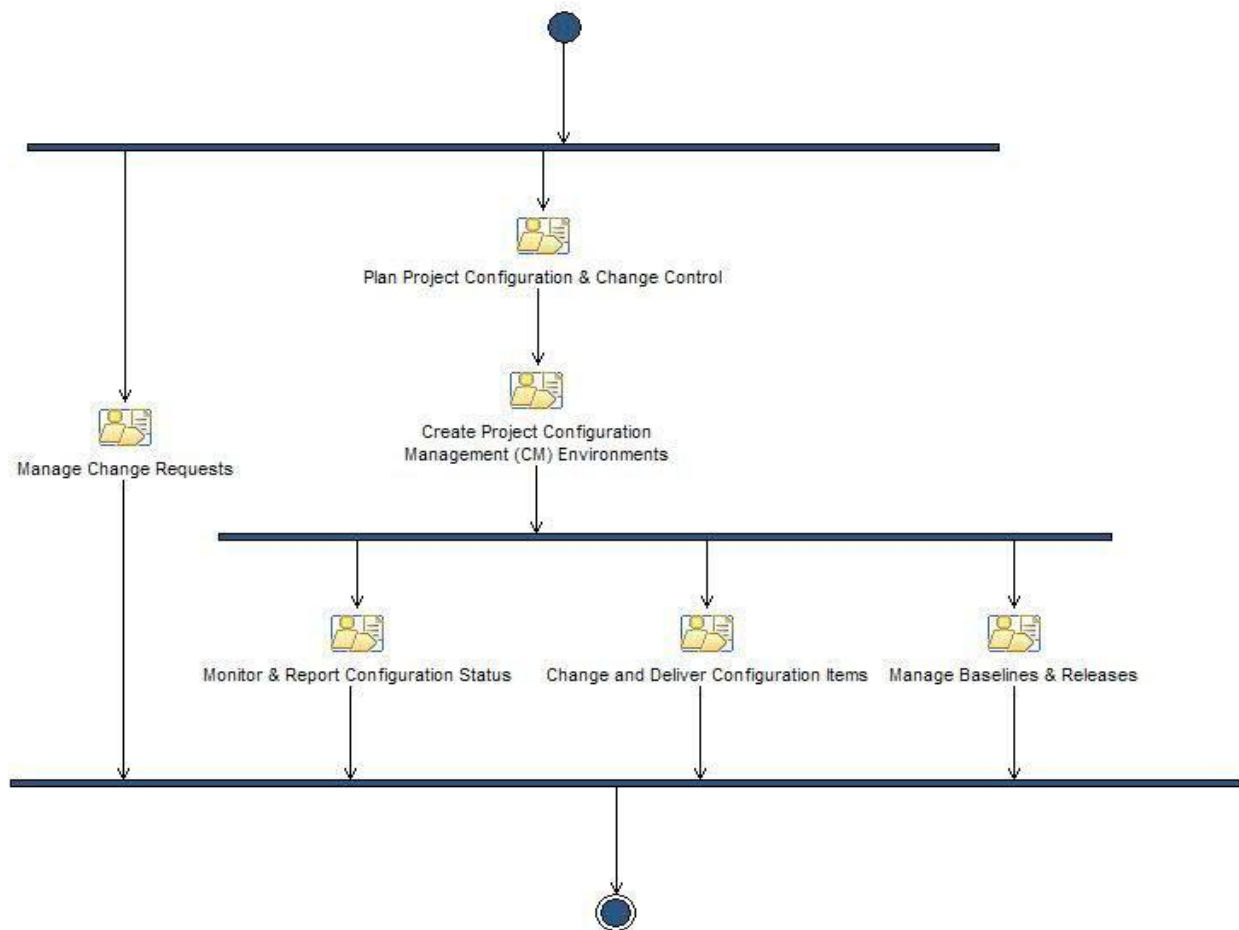
Vstupy: Build , Change Request (volitelné), Test log

Výstupy: Change Request, Test Result

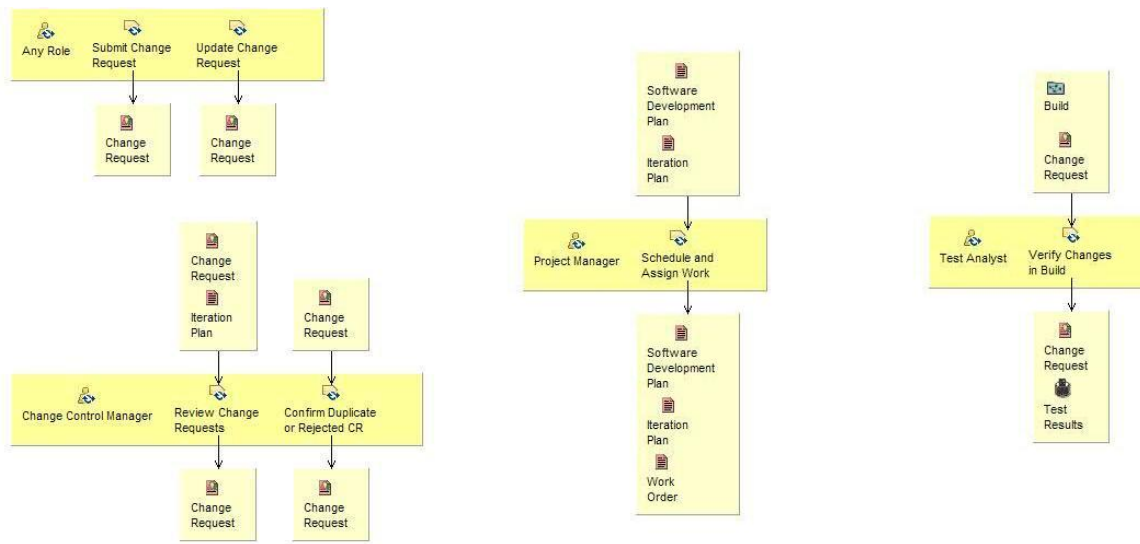
Kroky:

- Vyřešit CR
- Kontrola změn v Test Build
- Ověření změn v Release Build
- Ověřit a vyhodnotit výsledky

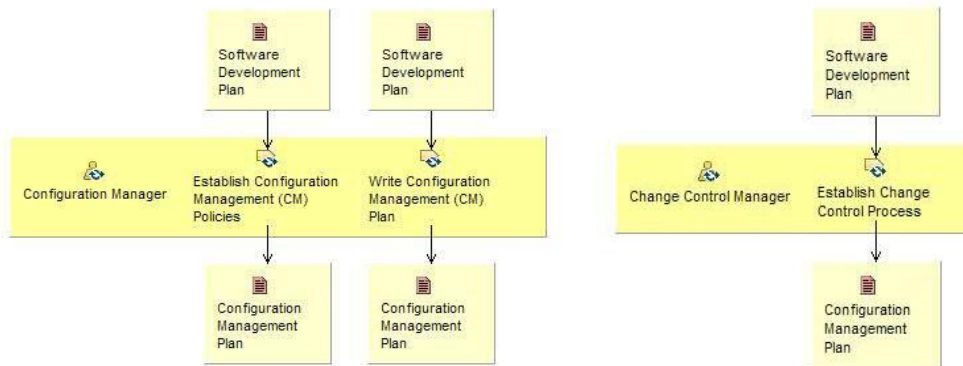
3 Činnosti



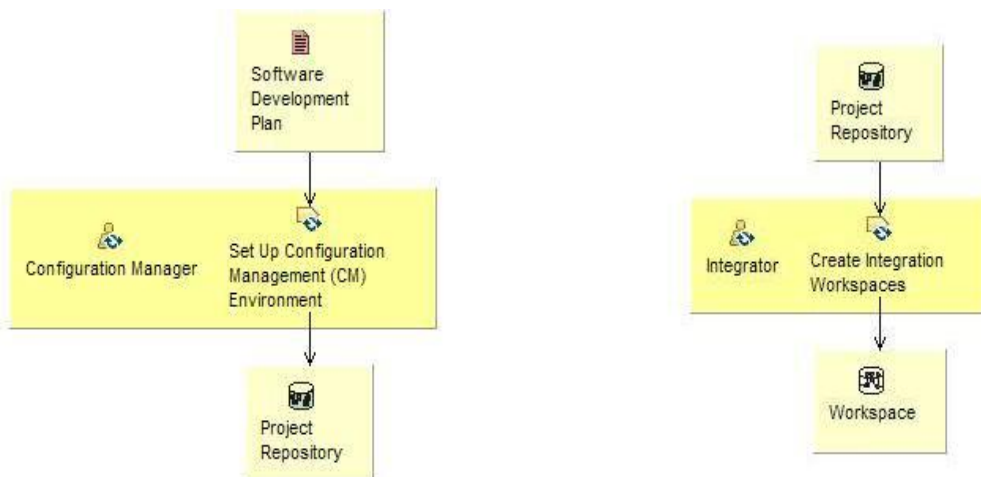
3.1 Manage Change Requests



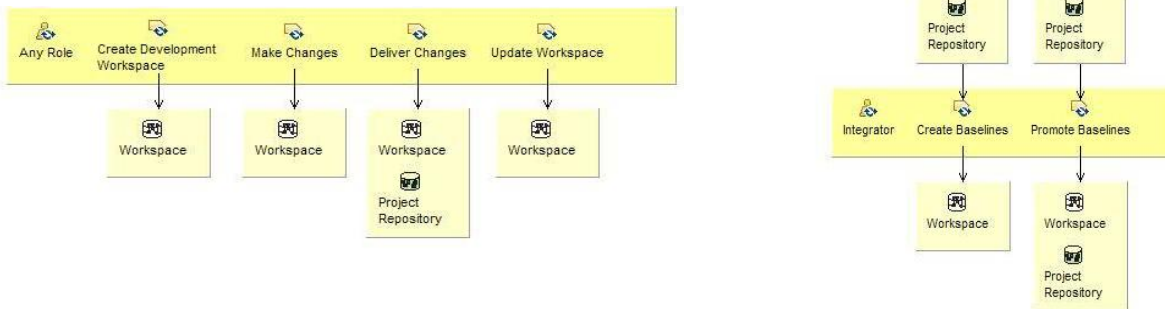
3.2 Plan Project CM



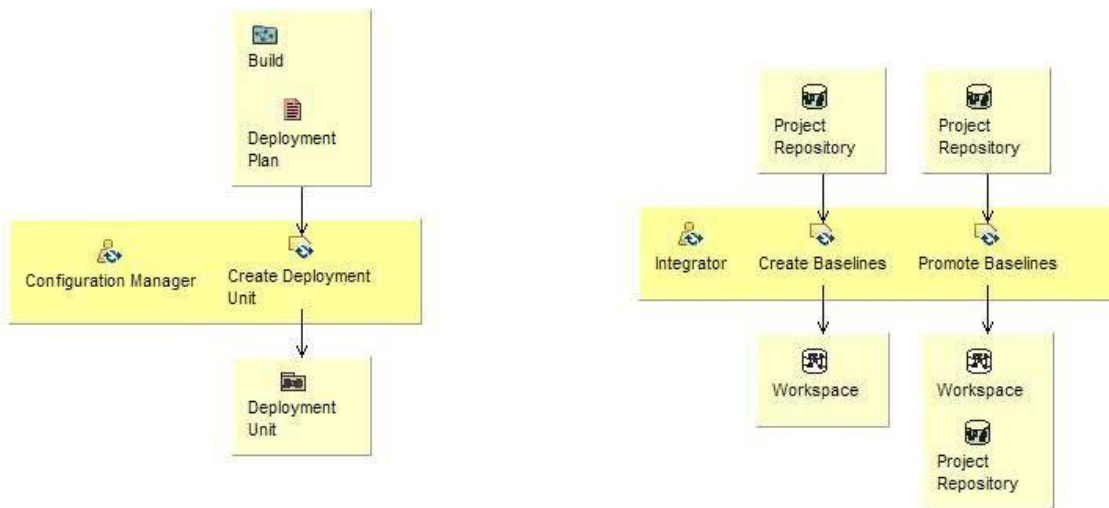
3.3 Create Project CM Environments



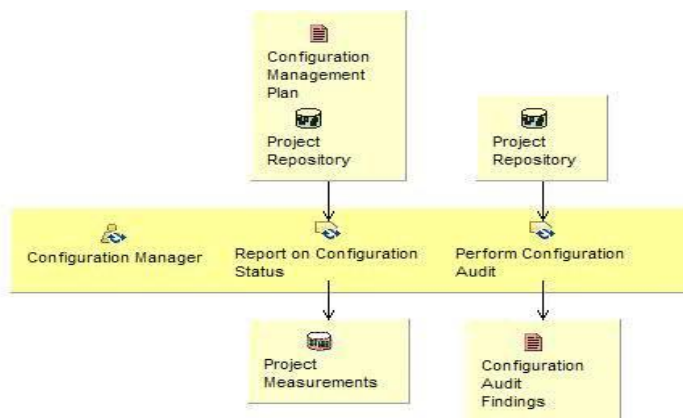
3.4 Change and Deliver Configuration Items



3.5 Manage Baselines & Releases



3.6 Monitor & Report Configuration Status



4 Produkty

4.1 Change Request

Potřeba provádět změny prochází celým procesem vývoje softwaru. Od úvodních fází tvorby systému až po každodenní správu a údržbu produkční verze. Proto je zde Change request - požadavek na změnu, který je záznamem o rozhodnutí provést nějakou změnu a jeho cílem je společně s navazujícími procesy uvést tuto změnu v život.

Pro požadavek na změnu můžeme nalézt mnoho různých označení, ať už jde o CR, defekt, bug nebo incident. Pomocí zaznamenávání a řádné správy požadavků na změnu je možné zajistit vhodné a kontrolované úpravy systému s předvídatelným dopadem na cílový systém.

Mezi důležité typy požadavků na změnu řadíme:

- **Enhancement Requests - žádosti o vylepšení**, které se používají především pro sběr požadavků na budoucí verze systému. Tj. na vlastnosti a funkcionalitu, kterou je požadováno doplnit do dalších verzí systému. Jedná se o požadavky od zainteresovaných osob, které tak vyjadřují své požadavky a potřeby.
- **Defects - defekty** jsou reporty nepředvídaných událostí a chyb, které se vyskytují v současném systému. Defekty obsahují nedostatky a problémy systému, které byly objeveny v úvodních fázích životního cyklu nebo chyby, které byly identifikovány a měly by být opraveny v rámci softwaru. Defekty také mohou obsahovat odchylky od toho, co bylo požadovaného a jak se software ve skutečnosti chová (například problémy s použitelností softwaru).

Účelem hlášení chyb je uvést co nejpodrobnější popis problému, který následně umožní spuštění opravné akce, vyřešení problému, to vše v dokumentované formě.

Požadavky na změnu používají následující role:

- **Analysts** používají požadavky na změnu k definování požadovaných změn v rámci vyšší úrovně abstrakce, tedy větších a komplexnějších změn. Pracují nejčastěji s požadavky na vylepšení (Enhancement Requests).
- **Managers** používají požadavky na změnu k řízení a kontrole práce na přiřazených úkolech

- **Testers** slouží požadavky na změnu k popisu chyb a defektů, na které narazí v rámci testování software.
- **Developers** používají požadavky na změnu k analýze a opravě nahlášených chyb v rámci systému.
- **Test Analyst** používají požadavky na změnu k plánování testů, které ověří zpracované požadavky na změnu. Slouží také k měření výsledné kvality softwaru i celého vývoje sledováním pomocí nahlášených chyb.

Ukázkový formulář pro požadavek na změnu:

1. Identifikace

- Projekt
- Číslo požadavku na změnu
- Typ požadavku (problém nebo vylepšení)
- Název
- Datum odeslání
- Původce (autor)
- Priorita požadavku

2. Samotný problém

- Popis problému
- Kritická chyba
- Závada
- Vylepšení
- Nový požadavek
- Podmínky, za kterých byl problém pozorován
- Současné prostředí (hardware)
- Kompilátor operačního systému
- Zdroj problému
- Náklady či úspory z tohoto problému

3. Navrhovaná změna (od původce požadavku)

- Popis navrhované změny
- Odhadované náklady na implementaci požadované změny

4. Navrhovaná změna (od Change review týmu)

- Akce
- Schváleno
- Zamítnuto
- Odloženo
- Popis navrhované změny
- Postižené konfigurační soubory

- Kategorie
- Oprava chyby
- Vylepšení
- Nová vlastnost
- Ostatní

5. Řešení

- Odhadované náklady na implementaci změny
- Implementátor
- Skutečný čas na implementaci změny
- Analýza
- Implementace
- Test
- Dokumentace
- Postižené řádky kódu

6. Posudek

- Testovací metody
- Inspekce
- Analýza
- Demonstrace
- Testy
- Testovací platformy
- Testovací případy

7. Change Review Team Disposition

- Změna potvrzena a přijata

4.2 Configuration Audit Findings

Účelem Configuration Audit Findings neboli nálezů konfiguračního auditu je reportovat výkon vyvinutého softwaru a porovnat jej s požadavky. Dále informovat o fyzické přítomnosti požadovaných částí systému.

Tento dokument identifikuje baseline, chybějící požadované části systému, a nedokončené testy nebo nesplněné požadavky.

Následující témata je třeba řešit v nálezech konfiguračního auditu:

- Úvod
- Datum auditu
- Účel auditu
- Celkové hodnocení
- Audit fyzické konfigurace

- Identifikace baseline
- Chybějící produkty
- Audit funkční konfigurace
- Netestované požadavky
- Neúspěšné požadavky
- Otevřené požadavky na změnu
- Akce k nápravě
- Akce
- Zodpovědná osoba
- Datum následné kontroly

4.3 Configuration Management Plan

Configuration Management Plan neboli plán řízení konfigurace shrnuje a upřesňuje plán aktivit, přiřazené zodpovědnosti a požadované zdroje včetně zaměstnanců, nástrojů a počítačového vybavení.

Jeho účelem je definovat nebo poukázat na aktivity, které popisují jak je řízení změny a konfigurací prováděno při vývoji softwarového produktu.

RUP nabízí ukázkovou šablonu dokumentu, která má 6 stran. Je rozdělena do šesti hlavních částí. V úvodu je definován účel a rozsah včetně potřebných definic. Druhá část se věnuje řízení konfigurace softwaru, kde definuje jak samotnou organizaci, odpovědnosti a rozhraní, tak také nástroje, prostředí a infrastrukturu. Další část popisuje samotný aplikovaný program. Od identifikace jednotlivých konfigurací přes jejich kontrolu po nakládání s konfiguracemi. V dalších sekcích jsou definovány milníky, zdroje a práce s dodavateli a zákazníky.

4.4 Project Repository

Projektový repozitář v sobě ukládá všechny verze souborů a složek projektu. Také nese veškerá odvozená data a metadata spojená s uloženými soubory a složkami.

Na základě velikosti projektu může být vytvořeno více projektových repozitářů a každý repozitář může obsahovat desítky tisíc souborů a složek. Počet souborů uložených na jednom projektovém repozitáři závisí na velikosti disků, na kterých běží repozitářový server a počtu uživatelů, kteří současně přistupují k datům. Repozitářový server se stará o čtení a zápis dat do projektového repozitáře.

Požadavky na repozitářový server jsou především v oblasti paměti, rychlosti čtení a zápisu, způsobu připojení a jeho velikosti a velikost diskového prostoru. Přičemž každý z těchto parametrů je možné individuálně parametrizovat dle požadavků projektu. Důležité pro hladký průběh je celková spolehlivost tohoto serveru.

4.5 Workspace

Tento produkt doplňuje kontrolovaný přístup k jednotlivým částem systému a dalším zdrojům, které jsou potřeba pro vývoj použitelného softwarového produktu. Workspace neboli pracovní prostor poskytuje bezpečný a exkluzivní přístup k verzovaným částem projektu.

Účel pracovního prostoru je umožnit přístup k vyvíjenému produktu a ke zdrojům potřebným k vývoji a sestavení dodatečného produktu. Vývojové pracovní prostory jsou soukromými prostory vývojářů, kteří zde mohou vyvíjet a testovat zdrojový kód v relativní izolaci od ostatních vývojářů. Integrační workspaces slouží jako veřejný prostor, kde se setkává individuální práce jednotlivých vývojářů a skládá se v jeden společný produkt.

Jak už bylo řečeno výše, můžeme definovat dva druhy workspace:

- **Vývojový workspace** je prostor pro individuální vývoj, kde může vývojář dělat změny bez toho aniž by se tyto změny dotkly ostatních vývojářů či jim byly vůbec viditelné.
- **Integrační workspace** je sdílený prostor přístupný všem členům projektového týmu. Hlavní produkt je sestavován a verzován právě v tomto integračním workspace.

V projektu je jedno sdílené integrační workspace nebo jich může být i více. Každý člen projektového týmu potřebuje pracovat v rámci takového workspace, ve kterém získá přístup k částem systému, kterou jsou uloženy v projektovém repozitáři. Integrátor následně vytváří buildy (sestavení) v rámci integračního workspace a vytváří hlavní verzi, která je přístupná celému vývojovému týmu.

Protože je za každé vývojové prostředí zodpovědný každý člen týmu individuálně, je možné přizpůsobit tento produkt potřebám každého individuálně dle specifik práce, která je dělána v rámci workspace.

5 Návody

5.1 Configuration and Change Request Management

Následující koncept popisuje rozsah a operace v rámci Configuration and Change Request Managementu. Ten se skládá z následujících částí:

- **Change Request Management (CRM)** - se zaměřuje na organizační infrastrukturu potřebnou k posouzení nákladů a harmonogramu a dopadu požadovaných změny na stávající produkt.
- **Configuration Status Accounting (Measurement)** - se používá k popisu "stavu" produktu na základě typu, počtu, míry a závažnosti zjištěných nedostatků a je stanovován, v průběhu vývoje produktu. Metriky odvozené na základě tohoto měření získané buď formou auditů nebo z dat jsou užitečné při určování celkové míry, do jaké je projekt hotov.
- **Configuration Management (CM)** - popisuje strukturu produktu a identifikuje jeho základní konfigurační vlastnosti, které jsou klasifikovány jako samostatné verze. CM se zabývá vymezením konfigurace, sestaveními, značení, evidencí verzí a udržování sledovatelnosti mezi jednotlivými verzemi.
- **Change Tracking** - popisuje, co se děje s jednotlivými prvky, z jakých důvodů jsou měněny a kdy. Slouží jako historie a zdůvodnění provedených změn.
- **Version Selection** - účelem dobrého "výběru verze" je zajistit, aby vždy byla správná verze konfigurace byla aplikována při změně implementace. Výběr verzí je založen na pevných základech "identifikace konfigurace".
- **Software Manufacture** - pokrývá potřebu automatizovat kroky k sestavení, testování a balení softwaru pro distribuci.

5.2 Baselining

Baseline je "snímek" jedné a každé verze produktu v určitém čase, který je uložen v projektovém úložišti. Poskytuje oficiální standard, na kterém je založena veškerá další práce, a všechny změny mohou být provedeny pouze autorizovaně. Po založení baseline je každá další změna zaznamenána jako rozdíl, dokud nedojde k nastavení další baseline.

Jsou tři hlavní důvody pro vytváření baseline a to reprodukovatelnost, sledovatelnost a reporting.

Reprodukovatelnost je schopnost jít zpět v čase a reprodukovat daný release softwarového systému, či reprodukovat vývojové prostředí v předchozí době v rámci projektu.

Sledovatelnost nám umožňuje sledovat vztah předchůdce - nástupce mezi jednotlivými verzemi. Jejím cílem je zajistit, že výsledná práce splňuje požadavky a finální program je postaven ze správného kódu.

Reporting je založen na srovnání obsahu jednoho baseline proti druhému. Baseline srovnání pomáhá při ladění a generování release notes.

Pokud jsou vytvořeny baseline, všechny prvky a baseline musí být označeny tak, aby byly jednoznačně identifikovatelné a znovu-vytvořitelné.

Doporučení: Vytvářejte baseline pravidelně, abyste se ujistil, že vývojáři jsou stále mezi sebou synchronizováni v jejich práci s ostatními. Je potřeba vytvářet baseline i rutinně na konci iterací (drobné milníky) a hlavních milníků, které jsou spojené s koncem jednotlivých fází životního cyklu.

5.3 Unified Change Management (UCM)

Unified Change Management (UCM) je přístup k řízení změn při vývoji softwarových systémů, od požadavků, až po release. UCM se klene nad životním cyklem vývoje, definuje jak řídit změnu požadavků, modelů, dokumentace, komponent, testovacích případů a zdrojového kódu.

Jedním z klíčových aspektů modelu UCM je, že sjednocuje činnosti sloužící k plánování a sledování pokroku projektu a výsledky práce, které se změnily. Model UCM je realizována jak procesy, tak nástroji. Produkty ClearCase a ClearQuest jsou základními technologiemi pro UCM.

ClearCase řídí všechny produkty vytvořené v rámci projektu.

ClearQuest řídí v projektu úkoly, vady, a požadavky na vylepšení a poskytuje vizualizační a reportovací nástroje potřebné ke sledování průběhu projektu.

5.4 Important Decisions in Configuration & Change Management

RUP nabízí i přehledovou tabulku s informacemi, které produkty bychom měli za jakých případech pro své projekty použít.

Následující tabulka udává, které nástroje jsou doporučené, a které jsou považovány za volitelné (tj. mohou být použity pouze v některých případech).

Produkt	Účel	Vhodnost (doporučené x volitelné)
Change Request	Pro sledování požadavků na změny v rámci projektu včetně chyb.	Doporučeno pro naprostou většinu projektů
Configuration Audit Findings	Pro záznam výsledků konfiguračního auditu.	Volitelné. Většinou spojeno s více formálními procesy.
Configuration Management Plan	Popisuje veškeré úkoly řízení konfigurace a změny, které mají být realizovány během vývoje produktu či v rámci životního cyklu produktu.	Volitelné
Project Repository	Úložiště pro všechny verze souborů a složek projektu.	Doporučeno Je doporučeno používat verzování vývoje pro všechny projekty.
Workspace	Poskytuje individuální prostor jednotlivým členům týmu pro vývoj aniž by se provedené změny staly viditelné pro ostatní členy týmu.	Doporučeno pro většinu projektů

6 Vlastní zhodnocení

Jelikož současný vývoj softwaru probíhá v týmech, je potřeba nástrojů a řešení, která umožní kolaborativní vývoj bez větších obtíží. Proto je zde Configuration and Change Management, který přináší koncept a best practices v oblasti správy konfigurací a změn. Umožňuje udržovat si jasný přehled nad vývojem a pomáhá řídit vývoj správným směrem na základě vhodného zpracování požadavků.

Jelikož je v průběhu vývoje vytvořena řada verzí, které se relativně rychle (dle druhu projektu) mění a vyvíjejí, je potřeba udržovat přehled o těchto verzích a mít k nim i odpovídající konfigurace potřebných zařízení. Z tohoto důvodu je v podstatě nutné využívat konceptů uvedených v této práci a řídit správu verzí a konfigurací.

Ve své podstatě se jedná o nástroje a koncepty, které nám umožňují efektivní řízení vývoje softwarového produktu. Především formou jednoznačně nastavených pravidel a rolí v rámci rozvoje produktu.

Jedná se o metodiku rigorózní, proto je pojata velice široce, obsahuje velké množství dokumentů, přesto bychom mohli říci, že jich obsahuje jen nezbytně nutné množství druhů pro efektivní fungování v rámci projektu. Nicméně použitím této metodiky samozřejmě vznikne velké množství dokumentů, hlavně change requestů.

RUP nám tak nabízí ucelený framework, sadu best practices, kterou můžeme aplikovat do našich projektů a umožňuje nám si vybrat jen to, co je pro nás nejvhodnější. Stává se tak důležitým prvkem při vývoji softwarového produktu a řízení spolupráce.

Hodnotíme Configuration and Change Management jako důležitý prvek při tvorbě a rozvoji softwarového produktu a můžeme jej doporučit pro vývoj každého softwaru. Samozřejmě ve formě odpovídající velikosti a zaměření projektu.

7 Zdroje

IBM CORPORATION. *Classic RUP for SOMA*. 2010. Dostupné z:
<https://kitscm.vse.cz/RUP/LargeProjects/index.htm>

KEBRLOVÁ, Linda. *Change Management*. Praha, 2011. Diplomová práce. Vysoká škola ekonomická v Praze. Vedoucí práce doc. Ing. Alena Buchalcevová, Ph.D.