

RUP - disciplína Test

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

ZS 2011/2012

4IT421 – Zlepšení budování procesů IS

Antonín Kučera

Prosinec 2011

1 Úvod

Disciplína test metodiky RUP působí jako poskytovatel služeb pro ostatní disciplíny. Disciplína test se především zaměřuje na hodnocení a posuzování kvality vytvářeného softwaru. Využívá tyto základní postupy:

- Najít a zdokumentovat chyby v softwaru
- Poradenství ohledně kvality softwaru
- Ověření a dokázání předpokladů u oblasti návrhu konkrétními demonstracemi
- Ověření správné funkčnosti odpovídající návrhu
- Ověření správně prováděných požadavků

Zajímavým rozdílem mezi disciplínou test a ostatními je to, že test má za úkol najít a odhalit nedostatky softwarových produktů. V podstatě disciplína test funguje na trochu jiné filosofii než ostatní disciplíny z metodiky RUP. Dalo by se říci, že ostatní disciplíny zajímá kompaktnost, ale test zajímá právě nekompaktnost. [RUP06]

Test se řídí určitými otázkami, které bychom si měli pokládat:

- Jak by se mohl software narušit?
- V kterých situacích nebude software fungovat správně?

Z mnoha výzkumů a zpracovaných prací vyplývá, že testování ušetří náklady zhruba mezi 30 – 50%. S podivem tedy může být, že lidé nevěří správnému otestování před tím, než je jim dodán. Testování softwaru není jednoduchá záležitost, naopak velmi důležitá, avšak podceňovaná. Testování v metodice RUP je postaveno hlavně na průběžném ověřování kvality a iterativním vývoji. *Každý artefakt, aktivita nebo vedlejší produkt procesu má metodikou jasně stanovené kontrolní body* [NES05]. Každý člen týmu je zapojen do vývoje. Tím že disciplína test uplatňuje iterativní vývoj, je možné testovat software po celou dobu projektu. Při každé aktivitě jsou srozumitelně definované cíle, tím také vzniká velmi dobrý přehled pro jednotlivé účastníky. Každý má přehled o svých povinnostech a na jejich základě vytváří potřebné artefakty. Nepřetržitý přístup ke kvalitě vyvíjeného softwaru po celý jeho životní cyklus v mnoha případech snižuje náklady na dokončení a údržbu. Správně používaná disciplína test v kooperaci s ostatními disciplínami, výrazně snižuje riziko na vytvoření nekvalitního a chybného softwarového produktu.

2 Role

V disciplíně test působí 4 role testerů (Tester, Test analyst, Test designer, Test manager). Každá z těchto rolí má na starosti a je odpovědná za jiné části. Každá z rolí je důležitá. Záleží vždy na konkrétním řešení, jak budou role využívány. U větších projektů bývá 1 role obsazena jedním nebo více lidmi, avšak u menších projektů je možné, aby 1 člověk působil ve více rolích.

2.1 Tester

Jak už ze samotného názvu role vyplývá, tak hlavní náplní této role je provádění testů. Mezi činnostmi testera také patří evidování výsledků z provedených testů. Při obsazování této role je nutné vzít v úvahu potřebné dovednosti pro úlohy, které se budou vykonávat. Níže na obrázku č. 1 lze vidět aktivity, které jsou s rolí testera propojeny. [RUP06]



Obrázek 1 – role testera

2.1.1 Analyze Test Failure

Analýza selhání testu je v podstatě zkoumání testovacího logu a hledání chyb, kterým došlo při zavádění a provádění testu. Dále je důležité upravit nedostatky, které vyplynuly z chyb v testovacím postupu. Úkolem testera je také vhodně zaznamenat důležité poznatky. [RUP06]

Vstupy:

- Povinné
 - Testovací log, Testovací suite (soubor testů)
- Volitelné
 - Testovací strategie, Test case, Testovací skript, atd...

Výstupy:

- Artefakt – Change request neboli dokument s požadavky na změny

Jednotlivé kroky analýzy selhání testu

- Zhlédnutí testovacích logů
- Zachycení netriviálních incidentů
- Identifikace procesních chyb
- Lokalizace a izolace selhání
- Charakterizovat a diagnostikovat chyby
- Identifikace kandidáta řešení
- Vhodně zdokumentovat zjištění
- Hodnocení a ověřování výsledků

2.1.2 Execute test suite

Provedení testovací suity je úloha, která popisuje jak provést příslušnou sbírku testů, jak zachytit výsledky testů, které umožňují průběžné hodnocení. Než se testovací suite provede, je nutné uvést testovací prostředí do výchozího stavu dle požadavků. Dále se musí připravit testovací nástroje, jedná-

li se o automatizovaný test. U manuálního to záleží na tom, jestli jsou nějaké nástroje použity. Neméně důležité je v této úloze správné načasování provedení, aby nedocházelo ke konfliktům s ostatními testy. Po provedení suity se kontroluje testovací log, který je výstupem. Prostředí se vrací zpátky do výchozího stavu. [KUC06]

Vstupy:

- Povinné
 - Build, Test suite
- Volitelné
 - Iterační plán, Testovací data, Testovací plán, Vývojová infrastruktura (SW, HW...) atd.

Výstupy:

- Artefakt – Testovací log

Jednotlivé kroky provedení testovací suity

- Nastavení testovacího prostředí do výchozího stavu
- Nastavení možností nástroje
- Plán testů
- Provedení testovací suity
- Vyhodnocení provedení
- Zastavení testů
- Kontrola úplnosti a správnosti testovacích protokolů
- Obnova testovacího prostředí do výchozího stavu
- Sledovat vztahy, které vznikly
- Vyhodnocení a ověřování výsledků

2.1.3 Implement test

Implementace testu je v podstatě vytvoření artefaktů respektive testovacích skriptů. Cílem je vypracovat testy, které mohou být provedeny ve spojení s dalšími testy jako součást větší testovací infrastruktury. Volí se způsob implementace skriptu, což se odvíjí od typu testu. Jsou vesměs 2 možnosti buď manuální, nebo automatické testování. U manuálního testování se většinou vytváří textové instrukce, kterými se tester řídí. U automatického testování se tvoří programové instrukce (programování, makro), Po té se samotný test implementuje. Pomocí UI se určí určité pozorovací a kontrolní body ve který ch při testu probíhá potvrzení testovacích případů. Pokud je potřeba rozsáhlejšího testování, tak je vhodné vytvořit dataset, což je zdroj testovacích dat. [RUP06]

Vstupy:

- Povinné
 - Testovací data, List testovacích nápadů, Testovací strategie
- Volitelné
 - Build, Vývojová infrastruktura, Test case, Testovací skript atd.

Výstupy:

- Artefakt – Testovací skript

Jednotlivé kroky implementace testu

- Výběr vhodné implementační techniky
- Nastavení předpokladů testovacího prostředí
- Provedení testů
- Tvorba externích dat
- Kontrola implementace
- Obnovení testovacího prostředí do výchozího stavu
- Sledovat vztahy, které vznikly
- Vyhodnocení a ověřování výsledků

2.1.4 Implement test suite

Implementace testovací suity je vytváření kombinace testů, které poskytují lepší data než samotné testy. [RUP06]Důležité je tedy správně vybrat testy, které bude suite obsahovat, aby testování a výsledky byly co nejužitečnější. Zjištění návazností a sjednocení vstupních podmínek testů je podstatné. Neméně důležité je, aby suite byla implementována, tak že při krachu jednoho testu vydrží a „poběží“ dál.[KUC06]

Vstupy:

- Povinné
 - Build, Testovací strategie
- Volitelné
 - Vývojová infrastruktura, Test case, Testovací suit, Testovací data, Model implementace atd.

Výstupy:

- Artefakt – Testovací suite

Jednotlivé kroky implementace testovací suity

- Kontrola „kandidátů“ do testovací suity
- Kontrola souvisejících testů a cílů testů
- Zjištění závislostí
- Zjištění příležitostí pro znovupoužití
- Aplikování nezbytných infrastrukturních nástrojů
- Určení obnovovacích požadavků
- Zavedení obnovovacích požadavků
- Stabilizování testovací suity
- Sledovat vztahy, které vznikly
- Vyhodnocení a ověřování výsledků

2.2 Test analyst

Role testového analytika spočívá ve správném určení potřebných procesů. Sleduje podrobně testování a hodnotí celkovou kvalitu.



2.2.1 Define Assessment and Traceability Needs

Definice potřeb a zhodnocení závislosti popisuje jak definovat požadavky na zhodnocení a sledování závislostí. Zhodnocení udává informace o určitém stavu aspektu, který souvisí s projektem. Porovnává skutečný stav s tím požadovaným. Sledování závislostí eviduje procesní vztahy a vztahy, které vzniknou v čase.[KUC06]

Vstupy:

- Povinné
 - Iterační plán, Plán testů
- Volitelné
 - Konfigurační plán, Plán na zajištění kvality, Plán vývoje sw atd.

Výstupy:

- Artefakt – Testovací plán

Jednotlivé kroky definice potřeb a zhodnocení závislosti

- Identifikace hodnocení a sledování
- Zvážení omezení
- Zvážení možných strategií
- Diskuze možných strategií
- Definování strategie hodnocení
- Definování nástrojů
- Vyhodnocení a ověřování výsledků

2.2.2 Define Test Details

Definice detailů testu představuje analýzu a návrh testu. Tester tyto testy se v pozdější fázi implementuje. V podstatě by se dalo říci, že tato úloha popisuje to jak nápady na testování přenést do

testů. Seberou se testové nápady, které se vyhodnotí a vyberou se ty, které budou mít největší užitek. Důležité jsou pozorovací a kontrolní body.[RUP06]

Vstupy:

- Povinné
 - Testovací data, Testovací strategie, List testovacích nápadů
- Volitelné
 - Požadavky na změny, Test case, Testovací data, Use Case, Storyboard atd.

Výstupy:

- Artefakt – Testovací case, Testovací data, Testovací skript, Výpočtový model pracovní zátěže

Jednotlivé kroky definice detailu testu

- Kontrola cíle testu a souvisejících testovacích nápadů
- Výběr vhodných testovacích nápadů
- Návrh testu pro testovací nápad
- Definování požadovaných zdrojů dat, hodnoty a rozsahu
- Dostatek potřebných testovacích dat
- Sledovat vztahy, které vznikly
- Vyhodnocení a ověřování výsledků

2.2.3 Determine test results

Určení výsledků testu popisuje jak přesně zaznamenávat nálezy při testování a jejich sledování. Váže se na testový cyklus, což je spojeno s jedním buildem. V iteraci může být více cyklů. Prozkoumávají se nalezené chyby z logů. K těmto chybám se zajistí změnové požadavky. Hodnotí se ukazatele, například ukazatele pokrytí proběhlými testy. Hodnotí se kvalita produktu a největší rizika. Výsledky a hodnocení je prezentováno zadavateli.[RUP06]

Vstupy:

- Povinné
 - Testovací log, Testovací strategie, List testovacích nápadů
- Volitelné
 - Požadavky na změny, Test case, Testovací data, Use Case, Storyboard atd.

Výstupy:

- Artefakt – Testovací case, Výpočtový model pracovní zátěže

Jednotlivé kroky určení výsledků testu

- Zkoumání možných testovacích případů a jejich selhání
- Tvorba a správa požadavků na změny

- Analýza a hodnocení stavu
- Provedení zhodnocení aktuální kvality
- Zhodnocení rizik
- Vyhodnocení pokrytí testu
- Návrh hodnocení
- Informování stakeholderů o klíčových zjištěních
- Vyhodnocení a ověřování výsledků

2.2.4 Identify Targets of Test

Identifikace testových cílů slouží k nalezení prvku, který je třeba otestovat.[RUP06] Dle plánu iterace dojde ke zjištění, co se bude vyvíjet nebo co se bude měnit a podle toho se určí testy. Změny se sledují, což je základem regresního testování. Dále se identifikují sw prvky k otestování. Nutné zjistit jejich vazby na sw, hw, os apod.. Závěrem se vytvoří seznam cílů k otestování.[KUC06]

Vstupy:

- Povinné
 - Model závislosti, Implementační model, Iterační plán
- Volitelné
 - Datový model, Testovací strategie, Use Case – realizace, atd.

Výstupy:

- Artefakt – Testovací strategie

Jednotlivé kroky identifikace testových cílů

- Zjistit jaký sw bude implementován
- Identifikovat vhodné prvky k otestování
- Upřesnit seznam kandidátů cílových položek
- Definovat seznam cílových položek
- Vyhodnocení a ověřování výsledků

2.2.5 Identify Test Ideas

Identifikace testových nápadů. Dle metodiky RUP je testový nápad jako myšlenka vyjádřená formou *krátkého sdělení, popisující test, jehož provedení je potenciálně užitečné*. Nebývá to přesný popis, ale spíše obecný. Slouží jako inspirace skutečným testům. Slouží k nalezení cílových objektů případného testu a testových motivátorů, které „přinutí“ členy týmu k testování.[RUP06]

Vstupy:

- Povinné
 - Testovací strategie, Iterační plán
- Volitelné

- Datový model, Model závislostí, Model návrhu, Rozhraní, Storyboard, Plán testu, Use Case (model, realizace), atd.

Výstupy:

- Artefakt – List testovacích nápadů

Jednotlivé kroky identifikace testových nápadů

- Identifikace motivátorů a cílových objektů
- Zkoumání dostupných testovacích nápadů
- Brainstorming dalších nápadů
- List kandidátů na testovací nápady
- Upřesnění testovacích nápadů
- Sledovat vztahy, které vznikly
- Vyhodnocení a ověřování výsledků

2.2.6 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:

- Povinné
 - Build, Požadavky na změny
- Volitelné
 - Testovací log

Výstupy:

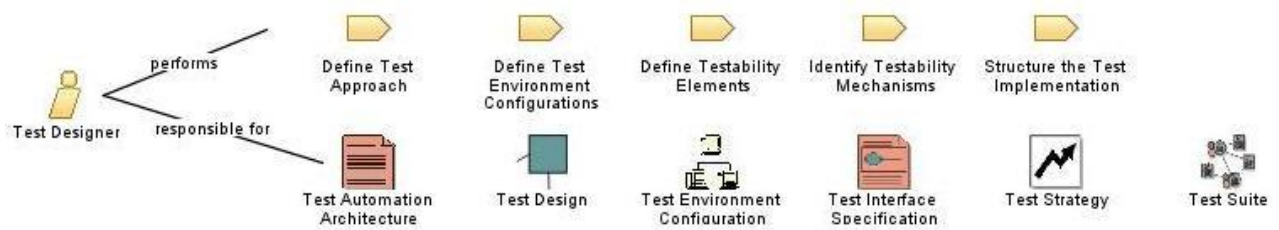
- Artefakt – Požadavky na změny, výsledek testu

Jednotlivé kroky

- Vyřešit požadavky na změny
- Kontrola změn v testovacím prostředí
- Ověření změn v repase build
- Ověřte a vyhodnotit výsledky

2.3 Test Designer

Tato role definuje testové přístupy a zajišťuje jejich realizaci. Zahrnuje určení vhodných metod, nástrojů a pokynů k provedení požadovaných zkoušek. Dále poskytnout rady při testování.



2.3.1 Define Test approach

Tato Definice přístupu k testování se zabývá výběrem technik, které se u testů používají. Definuje se testovací strategie a naznačuje se automatické testování architektury.[RUP06]

Vstupy:

- Povinné
 - Plán iterace, Vize, Dokumentace sw architektury
- Volitelné
 - Model nasazení, specifikace projektu, plán vývoje sw, plán testů, vize, use-case model apod...

Výstupy:

- Artefakt – Plán testů, testovací strategie, konfigurace testovacího prostředí

Jednotlivé kroky

- Prohlížení testových motivátorů a zkušební položky
- Prohlížení sw architektury
- Zvážení šíře a hloubky testů
- Identifikace zkušebních metod pro opakované využití
- Identifikace dalších technik
- Definice technik
- Nástin automatických testů architektury
- Definovat testovací strategii
- Přehled dostupnosti opakovaně dostupných aktiv
- Zachytě své poznatky
- Hodnotit a ověřovat výsledky

2.3.2 Define test environment configurations

Definice konfigurací testovacích prostředí se zabývá shromážděním požadavků na jejichž základě se konfiguruje testovací prostředí. Testové prostředí by se v maximální míře mělo podobat cílovému prostředí, ve kterém se bude provozovat daný produkt.[RUP06]

Vstupy:

- Povinné
 - Testovací strategie, Dokumentace sw architektury
- Volitelné

- Model nasazení, testovací data, testovací plán apod...

Výstupy:

- Artefakt – Konfigurace testovacího prostředí

Jednotlivé kroky

- Kontrola testovacího prostředí proti sw architektuře
- Identifikace každého konkrétního zaváděcího prostředí
- Konsolidovat seznam nezbytných prostředí
- Pro každé testovat konfigurační prostředí
- Hodnotit a ověřovat výsledky

2.3.3 Define testability elements

Definice prvků testovatelnosti má bližší souvislost s automatizovaným testováním. Identifikuje prvky, které slouží k podpoře testování.[KUC06] Dále identifikuje fyzické prvky testování realizace infrastruktury, které jsou nezbytné pro testování konfiguračního prostředí. Definují se požadavky programového vybavení.

Vstupy:

- Povinné
 - Testovací strategie, Dokumentace sw architektury
- Volitelné
 - Specifikace testovacího rozhraní, model nasazení, prototyp UI apod..

Výstupy:

- Artefakt – Automatický test architektury, Specifikace testovacího rozhraní

Jednotlivé kroky

- Pro každou požadovanou položku cílového testu identifikovat vztahy s testovacími mechanismy
- Identifikace dynamických prvků a událostí
- Určení hranic a rozhraní
- Identifikace prvků testovací infrastruktury
- Identifikovat testový specifický návrh
- Stanovené požadavků sw testovatelnosti
- Definice testovací infrastruktury
- Hodnotit a ověřovat výsledky

2.3.4 Identify testability mechanism

Identifikace mechanismů testovatelnosti popisuje jak identifikovat obecné mechanismy technického řešení. Tyto mechanismy jsou potřeba k usnadnění přístupu k testům. A jak naznačit obecný význam a hlavní znaky těchto mechanismů.[RUP06]

Vstupy:

- Povinné
 - Testovací strategie, Model návrhu

- Volitelné
 - Model nasazení, rozhraní, testovací případ, testovací data, automatický test architektury apod...

Výstupy:

- Artefakt – Automatický test architektury, Specifikace testovacího rozhraní, testovací návrh

Jednotlivé kroky

- Prohlédnou sw architektury a její cílové prostředí
- Identifikace mechanismů pro testování
- Inventarizovat zkušební mechanismy
- Definovat zkušební mechanismy
- Hodnotit a ověřovat výsledky

2.3.5

2.3.6 Structure the Test Implementation

Hlavním úkolem struktury implementace testů je definování celkové struktury pro test suitu. Struktura se stanoví a přidělí se odpovědnost pro danou testovací sadu a oblast provádění. Dále se uvedou požadované testy.[KUC06]

Vstupy:

- Povinné
 - Testovací strategie, testovací skript, testovací suite
- Volitelné
 - testovací případ, testovací data, test konfiguračního prostředí, testovací plán, apod...

Výstupy:

- Artefakt – Test skript, test suite

Jednotlivé kroky

- Kontrola zkušebnímu přístupu, položek cíle testu a hodnocení potřeb
- Prohlédnout testovatelnost mechanismů
- Tvorba počáteční struktury
- Přizpůsobení struktury, tak aby odrážela týmovou organizaci a nástroje omezení
- Identifikace testovacích mechanismů
- Definice počáteční závislosti mezi prvky test suity
- Vizually modelovat testování implementační architektury
- Upřesnění struktury test suity
- Udržovat sledování vztahů
- Hodnotit a ověřovat výsledky

2.4 Test manager

Role testovacího manažera řídí veškeré testové úsilí. Velmi důležitou vlastností testovacího manažera je, že dokáže prosadit, plánovat a řídit testy. Dokáže také řešit problémy, kterou vzniknou během testování.



2.4.1 Agree on the Mission

Souhlas neboli schválení s misí je aktivita, kde dochází k hledání správné rovnováhy mezi zdroji a cíli iterace. Cílem této aktivity je také vyjednat co nejlepší respektive nejefektivnější využití testovacích zdrojů pro každou iteraci. Dále také se dohodnout na vhodných a dosažitelných cílech a konečných výstupech pro iteraci.[RUP06]

Vstupy:

- Povinné
 - Testovací plán, plán iterace
- Volitelné
 - Požadavek na změnu, Vize, List rizik, Plán zajištění kvality (jakosti), plán vývoje sw, apod...

Výstupy:

- Artefakt – Testovací plán

Jednotlivé kroky

- Pochopit cíle iterace
- Zjistit možnosti rozsahu
- Možnosti zúčastněných stran
- Formulovat misi (poslání)
- Identifikovat výsledky testů
- Získ zúčastněných stran
- Hodnotit a ověřovat výsledky

2.4.2 Assess and Advocate Quality

Hodnocení a prosazení kvality je aktivita, která se zaměřuje především na celkovou podporu identifikace mezer kvality, posouzení jejich dopadů a rizik a nalezení vhodného řešení.[RUP06]
Sleduje pokrok, identifikuje vady, podporuje vhodné řešení pro zlepšení kvality sw.[KUC06]

Vstupy:

- Povinné
 - Souhrnné hodnocení testu, plán iterace

- Volitelné
 - Měření projektu, Plán zajištění kvality(jakosti), Výsledky testů , List rizik, apod...

Výstupy:

- Artefakt – Souhrnné hodnocení testu

Jednotlivé kroky

- prohlédnout aktuálních testů
- prohlédnou vybrané výsledky testů
- - || - klíčové požadavky na změny
- Určit mezery kvalit a hodnotit rizika
- - ||- určit základní opatření na řešení špatné kvality
- Identifikovat a diskutovat hlavní otázky
- Jednat o prioritách
- Monitorovat postup
- Potvrdit vhodné řešení klíčových otázek
- Hodnotit a ověřovat výsledky

2.4.3 Assess and Improve Test Effort

Zhodnocení a zlepšování testovacího úsilí je aktivita, která se zaměřuje na aktuální změny v testovacím úsilí a snaží se zvyšovat jeho součinnost. Hodnotí se produktivita, efektivita a úplnost.[RUP06]

Vstupy:

- Povinné
 - Testovací plán, plán iterace, Souhrnné hodnocení testu,
- Volitelné
 - Požadavky na změny, Měření projektu, Výsledky testů, Iterační hodnocení, apod...

Výstupy:

- Artefakt – Souhrnné hodnocení testu, Testovací plán

Jednotlivé kroky

- Zachytit stav práce
- Shromáždit testovací ukazatele produktivity a efektivity
- Shromáždit žádosti o změnu distribuce, vývoje, metriky...
- Shromáždit sledování, rozsah a závislost metrik
- Hodnotit ukazatele
- Záznam výsledků
- Průběžné hodnocení a získávání výsledků
- Plánování a realizace zlepšení iniciativy
- Sledovat a podporovat zlepšení iniciativy
- Hodnotit a ověřovat výsledky

2.4.4 Identify Test Motivators

Identifikace testovacích motivátorů popisuje jak identifikovat konkrétní seznam věcí včetně událostí a pracovních výrobků, které budou sloužit k motivaci testování v této iteraci. Výchozím krokem je se plán iterací a prozkoumání cílů iterace.[RUP06]

Vstupy:

- Povinné
 - Vize, plán iterace, sw požadavky, list rizik, dokument sw architektury
- Volitelné
 - Business case, seznam problémů, testovací plán, pracovní příkaz apod...

Výstupy:

- Artefakt – Testovací plán

Jednotlivé kroky

- Identifikace položek cílu iterace
- Shromažďovat a zkoumat související informace
- Identifikace motivátorů
- Určení rizik pro jakost
- Definovat seznam motivací
- Udržovat sledování vztahů
- Hodnotit a ověřovat výsledky

2.4.5 Obtain Testability Commitment

Získání záruk testovatelnosti je aktivita zaměřena na definování, stanovení priorit a podporu testovatelnosti. Podpora tvorby testovatelného softwaru, který podporuje potřeby úsilí testu. Dále je úkolem aktivity propagovat a podporovat užívání vhodných automatizačních technik a nástrojů.[RUP06]

Vstupy:

- Povinné
 - Testovací strategie, Seznam problémů
- Volitelné
 - Testování specifikace rozhraní, seznam problémů, plán vývoje sw, automatizace testování architektury, apod....

Výstupy:

- Artefakt – Testovací plán

Jednotlivé kroky

- Kontrola potřeby testovatelnosti
- Posouzení dopadů a priorit
- Definovat výhody testovatelnosti

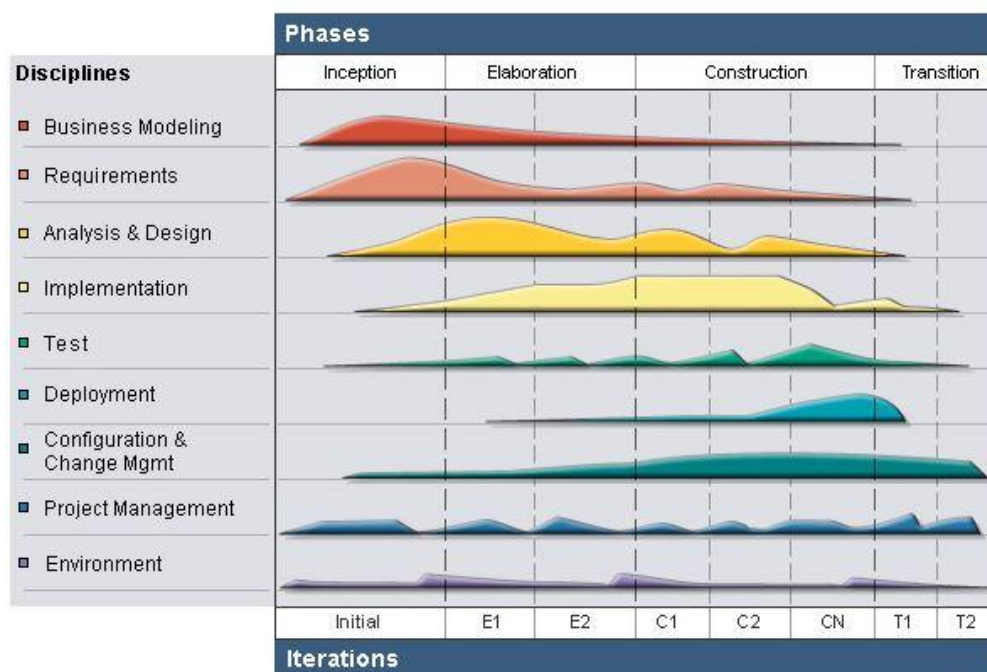
- Identifikovat a zapojit testovatelnost „mistrů“
- Podporovat testovatelnost potřeba a výhod
- Zvýšit odhodlání podporovat a udržovat testovatelnost
- Advokát řešení problémů testovatelnosti
- Hodnotit a ověřovat výsledky

3 Vztah k ostatním disciplínám

Disciplína testování není od ostatních disciplín oddělena žádným „osnatým drátem“, ale naopak je s některými disciplínami spjata. Níže uvedu několik příkladů:[KUC06]

- Konfigurační a změnový management kontroluje změny a testování zároveň ověřuje, jestli změny proběhly v pořádku.
- V implementaci se vytvářejí buildy. Během iterací se buildy testují
- Ve fázi požadavků se formulují požadavky na sw, které jsou dále vstupem mnoha činností testování
- Ve fázi předání se využívají beta testy a akceptační testy
- Atd...

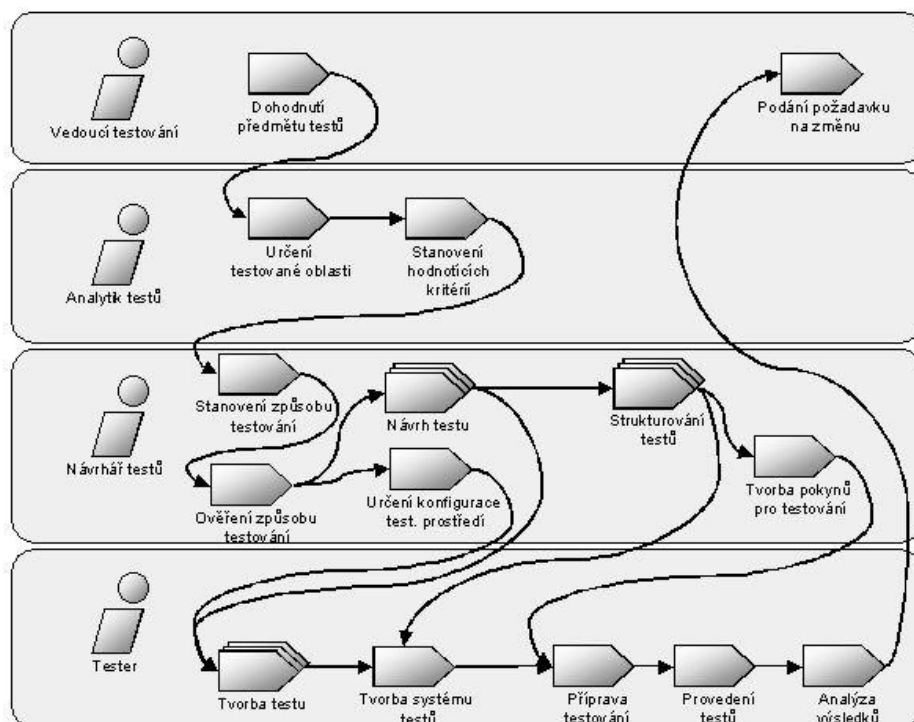
4 Fáze metodiky



5

Obrázek 2 - fáze metodiky, zdroj [RUP06]

6 Schémata návazností



Obrázek 3 - schema, zdroj [ALD05]

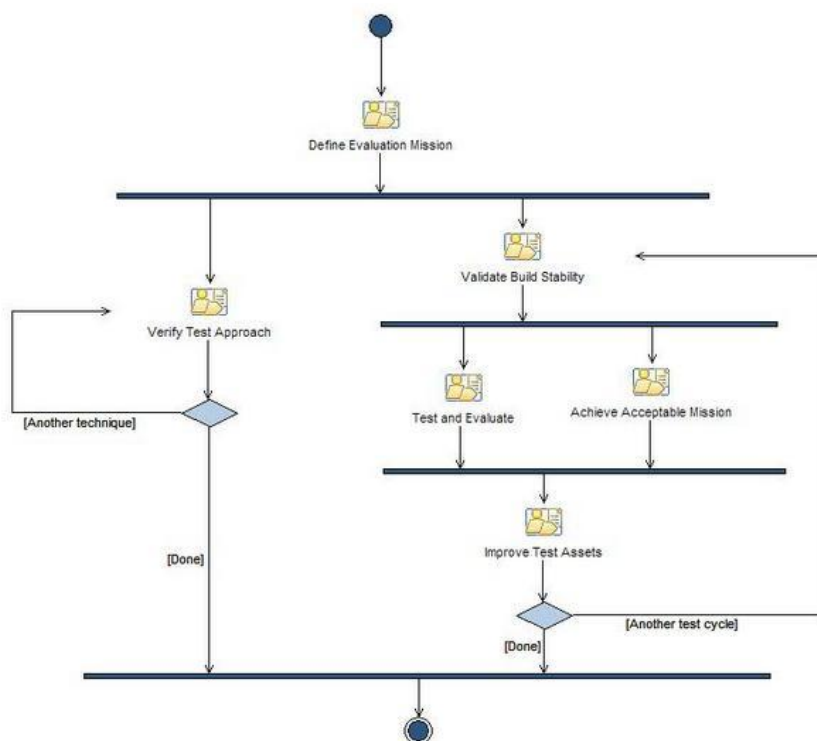
7 Zhodnocení

Před psaním této práce jsem metodiku RUP znal jen okrajově a moje ponětí o ní bylo spíše laické. Po zpracování této práce jsem se metodice trochu přiblížil. Náplní mojí práce byla disciplína test. Mnoho lidí by mohlo napadnout, že je to jen okrajová disciplína, která se využívá na konci projektu nebo v mimořádných případech. Mohu je ujistit, že tomu tak rozhodně není, jak je vidět z obrázku, který popisuje celkovou metodiku RUP. Disciplína test je součástí projektu de facto po celý životní cyklus a je reálně využívána v každé fázi projektu. Disciplína test není využívána jen jako „hasící“ nástroj v případě problémů, ale slouží jako preventivní opatření. Jedním z důvodů je to, že je daleko snazší chyby opravit v počátečních fázích, než kdybychom je odhalili na konci projektu nebo po jeho dokončení. Při prvním pohledu mi tato disciplína přišla velmi složitě popsaná ostatně jako celá metodika, ale při hlubším zkoumání jsem zjistil, že je naopak popsána velmi dobře a přehledně. Proto bych dle vlastního názoru metodiku RUP doporučil využívat jak u malých, tak i u velkých projektů.

V této práci jsem na metodiku RUP respektive na její disciplínu test pohlížel zejména z pohledu rolí. To znamená, že jsem v práci uvedl, jaké role se týkají této disciplíny, jaké činnosti vykonávají, co je při těchto činnostech třeba (vstup), co vzniká (výstupy) a jak se zhruba postupuje. Na závěr práce jsem uvedl pár schémat propojení jednotlivých rolí a artefaktů

8 Kompletní workflow

Obrázek níže zobrazuje workflow disciplíny test. Jednotlivé činnosti jsou popsány na obrázcích v přílohách tohoto dokumentu.



Obrázek 4 - workflow

9 Zdroje

[ALD05]

ALDORF, Filip. *Objekty - Objekty - Metodika RUP (Rational Unified Process)*

[online]. 2005, cit.18.10.2007. Dostupné z WWW:

<http://objekty.vse.cz/Objekty/RUP>

[KUC06]

KUČERA, M. *Testování softwaru a metodika RUP. Praha, 2006. 67 s. Bakalářská práce na Fakultě informatiky a statistiky Vysoké školy ekonomické na Katedře informačních technologií. Vedoucí bakalářské práce Ing. Martin Kosejk.*

[NES05]

NESS, Pete, THOMAS, Lee. *The Rational Unified Process for testers [online].*

2005, cit. 5.7.2006. Dostupné z WWW: [http://www-](http://www-128.ibm.com/developerworks/rational/library/04/r-3239/index.html)

[128.ibm.com/developerworks/rational/library/04/r-3239/index.html](http://www-128.ibm.com/developerworks/rational/library/04/r-3239/index.html)

[RUP06]

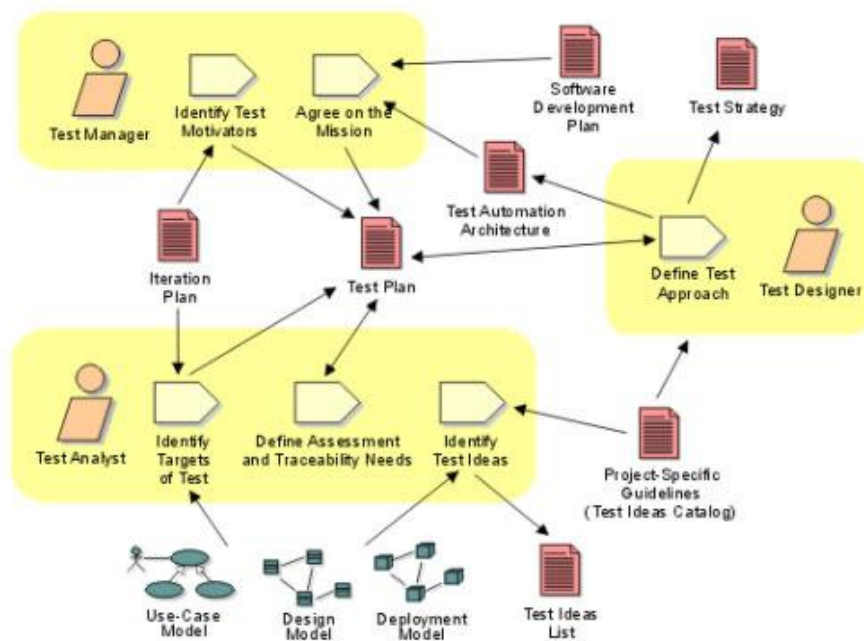
Rationl unified process [online]. 2006 [cit. 2011-12-08]. Dostupné z WWW: <http://kitscm.vse.cz/RUP>.

10 Seznam obrázků

Obrázek 1 – role testera.....	3
Obrázek 2 - fáze metodiky, zdroj [RUP06].....	17
Obrázek 3 - schema, zdroj [ALD05]	17
Obrázek 4 - workflow.....	18
Obrázek 5 - definování hodnotící mise, www.ibm.com.....	20
Obrázek 6 - dosažení misí, www.ibm.com.....	20
Obrázek 7 – ověření zkušebního přístupu, www.ibm.com.....	21
Obrázek 8 – testování a vyhodnocování, www.ibm.com	22
Obrázek 9 – ověření a vytvoření stability, www.ibm.com	23
Obrázek 10 – testovací prostředky, www.ibm.com.....	23

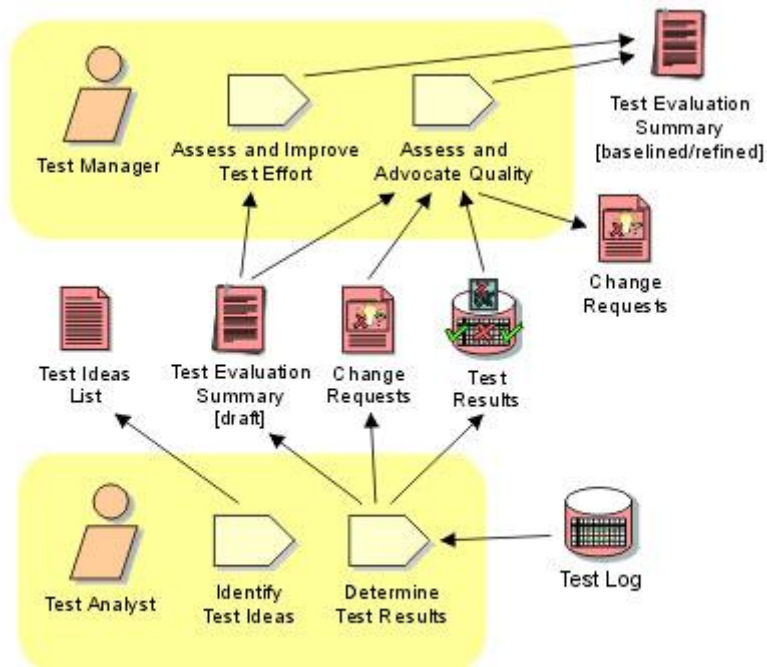
11 Přílohy

11.1 Příloha č.1 Definování hodnotící mise



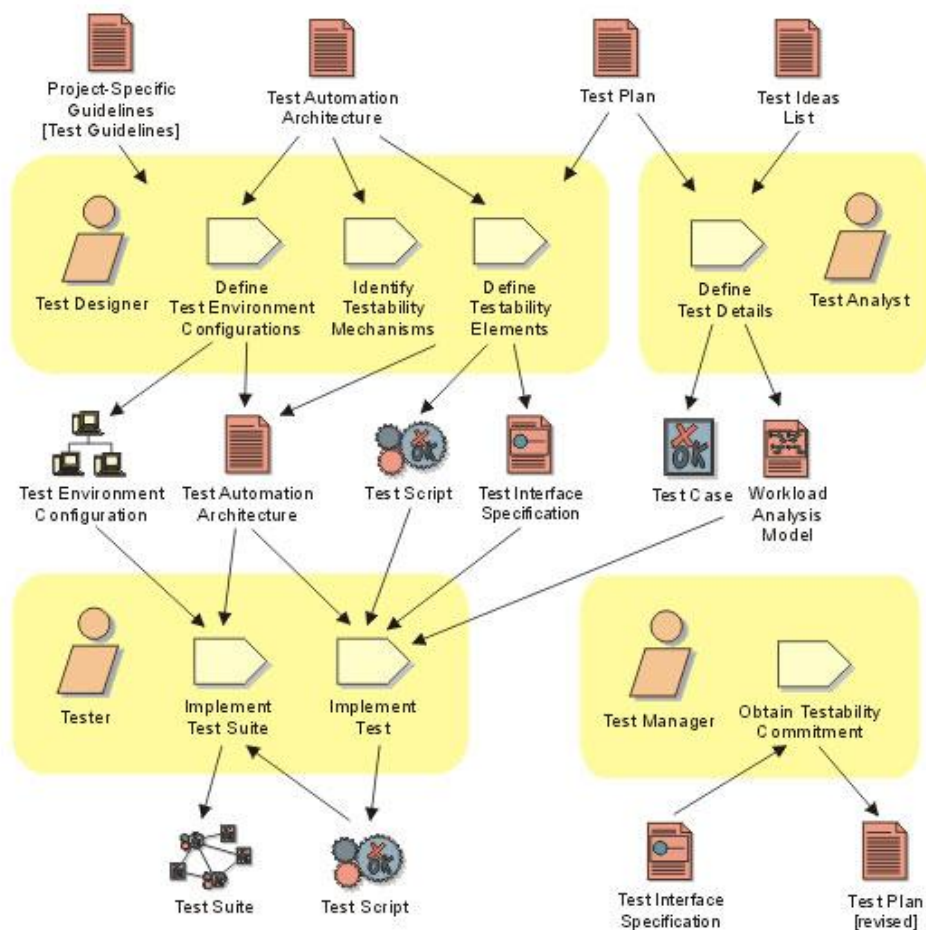
Obrázek 5 - definování hodnotící mise, www.ibm.com

11.2 Příloha č.2 - Dosažení přijatelných misí



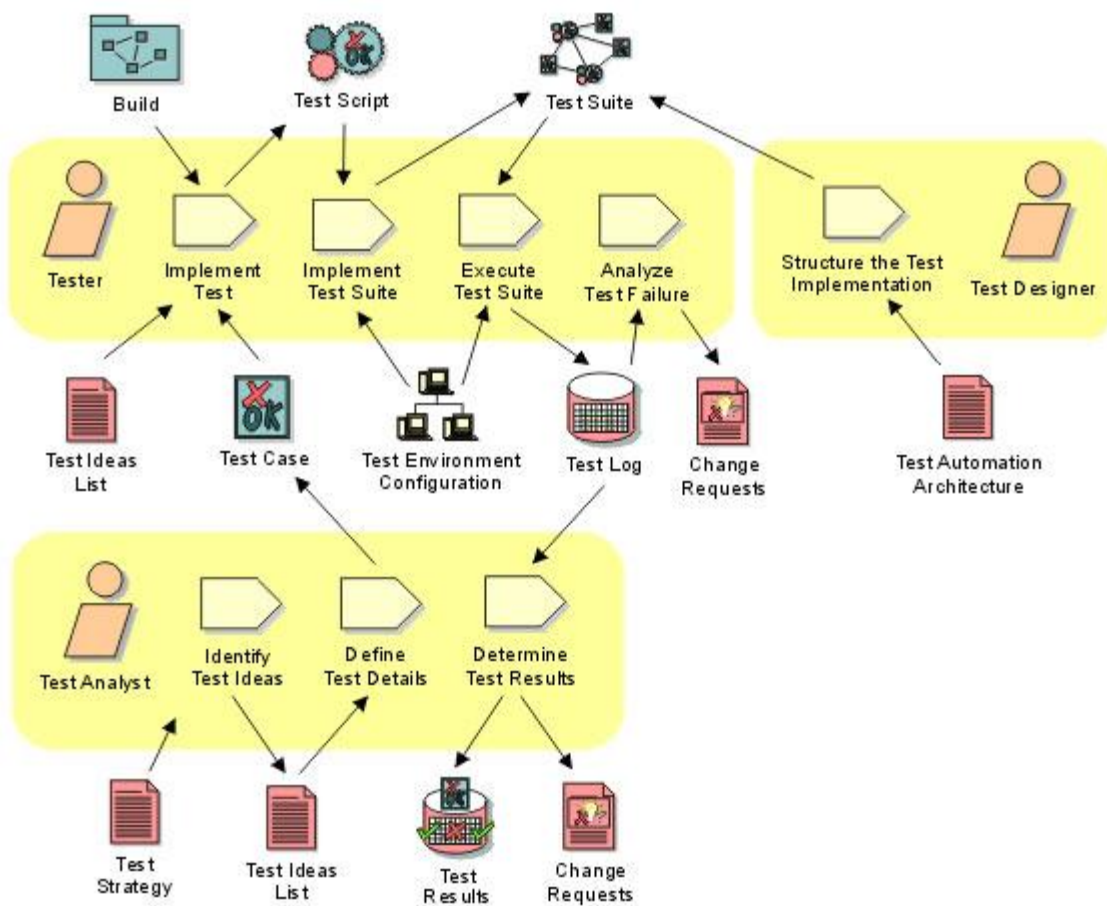
Obrázek 6 - dosažení misí, www.ibm.com

11.3 Příloha č.3 - Ověření zkušebního přístupu



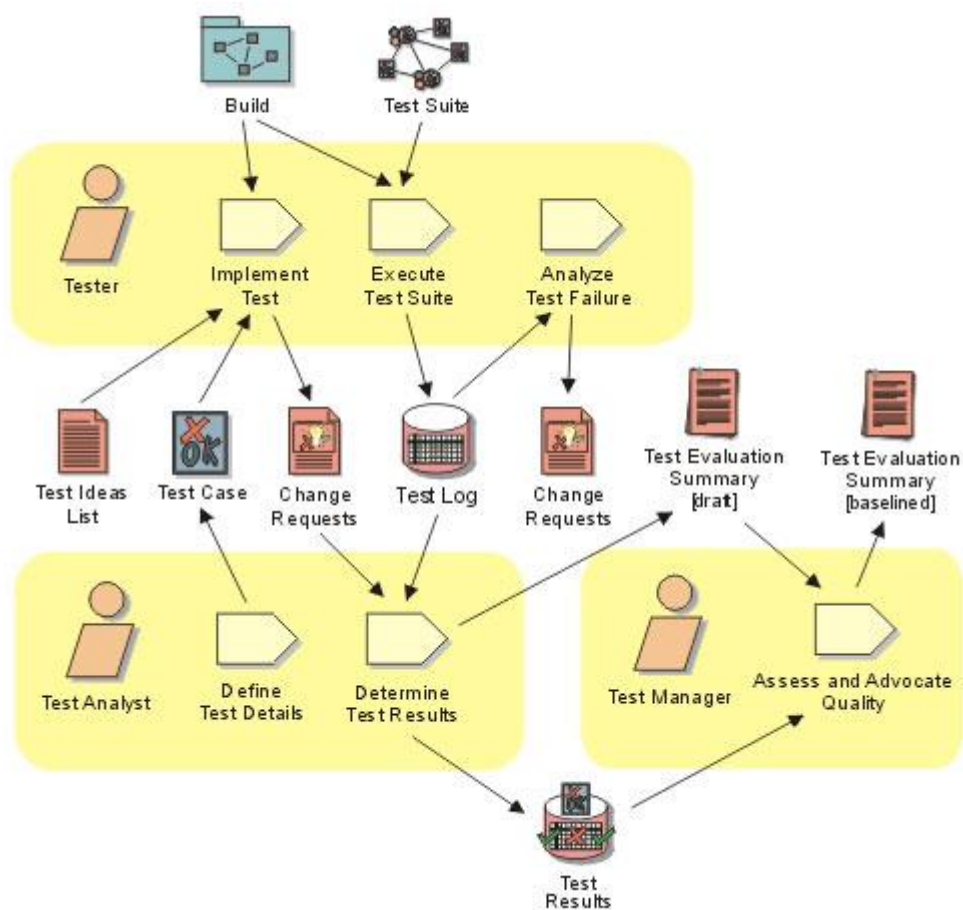
Obrázek 7 – ověření zkušebního přístupu, www.ibm.com

11.4 Příloha č.4 - Testování a vyhodnocování



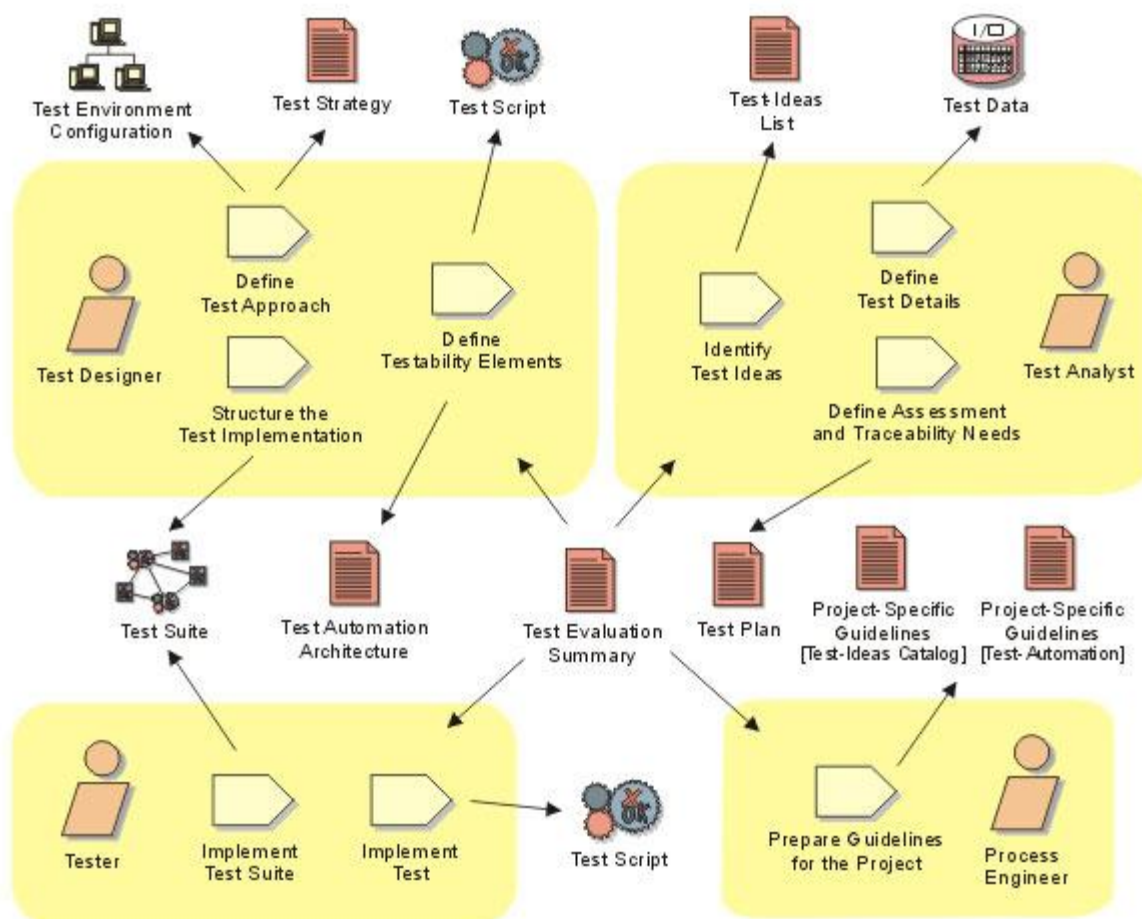
Obrázek 8 – testování a vyhodnocování, www.ibm.com

11.5 Příloha č.5 - Ověření a vytvoření stability



Obrázek 9 – ověření a vytvoření stability, www.ibm.com

11.6 Příloha č.6 - Testovací prostředky



Obrázek 10 – testovací prostředky, www.ibm.com