

**Vysoká škola ekonomická v Praze**

**Fakulta informatiky a statistiky**

**Katedra systémové analýzy**

**NÁVRH METODIKY  
KOMPETENČNÍHO CENTRA ICT  
PROVOZNÍCH PROJEKTŮ**

(Aplikace principů programme managementu při budování a provozování  
kompetenčního centra provozních projektů v bankovním sektoru)

**Doktorská disertační práce**

Doktorand : Ing. Petr Marounek  
Školitel : doc. Ing. Prokop Toman, CSc.  
Obor : Informatika

© 2007 Ing. Petr Marounek

marounek@vse.cz

při citaci uvádějte odkaz:

*Marounek, P. NÁVRH METODIKY KOMPETENČNÍHO CENTRA ICT PROVOZNÍCH  
PROJEKTŮ, disertační práce, VŠE-FIS, Praha, 2007*

Praha, únor, 2007

## Poděkování

Rád bych vyjádřil poděkování svému vedoucímu disertační práce panu doc. Ing. Prokopovi Tomanovi, CSc. za jeho trpělivost, nezbytnou odbornou a metodickou pomoc a ochotu, se kterou se mi věnoval.

Dále děkuji společnostem Unicorn Consulting, spol. s r.o., LogicaCMG, s.r.o. a Internal Business Machine (IBM) za možnost získat praktické zkušenosti v oblastech managementu a projektování informačních systémů.

Můj vděk patří i rodičům za bezmeznou podporu a trpělivost.

# Prohlášení

Prohlašuji, že jsem svou disertační práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

Praha, dne 17.12.2006

Ing. Petr Marounek

# Abstrakt

Cílem této disertační práce je navrhnout metodiku, která podpoří realizaci několika vzájemně provázaných ICT provozních projektů. Na základě tzv. dobrých praktik v odvětví, principů projektového a programového řízení, vlastních znalostí a zkušeností, vytvořil autor návrh metodiky kompetenčního centra ICT provozních projektů jako virtuálního útvaru společnosti, které má za cíl dodat takový typ projektů.

Autor této práce formuloval následující cíle:

- Na základě analýzy použitelnosti vybraných metodik vymezit provozní procesy (řídící, realizační, podpůrné a QA procesy) s použitím vstupů vybrané metodiky.
- Navržení procesu vytvoření a provozu kompetenčního centra na principech programme managementu.
- Definice vhodných indikátorů z pohledu vyhodnocení kvality služeb dodávaných v rámci kompetenčního centra.
- Definice oblastí, které autor doporučuje po vytvoření kompetenčního centra dohledovat, aby nebyla ohrožena kvalita dodávky projektů.
- Nastínění směru dalšího rozvoje disertační práce a jejich výsledků.

Klíčové závěry:

- Analýzou vybraných metodik dle definovaných kritérií (viz. kapitola 6 – Porovnání metodik), vyhodnocuje autor metodiku CORTEX jako nejvhodnější pro vytváření a provoz kompetenčního centra. Tuto metodiku upravuje o vlastní návrh konkrétních procesů a aktivit, identifikaci jejich výstupů a doporučení, jakým způsobem vytvářet kompetenční centrum.

- Analýzou vybraných metodik vyhodnocuje autor metodiku IBM Rational Unified Process (RUP) jako nejvhodnější vstup pro návrh jednotlivých provozních procesů kompetenčního centra provozních projektů. Vybrané procesy metodiky autor přepracoval nebo kompletně nově navrhl. Autor také identifikoval a vyprojektoval procesy dosud nepokryté metodikami.
- Návrh metodiky kompetenčního centra byl úspěšně vyzkoušen na vybraných mezinárodních projektech u zákazníka s pozitivním dopadem na dodávku těchto projektů.

Přínosy práce:

- Autor navrhuje vlastní design procesů (včetně jejich výstupů) následujícím způsobem:
  - o Řídící procesy - Řízení problémů (incidentů), Utilizace zdrojů, Řízení očekávání zákazníka, Ověřování kvality, Reportování, Účtování.
  - o Realizační procesy - Proces poskytování podpory, Životní cyklus požadavku, Proces poskytování podpory první úrovně, Proces poskytování podpory druhé úrovně, Analýza, Design, Implementace, Testování, Nasazení.
  - o Podpůrné procesy - Příprava a správa prostředí, Řízení konfigurací, Backup a archivace.
  - o QA procesy - Proces řízení kvality, Definice interních a externích indikátorů kvality.
- Identifikuje místa, která je vhodné aktivně monitorovat při přechodu od jednotlivých provozních projektů po jejich dodávku kompetenčním centrem (**autorův návrh**) s cílem zlepšit kvalitu a splnitelnost termínů.

- Zjištění, že ideu kompetenčního centra je vhodné použít na dodávku libovolných, spjatých projektů i mimo oblast ICT. Pro takový případ konstituční procesy a procesy řízení zůstanou stejné. Rozdíl je v realizačních, podpůrných a částečně procesech kvality, které musí být nově přizpůsobeny náplni centra. Při jejich designu je možné se inspirovat procesy a jejich návrhem, neboť autorův přístup a použitá notace diagramů je univerzální.
- Vědecké přínosy práce jsou detailně rozpracovány v kapitole Závěr.

## Struktura práce

- Cíl práce.
- Popis řešeného problému.
- Metodika práce.
- Analýza a vyhodnocení vybraných metodik.
- Aplikace programme managementu na KC.
- Návrh procesů KC.
- Přehled procesů.
- Závěr.

## Klíčová slova

- Kompetenční centrum.
- Programme management.
- IS Servisní projekt.
- IS Provozní projekt.
- Metodika.
- Návrh procesu.
- Design.
- Implementace.

- Dodávka servisních projektů.
- Indikátor kvality.
- Rational Unified Process.
- ITIL.
- COBIT.
- PMM.

# Abstract

Main objective of this work is to define the methodology, how to manage and deliver several ICT service and maintenance projects, which are dependent together. Author invented methodology of competence center (further KC) based mainly on the best practices and standards, which are currently used in ICT industry, project and programme management principles and on his knowledge and experience as well.

Author of this work formulated following objectives in detail:

- Based on results of “evaluation of usability of brand names methodologies” analysis to define production processes (management, production, supporting and QA processes) using inputs from selected methodology.
- Design of process of establishing and production of competence centre based on programme management principles.
- To define suitable indicators for measurement and evaluation of quality of delivered services by competence centre.
- To define critical areas, which have to be monitored after establishing of competence centre to prevent lower quality of delivery.
- Identifying of potential ways of development of this work and its results.

Key conclusions:

- Based on results of “evaluation of usability of brand names methodologies” analysis (see capture 6) author evaluates CORTEX methodology as the most suitable input for design of processes of establishing and production of competence centre. This input is extended by author about detailed design of key processes, activities, identification of their outputs and his own proposal, how to establish competence centre.



- Based on results of “evaluation of usability of brand names methodologies” analysis author evaluates IBM Rational Unified Process methodology as the most suitable input for design of each production processes of competence centre of service and maintenance projects. Selected RUP processes were completely redesigned or newly designed by author. Author also identified and designed missing processes as well.
- New methodology was tested on selected multinational customer ICT projects with positively affected results.

Contribution of this work:

- Author introduces his design of following processes (including their outputs):
  - o Management processes – Incident management, Resource utilization, Customer expectation management, QA, Reporting, Invoicing.
  - o Production processes – Process of service and maintenance providing, Requirement lifecycle, First line service and maintenance process, Second line service and maintenance process, Analysis, Design, Implementation, Test, Deployment.
  - o Supporting processes – Environment management, Configuration Management, Backup and archiving.
  - o QA processes – QA process, internal and external quality indicators definition.
- Identification of potential critical areas, which have to be monitored after establishing of competence centre to improve quality of delivery and schedule.

- Inventing, that concept of competence centre is reusable for delivery of arbitrary dependent projects, which can also be out of ICT area. In that case, constitutional and management process remains unchanged. The difference is in production, supporting and partly in QA processes, which have to be newly redesigned according to the scope of competence centre. The author's concept can be fully reused, because introduced access and used notation as well are universal.
- All contributions of this work are detaily described in the last chapter.

## Structure of the work

- Objective.
- Description of problem.
- Methodology.
- Analysis and evaluation of applicable methodologies.
- Applying of programme management principles to competence centre.
- Design of competence centre processes.
- Overview of processes.
- Conclusions.

## Key words

- Competence centre.
- Programme management.
- IS Service project.
- IS Support project.
- Methodology.
- Process design.
- Design.
- Implementation, Development.

- Delivery of service and maintenance projects.
- Quality Indicator.
- Rational Unified Process.
- ITIL.
- COBIT.
- PMM.

# Obsah:

<b>1. ÚVOD</b> .....	<b>18</b>
1.1. KLÍČOVÉ CÍLE PRÁCE.....	21
<b>2. PODNIK A INFORMAČNÍ SYSTÉM</b> .....	<b>23</b>
2.1. ÚVOD DO ŘÍZENÍ PODNIKU.....	23
2.1.1. <i>Globální strategie</i> .....	25
2.1.2. <i>Informační strategie</i> .....	26
2.1.3. <i>Model řízení podnikové informatiky - SPSPR</i> .....	26
2.2. ŽIVOTNÍ CYKLUS PROJEKTU IS.....	29
2.2.1. <i>Životní cyklus - Tunel</i> .....	32
2.2.2. <i>Životní cyklus - Vodopád</i> .....	32
2.2.3. <i>Životní cyklus – PMBOK</i> .....	33
2.2.4. <i>Životní cyklus - Prototyp</i> .....	35
2.2.5. <i>Životní cyklus - Spirálový vývoj</i> .....	35
2.2.6. <i>Životní cyklus - Iterativní vývoj</i> .....	36
2.2.7. <i>Životní cyklus - Extrémní programování a agilní techniky</i> .....	37
<b>3. POPIS ŘEŠENÉHO PROBLÉMU</b> .....	<b>38</b>
3.1. PROVOZNÍ PROJEKTY A JEJICH VLASTNOSTI.....	39
3.1.1. <i>Provoz a údržba osmi bankovních IS</i> .....	39
3.1.2. <i>Provoz IS mimo území ČR</i> .....	39
3.1.3. <i>Maximální „utilizace“ servisního týmu</i> .....	40
3.1.4. <i>Předdefinované požadavky na služby</i> .....	40
3.1.5. <i>Profitabilita projektů</i> .....	40
3.1.6. <i>Nespokojenost zákazníka s kvalitou dodávaných patchů (záplat)</i> .....	40
3.1.7. <i>Formalizace procesu testování</i> .....	41
3.1.8. <i>Formalizace procesu řízení požadavků</i> .....	41
3.1.9. <i>Aktualizace uživatelské dokumentace</i> .....	41
3.2. NÁVRH ŘEŠENÍ.....	41
<b>4. METODIKA DISERTAČNÍ PRÁCE</b> .....	<b>42</b>
4.1. NOTACE DIAGRAMŮ.....	44
<b>5. PROGRAMME MANAGEMENT</b> .....	<b>45</b>

<b>6. POROVNÁNÍ METODIK.....</b>	<b>47</b>
6.1. ÚVOD .....	47
6.1.1. Stav poznání řešené problematiky v rámci akademických institucí.....	47
6.1.2. Stav poznání řešené problematiky v rámci komerčních institucí .....	49
6.2. VÝBĚR KRITÉRIÍ PRO ANALÝZU METODIK .....	51
6.2.1. Kritéria pro oblast programme managementu.....	51
6.2.2. Kritéria pro oblast provozních projektů.....	51
6.3. METODIKA POPISU METODIK ICT .....	52
6.4. ITIL - INFORMATION TECHNOLOGY INFRASTRUCTURE LIBRARY [ITIL] .....	53
6.4.1. Úvod.....	53
6.4.2. Metamodel.....	54
6.4.3. Struktura ITIL .....	54
6.4.4. Proces - příklad.....	58
6.4.5. Implementace ITIL.....	59
6.4.6. Programme management .....	59
6.4.7. ITIL a realizace provozních projektů.....	64
6.4.8. Vyhodnocení aplikovatelnosti metodiky ITIL pro potřeby KC.....	65
6.5. COBIT- CONTROL OBJECTIVES FOR INFORMATION AND RELATED TECHNOLOGY ....	65
6.5.1. Úvod.....	65
6.5.2. Metamodel.....	66
6.5.3. Struktura COBIT.....	67
6.5.4. Proces – příklad.....	71
6.5.5. Implementace COBIT.....	71
6.5.6. Programme management .....	71
6.5.7. COBIT a realizace provozních projektů.....	72
6.5.8. Vyhodnocení aplikovatelnosti metodiky COBIT pro potřeby KC.....	72
6.6. CORTEX [COR].....	73
6.6.1. Úvod.....	73
6.6.2. Metamodel.....	73
6.6.3. Struktura CORTEX .....	73
6.6.4. Popis procesních oblastí [COR] .....	73
6.6.5. Proces - příklad.....	75
6.6.6. Implementace CORTEX .....	77
6.6.7. CORTEX a programme management.....	77
6.6.8. CORTEX a realizace provozních projektů .....	77

6.6.9.	<i>Vyhodnocení aplikovatelnosti metodiky</i> .....	78
6.7.	<b>PROJECT MANAGEMENT METODOLOGY – PMM [PMM]</b> .....	78
6.7.1.	<i>Úvod</i> .....	78
6.7.2.	<i>Metamodel</i> .....	80
6.7.3.	<i>Struktura PMM</i> .....	80
6.7.4.	<i>PMM a programme management</i> .....	86
6.7.5.	<i>Metodika a realizace provozních projektů</i> .....	86
6.7.6.	<i>Vyhodnocení aplikovatelnosti metodiky</i> .....	87
6.8.	<b>RATIONAL UNIFIED PROCESS – RUP [RUP]</b> .....	87
6.8.1.	<i>Úvod</i> .....	87
6.8.2.	<i>Metamodel</i> .....	89
6.8.3.	<i>Základní struktura procesů</i> .....	90
6.8.4.	<i>Ukázka procesu</i> .....	91
6.8.5.	<i>Metodika a programme management</i> .....	92
6.8.6.	<i>Metodika a realizace provozních projektů</i> .....	92
6.8.7.	<i>Vyhodnocení aplikovatelnosti metodiky</i> .....	92
6.9.	<b>POROVNÁNÍ METODIK</b> .....	93
6.10.	<b>VYHODNOCENÍ METODIK</b> .....	95
6.11.	<b>VYHODNOCENÍ METODIK (ZADÁNÍ PRÁCE) – PROGRAMME MANAGEMENT</b> .....	95
6.12.	<b>VYHODNOCENÍ METODIK (ZADÁNÍ PRÁCE) – PROVOZNÍ PROJEKTY</b> .....	96
<b>7.</b>	<b>APLIKACE PROGRAMME MANAGEMENTU</b> .....	<b>97</b>
<b>8.</b>	<b>NÁVRH KOMPETENČNÍHO CENTRA PROVOZNÍCH PROJEKTŮ</b> .....	<b>99</b>
8.1.	<i>CÍLE</i> .....	100
8.2.	<i>HARDWARE A SOFTWARE</i> .....	101
8.3.	<i>ORGANIZAČNÍ STRUKTURA</i> .....	102
8.4.	<i>PROCESY</i> .....	104
8.4.1.	<i>Návrh procesu Poskytování podpory</i> .....	105
<b>9.</b>	<b>AUTORŮV NÁVRH ŘÍDÍCÍCH PROCESŮ KC</b> .....	<b>111</b>
9.1.	<i>ŘÍZENÍ PROBLÉMŮ (INCIDENTŮ)</i> .....	111
9.2.	<i>VYUŽITÍ ZDROJŮ</i> .....	112
9.3.	<i>ŘÍZENÍ OČEKÁVÁNÍ ZÁKAZNÍKA</i> .....	113
9.4.	<i>OVĚŘOVÁNÍ KVALITY</i> .....	114
9.5.	<i>AUDITOVÁNÍ</i> .....	115

9.6.	MANAGEMENT KC .....	116
9.6.1.	Agenda a Zázpis z jednání .....	116
9.6.2.	Registr rizik.....	116
9.6.3.	Přehled požadavků.....	117
9.6.4.	Plán projektu.....	117
9.6.5.	Plán kompetenčního centra.....	117
9.7.	REPORTOVÁNÍ .....	118
9.7.1.	Hodnocení projektu interní .....	118
9.7.2.	Hodnocení projektu externí.....	118
9.8.	ÚČTOVÁNÍ .....	119
9.8.1.	Finanční stav projektu .....	119
9.8.2.	Vydané faktury .....	119
9.9.	SPECIFICKÉ AKTIVITY ŘÍZENÍ PROVOZNÍCH PROJEKTŮ .....	121
9.9.1.	Delegace práce a monitorování průběhu realizace .....	123
9.9.2.	Řízení subdodavatelů .....	123
9.9.3.	Obnova kontraktu.....	123
9.9.4.	Měření spokojenosti zákazníka .....	124
9.9.5.	Plánování kontroly projektu.....	124
9.9.6.	Risk management .....	124
<b>10.</b>	<b>AUTORŮV NÁVRH REALIZAČNÍCH PROCESŮ KC .....</b>	<b>125</b>
10.1.	PROCES POSKYTOVÁNÍ PODPORY - REALIZACE .....	125
10.1.1.	Provoz a podpora – první úroveň .....	128
10.1.2.	Provoz a podpora – druhá úroveň .....	130
10.2.	ANALÝZA .....	132
10.2.1.	Analýza a design dle metodiky RUP .....	132
10.2.2.	Návrh procesu analýza .....	133
10.3.	DESIGN .....	137
10.3.1.	Design dle metodiky RUP .....	137
10.3.2.	Návrh procesu design pro KC.....	138
10.4.	IMPLEMENTACE .....	140
10.4.1.	Implementace dle metodiky RUP .....	140
10.4.2.	Návrh procesu implementace pro KC.....	141
10.5.	TESTOVÁNÍ.....	144
10.5.1.	Testování dle metodiky RUP.....	144
10.5.2.	Návrh procesu testování pro KC.....	146

10.6.	NASAZENÍ.....	149
10.6.1.	<i>Nasazení dle metodiky RUP.....</i>	<i>149</i>
10.6.2.	<i>Návrh procesu nasazení pro KC.....</i>	<i>150</i>
<b>11.</b>	<b>AUTORŮV NÁVRH PODPŮRNÝCH PROCESŮ KC.....</b>	<b>153</b>
11.1.	PŘÍPRAVA A SPRÁVA PROSTŘEDÍ DLE METODIKY RUP.....	153
11.2.	NÁVRH PROCESU SPRÁVA PROSTŘEDÍ PRO KC.....	154
11.3.	ŘÍZENÍ KONFIGURACÍ DLE METODIKY RUP.....	156
11.4.	NÁVRH PROCESU ŘÍZENÍ KONFIGURACÍ PRO KC.....	157
11.5.	BACKUP A ARCHIVACE DLE METODIKY RUP.....	159
11.6.	NÁVRH PROCESU BACKUP A ARCHIVACE PRO KC.....	159
11.7.	ŘÍZENÍ KVALITY DLE METODIKY RUP.....	161
11.8.	NÁVRH PROCESU ŘÍZENÍ KVALITY PRO KC.....	162
11.8.1.	<i>Ověření procesů z pohledu kvality.....</i>	<i>163</i>
11.8.2.	<i>Ověření výstupů z pohledu kvality.....</i>	<i>163</i>
11.8.3.	<i>Definice a kontrola použití indikátorů.....</i>	<i>163</i>
11.8.4.	<i>Indikátory.....</i>	<i>163</i>
11.8.5.	<i>Doporučení pro implementaci procesů.....</i>	<i>169</i>
<b>12.</b>	<b>PŘEHLED NAVRŽENÝCH PROCESŮ.....</b>	<b>171</b>
<b>13.</b>	<b>VYUŽITÍ KONCEPTU KOMPETENČNÍHO CENTRA MIMO PROJEKTY ICT</b>	<b>173</b>
<b>14.</b>	<b>ZÁVĚR.....</b>	<b>174</b>
<b>15.</b>	<b>ZKRATKY A VYSVĚTLIVKY.....</b>	<b>180</b>
<b>16.</b>	<b>SEZNAM POUŽITÝCH ZDROJŮ.....</b>	<b>186</b>
16.1.	PUBLIKACE.....	186
16.2.	ELEKTRONICKÉ ZDROJE A INTERNET:.....	189
<b>17.</b>	<b>PŘÍLOHA Č. 1 – SEZNAM OBRÁZKŮ.....</b>	<b>191</b>



# 1. Úvod

Vzrůstající komplexita požadavků uživatelů a penetrace ICT technologií do života každého jedince, vyžadují stále rozsáhlejší, komplexnější řešení, která jsou dodávána pomocí projektů. Tyto projekty již nejsou doménou několika osamělých jedinců, kteří se rozhodují ad hoc nebo pouze na základě vlastní zkušenosti. Pro zaručení úspěchu jsou projekty realizovány pomocí ucelené množiny pravidel, metodiky.

Metodiky, které jsou dnes používány ve světě vývoje software zahrnují pravidla, doporučení a návody, jak dodat jeden izolovaný projekt – jeden informační systém od formulace a analýzy požadavků zákazníka až po jejich realizaci a nasazení do provozu. Co tyto metodiky zpravidla nezahrnují, nebo pouze v omezené míře, je programme management neboli dodání několika vzájemně propojených projektů jednomu zákazníkovi. Oblast, kterou metodiky naprosto ignorují (alespoň nejznámější, nejvíce používané a autorem této práce zkoumané), je vlastní každodenní provoz a další rozvoj již zákazníkem rutinně používaných informačních systémů. Efektivní realizace (z pohledu času, nákladu, využití lidských zdrojů atd.) provozu informačních systémů je strategickou záležitostí v dnešní době, kdy podnikové procesy jsou nedílně spjaté s podnikovou informatikou a na kvalitě jejího fungování přímo závislé.

Tato práce mapuje základní principy programme managementu pro účely vytvoření kompetenčního centra pro dodávku provozních projektů ICT. Na základě studia vybraných metodik (ITIL, Cobit, RUP, Cortex, PMM a dalších) pojednává autor o programme managementu a jeho praktické aplikaci při řešení problému vybudování kompetenčního centra pro řízení a realizaci provozních projektů nad bankovními IS.

Aplikací uvedených principů je výsledné zjištění, že metodiky zpravidla poskytují rady a doporučení pouze na vysoké, obecné úrovni, které nejdou příliš do hloubky. To je nevýhoda těchto metodik, neboť detailní pohled na činnosti a výstupy v rámci kompetenčního centra a jejich specifikace, jsou počátečním klíčem k úspěchu. Toto si dovoluje tvrdit autor této práce nejen na základě své zkušenosti z realizace ICT projektů, ale i výsledků své diplomové práce, jejímž zjištěním bylo, že správně nastavené pracovní procesy podpořené vhodným intranetovým informačním systémem přináší podniku konkurenční výhodu a to nejen v podobě efektivního sdílení informací [dle MAROUNEK 2002].

Práce popisuje vztah podniku a informačního systému pohledem přes jednotlivé úrovně strategie řízení IS. Dále zachycuje vybrané přístupy v oblasti životních cyklů projektu.

V rámci této disertační práce autor zkoumal řešení provozu a rozvoje IS ve vybraných společnostech, nicméně není oprávněn k tomu, aby publikoval seznam zkoumaných společností ani zjištěné výsledky.

Autor navazuje na principy a procesy programme managementu a rozšiřuje je o jím definované provozní procesy kompetenčního centra. Ty zahrnují nejen manažerské přístupy, ale i procesy, jak např. opravovat chyby v produkčním prostředí a jak tyto opravy následně promítnout do procesu správy verzí.

Pojem kompetenční centrum je široce používán nejen v oblasti ICT pro útvar, který má svůj cíl, rozpočet, zdroje a metodiku fungování. Procesy spjaté s realizací provozních projektů nad IS dosud nejsou pokryty žádnou metodikou. Tato práce zachycuje proces vytváření kompetenčního centra na vybraných principech programme managementu. Metodika provozu kompetenčního centra pro řízení a realizaci provozních projektů nad bankovními IS je vlastním výsledkem autorovy práce.

Tato disertační práce také identifikuje specifické vlastnosti provozních ICT projektů, jejich rizika a vybraná ekonomická a technologická omezení.

Postup vytvoření kompetenčního centra pro jiné, než-li provozní projekty (např. vývojové projekty) je definován v této práci. Vnitřní procesy, které jsou potřebné pro kompetenční centrum a dosažení výsledků, je třeba vyprojektovat analogickým způsobem jako je tomu v této práci pro potřeby provozních projektů.

Postup vytváření kompetenčního centra a notace pro zachycení jeho vnitřních procesů je univerzální a plně nezávislá na typech projektů realizovaných kompetenčním centrem. Autor se domnívá, že uvedený přístup je použitelný i pro řešení problematiky mimo oblast ICT.

Pro potřeby kompetenčního centra provozních projektů, mapuje tato práce vybranou nezbytnou množinu pracovních postupů, které je třeba vyprojektovat, jejich činnosti a výstupy pro splnění cílů, pro které KC vzniklo. Procesy byly projektovány nejen s ohledem na jejich realizovatelnost v praxi, ale i s přihlédnutím k ekonomické efektivitě navrženého řešení. Projektování procesů je vlastním výsledkem autorovy práce.

Rámec projektování pracovních postupů je omezen vnitřními předpisy a požadavky instituce (ve které je KC pod autorovým vedením vytvářeno), požadavkem zákazníka na množství a kvalitu výstupů, manažerských přehledů a neporušením a neohrožením re-certifikace pro již získané ISO a TickIT certifikáty kvality.<sup>1</sup>

Kromě nezbytných řídicích a pracovních procesů, byl autorem vytvořen životní cyklus řízení požadavků zákazníka. Při definici procesu byly zohledněny termíny dodání záplat (patchů) a nových verzí IS dle předem smluvně dohodnutých SLA.

Vzhledem k tomu, že servis je poskytován na bankovní informační systém, kterým denně protéká objem financí v ceně několika miliard korun a který je přímo napojen na clearing národních bank, není překvapením, že SLA pro případ hlášení a opravu problému na produkčním prostředí definuje, že je třeba začít řešitelské práce na uvedeném incidentu do 2 hodin od jeho nahlášení.

---

<sup>1</sup> Ačkoli byla re-certifikace prováděna v pilotní fázi ověřování disciplín managementu a řízení požadavků zákazníka, nadnárodní auditor konstatoval, že nejsou porušena žádná pravidla a že dosud vytvořená metodika je v souladu s uvedenými normami.

Pro zabezpečení spokojenosti zákazníka s výsledky činností autor věnuje pozornost procesu řízení kvality (Quality Assurance Process) a disciplíně testování.

Nedílnou součástí této práce je návrh indikátorů, procesu jejich měření a to jak s cílem měření kvality procesů, výstupů tak i spokojenosti zákazníka.

Jelikož je systém instalován na několika místech v Evropě a existuje plán jeho rozšiřování i na kontinent Jižní Ameriky, otevírá se otázka globalizace tohoto konceptu kompetenčního centra, která není řešena touto prací.

Výstupem této práce je navržená metodika vytváření a fungování kompetenčního centra. Metodika vytváření kompetenčního centra reflektuje vybrané principy programme managementu metodiky CORTEX. Pro návrh realizačních procesů autor provedl průzkum nejznámějších metodik ICT. Výsledkem analýzy (viz. Kapitola 6 této práce) je zjištění, že metodiky nepokrývají oblast realizace provozních projektů. Vybrané procesy metodiky autor přepracoval nebo kompletně nově navrhl. Autor také identifikoval a vyprojektoval procesy metodikami dosud nepokryté.

Návrh metodiky kompetenčního centra pro realizaci provozních projektů je výstupem této práce.

Autor této práce si je vědom možných rozdílů v interpretaci použité ICT terminologie. Jelikož tato terminologie není obecně standardizována a také metodika kompetenčního centra je postavena na konceptech a principech metodiky Rational Unified Process, použil autor v této práci její terminologii. Klíčové termíny jsou uvedeny v kapitolách 6.8., 15 anebo přímo v metodice RUP[RUP].

## **1.1. Klíčové cíle práce**

Autor této práce formuloval následující cíle:

- Na základě analýzy použitelnosti jednotlivých metodik vymezit provozní procesy (řídící, realizační, podpůrné a QA procesy) s použitím vstupů z vybrané metodiky.
- Navržení procesu vytvoření a provozu kompetenčního centra na principech programme managementu.

- Definice indikátorů vhodných pro sběr a vyhodnocení kvality služeb dodávaných v rámci kompetenčního centra.
- Definice oblastí, které je třeba po uvedení konceptu kompetenčního centra dohledovat, aby nebyla ohrožena kvalita dodávky projektů.
- Nastínění směru dalšího rozvoje zaměření disertační práce a jejich výsledků.

## 2. Podnik a informační systém

Tato kapitola popisuje pohled na podnik a jeho podnikání s cílem zachytit posloupnost formulace vize a strategického řízení podniku až po vytvoření, dodávku a provoz informačního systému. Tato posloupnost je důležitá, neboť zobrazuje místo, ve kterém jsou realizovány provozní projekty. Navržení realizace provozních projektů pomocí kompetenčního centra je cílem této disertační práce.

### 2.1. Úvod do řízení podniku

Řízení podniku na strategické úrovni je klíčovým posláním podnikových manažerů. Definice strategie, její zachycení, měření a provádění korektivních opatření s cílem dosáhnout co nejlepších výsledků, jsou založeny na dobře zvládnutých teoretických postupech, přístupech a pravidlech, ale také i na osobní zkušenosti manažera.

Pojem strategie je definován např. ve slovníku cizích slov, který jej definuje jako „posloupnost kroků vedoucích k předem definovaným cílům“, [LINHART 2002].

Definice strategie dle Kennetha Andrewse: "Corporate strategy is the pattern of decisions in a company that determines and reveals its objectives, purposes, or goals, produces the principal policies and plans for achieving those goals, and defines the range of business the company is to pursue, the kind of economic and human organization it is or intends to be, and the nature of the economic and non-economic contribution it intends to make to its shareholders, employees, customers, and communities“, [ANDREWS 1987].

„Podniková strategie je rozhodovací předlohou ve společnosti, která zachycuje cíle, hlavní politiky a plány, jak dosáhnout uvedených cílů. Také definuje rozsah podnikání společnosti ve vztahu k ekonomické a lidské organizaci, jaký aktuálně je a jaký by měl být a také ekonomické a neekonomické přínosy akcionářům, zaměstnancům, zákazníkům a komunitám“, [ANDREWS 1987].

Definice strategie dle Michaela Portera: „Strategy is about being different. It means deliberately choosing a different set of activities to deliver a unique mix of value“, [PORTER 1986].

„Strategie je o tom být jiný. To v praxi znamená promyšlený výběr množiny činností s cílem zabezpečit dodání unikátní hodnoty“, [PORTER 1986].

Při definici strategie je obecně používáno těchto pět základních kroků:

1. Definice podnikání, nastavení cílů, formulace vize.
2. Nastavení měřitelných cílů.
3. Vytvoření souladu mezi strategií a cíli.
4. Implementace a realizace strategie.
5. Vyhodnocení přínosů strategie, definice a implementace korektivních opatření.

Hlavními výstupy této úrovně řízení podniku jsou [dle VORISEK 2001/2] následující rozhodnutí:

- Jaké cíle a priority bude podnik sledovat.
- Jaké produkty/služby a jakému okruhu zákazníků bude podnik poskytovat.
- Do jakých aliancí a kooperačních vztahů podnik vstoupí. Které kompetence a zdroje podnik v kooperaci uplatní a které naopak očekává od svých partnerů.
- Jak na hrubé úrovni budou probíhat materiálové a informační toky mezi partnery v řetězci.
- Jaké lidské, znalostní/informační a finanční zdroje budou v podniku zapotřebí a jak budou získány (a přebytečné uvolněny).
- Jaké metriky budou použity k měření stupně dosažení cílů.

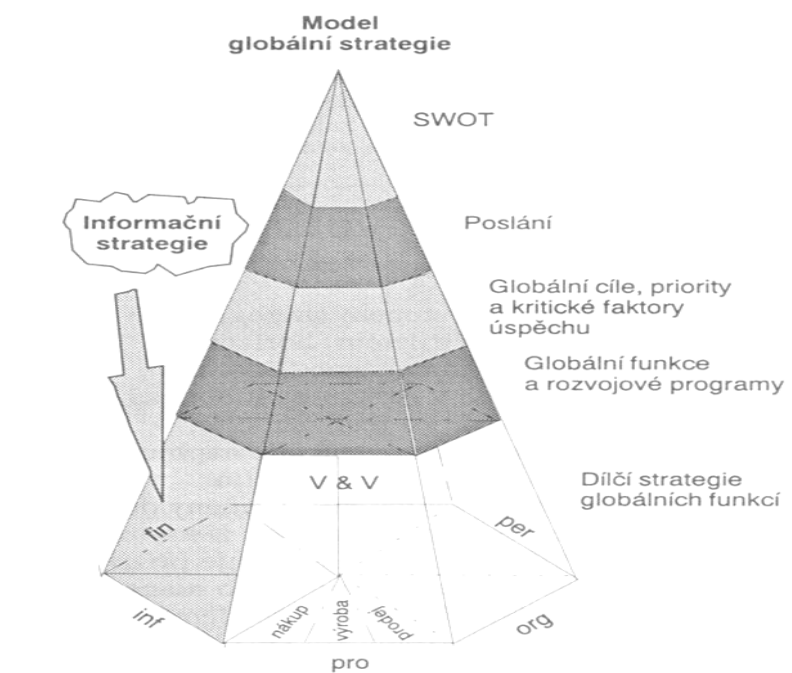
Konkrétní strategii podnikání zachycuje globální strategie podniku.

### 2.1.1. Globální strategie

Globální strategie podniku (GST) zachycuje klíčové vize a koncepce podnikání. Je strukturována na analýzu dosavadního stavu, návrh budoucího stavu a transformaci přechodu mezi stavy. GST primárně stanovuje hlavní směry, priority a cíle rozvoje podniku.

Postup formulace globální strategie:

- SWOT analýza podniku - (strengths, weaknesses, opportunities, threats).
- Definice poslání podniku.
- Definice globálních podnikových cílů (management, marketing, nákup, výroba a služby, prodej, výzkum a vývoj, HR, finance, logistika, informatika).
- Strategie globálních podnikových funkcí.



Obrázek 1: Model globální strategie [dle VORISEK 1997]



Při vyhodnocování globální strategie je kladen důraz především na následující aspekty:

- Míra splnění podnikových cílů.
- Ověření indikátorů vzhledem k jednotlivým cílům.
- Identifikace klíčových akcí a situací, které vedly k zásadním úspěchům a neúspěchům.
- Revize strategie vůči očekávaným změnám hospodářského prostředí.

Na GST navazují další strategie. Pro potřeby této práce je významná informační strategie.

### **2.1.2. Informační strategie**

Informační strategie (IST) navazuje na globální strategii. Jejím posláním je podpořit koncepci podnikání formulovanou v GST pomocí ICT. Analogicky jako GST i IST je strukturována na analýzu dosavadního stavu, návrh budoucího stavu a transformaci přechodu mezi stavy.

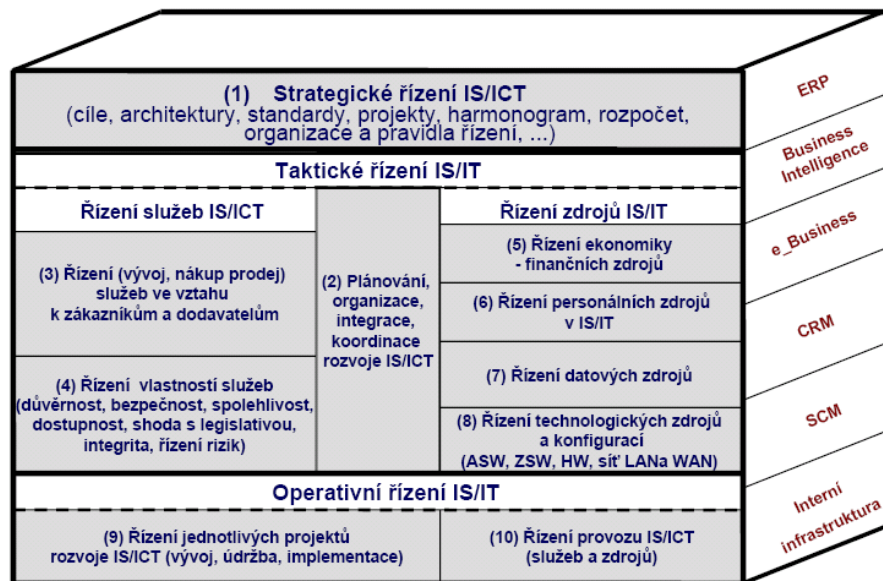
Postup formulace informační strategie:

1. Plánování informační strategie.
2. Převzetí a ověření závěrů globální strategie.
3. Odvození a formulace vize ICT.
4. Reengineering podnikových procesů.
5. Reengineering ICT.

Informatické potřeby plynoucí z IST jsou realizovány formou ICT projektů.

### **2.1.3. Model řízení podnikové informatiky - SPSPR**

Oblasti řízení podnikové informatiky zachycuje následující schéma:

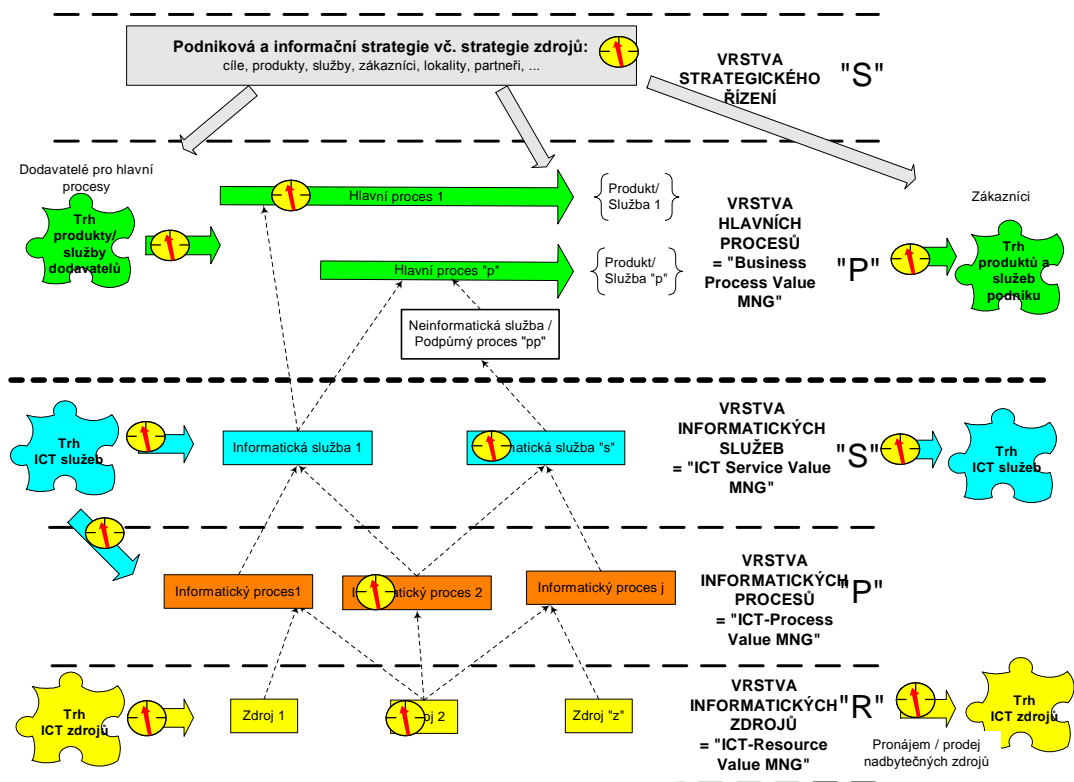


**Obrázek 2: Oblasti řízení ICT [POUR 1996]**

Model řízení – SPSPR (S - Strategy, P - Business Processes, S – ICT Services, P - ICT Processes, R – ICT Resources) definuje vztah mezi řízením podnikových procesů a řízením podnikových technologií. Model je strukturován do pěti vrstev.

Cílem rozdělení řídicích aktivit do pěti vrstev (úrovní) je [dle VORISEK 2001]:

- Taková struktura podnikových činností a zodpovědností, která optimálně odpovídá současným požadavkům na flexibilní a efektivní podnikové řízení.
- Jasně určené zodpovědnosti různých typů manažerů/specialistů v podniku.
- Zprůhlednění způsobu dekompozice podnikových cílů až na úroveň řízení provozu ICT.
- Vytvoření schématu, ze kterého je možné odvodit vhodné metriky úspěšnosti jednotlivých typů procesů a za ně odpovědných manažerů – viz místa označená budíkem.



Obrázek 3: Model SPSPR [VORISEK 2001]

Smysl a cíle své existence naplňuje organizace prostřednictvím hlavního předmětu podnikání, tj. prostřednictvím hlavních podnikových procesů. Úlohou druhé vrstvy řízení – tj. vrstvy hlavních a podpůrných procesů - je navrhnout a řídit podnikové procesy tak, aby organizace dosáhla strategických cílů definovaných první úrovní řízení. Hlavními aktivitami této úrovně řízení jsou: a) definice a optimalizace podnikových procesů, b) operativní řízení procesů a kapacit, c) monitoring procesů a d) realizace (vykonávání) procesů [Scheer, 1999], [Voříšek, 2000].

Zde se dostáváme k významné charakteristice modelu SPSPR, která reaguje na současný trend v rozdělování pravomocí mezi business a ICT manažery [ROSS 2003].

Manažer zodpovědný za definici a optimalizaci podnikového procesu je zodpovědný za navržení procesu (jednotlivých činností, jejich návaznosti, zodpovědností za jednotlivé činnosti atd.) tak, aby proces produkoval konkurenceschopný produkt/službu v optimálním čase, objemu a kvalitě s přijatelnými náklady. K měření efektivnosti procesu a tedy i k efektivnosti práce tohoto manažera mohou sloužit metriky typu: objem prodané produkce/služby, zisk z prodeje produktu/služby atd. Součástí návrhu podnikového procesu musí být i návrh takových informatických služeb, které budou optimálně podporovat příslušný proces. Tím je explicitně vyjádřena zodpovědnost manažera (vlastníka) podnikového procesu za „objednaný rozsah a objednanou kvalitu“ informatických služeb, [VORISEK 2000].

Zabezpečení informatických služeb pro podnikové procesy je zpravidla realizováno pomocí IS projektů.

## **2.2. Životní cyklus projektu IS**

Pojem projekt definuje norma ČSN ISO 10006:2004 Systémy managementu jakosti Směrnice pro management jakosti projektů jako „jedinečný proces sestávající z řady koordinovaných a řízených činností s daty zahájení a ukončení, prováděný pro dosažení cíle, který vyhovuje specifickým požadavkům, včetně omezení daných časem, náklady a zdroji.“

Projekt také definuje např. PMBOK: „A project is a temporary endeavour undertaken to create a unique product, service or result“, [PMBOK].

"Projekt je dočasné úsilí s cílem vytvořit unikátní produkt nebo službu", [PMBOK].

Projekt je tedy definován svým cílem, časem a rozpočtem. Na rozdíl od liniové činnosti (což je činnost, která nemá pevně stanovený začátek a konec a je realizována opakovaně), lze identifikovat následující typické vlastnosti projektů:

- Projekt je unikátní – má definovaný začátek, konec.
- Měřitelný.
- Kontrolovatelný.
- Definované zdroje – lidské, rozpočtové.

- Rizika a omezení.

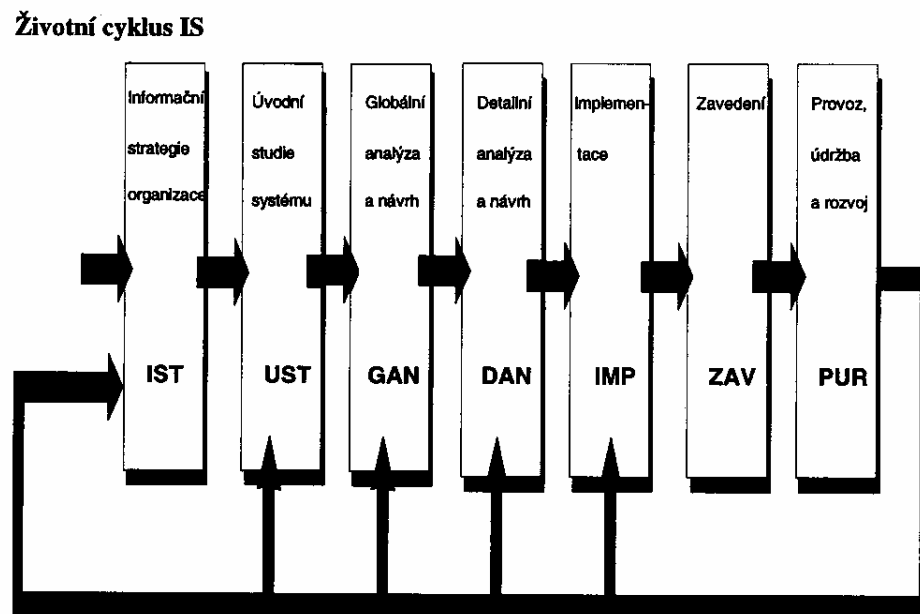
V rámci ICT projektů lze rozlišit několik typů projektů (Marounkova typologie ICT projektů):

- Vývojový projekt – cílem je dodat zákazníkovi nový informační systém, aplikaci.
- Konzultační projekt – strategické, manažerské a odborné konzultace v oblasti ICT.
- Integrovaný projekt – cílem je integrovat propojit informační systémy na různé úrovni.
- Školící projekt – náplní projektu je připravit školení a vyškolit určitý objem uživatelů.
- Provozní, servisní projekt – cílem je zabezpečit údržbu a rozvoj informačního systému, který je nasazen a provozován v produkčním prostředí.
- Infrastrukturní projekt – cílem posílení výkonu nebo optimalizace HW.

Mezi vývojovým a provozním projektem je zásadní rozdíl. V rámci vývojového projektu je aplikován některý z životních cyklů projektů a na jeho konci je dodán informační systém (vybrané životní cykly projektu jsou pospány v následujících kapitolách) zákazníkovi. Co vývojový projekt zpravidla neřeší, jsou věci spjaté s nasazením výstupu do produkčního prostředí (nasazení na pobočky, vyškolení uživatelů, postupný náběh atd.), neboť většinou životní cyklus skončí u výroby verze produktu a dokumentace.

Životní cykly vývojových projektů vůbec neřeší, jak na nasazeném produktu v produkčním prostředí opravovat identifikované chyby a realizovat nové požadavky zákazníka. Toto je posláním provozních projektů. Ze závěrů kapitoly 6, ve které autor této práce zkoumal vhodnost známých metodik a nejlepších praktik ve vztahu k realizaci provozních projektů plyne, že tyto metodiky bohužel nepokrývají životní cyklus dodávky servisních projektů ani nedávají doporučení, jak tyto projekty realizovat.

Vztahem životního cyklu IS a metodik se zabýval Václav Řepa v [REPA 1999], kde názorně rozděluje životní cyklus do několika etap.



**Obrázek 4: Řepův životní cyklus IS projektů [REPA 1999]**

U každé etapy životního cyklu IS V. Řepa stanovuje [viz REPA 1999]:

- Cíl etapy.
- Účel a obsah etapy.
- Předpoklady zahájení etapy.

- Kritéria ukončení etapy.
- Klíčové dokumenty etapy.
- Kritické faktory etapy.
- Činnosti etapy.
- Návaznosti činností v etapě.

### 2.2.1. Životní cyklus - Tunel

Princip metody tunelu je založen na tom, že po zahájení projektu projekt postupuje do neznáma – tunelu, [COAD 1991]. Na základě aktuálních vstupů a celkové situace projektu jej vedení řídí na operativní úrovni. Úkolem vedení je dohlédnout na konec tunelu a stanovit takový postup, aby byl projekt zdárně ukončen.



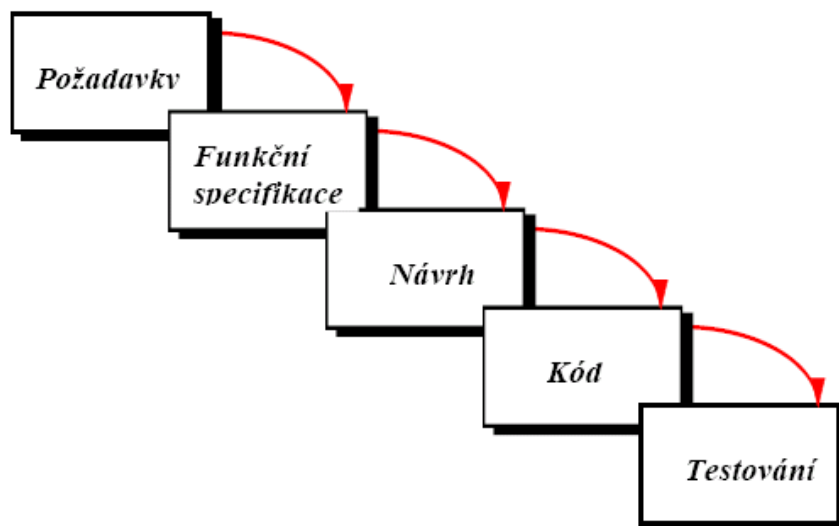
**Obrázek 5: Řízení projektu tunelovým způsobem [COAD 1991]**

Zřejmou nevýhodou této metody, kromě nejistoty, kdy a jak projekt skončí, je také nízká kvalita výsledku – dodaného řešení. Výsledek je zatížen velkou chybovostí a velkým množstvím nedodělků.

### 2.2.2. Životní cyklus - Vodopád

Metoda vodopád rozděluje životní cyklus projektu dle jednotlivých disciplín softwarového vývoje. Jednotlivé etapy jsou odděleny milníky. Životní cyklus získal svůj název proto, že výstupy z jedné fáze se předávají jako vstup do druhé fáze a to odshora dolů – viz. obr.6.

Nevýhodou tohoto přístupu je znemožnění realizace změn v průběhu dodávky IS, které jsou dnes zákazníci velmi požadované.



Obrázek 6: Životní cyklus vodopád

### 2.2.3. Životní cyklus – PMBOK

Životní cyklus PMBOK je rozdělen do 4 hlavních fází [dle PMBOK]:

- Proveditelnost.
- Plánování a návrh.
- Zavedení a spuštění.
- Uzavření.

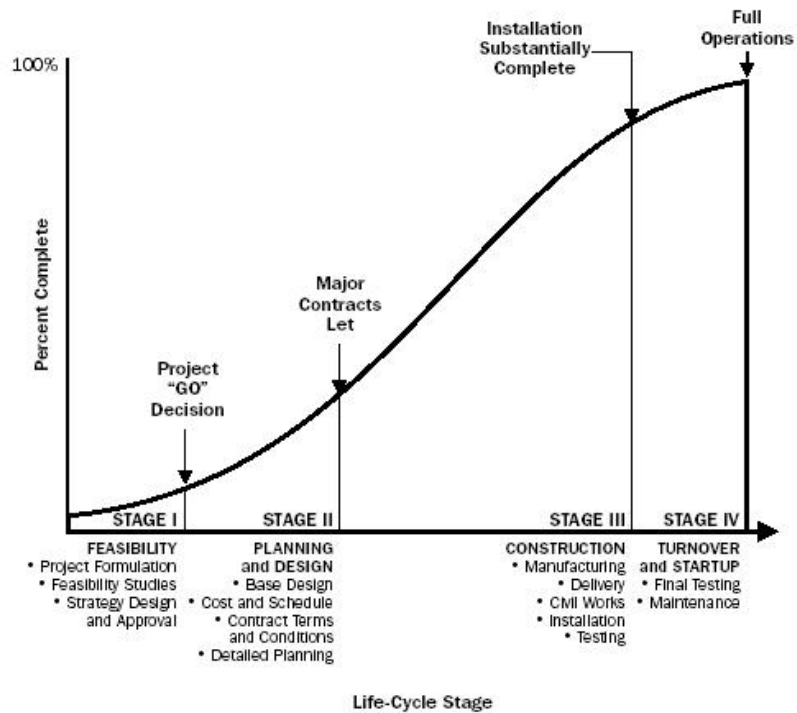
Dle PMBOK “Project life cycle generally defines:

- What technical work to do in each phase.
- When the deliverables are to be generated in each phase and how each deliverable is reviewed, verified and validated.
- Who is involved in each phase.
- How to control and approve each phase“, [PMBOK].



Dle PMBOK "Životní cyklus projektu obecně definuje:

- Jakou činnost je třeba realizovat v každé fázi.
- Kdy mají být vytvořeny výstupy projektu v rámci fáze a jak má každý výstup být revidován, ověřen a odsouhlasen.
- Kdo je ovlivněn v jaké fázi.
- Jak kontrolovat a odsouhlasit každou fázi", [PMBOK].



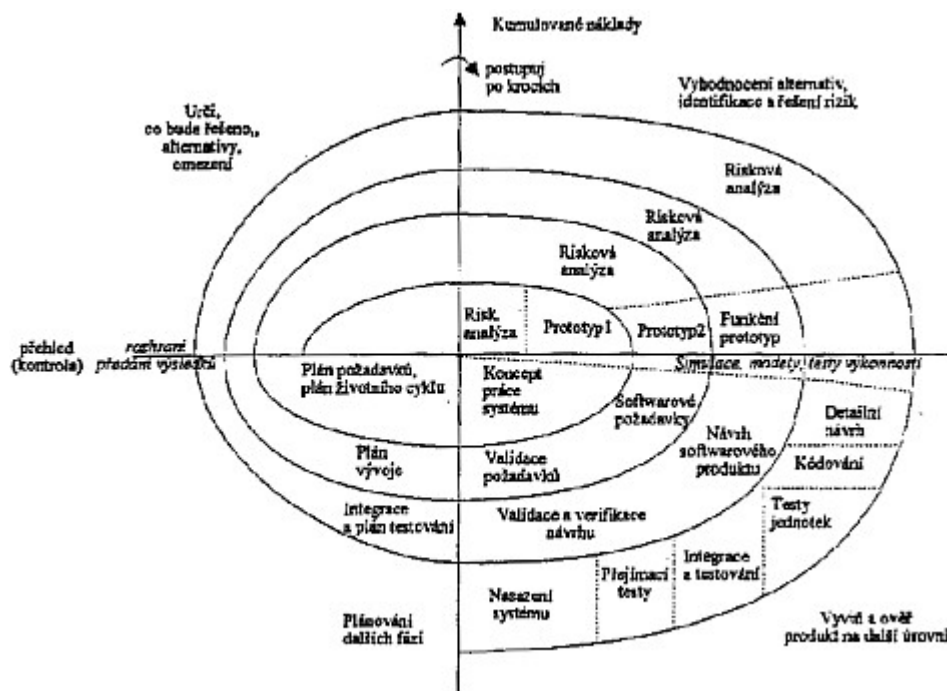
Obrázek 7: Životní cyklus PMBOK [PMBOK]

## 2.2.4. Životní cyklus - Prototyp

Filosofie tohoto životního cyklu je založena na co nejdřívější dodávce uživatelům konkrétního návrhu řešení – prototypu. Tito uživatelé prototyp testují a doplňují, zpřesňují zadání. Několik týmů paralelně pracuje na několika verzích prototypu a požadavky zapracovávají. Některé prototypy jsou pak použity, jiné ne. Zřejmou nevýhodou tohoto přístupu je vysoký náklad na pořízení výsledku, velkou výhodou je, že výsledný produkt zpravidla splní očekávání koncových uživatelů.

## 2.2.5. Životní cyklus - Spirálový vývoj

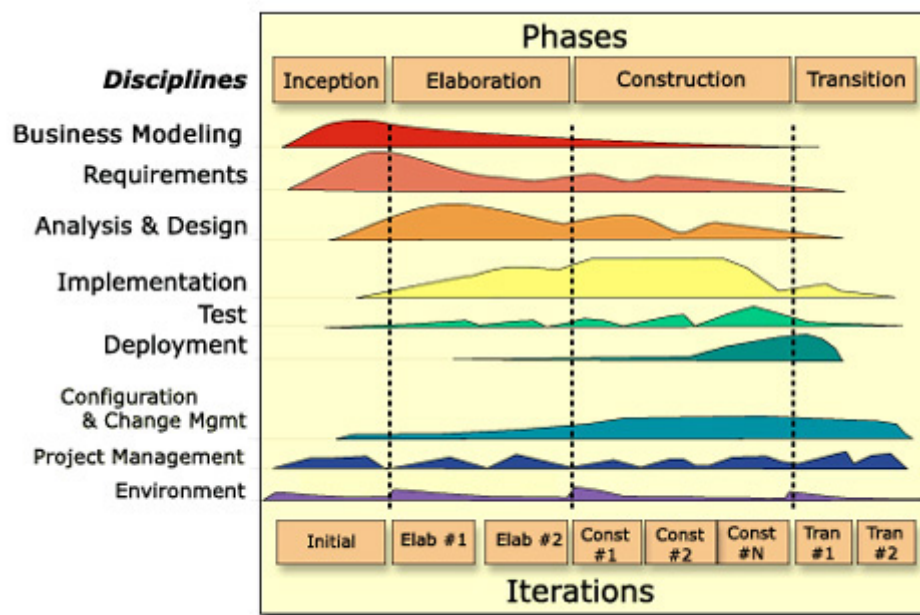
Tento životní cyklus je vylepšenou variantou prototypu o zakomponování ošetření rizik. Název spirálový zachycuje posloupnost opakování jednotlivých kroků ve spirále, kdy každá vyšší úroveň spirály evokuje vyšší verzi řešení (zvládnutí problému) a eliminaci rizik.



Obrázek 8: Spirálový model [ROYCE 1998]

## 2.2.6. Životní cyklus - Iterativní vývoj

Idea iterativního vývoje je založena na eliminaci rizik plynoucích ze změny zadání projektu v průběhu jeho realizace. Dle iterativního přístupu, je projekt dekomponován na menší uzavřené celky, které jsou potom po částech – v iteracích realizovány. Každá iterace končí výstupem a spustitelnou verzí (ve fázi analýzy se jedná např. o prototyp GUI nebo prototyp(y) eliminující technologická nebo kapacitní rizika daného řešení). V rámci iterace je realizován tzv. malý vodopád, kdy funkcionality, která byla vytvořena v předchozí iteraci je dále rozšířena o např. alternativní scénáře nebo o novou, méně prioritní funkcionality.



Obrázek 9: Iterativní vývoj [RUP]

### **2.2.7. Životní cyklus - Extrémní programování a agilní techniky**

Filosofie extrémního programování je založena na snaze co nejrychleji dodat řešení zadavateli a proto používá vybrané praktiky.

Vývoj řešení např. probíhá nejdříve napsáním procedury testování (tzv. test case) a následně je teprve vytvářena vlastní funkcionálníta.

Nedílnou součástí extrémního programování je spolupráce v týmu, kdy dochází k mixu zkušených a méně zkušených programátorů s cílem optimalizovat transfer know-how. Extrémní programování klade vysoký důraz na kvalitu výstupů a proto je jednou z jeho technik kontrola zdrojového kódu programátorem, který tento kód nevytvářel.

Uvedené životní cykly projektů lze aplikovat na vývojové projekty. Není cílem této práce podat jejich vyčerpávající množinu. Jedním z cílů této práce je aplikace iterativního přístupu na provozní projekty realizované v rámci kompetenčního centra.

### 3. Popis řešeného problému

Tato kapitola popisuje klíčové atributy problému, který se vyskytl v praxi. Autorova disertační práce poskytuje návod na vyřešení na tohoto problému.

Při standardní implementaci a customizaci informačního systému je výsledek, finální dílo, po akceptačních testech, pilotním provozu a plošném nasazení předáno koncovým uživatelům pro rutinní, každodenní používání. Mnoho metodik vývoje software v této chvíli projekt ukončí a tím i svá doporučení. Pro méně zkušeného pozorovatele z oblasti ICT situace působí dojmem, že projekt byl dodán, zákazník zaplatil a tudíž je spolupráce ukončena a je třeba hledat nový prostor pro další spolupráci.

Naštěstí nejen pro koncové uživatele, manažery IT útvarů a dodavatele tomu tak ve skutečnosti není. Právě rutinně provozovaný software je třeba rozšiřovat v souladu s rostoucími požadavky jednotlivých útvarů podniku, kterým slouží. To je možné dělat mnoha způsoby a to jak v rámci svých vlastních lidských zdrojů, tak originálním dodavatelem, implementátorem nebo outsourcingem od specializované společnosti.

Servisní, supportní, provozní a podobně nazývané projekty jsou takové projekty, které pokrývají životní cyklus informačního systému po jeho uvedení do rutinního provozu. To, že se jedná o vysoce podnikatelsky zajímavou činnost, dokládají nově vznikající společnosti, útvary nebo dceřiné společnosti velkých softwarových firem.

Klíčové vlastnosti provozních projektů (v kontextu této práce):

- Provoz a údržba osmi bankovních IS.
- Provoz IS mimo území ČR.
- Maximální utilizace servisního týmu.
- Předdefinované požadavky na služby.
- Profitabilita projektů.
- Nespokojenost zákazníka s kvalitou dodávaných patchů (záplat).
- Formalizace procesu testování.

- Formalizace procesu řízení požadavků.
- Aktualizace uživatelské dokumentace.

### **3.1. Provozní projekty a jejich vlastnosti**

Následující podkapitoly detailněji popisují vlastnosti provozních projektů v kontextu řešené problematiky. Nejedná se o celkový výčet požadavků, spíše o jejich průřez.

#### **3.1.1. Provoz a údržba osmi bankovních IS**

Jako první problém je třeba vyřešit podporu a údržbu osmi bankovních IS, které jsou provozovány u různých zákazníků. Těchto osm řešení je nejen funkčně, ale i architektonicky různých.

Vzhledem k evolučnímu vývoji systémů, existuje i případ, kdy dvě instalace téhož systému, ale různých verzí, jsou provozovány u jednoho klienta na jeho dvou různých pobočkách. U starší verze systému bohužel není triviální přechod na jeho novější verzi. Každá oprava nebo nová funkcionality se musí de-facto vytvářet dvakrát, jednou pro starší a jednou pro novější verzi systému. Sjednocení verzí toto eliminuje.

#### **3.1.2. Provoz IS mimo území ČR**

Dvě instalace IS jsou provozovány mimo území ČR, přičemž existuje plán na expanzi nejen v rámci Evropy, ale i do Jižní Ameriky. Robustnost řešení musí umožnit podporu až osmi instalací mimo území ČR. Autor si dovoluje připomenout, že se pohybujeme v kontextu bankovních IS, které obsahují vysoce citlivá data a proto nepřipadá v úvahu varianta podporovat tyto IS přímo z centra dodavatele IS.

### **3.1.3. Maximální „utilizace“ servisního týmu**

Tento poněkud vágně definovaný požadavek v sobě skrývá snahu o maximální využití lidských zdrojů. Vzhledem k vysoké kvalifikaci specialistů, jsou zdroje sdíleny s klasickými vývojovými projekty. To má své ekonomické výhody, nicméně obrovskou nevýhodou je zátěž, která je kladena na bedra manažerů při plánování.

### **3.1.4. Předdefinované požadavky na služby**

Průběžně dohodnuté požadavky na služby, tzv. „SLA“ nejsou jednotná ze dvou důvodů, jednak proto, že každý zákazník je specificky „náročný“ a jednak v době vzniku smluv neexistoval žádný předpis nebo limit, jak je nastavit.

Nejtvrdějším SLA je garance, že je třeba začít řešitelské práce na hlášeném incidentu v produkčním prostředí do 2 hodin od jeho nahlášení. Pro běžné problémy nebo rozvoj nové funkcionality je SLA 16 pracovních dní, jinak dle vzájemné dohody. Tato kritéria musí být za každou cenu dodržena.

### **3.1.5. Profitabilita projektů**

Jelikož se jedná o problém z podnikatelské praxe, není překvapující, že každý provozní projekt musí být profitabilní. Zisk musí být vytvořen i přes projekty napříč což je jedním ze základních posláních programme managementu [detailněji pojednává COR].

### **3.1.6. Nespokojenost zákazníka s kvalitou dodávaných patchů (záplat)**

Toto považuje autor za první specifickou vlastnost řešeného problému, i když z praxe ne nijak ojedinělou. Zákazník je velmi nespokojen s termíny dodávání záplat (patchů) nebo nových verzí, jejich kvalitou, obtížnou testovatelností, absencí release notes a absencí testovacích scénářů.

Bohužel nejsou výjimkou situace, kdy záplata funguje bez problémů na straně dodavatele, ale u zákazníka je po nainstalování nefunkční. Často se problémy objevují již při pouhé instalaci.

Není také řídkým jevem, že v průběhu testování zákazníkem, jsou objeveny chyby vzniklé (zavlečené) opravou jiné chyby.

Výše uvedené aspekty je třeba eliminovat a garantovat tak maximální kvalitu zákazníkovi.

### **3.1.7. Formalizace procesu testování**

Absencí formalizovaného procesu testování, postupu, testovacích scénářů a vzorku reálných dat pro testovací účely, dochází k nedokonalému testování. To ještě zhoršuje časté neověření instalovatelnosti záplaty nebo nové verze.

Řešením je zavést použitelný a opakovatelný proces testování, který bude ekonomicky únosný.

### **3.1.8. Formalizace procesu řízení požadavků**

Proces správy požadavků musí zachycovat jednotlivé požadavky zákazníka a jejich řešení. Na základě tohoto procesu lze odvodit podklady pro stanovení nákladů a tedy i fakturace pro zákazníka. Nezbytným výstupem musí být také podklady pro plánování zdrojů.

### **3.1.9. Aktualizace uživatelské dokumentace**

Zákazník je velmi nespokojen s tím, že uživatelská dokumentace je již přes rok zastaralá a průběžně nereflektuje změny aplikace.

## **3.2. Návrh řešení**

Na základě analýzy požadavků autor prozkoumal dostupné zdroje, navštívil několik společností a došel k závěru, že nejvhodnějším řešením je vyprojektovat kompetenční centrum pro řízení provozních projektů na základě principů programme managementu.



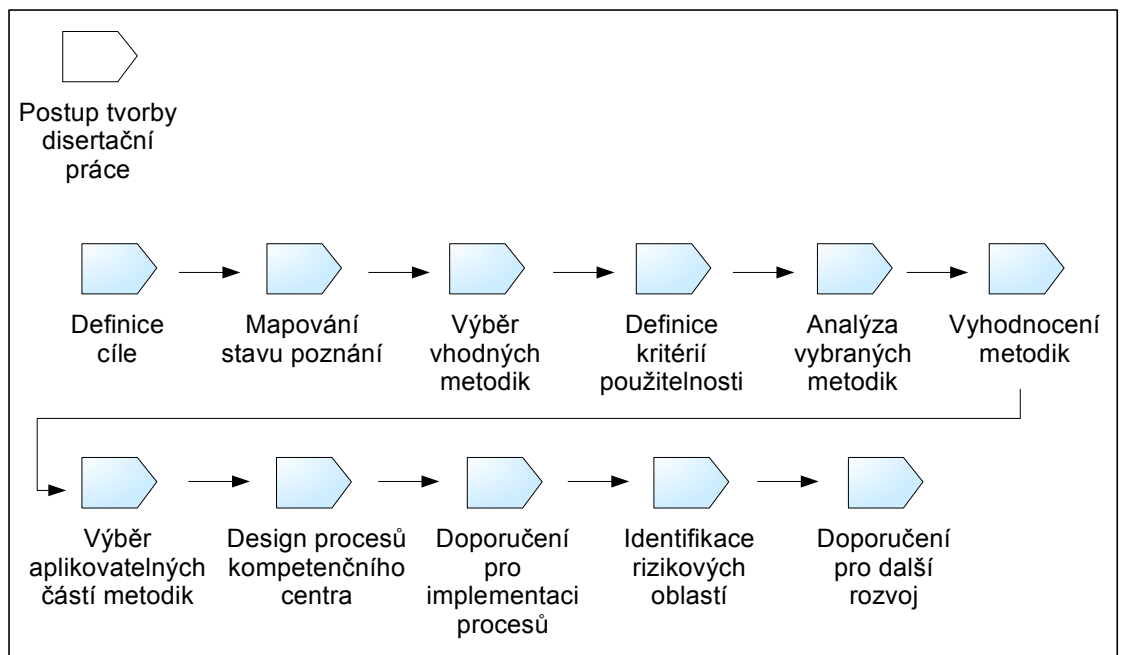
## 4. Metodika disertační práce

Pro vytvoření této disertační práce byly autorem zvoleny standardní vědecké metody a přístupy, mezi které patří převážně pozorování, deskripce, explanace, měření a komparace, modelování, analýza, syntéza, indukce a dedukce.

Postup tvorby této disertační práce:

- Definice cílů práce.
- Mapování stavu poznání ve světě i u nás.
- Výběr metodik ve vztahu k řešené problematice.
- Definice sledovaných kritérií a atributů metodik.
- Analýza vybraných metodik.
- Stanovení použitelných částí metodik.
- Obecné procesy programme managementu.
- Konkrétní procesy Provozu a Servisu.
- Stanovení vlastní práce.
- Definice chybějících procesů programme managementu.
- Definice chybějících konkrétních procesů Provozu a Servisu.
- Reengineering procesů pro potřeby kompetenčního centra.
- Postup nasazení.
- Rizika nasazení.
- Doporučení pro další rozvoj a použití.

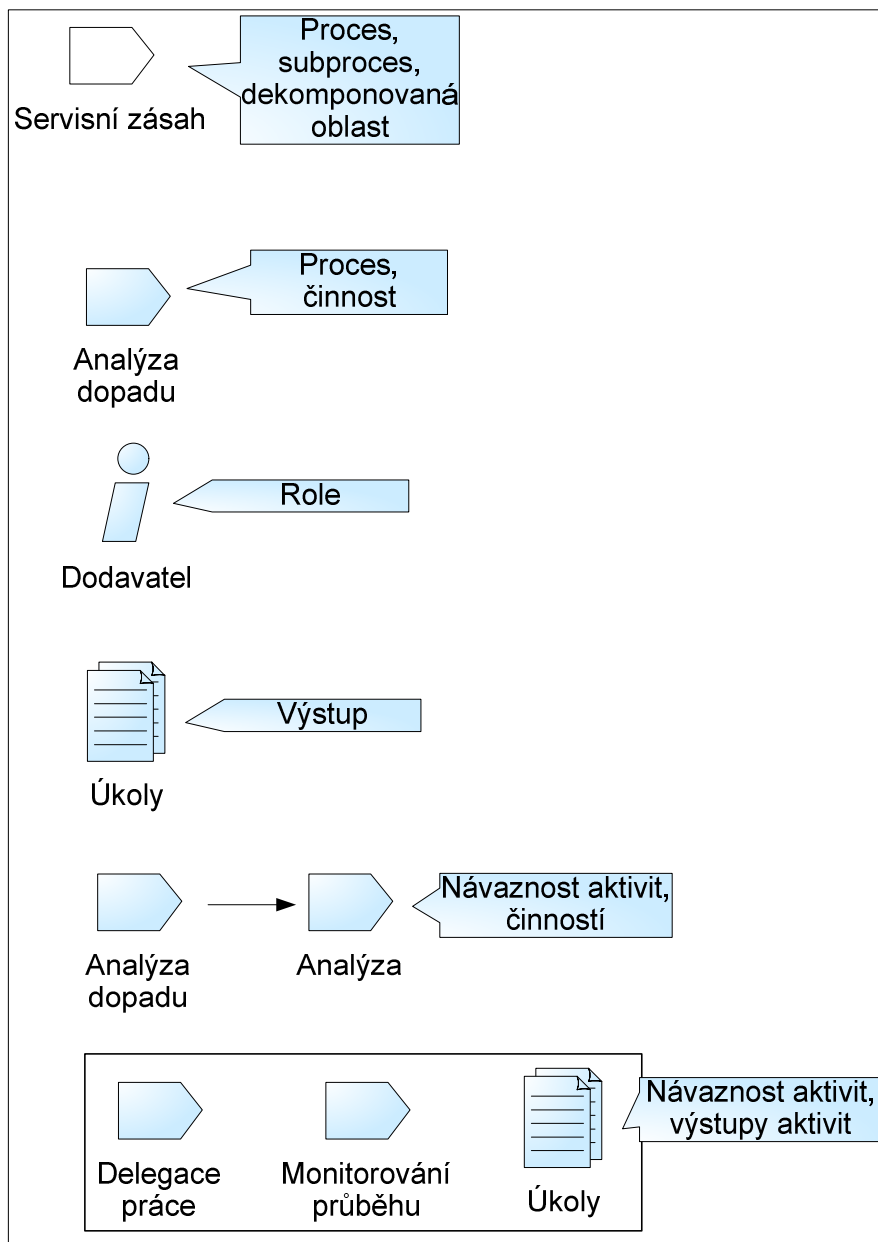
Aplikací výše uvedených činností vznikla metodika pro vytváření a provoz kompetenčního centra, která podporuje realizaci projektů v oblasti Podpory a údržby ICT. Metodika je zachycena formou psaného textu a procesních diagramů (v této práci).



**Obrázek 10: Klíčové aktivity použité při vytváření této disertační práce**

Autor také připravil prototyp intranetového site (k dalšímu dopracování), ve kterém naznačuje možnosti převedení metodiky do elektronické podoby. Tento krok umožňuje další rozvoj metodiky a její rozšíření pro praktické využití mezi podniky podnikajícími v oblasti ICT.

## 4.1. Notace diagramů



Obrázek 11: Vysvětlení notace použité ve schématech diagramů

## 5. Programme management<sup>1</sup>

Vzhledem ke stále rostoucí komplexitě projektů a jejich klíčové úloze pro dosažení business cílů, pokouší se již několik let přední guru projektového managementu formulovat koncepty pro zabezpečení bezproblémové realizace klíčových projektů.

Programme management je výsledkem těchto pokusů. Je založen na filosofii, kdy je snazší koordinovat několik menších projektů pro jednoho zákazníka, než-li mu dodávat jeden velký, složitý projekt. Tato úvaha je založena na předpokladu, že existuje poměr mezi složitostí projektu a mírou problémů a rizik, které se na něm mohou objevit [blíže řešeno ve BOOCH 1999].

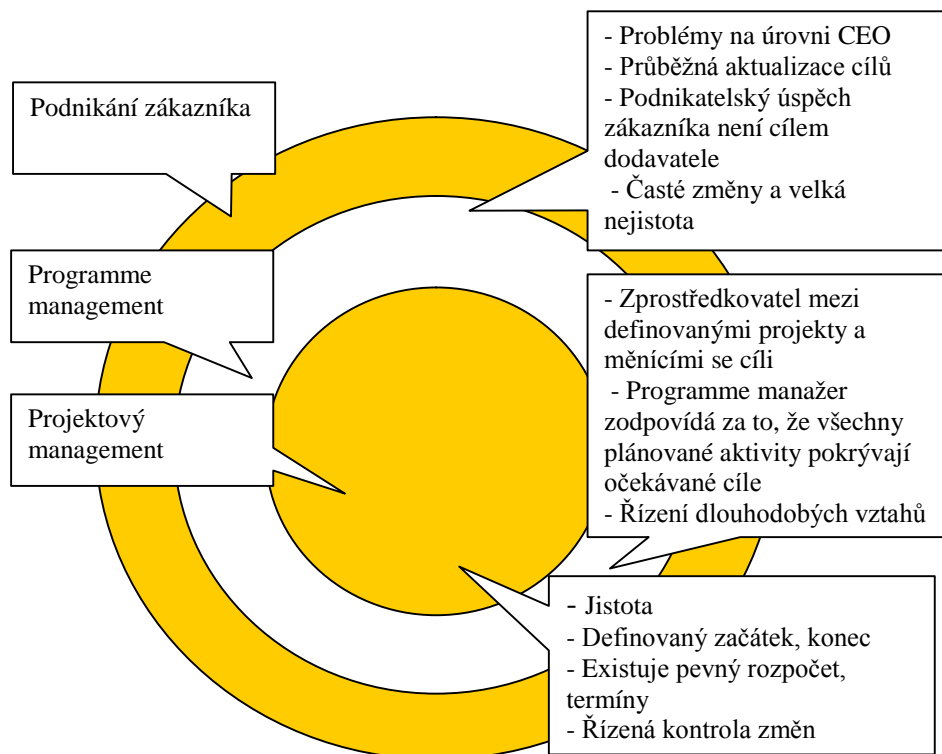
Na základě prostudování literatury [GULDENTOPS 2000], [HAREN 2005], [UČEŇ 2001] a [PMM] lze shrnout definice tohoto manažerského přístupu, do následujícího znění:

**Programme management je portfolio projektů a aktivit, které jsou koordinovány a řízeny dohromady takovým způsobem, aby dosahovali svých cílů a přínosů jak jednotlivě, tak i dohromady a to v podobě synergického efektu na strategické úrovni.**

Metodiky programme managementu doporučují, jak uvedeného cíle dosáhnout.

---

<sup>1</sup> Programme management je definován v této kapitole. Pro tento pojem v češtině neexistuje ekvivalentní výraz, a proto se nepřekládá. Je běžně používán v ICT oblasti.



**Obrázek 12: Programme management [BOOCH 1999]**

Programme management zahrnují komplexní metodiky pro řízení IT útvarů a dodávky jejich služeb (ITIL, Cobit, Cortex, PMM a další), přičemž u nich existuje minimální rozdíl jak v samotném popisu „frameworku“, tak i v obecných doporučeních.

Zásadní rozdíl existuje v akcentu na klíčové části programme managementu, jak programme management implementovat a doporučení oblastí vhodných pro outsourcing. Složitost dané problematiky výrazně přesahuje rámec této disertační práce, a proto se jí autor nadále nebude zabývat.

## **6. Porovnání metodik**

### **6.1. Úvod**

Pro dosažení odborné úrovně a naplnění cílů této práce, autor nejprve realizoval výzkum řešení provozních projektů a to jak v lokálních, tak i nadnárodních společnostech. Bohužel autor není oprávněn publikovat konkrétní výsledky svého zjištění, neboť k tomu nedostal souhlas od uvedených společností.

Realizace provozních projektů probíhá u vybraných subjektů v zásadě dvojitým způsobem (zobecnitelné závěry z analýzy vybraných finančních institucí):

- Realizace empiricky na základě zkušeností. Zpravidla existuje pracovník, který je kontaktní osobou a správcem aplikace najednou, na kterého se všichni obrací s žádostí o vyřešení svých problémů a přání. Tento pracovník rozhoduje kdy a jak budou požadavky vyřešeny. Kompetencí pro jeho rozhodování je hluboká znalost aplikace, neboť tento pracovník bývá zpravidla jejím autorem a zná její historii.
- Realizace na základě doporučení standardů ICT metodik (např. COBIT, ITIL atd.) V tomto případě se zpravidla jedná o implementaci vybraných procesů, které často nemají vzájemná rozhraní a nejsou podpořeny odpovídajícími nástroji. Velkou nevýhodou těchto metodik je, že pokrývají problematiku ICT provozu na vysoké úrovni a doporučují pouze dobré praktiky daného odvětví. Neobsahují design procesů, jejich výstupů, nástrojovou podporu, metriky, guidelines atd.

#### **6.1.1. Stav poznání řešené problematiky v rámci akademických institucí**

Vzhledem k orientaci náplně této práce více do praxe a na základě prostudování webových stránek předních institucí, si autor dovoluje tvrdit, že k datu tvorby této práce, se žádná ze zkoumaných akademických institucí dosud touto problematikou nezabývá.

Autor intenzivně analyzoval níže uvedené webové stránky v období dvou měsíců (únor a březen 2005), kdy začínal pracovat na výzkumu své práce. Tato analýza probíhala kontinuálně v průběhu tvorby této práce, tj. do konce roku 2006.

Seznam akademických institucí v ČR (řazeno abecedně), které autor této práce analyzoval s cílem zmapovat stav poznání problematiky této disertační práce:

- České vysoké učení technické v Praze – [www.cvut.cz](http://www.cvut.cz).
- Česká zemědělská univerzita v Praze – [www.czu.cz](http://www.czu.cz).
- Masarykova univerzita – [www.muni.cz](http://www.muni.cz).
- Slezská univerzita v Opavě – [www.slu.cz](http://www.slu.cz).
- Technická univerzita v Liberci – [www.vslib.cz](http://www.vslib.cz).
- Vysoká škola ekonomická v Praze – [www.vse.cz](http://www.vse.cz).
- Vysoká škola manažerské informatiky a ekonomie, a.s. v Praze – [www.vsmie.cz](http://www.vsmie.cz).
- Vysoké učení v Brně – [www.vutbr.cz](http://www.vutbr.cz).
- Západočeská univerzita v Plzni – [www.zcu.cz](http://www.zcu.cz).

Seznam akademických institucí v zahraničí (řazeno abecedně), které autor této práce analyzoval s cílem zmapovat stav poznání problematiky této disertační práce:

- Berkeley (University of California) – USA – [www.berkeley.edu](http://www.berkeley.edu).
- Bert-Ludwigs-Universität Freiburg – D - <http://www.uni-freiburg.de>.
- Boston University – USA - <http://www.bu.edu>.
- California Institute of Technology – USA – [www.caltech.edu](http://www.caltech.edu).
- Columbia University – USA – [www.columbia.edu](http://www.columbia.edu).
- Eidgenössische Technische Hochschule Zürich – CH - <http://www.ethz.ch>.
- Harvard University – USA - [www.harvard.edu](http://www.harvard.edu).
- Lomonosov Moscow State University - RU - <http://www.msu.ru/en>.
- Ludwig-Maximilians-Universität in München – D - <http://www.uni-muenchen.de/index.html>.
- Massachusetts Institute of Technology – USA – [www.mit.edu](http://www.mit.edu).

- Osaka University – JAP - <http://www.osaka-u.ac.jp/eng/index.html>.
- Pierre & Marie Curie University – F- <http://english.upmc.fr/UK/info>.
- Princeton University – USA- [www.princeton.edu/main](http://www.princeton.edu/main).
- Stanford University – USA – [www.stanford.edu](http://www.stanford.edu).
- The Australian National Univerisity – AU - <http://www.anu.edu.au>.
- University of Copenhagen – DN - <http://www.ku.dk/english>.
- The University of Tokio – JAP - [http://www.u-tokyo.ac.jp/index\\_e.html](http://www.u-tokyo.ac.jp/index_e.html).
- Tokyo Institute of Technology – JAP - <http://www.titech.ac.jp>.
- Uppsala Universitet – SW - <http://www.uu.se>.
- University of Cambridge – UK - [www.cam.ac.uk](http://www.cam.ac.uk).
- University of Oxford – UK - <http://www.ox.ac.uk>.
- University of Toronto – CA - <http://www.utoronto.ca>.
- Universiteit Utrecht – N -  
<http://www.uu.nl/uupublish/homeuu/homeenglish/1757main.html>.

Výše uvedené akademické instituce se k datu tvorby této disertační práce nezabývají řešeným tématem.

### **6.1.2. Stav poznání řešené problematiky v rámci komerčních institucí**

Kapitola 6 detailněji pojednává o vybraných metodikách a jejich použitelnosti vzhledem k náplni a zaměření této práce. Závěry ze zkoumání těchto metodik lze shrnout do následujících bodů:

- Metodiky obecně pokrývají životní cyklus ICT řešení od počátku až po nasazení do produkčního prostředí.
- ITIL, COBIT a PMM pokrývají problematiku provozu ICT na vysoké úrovni.
- RUP a CORTEX provoz ICT neřeší vůbec.
- Neexistuje metodika, která pokrývá vývoj a provoz ICT najednou



- Neexistuje metodika, která pokrývá provoz ICT (procesy, činnosti, výstupy, role, šablony, guideline, koncepty, metriky, doporučení pro implementaci atd.).

Vývoj metodik směřuje k procesní orientaci IT a k vzájemné integraci:

- ISACA/ITGI - COBIT 4.0 (prosinec 2005) – orientace směrem na IT Governance (informace, aplikace, infrastruktura, proces, SLA atd.) a měření výkonnosti – SLA, KPI, KGI.
- OGC/ITSMF – ITIL3.0 (2006) – integrace ITIL a COBIT, integrace s normou ISO 20000.

Pro potřeby této práce, autor detailně analyzoval následující vybrané metodiky spjaté přímo s vývojem software nebo ICT<sup>1</sup>:

- ITIL (Information Technology Infrastructure Library), [ITIL].
- COBIT (Control Objectives for Information and Related Technology), [COBIT].
- CORTEX, [COR].
- PMM (Programme Management Methodology), [PMM].
- RUP (Rational Unified process), [RUP].

Výše uvedené metodiky byly vybrány autorem na základě následujících kritérií:

- Dostupnost metodiky, dostupnost podkladů.
- Relevantnost metodiky pro řešenou oblast.
- Zastoupení jak světově známých, tak i lokálních (PMM) metodik.
- Zahrnutí metodik od velkých společností, stejně tak i od malých.
- Vyjma metodiky CORTEX zkušenosti s praktickou implementací.

---

<sup>1</sup> Výběr metodik podléhá plně autorovi této práce. Každá z vybraných metodik je dále představena a analyzována v samostatných kapitolách.

Na základě analýzy metodik v kontextu definovaných kritérií, vymezil autor procesy programme managementu, které je možné použít pro vytváření kompetenčního centra.

Dále vymezil procesy, které je potřeba vyprojektovat pro realizaci provozních projektů. Vyprojektování chybějících procesů patří mezi cíle této práce.

## **6.2. Výběr kritérií pro analýzu metodik**

### **6.2.1. Kritéria pro oblast programme managementu**

- Definice programme managementu.
- Obecný proces programme managementu.
- Konkrétní procesy programme managementu.
- Výstupy.
- Doporučení pro implementaci (pokrytí vytvoření KC).

### **6.2.2. Kritéria pro oblast provozních projektů**

- Management provozních projektů.
- Správa prostředí.
- Správa požadavků.
- Analýza a Návrh.
- Oprava chyb/Vývoj.
- Testování.
- Nasazení.

Každý proces z výše uvedených bude analyzován z následujícího úhlu pohledu:

- Obecná definice procesu.
- Konkrétní definice procesu.
- Definice aktivit (činností, kroků) a jejich popis.
- Definice výstupů a jejich popis.

- Přítomnost šablon výstupů.
- Využitelnost procesu bez dodatečných úprav.
- Implementovatelnost procesu bez dodatečných úprav.

### **6.3. Metodika popisu metodik ICT**

Každá metodika je popsána v následující struktuře:

#### **1. Úvod.**

Cílem této kapitoly je představit metodiku zejména z následujících úhlů pohledu:

- Historie vzniku.
- Metamodel metodiky.
- Základní struktura procesů.
- Ukázka.

#### **2. Metodika a programme management.**

Cílem této kapitoly je analyzovat vhodnost metodiky pro aplikaci programme managementu při vytváření a řízení kompetenčních center.

Analýza metodiky pokrývá následující:

- Teorie programme managementu.
- Doporučení metodiky pro implementaci.

#### **3. Metodika a realizace provozních projektů.**

Cílem této kapitoly je analyzovat vhodnost metodiky pro aplikaci při realizaci provozních projektů.

Analýza metodiky pokrývá následující:

- Procesy provozních projektů.
- Doporučení pro implementaci.

#### **4. Vyhodnocení aplikovatelnosti metodiky.**

Cílem této kapitoly je vyhodnotit metodiku a stanovit vhodnost použití jednotlivých procesů pro potřeby KC.

Na závěr jsou v dedikované kapitole vyhodnoceny všechny metodiky s cílem stanovit oblasti, které budou převzaty (a upraveny pro potřeby KC) a procesy, které bude třeba od začátku tzv. „na zelené louce“ vyprojektovat. Velikost úprav závisí na konkrétní aplikaci projektu.

## **6.4. ITIL - Information Technology Infrastructure Library [ITIL]**

### **6.4.1. Úvod**

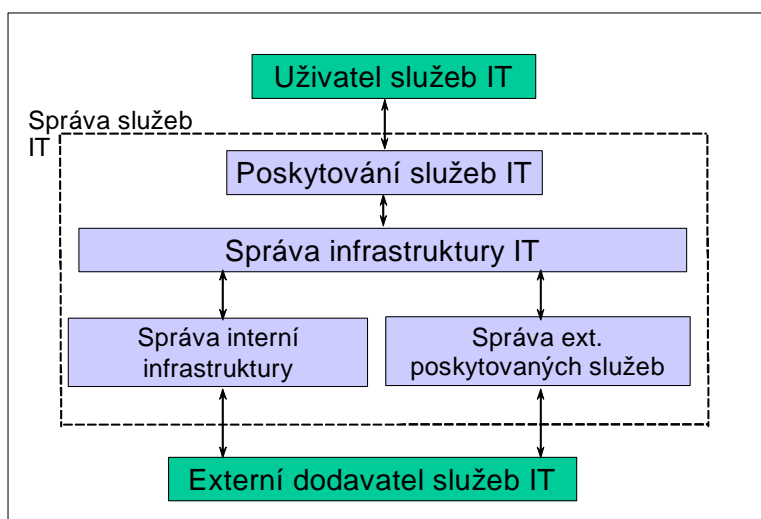
Information Technology Infrastructure Library (dále jen ITIL) je metodika vyvinutá podle standardů ISO 9001 britskou agenturou Central Computer and Telecommunications Agency (CCTA) před více než 25 lety. To, že tato metodologie existuje již takovou dobu neznamena, že je zastaralá, neboť průběžně dochází k její aktualizaci.

Procesy v oblasti IT jsou stále složitější a je stále důležitější mít zavedené všechny takové procesy, které zajistí, že IT bude dodávat stabilně kvalitní služby za optimální náklady.

Metodika ITIL je v Čechách rozšířena ve velkých finančních ústavech a společnostech, které jsou závislé na ICT. Metodika bude stále více používána neboť je standardem pro ISO certifikaci dle normy BS 15000 a ISO 20000. Dosud masovému rozšíření zabraňují pořizovací náklady této metodiky, implementační náklady a přirozená obava managementu ze změn.

## 6.4.2. Metamodel

V ITIL je oddělení IT chápáno jako poskytovatel služeb a to jak vně, tak dovnitř firmy. Je nezbytné monitorovat, zda IT splňuje požadavky, které jsou na něj kladeny. K tomu je nutné nějakým způsobem kategorizovat služby a měřit jejich efektivitu, resp. dostupnost. Tím je možné zjistit velký rozdíl ve výkonnosti jednotlivých pracovníků, případně jednotlivých oddělení, které se problematikou IT zabývají.



Obrázek 13: IT jako poskytovatel služeb [ITIL]

## 6.4.3. Struktura ITIL

Metodika ITIL popisuje IT problematiku strukturovaným způsobem pomocí procesů. Celá problematika provozu a správy služeb v oblasti IT je rozdělena na několik základních procesů. Zaměřuje se na všechny aspekty, které je třeba vzít v úvahu při plánování a implementaci procesů a jak procesy koordinovat v rámci celého IT.

Procesy ITIL prochází vertikálně napříč útvary IT a pokrývají celou organizační strukturu. Každý proces má vždy nějakého koordinátora - osobu určenou za koordinaci procesu napříč organizační hierarchií.

Metodologie je koncipována modulárním způsobem, což v praxi znamená, že je možno ji použít buď jako celek nebo implementovat pouze izolované moduly.

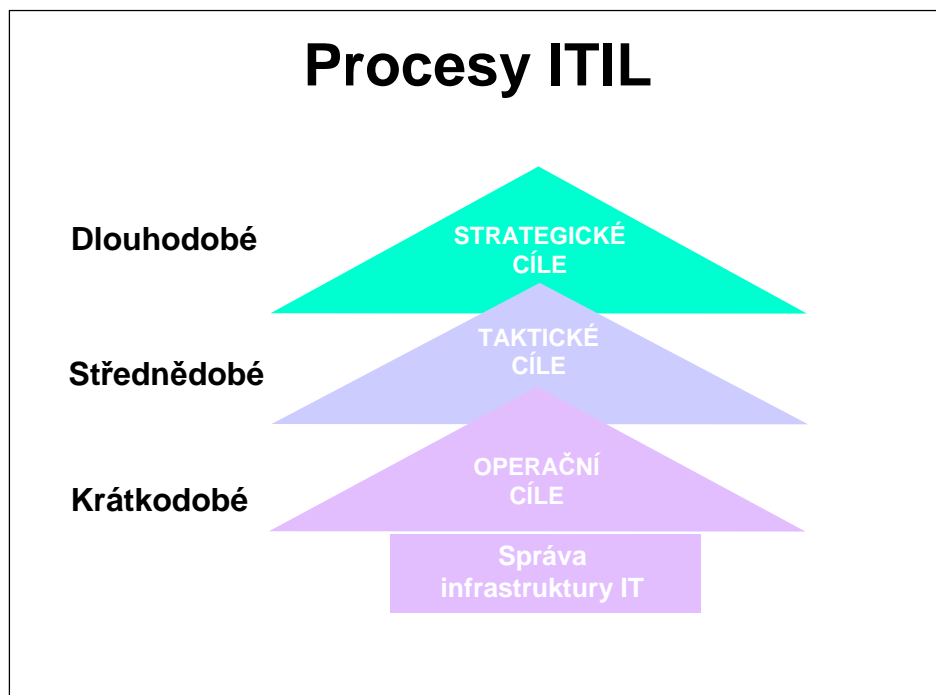
Tzv. „Service Delivery Set“ zahrnuje procesy, které zajišťují dodávku kvalitních služeb na základě SLA (Service level agreement). SLA lze chápat jako smlouvy o poskytovaných službách, jejich kvalitě a dostupnosti mezi vedením IT a jeho zákazníky.

Mezi procesy, které je nezbytné dle ITILu takto řídit patří:

- „Správa smluv o poskytovaných službách.
- Řízení dostupnosti.
- Řízení kapacit.
- Plánování a správa rizik“, [BARTLET 2003].

Jako „Service Support Set“ jsou označovány procesy, které se zabývají identifikací a udržováním konfigurace, každodenní komunikací s uživateli, řízením a koordinací incidentů, problémů a změn.

ITIL člení funkce Řízení služeb IT do tří úrovní - Operační, Taktické a Strategické procesy [BARTLET 2003].



**Obrázek 14: Členění funkce řízení služeb [BARTLET 2003]**

Krátkodobými operačními cíli se rozumí každodenní operativa, střednědobými (taktickými) horizont měsíců, kvartálů, maximálně však jednoho roku a dlouhodobými (strategickými) cíli horizont několika let.

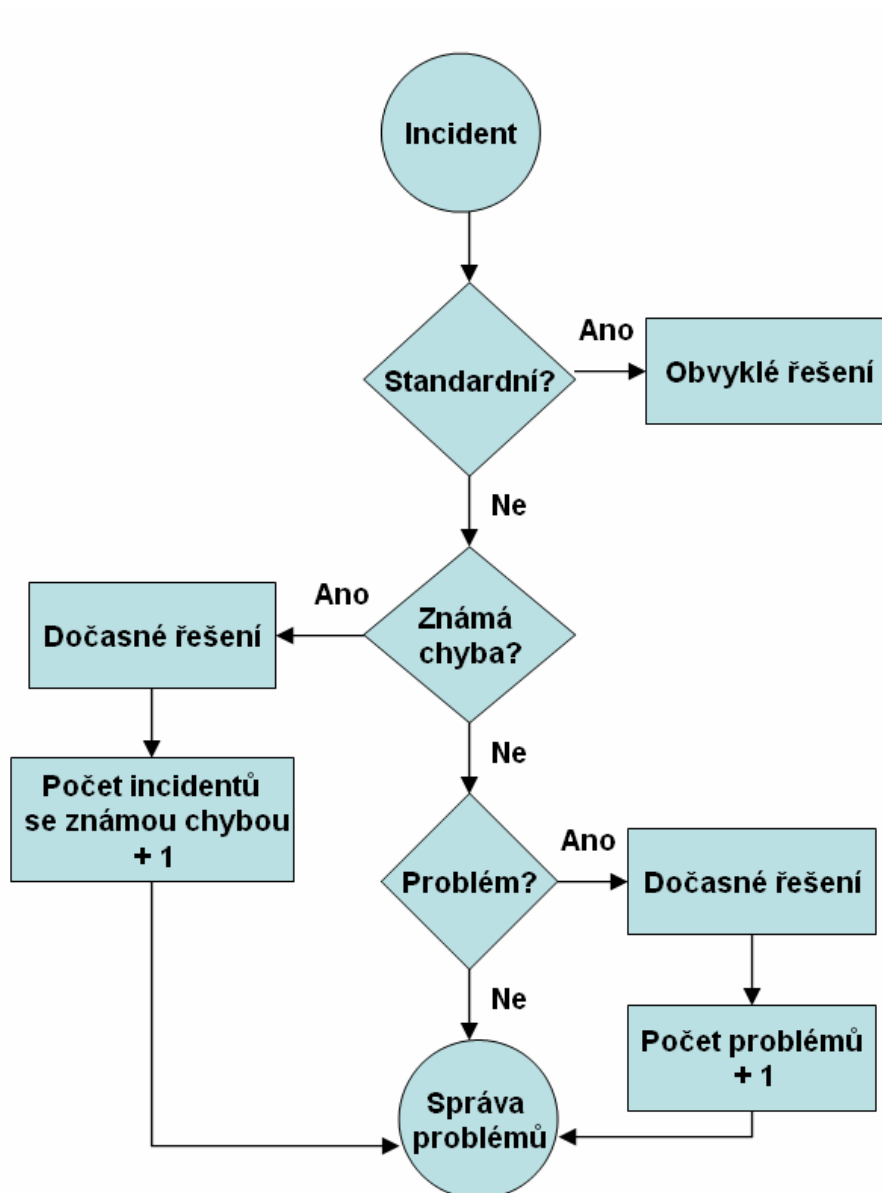
Krátkodobé funkce zajišťují stabilitu a flexibilitu. Střednědobé funkce zabezpečují efektivní správu služeb, což zaručuje dodávku služeb v rozumné kvalitě a ceně. Dlouhodobé funkce se týkají formování strategie IT v souladu s formováním strategie obchodních procesů podniku na několik let dopředu.

Rozdělení na tři skupiny procesů reflektuje různé pohledy na stejnou problematiku od různých úrovní zaměstnanců podniku. Např. zaměstnanci Service Desku se starají především o každodenní poskytování služeb IT, Service manažeři se starají o uzavírání smluv o poskytování služeb (SLA), o pravidelné schůzky s koncovým uživatelem a o pravidelné vyhodnocování kvality poskytovaných služeb.

Naproti tomu Manažer IT (CIO) se zabývá informatickým zabezpečením obchodních procesů firmy v dlouhodobém pohledu. Všechny skupiny by samozřejmě měly vědět i o cílech, které jim přímo nepřísluší.



#### 6.4.4. Proces - příklad



Obrázek 15: Ukázka procesu ITIL - Řešení incidentů na operativní úrovni

### **6.4.5. Implementace ITIL**

Cílem implementace metodiky je zavedení standardních pracovních procesů, standardních postupů a pracovních instrukcí, podle kterých se skutečně postupuje. Dále také stanovení rolí v rámci procesů, zodpovědností a pravomocí. Úspěšné zavedení procesů v IT závisí na angažovanosti vedení IT.

ITIL je nejvhodnější implementovat pomocí studie proveditelnosti, realizace a post-implemenční revize. Základním cílem Studie proveditelnosti je stanovit rozsah, vyčíslit náklady, čas a přínosy, identifikovat rizika a doporučit detailnější kroky, pro úspěch implementace.

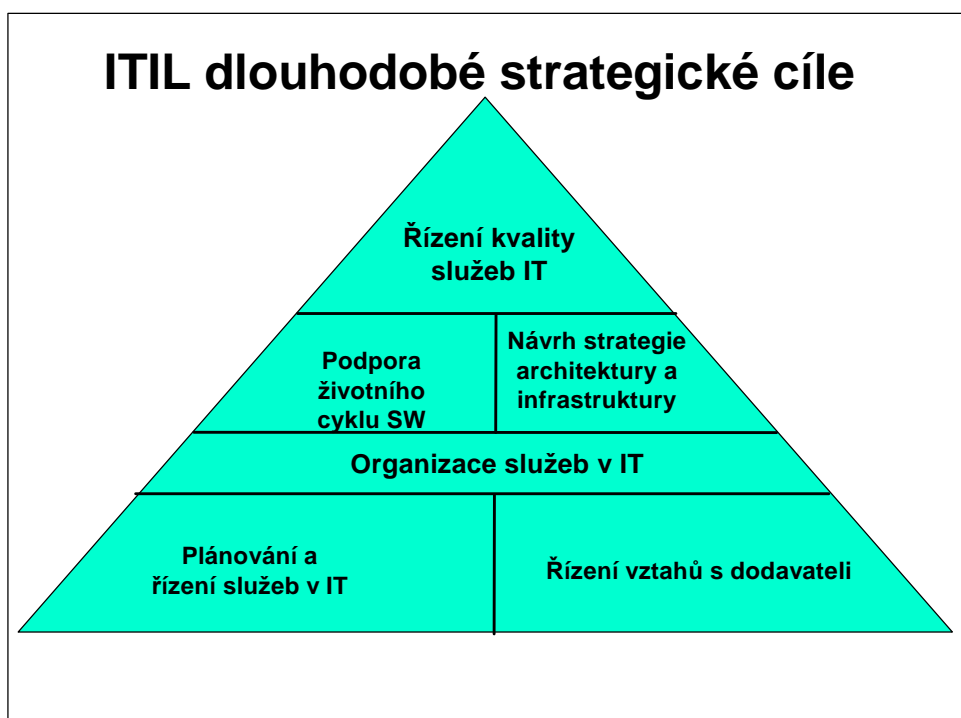
Revize po zavedení (resp. audit) je doporučován proto, aby bylo možno stanovit, zda projekt zavedení "Procesu" splnil svůj účel a případné chyby, které byly identifikovány, mohly být eliminovány při příštích projektech. Tato revize by měla být realizována jednak bezprostředně po ukončení projektu, jednak jako post-implemenční audit cca 3-6 měsíců po ukončení projektu.

Kromě jednorázových revizí je vhodné provádět další procedury zahrnující denní, měsíční, čtvrtletní a roční akce, včetně ad-hoc revizí.

### **6.4.6. Programme management**

ITIL jako takový definuje, jakým způsobem řídit a poskytovat služby ICT. Výsledkem detailní analýzy je zjištění, že nezahrnuje ani projektový ani programový management. Přesto autor této práce považuje za vhodné detailněji popsat rozpad úrovní procesů, neboť jej může dále použít jako vstup při vytváření kompetenčního centra.

#### 6.4.6.1. Dlouhodobé (strategické) cíle

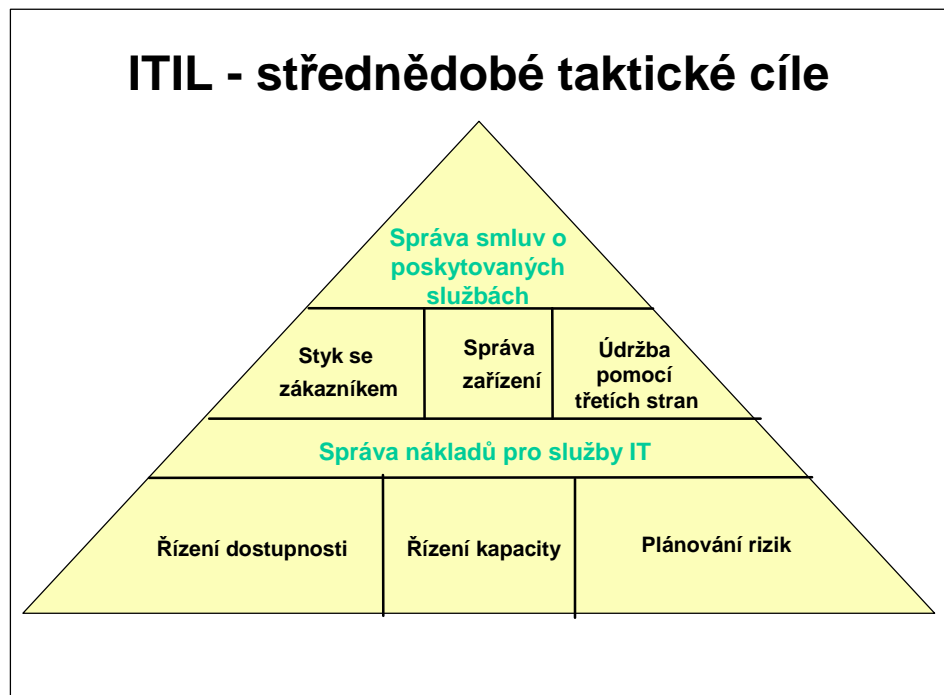


**Obrázek 16: Dlouhodobé (strategické) cíle [BARTLET 2003]**

- Plánování a řízení služeb v IT (Planning & Control for IT Services).
  - o Procesy efektivního plánování a řízení podpory a poskytování služeb IT.
- Řízení vztahů s dodavateli (Managing Supplier Relationships).
  - o Procesy správy vztahů k dodavatelům a udržovatelům HW, SW, podpůrných služeb a síťových komponent od prvního kontaktu.
- Organizace služeb v IT (IT Services Organisation).
  - o Procesy organizace personálu IT.
  - o Požadavky na znalosti a zkušenosti.

- Podpora životního cyklu SW (Software Lifecycle Support).
  - o Procesy zapojení zaměstnanců do životního cyklu SW včetně analýzy a návrhu, testování a údržby.
- Návrh strategie architektury a infrastruktury.
  - o Procesy návrhu a optimalizace architektury a infrastruktury.
- Řízení kvality služeb IT (Quality Management for IT Services).
  - o Procesy plánování a implementace systému řízení kvality služeb.

#### 6.4.6.2. Střednědobé (taktické) cíle



**Obrázek 17: Střednědobé (taktické) cíle [BARTLET 2003]**

- Řízení dostupnosti (Availability Management).
- Řízení kapacity (Capacity Management).
- Plánování rizik (Contingency Management).

- Procesy obnovy dostupnosti služeb v co nejkratší době.
- Správa smluv o poskytovaných službách (Service Level Management).
  - Procesy zabezpečující spolehlivost a dostupnost služeb.
- Styk se zákazníkem (Customer Liaison).
  - Procesy zefektivňující investice do IT zaměřené na použití IT zákazníkem.
- Správa zařízení (Managing Facilities Management).
  - Procesy externí správy poskytované externími organizacemi.
- Údržba pomocí třetích stran (Third Party Maintenance).
  - Procesy outsourcingu.
- Správa smluv o poskytovaných službách (Service Level Management).
  - Procesy správy smluv o poskytování služeb v oblasti IT.

#### 6.4.6.3. Krátkodobé operační cíle

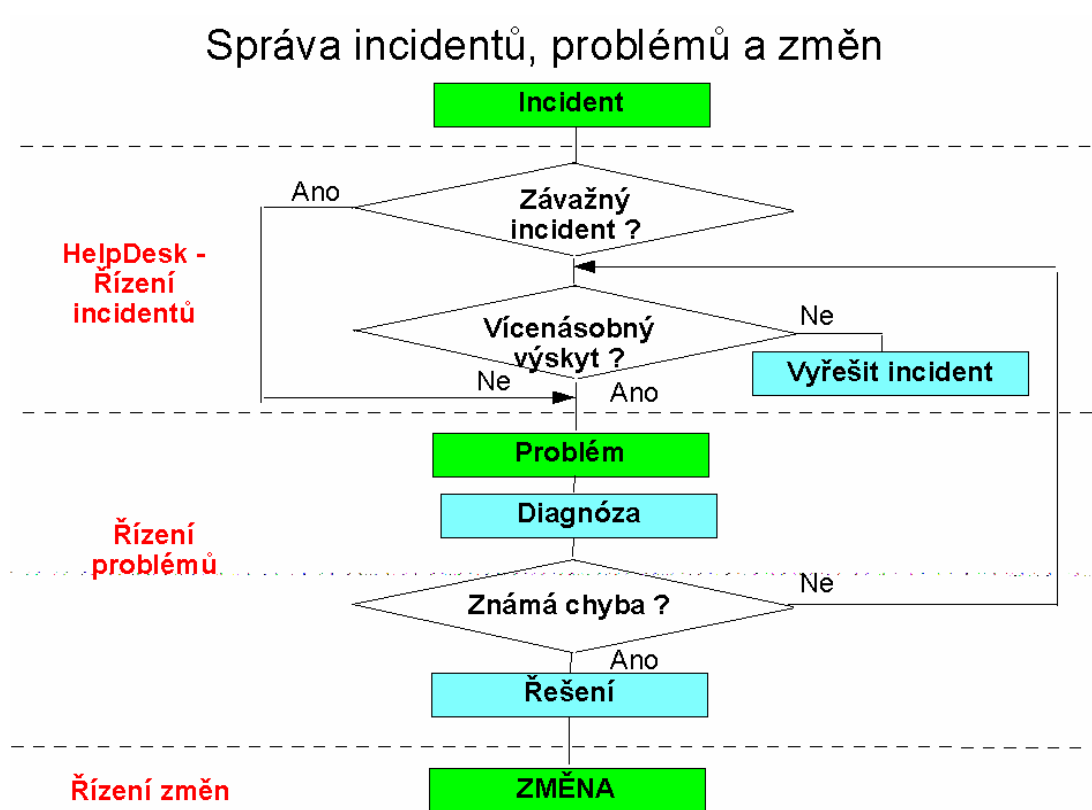


**Obrázek 18: Krátkodobé (operační) cíle IT útvaru [BARTLET 2003]**

- Řízení konfigurace.
  - o Procesy řízení konfigurací.
- Help Desk.
  - o Proces centralizovaného hlášení problémů.
  - o Řešení cca 80 – 90% problémů.
- Provoz počítačů (Computer Operations).
  - o Procesy spjaté s každodenním provozem počítačů.
- Neobsluhovaný provoz (Unattended Operating).
  - o Procesy pro plánování a implementování neobsluhovaných provozů.
- Správa síťových služeb.
  - o Procesy instalace, konfigurace a provozu síťových služeb.
- Správa lokálních počítačů a stanic (Management of Local Processors and Terminals).
  - o Proces správy uvedených zařízení, která jsou fyzicky umístěna u zákazníků.
- Správa problémů (Problem Management).
  - o Proces řešení incidentů, které se opakují častěji.
- Správa a distribuce SW (SW Control and Distribution).
  - o Procesy řízení, ukládání, distribuce SW.
- Instalace PC a akceptace.
  - o Proces instalace a nasazení PC do vybraného prostředí.
- Testování služeb IT pro použití v provozu.
  - o Proces ověření nastavení služeb IT.
- Řízení změn (Change Management).
  - o Procesy řízení změn od návrhu přes dopad, rizika a autorizace až po implementaci.

### 6.4.7. ITIL a realizace provozních projektů

Z pohledu aplikovatelnosti metodiky ITIL je třeba realizaci provozních projektů rozlišit na dvě oblasti – management problémů, kde je ITIL vhodné použít (viz. následující schéma) a management provozních projektů a jejich realizaci, kde je ITIL naprosto nevhodným nástrojem.



Obrázek 19: Řízení incidentů

## 6.4.8. Vyhodnocení aplikovatelnosti metodiky ITIL pro potřeby KC

Metodika ITIL je renomovanou a kvalitní metodikou pro řízení IT útvarů. Vzhledem k záměru analýzy této práce je třeba podotknout, že záběr pro potřeby programme managementu a řízení a dodávku provozních projektů je téměř nevhodný. Nevhodnost ITILu spočívá v nepokrytí detailu zkoumané problematiky.

DISCIPLÍNA/METODIKA	ITIL
<i>Programme management</i>	<i>Nevhodná</i>
<b>Supportní projekty</b>	<b>N/A</b>
<i>Management provozních projektů</i>	<i>Nevhodná</i>
<i>Správa prostředí</i>	<i>Nutno upravit</i>
<i>Správa požadavků</i>	<i>Nutno upravit</i>
<i>Analýza a Design</i>	<i>Nutno upravit</i>
<i>Oprava chyb/Vývoj</i>	<i>Nutno upravit</i>
<i>Testování</i>	<i>Nutno upravit</i>
<i>Nasazení</i>	<i>Nutno upravit</i>

## 6.5. COBIT- Control Objectives for Information and Related Technology

### 6.5.1. Úvod

*„The COBIT Mission: To research, develop, publicize and promote an authoritative, up-to- date international set of generally accepted information technology control objectives for day-to-day use by business managers and auditor“, [GULDENTOPS 2000].*

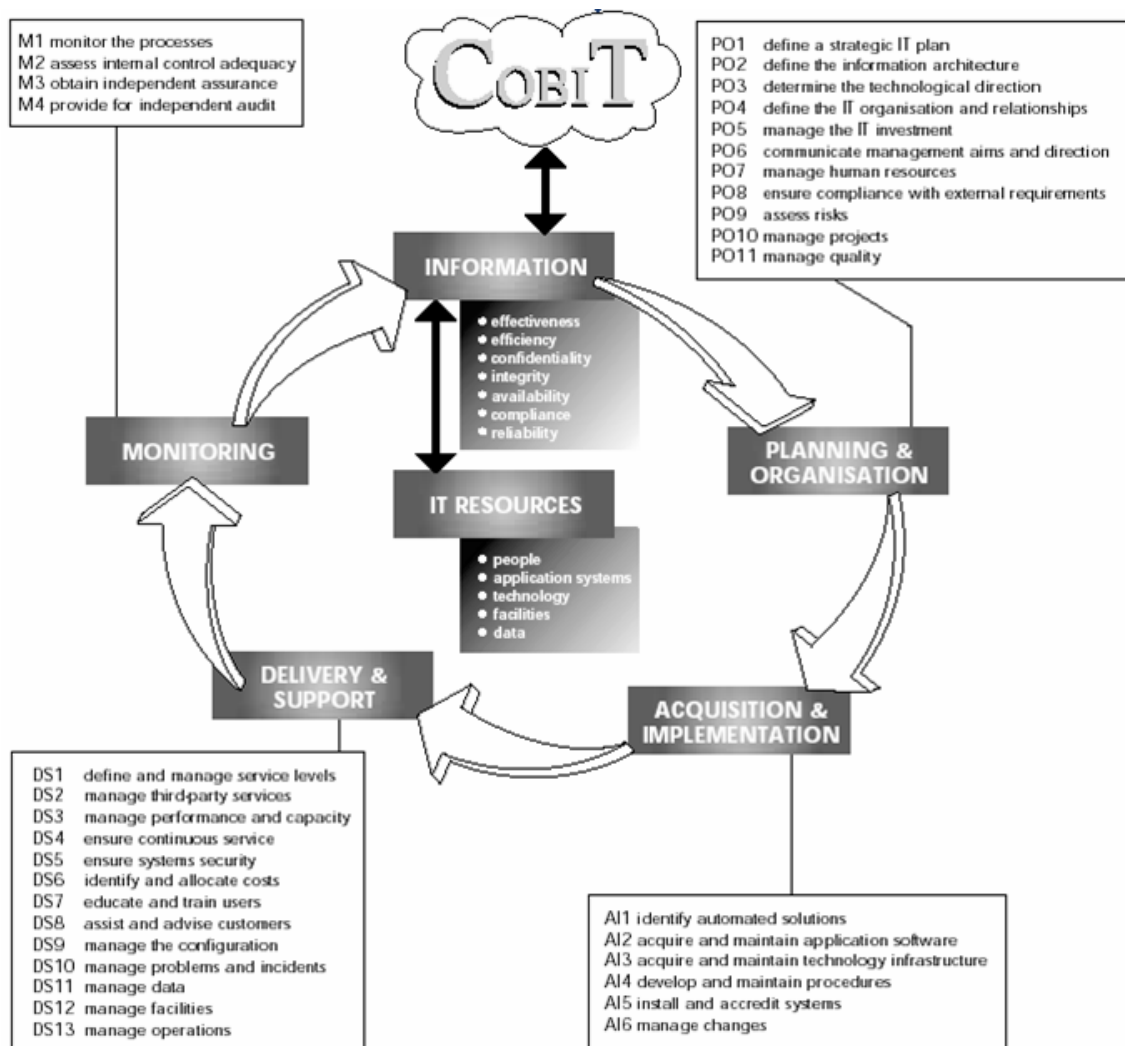


„Posláním metodiky COBIT (Control Objectives for Information and Related Technology) je zkoumat, vyvíjet, publikovat a rozšiřovat směrodatný, aktuální a mezinárodně přijímaný komplex cílů řízení informačních technologií pro každodenní používání obchodními manažery a auditory“, [GULDENTOPS 2000].

Metodika COBIT je používána zpravidla ve velkých finančních ústavech a společnostech, které jsou závislé na ICT a mají implementován ITIL. COBIT slouží mj. pro hodnocení kvality implementovaných procesů dle ITIL.

### **6.5.2. Metamodel**

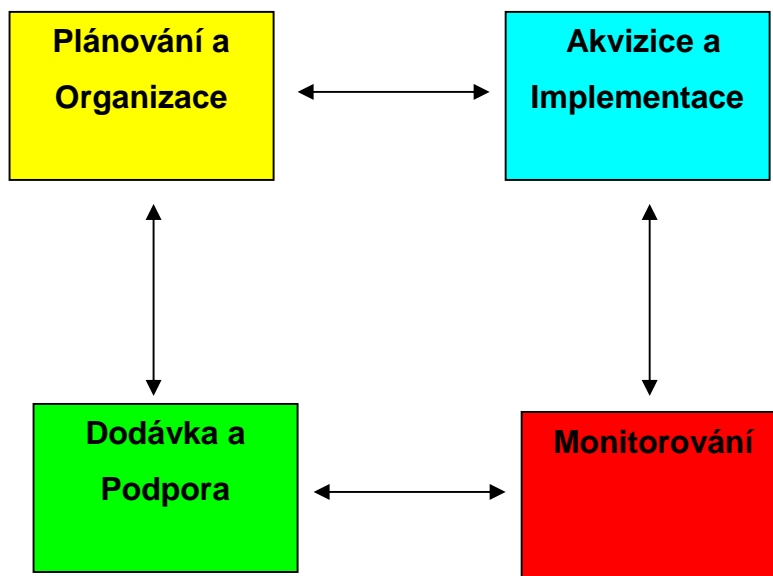
COBIT definuje velmi přesně výkonnostní indikátory a způsoby měření a řízení jednotlivých procesů. Kromě toho také popisuje parametry procesu z hlediska procesní vyspělosti dle CMMi (Capability Maturity Model Integrated). Metodika COBIT není založena na metamodelu.



Obrázek 20: Přehled COBIT [GULDENTOPS 2000]

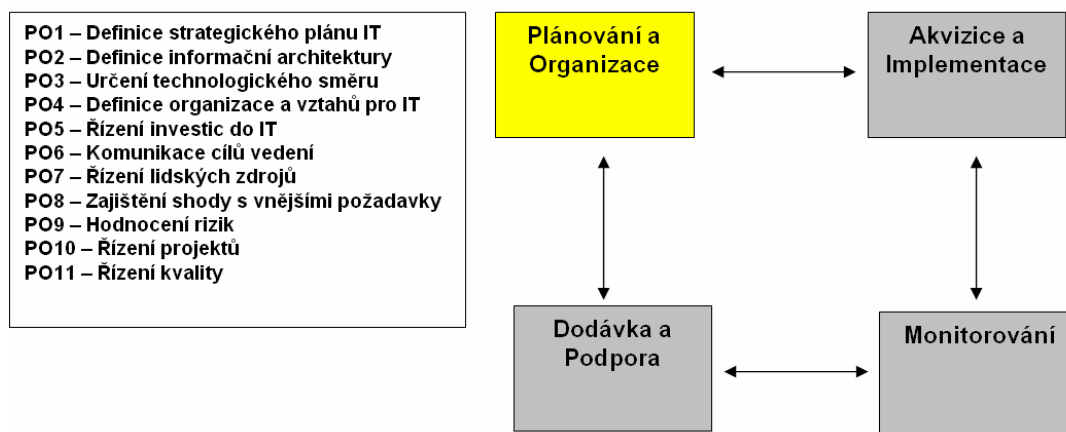
### 6.5.3. Struktura COBIT

COBIT je složen ze čtyř základních domén, které obsahují 34 klíčových procesů [platí pro COBIT – 3 ed., GULDENTOPS 2000].



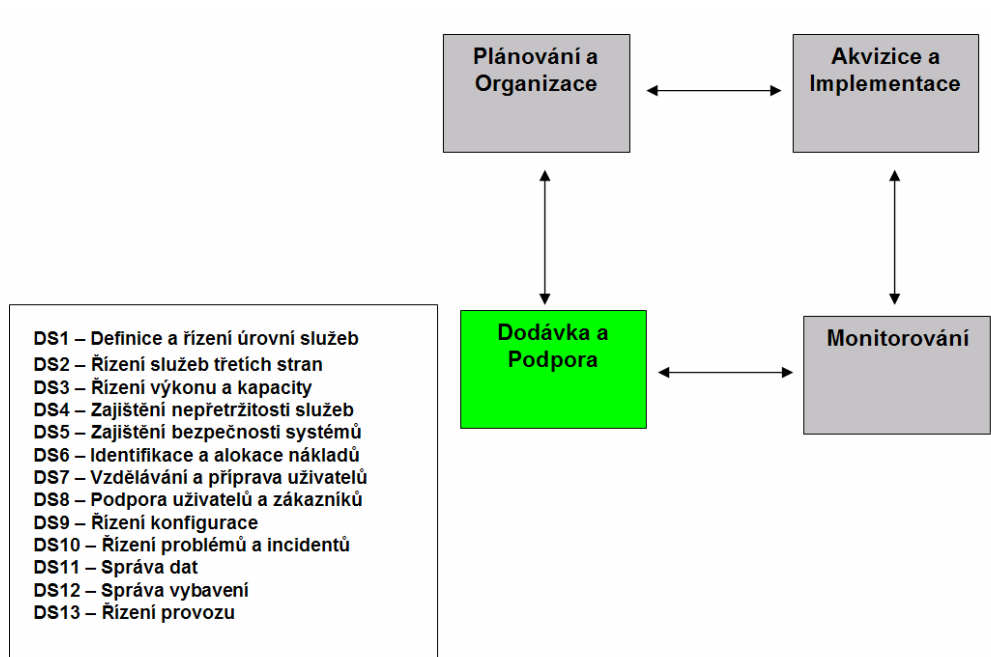
**Obrázek 21: Čtyři domény COBIT [GULDENTOPS 2000]**

Cobit je primárně systém směrnic stanovujících, které procesy, postupy a řídicí mechanismy je třeba implementovat do ICT a jaké výsledky by měly tyto procesy IT přinést. Z tohoto úhlu pohledu používají metodiku COBIT manažeři IT útvarů (v těsné vazbě na IT GOVERNANCE) a auditoři pro ověření správnosti, bezpečnosti a spolehlivosti ICT.



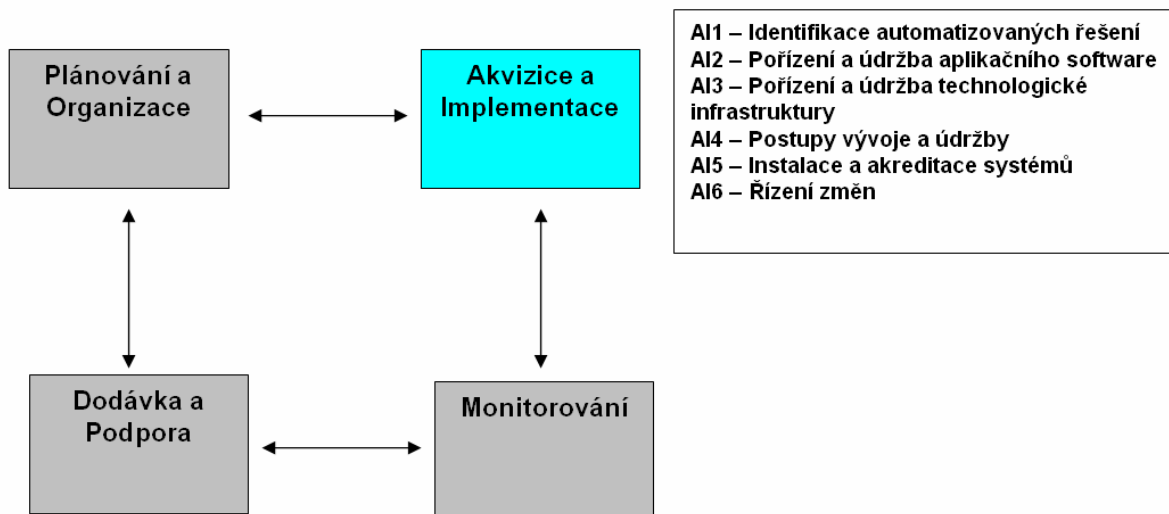
**Obrázek 22: Seznam procesů v doméně „Plánování a Organizace“ [GULDENTOPS 2000]**

COBIT je vhodný jako auditovací nástroj výkonnosti IT. Výkonnost IT je auditována pomocí rozsáhlého kontrolního seznamu, který zahrnuje všechny ICT procesy a stanovuje jejich pravidla a metriky výkonnosti.



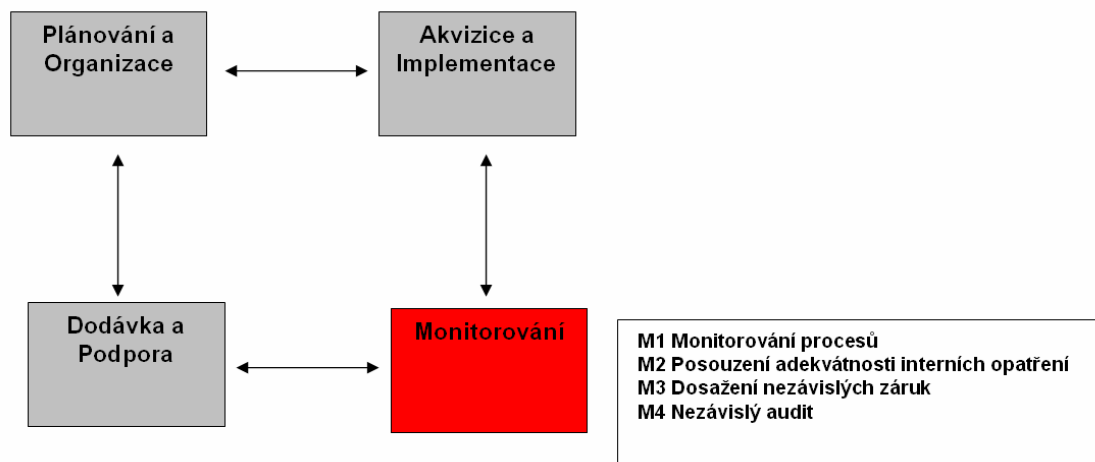
**Obrázek 23: Seznam procesů v doméně „Dodávka a Podpora“ [GULDENTOPS 2000]**

COBIT definuje pomocí vlastnictví procesů odpovědnosti, pravomoci a obsah IT procesů včetně vazeb na kritické faktory, podnikové cíle. Formalizuje také procesní řízení a zamezuje absence nebo nejasnosti v odpovědnostech.



**Obrázek 24: Seznam procesů v doméně „Akvizice a implementace“ [GULDENTOPS 2000]**

COBIT obsahuje komplexní návrh výkonnostních i výsledkových ukazatelů/metrik pro měření výsledků IT:



**Obrázek 25: Seznam procesů v doméně „Monitorování“ [GULDENTOPS 2000]**

COBIT neobsahuje explicitní specifikaci vstupních a výstupních dokumentů/formulářů. Ty je třeba definovat při dodržení obsahu procesu ve vazbě na cíle a předpoklady. COBIT také neobsahuje detailní návrh pracovních toků a detailních rolí.

#### 6.5.4. Proces – příklad

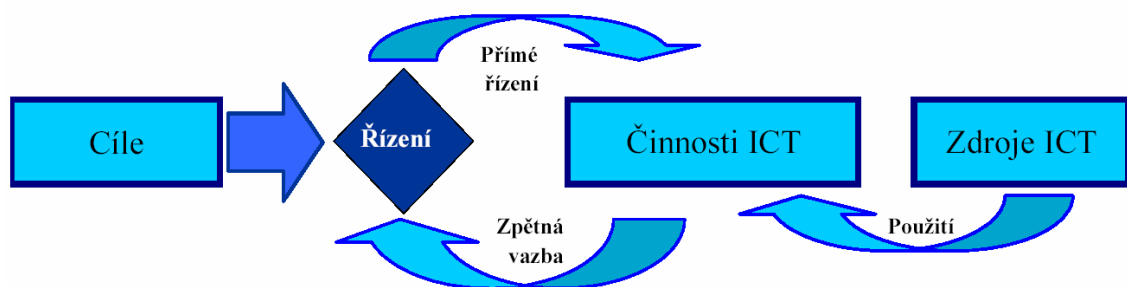
COBIT neobsahuje design jednotlivých procesů. Obsahuje pouze jejich identifikaci, cíl řízení a metriky pro jejich audit.

#### 6.5.5. Implementace COBIT

Implementace COBIT je velmi obtížná, neboť COBIT tvoří pouze doporučení, jak má IT fungovat a jak ověřit, že tomu tak je. Metodika obsahuje 34 procesů vyšší úrovně, které je třeba vyprojektovat a implementovat.

#### 6.5.6. Programme management

Metodika COBIT nepokrývá programové řízení; pokrývá řízení IT útvarů na strategické úrovni.



Obrázek 26: Řízení IT [GULDENTOPS 2000]

### 6.5.7. COBIT a realizace provozních projektů

COBIT nedefinuje terminologii, aktivity, role, vstupy, výstupy a proto je nevhodný pro řízení nebo realizaci provozních projektů.

### 6.5.8. Vyhodnocení aplikovatelnosti metodiky COBIT pro potřeby KC

Metodika COBIT slouží primárně pro business managery a auditory jako nástroj pro ověření kvality poskytovaných služeb. Je vhodná pro řízení IT na strategické úrovni. Pomocí úrovní CMMi pomáhá identifikovat manažerům kvalitu ICT služeb a dodává doporučení, pro přechod na vyšší úroveň. Pro potřeby této práce je metodika nevhodná.

<b>DISCIPLÍNA/METODIKA</b>	<b>COBIT</b>
<i>Programme management</i>	<i>Nevhodná</i>
<b>Supportní projekty</b>	<b>N/A</b>
<i>Management provozních projektů</i>	<i>Nevhodná</i>
<i>Správa prostředí</i>	<i>Nevhodná</i>
<i>Správa požadavků</i>	<i>Nevhodná</i>
<i>Analýza a Design</i>	<i>Nevhodná</i>
<i>Oprava chyb/Vývoj</i>	<i>Nevhodná</i>
<i>Testování</i>	<i>Nevhodná</i>
<i>Nasazení</i>	<i>Nevhodná</i>

## **6.6. CORTEX [COR]**

### **6.6.1. Úvod**

*CORTEX vznikl v roce 1975 s cílem zachytit vnitřní procesy společnosti LogicaCMG (dále jen LCMG). Framework těchto procesů pokrývá jak procesy spjaté s dodávkou výstupů zákazníkovi, tak i nezbytnou množinu aktivit spjatých s vnitřním během společnosti. Procesy CORTEX prochází kontinuálním reengineeringem.*

### **6.6.2. Metamodel**

Metodika CORTEX nemá metamodel, nicméně je strukturovaná do 13 klíčových skupin procesů, tzv. Key process area (KPA's).

### **6.6.3. Struktura CORTEX**

Metodika Cortex je složena ze 13 klíčových skupin procesů. Každá skupina je reprezentována množinou aktivit a výstupů, které je třeba v rámci procesu realizovat. Metodika také obsahuje i šablony výstupů.

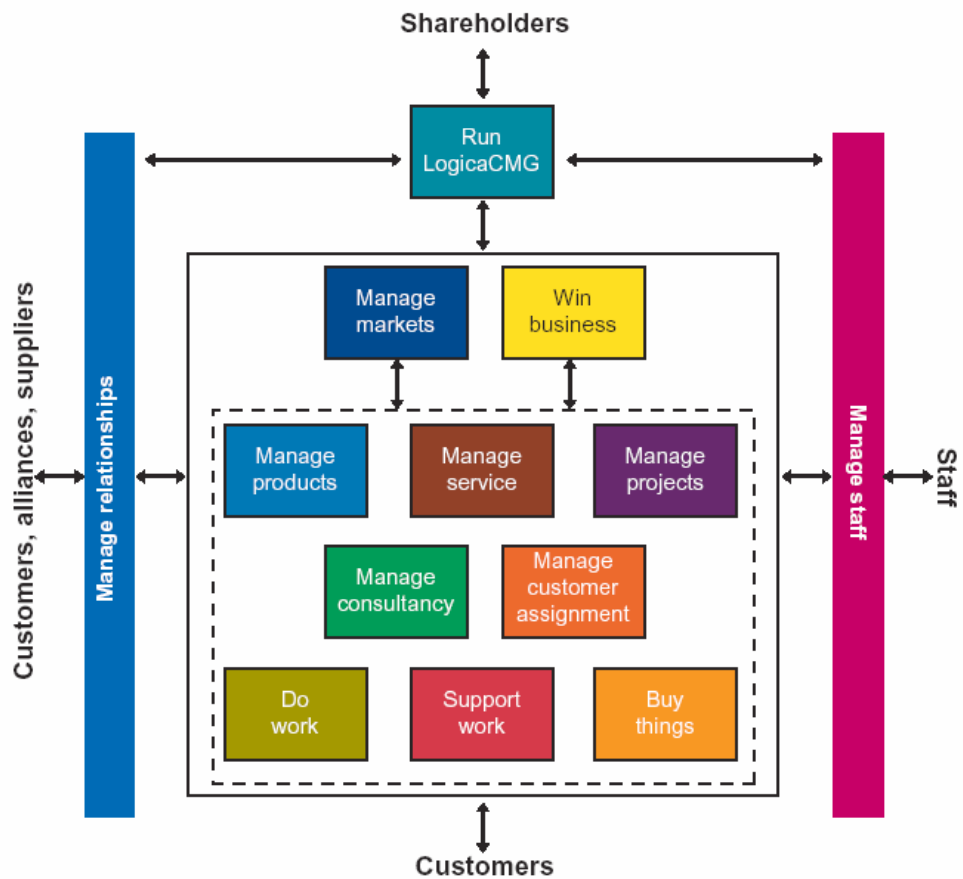
### **6.6.4. Popis procesních oblastí [COR]**

- Logica CMG (Run LogicaCMG).
  - o Definice zásady, strategie, operací, chování a kultury LCMG.
  - o Záležitosti jak např. budovat novou pobočku, definovat její cíl, marketing, finance, management znalostí, ale také práci s „klasifikovanými“ informacemi.
- Řízení lidských zdrojů (Manage Staff).
  - o Procesy spjaté s výběrem, zaměstnáním, přiřazováním na pozici, školením a růstem.
- Řízení vztahů (Manage Relationships).
  - o Procesy spjaté s account managementem a klient managementem.



- Tato oblast také pokrývá programme management.
- Řízení trhů (Manage Markets).
  - Pokrývá segmentaci trhů, marketing atd.
- Obchod (Win Business).
  - Proces zahrnující identifikaci zakázky, konceptuálního designu řešení, řízení nabídky.
- Správa produktu (Manage products).
  - Procesy poskytování, správy a rozvoje produktů.
- Správa služeb (Manage service).
  - Procesy poskytování služeb od podpory produktů, management aplikací a outsourcing.
  - Produktová, call a datová centra.
- Řízení projektů (Manage projects).
  - Procesy projektového řízení.
  - Řízení očekávání zákazníka.
- Řízení konzultací (Manage consultancy).
  - Procesy spjaté s poskytováním konzultačních služeb.
- Výběr pracovníků do týmu zákazníka (Manage customer assignment)
  - Procesy a podmínky pro výběr a uvolnění pracovníků do externích týmů.
  - Zabezpečení výběru „vhodně zkušených a znalých“ pracovníků.
- Realizace (Do work).
  - Skupina procesů realizace od analýzy po nasazení IS.
- Podpůrné procesy (Support work).
  - Rodina podpůrných procesů pro externí i interní zákazníky, help desk, řízení změn, řízení konfigurací, řízení dokumentů, řízení kvality atd.
- Nákup (Buy things).
  - Zprostředkování nebo nákup zboží a služeb.
  - Procesy pro vyhledávání, vyhodnocení a řízení subdodavatelů.

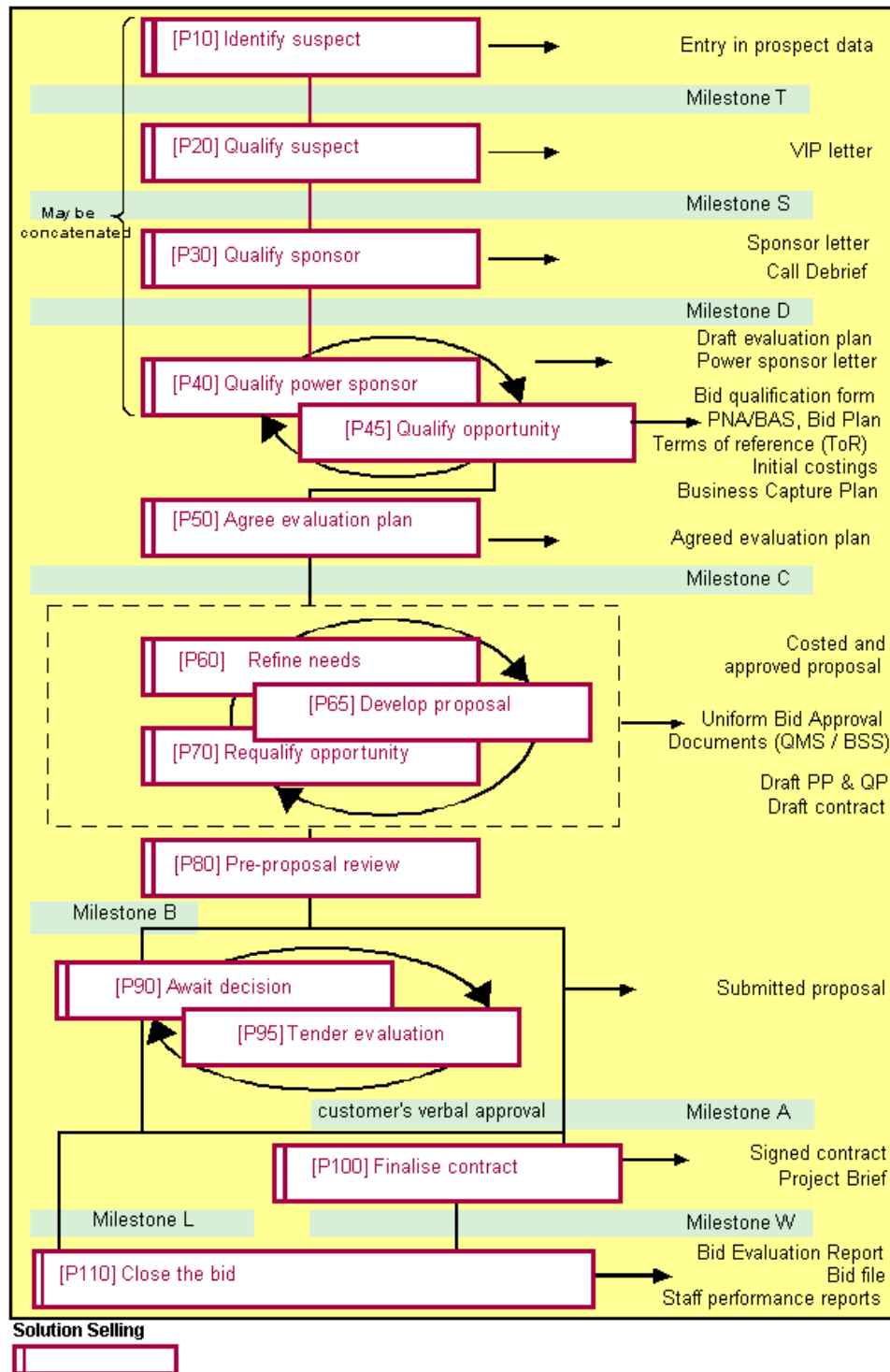
## The Cortex business management system



Obrázek 27: Struktura CORTEX [COR]

### 6.6.5. Proces - příklad

Níže uvedené schéma zachycuje detailní rozpad procesu „Obchod“ na jednotlivé kroky a jejich výstupy. Každý krok je strukturovaně popsán, jsou definovány jeho výstupy a šablony těchto výstupů. Každý proces je také doplněn guideline.



**Obrázek 28: Detail procesu Obchod (Win Business) [COR]**

### **6.6.6. Implementace CORTEX**

CORTEX je interním předpisem společnosti LCMG, přičemž není v zájmu LCMG jej externě uvolnit pro komerční využití. V rámci interní implementace je přípustné upravit procesy a šablony vzhledem k potřebám lokálního prostředí.

### **6.6.7. CORTEX a programme management**

CORTEX pokrývá programme management, ačkoli ne v celé šíři. V pojetí této metodiky je programme management chápán jako řízení několika projektů pro jednoho zákazníka s cílem balancovat přidanou hodnotu a zisk pro zákazníka.

Tradiční pojetí programme managementu ještě vyžaduje vzít v potaz provázanost projektů, sdílení zdrojů, externí a interní závislosti. Toto tradiční pojetí není metodikou CORTEX pokryto.

Obecně jsou doporučovány čtyři základní kroky pro založení programme managementu (dle metodiky CORTEX) [COR]:

1. Definice konceptu.
2. Přípravení Business Case.
3. Schválení programme managementu.
4. Vytvoření programme managementu.

### **6.6.8. CORTEX a realizace provozních projektů**

Cílem této kapitoly je analyzovat vhodnost metodiky pro aplikaci při realizaci provozních projektů.

Metodika CORTEX poskytuje fragmentární informace ohledně procesů vývoje software, provozu i správy. Nicméně při jejich detailní analýze autor zjistil, že procesy provozu a správy IS jsou definovány na vysoké, obecné úrovni. U těchto procesů také chybí guidelines i šablony výstupů.

Kvalita designu provozních procesů byla jedním z důvodů, proč vznikla tato práce.

## 6.6.9. Vyhodnocení aplikovatelnosti metodiky

CORTEX je kvalitní, propracovaná metodika s ohledem na detail a výstupy. Guidelines pomáhají nově zainteresovaným ve snadné orientaci. Z pohledu zkoumaného problému je třeba zmínit následující – metodika dostatečně detailně nepokrývá jednotlivé dílčí procesy, aktivity a výstupy pro provozní projekty. Její interpretace programme managementu je omezená – tj. metodika nezohledňuje provázanost projektů, sdílení zdrojů, jejich externí a interní závislosti. Přesto autor vidí vysokou použitelnost této metodiky pro další dopracování.

DISCIPLÍNA/METODIKA	CORTEX
<i>Programme management</i>	<i>Nutno upravit</i>
<b>Supportní projekty</b>	<b>N/A</b>
<i>Management provozních projektů</i>	<i>Nutno upravit</i>
<i>Správa prostředí</i>	<i>Nutno upravit</i>
<i>Správa požadavků</i>	<i>Nutno upravit</i>
<i>Analýza a Design</i>	<i>Nutno upravit</i>
<i>Oprava chyb/Vývoj</i>	<i>Nutno upravit</i>
<i>Testování</i>	<i>Nutno upravit</i>
<i>Nasazení</i>	<i>Nutno upravit</i>

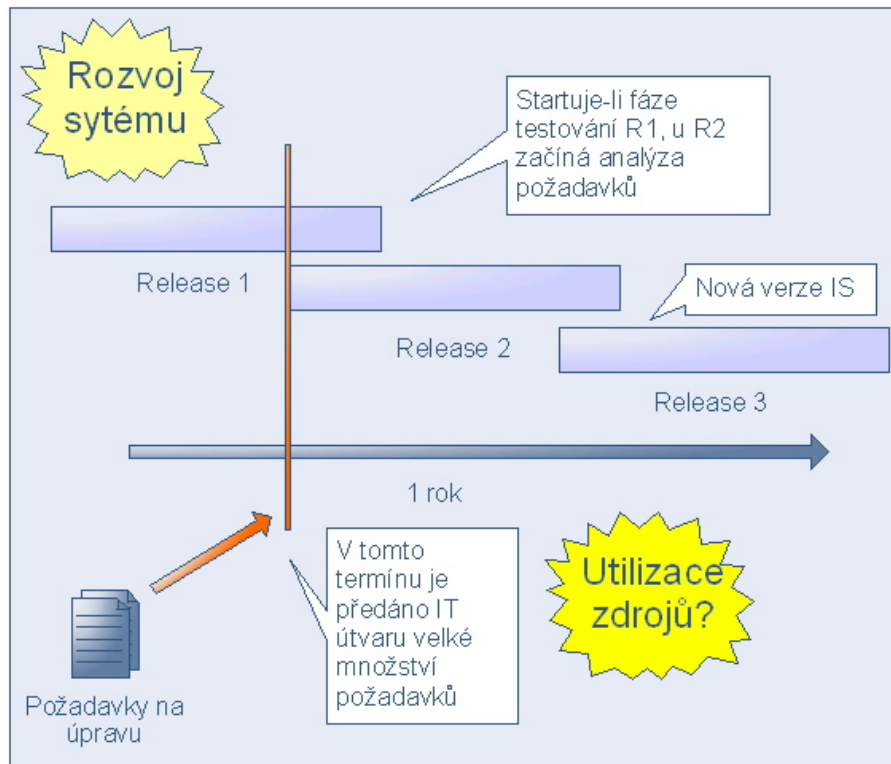
## 6.7. Project Management Methodology – PMM [PMM]

### 6.7.1. Úvod

Metodika Project Management Methodology (PMM) byla vytvořena společností UNICORN v letech 2003/2004 pro velký, nadnárodní finanční ústav. Vznikla v období, kdy krátce předtím tento ústav převzal nový nadnárodní vlastník s cílem vyřešit klíčové IT problémy tohoto ústavu:

- Nepromyšlenost zadání do detailu.
- Požadavky zpracované s různou granularitou.
- Často si vzájemně odporující zadání.
- Nerealistické odhady nákladů, přínosů.

- Nesplnitelné termíny realizace.
- Požadavky nejsou dodávány rozloženě v čase, ale vždy ve velkém množství k datu uzávěrky.



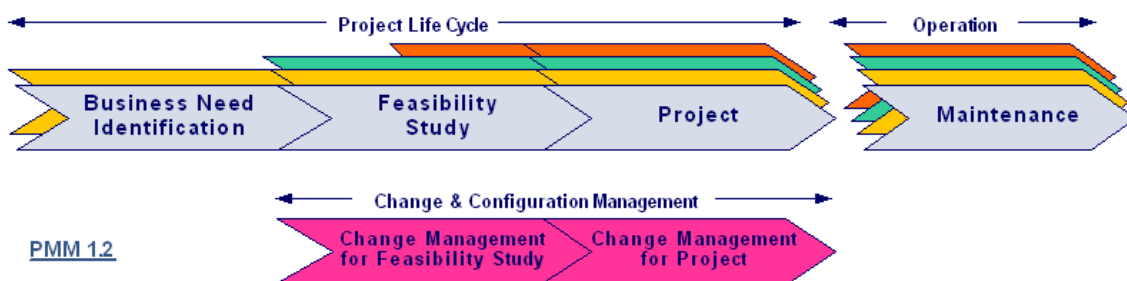
**Obrázek 29: Typický rozvoj systému v produkčním prostředí [MAROUNEK 2004/1]**

Nová metodika vznikla na základě následujících složek:

- Výběr použitelných stávajících předpisů (hlavně pro bankovní oblast).
- Programme management byl převzat z proprietární metodiky mateřské nadnárodní instituce.
- Jednotlivé postupy byly detailně rozpracovány na základě metodiky Rational Unified Process (RUP).
- Provozní procesy a SLA byly inspirovány metodikou COBIT.

## 6.7.2. Metamodel

Metodika je navržena formou intranetovské website, který je provázán hypertextovými odkazy. Jsou zde zachyceny fáze projektů, pracovní postupy, role, výstupy, šablony výstupů, toolmentory a samozřejmě návod, jak s danou metodikou pracovat. Autor se domnívá, že je významně za rámec této práce, metamodel metodiky dále detailněji rozebírat a popisovat.



Obrázek 30: Struktura PMM [PMM]

## 6.7.3. Struktura PMM

Životní cyklus požadavku je rozdělen na čtyři základní etapy:

- Obchodní příležitost.
- Studie proveditelnosti.
- Realizace.
- Provoz a údržba.

První dvě fáze lze chápat jako prefáze projektu, ve kterých je cílem specifikovat požadavky na nový systém, případně rozšíření stávajícího, rozdělit je do jednotlivých projektů (jeden požadavek může ovlivnit několik IS najednou) a u nejrizikovějších z nich, provést elaboraci.

Pod pojmem elaborace je chápáno ověření realizovatelnosti daného požadavku [detailněji pojednává ROYCE 1998, CANTOR 1999 a RUP]. Čím dříve je tato fáze realizována, tím lépe, neboť s jejím skončením riziko neúspěchu projektu (hlavně s technologického hlediska) významně klesá.

Fáze Realizace řeší klasický životní cyklus projektu od analýzy a specifikace požadavků až po nasazení do produkčního prostředí. Zde byla nejvíce využita metodika Rational Unified Process. Daná instituce používá manažerský informační systém EIS a mnoho podpůrných nástrojů společnosti Rational.

Jako největší nevýhodu metodiky Rational Unified Process a všech aplikovaných řešení lze vidět absenci fáze Provoz a údržba. Tato fáze řeší provoz, údržbu a drobný rozvoj již nasazených systémů.

#### **6.7.3.1. Fáze I. - Obchodní příležitost**

Cílem této fáze je zachytit požadavky uživatelů na nový systém nebo úpravu stávajícího systému. Každý pracovník má právo podat jakýkoli námět kontaktní osobě za stranu obchodu, tzv. Single point of contact. Tato role udržuje všechny požadavky na jednom místě a rozhoduje o jejich realizaci či odmítnutí z obchodního úhlu pohledu. Je zodpovědná za stanovení jejich priority a ocenění pracnosti (po konzultaci se stranou IT). V případě, že je požadavek nesmyslný nebo jde proti politice banky, je časově nebo finančně náročný, dochází k jeho revizi. Na základě přehodnocení může dojít k úpravě požadavku nebo případně k zamítnutí. Výstupem této fáze je schválená množina kategorizovaných, oprioritizovaných (z úhlu pohledu zadavatele) a oceněných požadavků.

*Pozn. Ocenění požadavku je provedeno na základě jeho kategorizace – nebo-li s přesností v řádech desítek tisíc, stovek tisíc a milionu korun dle kategorií A, B a C.*

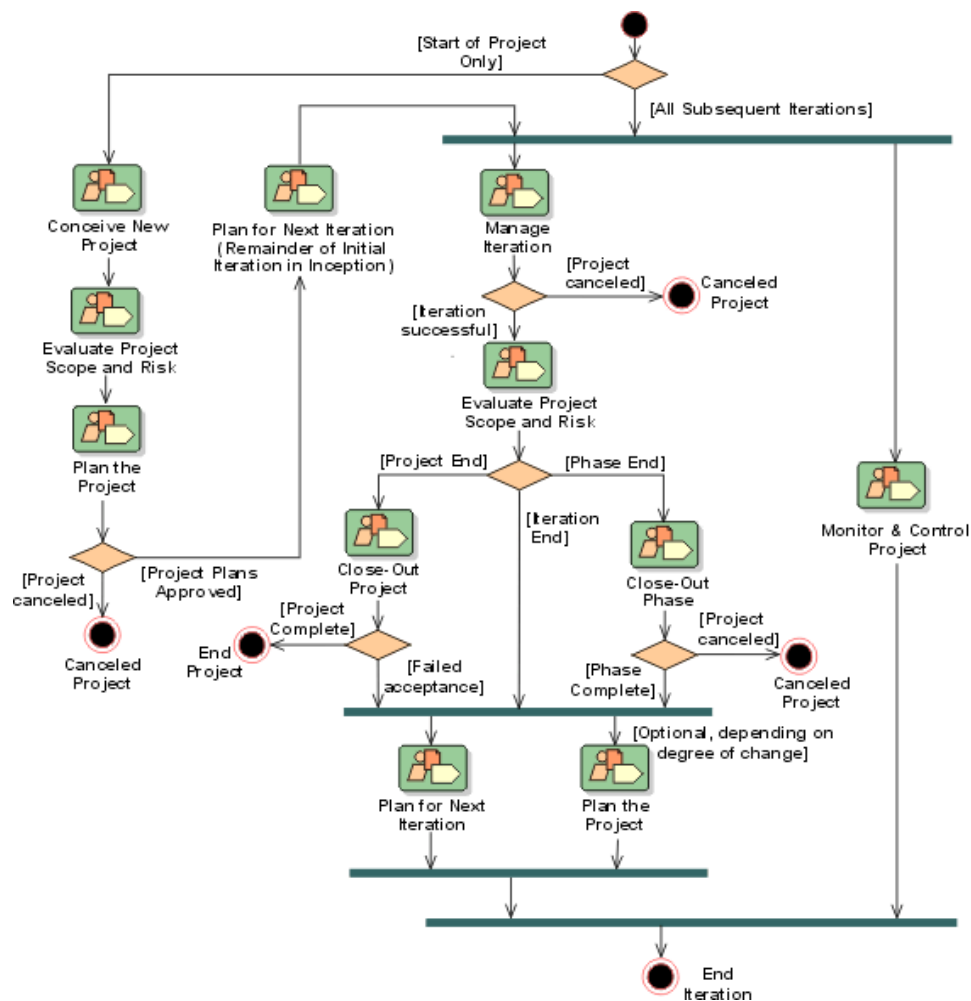


### **6.7.3.2. Fáze II. – Studie proveditelnosti**

Ve této fázi jsou jednotlivé požadavky zrevidovány a rozděleny do kategorií podle stavu náročnosti a rizik realizace. Velkou neznámou a potenciálním problémem do budoucna jsou požadavky, které se jeví na první pohled jako jednoduché, přičemž jejich záležitost spočívá v dopadu do několika systémů. Tato fáze by měla právě takové požadavky pomoci identifikovat, ale osobně zde autor vidí jednu z mála slabín navržené metodiky. Slabina spočívá v tom, že je očekávána detailní znalost požadavku a zároveň detailní, architektonická znalost systémů v bance. Vzhledem k jejich počtu a složitosti hrozí riziko, že budou chybně stanoveny dopady navrhovaného řešení do jednotlivých systémů. Chybný odhad může velmi ztížit koordinaci vývoje jednotlivých verzí různých systémů.

Technologická rizikovost a složitost požadavků je částečně nebo úplně řešena pomocí prototypu [detailněji pojednává ROYCE 1998, CANTOR 1999 a RUP]. Na základě úspěšnosti prototypu, realizovatelnosti požadavků, průniků a dopadů do systémů, rozpočtu, času a dostupných zdrojů jsou požadavky rozděleny do jednotlivých releasů a připravena odpovídající schvalovací dokumentace projektu.

### 6.7.3.3. Fáze III. - Realizace



**Obrázek 31: Fáze Realizace dle metodiky PMM a RUP [ROYCE 1998, RUP]**

Komise schválí projekt a tým začíná jeho realizace. Typ schvalovací komise je stanoven v závislosti na velikosti projektu.

Zadání je roztrženo na tzv. koncepty – menší, snáze realizovatelné, ale hlavně snáze říditelné části a je přiřazeno jednotlivým iteračním týmům. Iterační tým je malý vývojový tým, který realizuje všechny disciplíny softwarového vývoje mimo nasazení výsledného software do produkčního prostředí. Jeho velikost je 4 až 8 lidí, řídí jej vedoucí týmu.

Jeho kapacita je cca 1000 až 1500 ČLH hodin a životnost čtyři až šest týdnů. Cílem iteračního týmu je zrealizovat řešení konceptu. Po realizaci konceptu A, získává tým nový koncept nebo zaniká. Za koordinaci týmů a dohled nad kvalitou jejich výstupu je zodpovědný projektový manažer a jednotka SI (Systémová integrace).

Iterační tým vede vedoucí, který detailněji naplňuje iteraci - životní cyklus konceptu. V kompetenci iteračního týmu je pak detailně analyzovat, navrhnout ve spolupráci s jednotkou SI design, vyrobit, otestovat a dodat řešení konceptu. Výstupem z každého iteračního týmu je iterační (projektová) dokumentace, otestovaná a integrovatelná verze software, testovací skripty, scénáře a jejich vyhodnocení, technická dokumentace.

Jednotka SI je zodpovědná za integraci všech iteračních výstupů dohromady. Tím vzniká nová verze systému. Novou verzi testuje pomocí testů, které dostala od jednotlivých iteračních týmů (testy funkčnosti) a nebo pomocí svých vlastních (zátěžové testy atd.) Jsou-li identifikovány při testech chyby, je rozhodnuto, zda je opravuje iterační tým nebo SI. Pro rozklíčování chyb je použit jednoduchý proces – je-li zřejmé, který tým chybu udělal, je mu zadána k opravení. Není-li toto zřejmé, chybu řeší SI. V případě chybného zadání provádí dospecifikaci a úpravu funkcionality iterační tým.

Je-li výstup úspěšně otestován SI, koncovými uživateli a pilotně ozkoušen, je nasazen na jednu pobočku a živě používán. Pokud i zde nejsou problémy, přechází všechny pobočky postupně na novou verzi. V případě, že se v jakékoli fázi objeví problémy nebo chyby, jsou ihned řešeny a to na úrovni SI nebo iteračního týmu, který je způsobil.

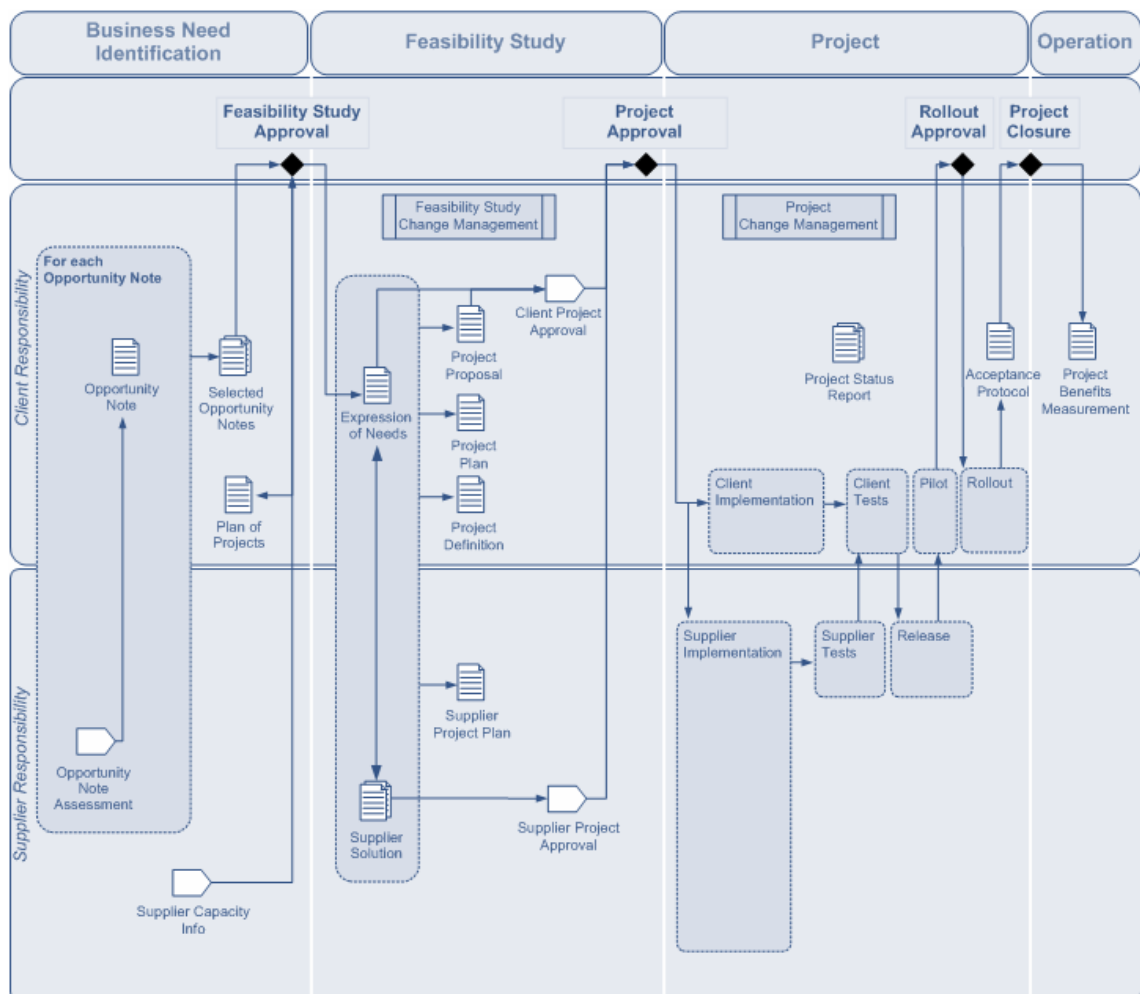
Je v zájmu iteračního týmu, aby dodával co nejkvalitnější řešení, neboť mezi tím, co je jeho výstup přetestován, pilotně testován a nasazen na jednu pobočku, řeší tento tým již další koncept anebo je rozpuštěn do několika dalších týmů. Tím nemá vedoucí týmu k dispozici volné zdroje na opravu chyb. Je předvídatelné, že v každém softwaru jsou a budou chyby, a proto bylo vedoucím týmů doporučeno plánovat kapacitu týmu maximálně na 90%. Zbýlých deset procent slouží jako rezerva, která může být použita právě na opravu chyb předchozích verzí.

Odevzdáním verze odpovídající zadání a jejím nasazením do ostrého provozu, dochází k uzavření projektu. Jako poslední krok je vyhodnocena odpovídající projektová dokumentace a řešení předáno do servisu.

#### **6.7.3.4. Fáze IV. – Provoz a údržba**

Cílem provozu a servisu je podpora uživatelů, oprava chyb a drobný rozvoj a měření přínosu řešení. Z projektového úhlu pohledu je servis a provoz nový projekt, který navazuje na vývojový. Cílem takového pohledu je oddělit pracovníky vývoje od podpory. Pracovníci podpory produkt provozují. V případě, že je nashromážděno dostatečné množství požadavků, je otevřen nový vývojový projekt s cílem vyrobít novou verzi IS (v souladu s výše popsány mi kroky).

Největší výhodou oddělení pracovníků podpory a vývoje spočívá v tom, že vývojáři mohou efektivně pracovat na jiných projektech a nejsou neplánovaně zatěžováni řešením chyb a oprav z předchozích projektů. Takový proces předpokládá, že existuje odpovídající technická dokumentace řešení a že pracovníci podpory byli s produktem dostatečně seznámeni.



Obrázek 32: Životní cyklus projektu dle PMM [PMM]

Pozn. Fáze Provoz a údržba není v době psaní této práce stále definována.

#### 6.7.4. PMM a programme management

Metodika PMM neřeší problematiku programme managementu.

#### 6.7.5. Metodika a realizace provozních projektů

Metodika PMM dosud nepokrývá problematiku provozních projektů.

## 6.7.6. Vyhodnocení aplikovatelnosti metodiky

Metodika je zatím nevhodná pro použití jednotlivých procesů pro potřeby KC. Pro potřeby realizace provozních projektů je vhodnější použít procesy přímo z metodiky Rational Unified Process (RUP).

DISCIPLÍNA/METODIKA	PMM
<i>Programme management</i>	<i>Nevhodná</i>
<b>Supportní projekty</b>	<b>N/A</b>
<i>Management provozních projektů</i>	<i>Nevhodná</i>
<i>Správa prostředí</i>	<i>Nevhodná</i>
<i>Správa požadavků</i>	<i>Nutno upravit</i>
<i>Analýza a Design</i>	<i>Nutno upravit</i>
<i>Oprava chyb/Vývoj</i>	<i>Nevhodná</i>
<i>Testování</i>	<i>Nevhodná</i>
<i>Nasazení</i>	<i>Nevhodná</i>

## 6.8. Rational Unified Process – RUP [RUP]

### 6.8.1. Úvod

Metodika Rational Unified Process (RUP) je komerčním produktem společnosti IBM. Jedná se o komplexní sadu konceptů, postupů, nástrojů a technik používaných v softwarovém inženýrství. RUP vznikl na základě podrobné analýzy mnoha projektů a je založen na rozsáhlých praktických poznatcích. RUP představuje disciplinovaný přístup k přidělování úkolů a odpovědností v rámci vývojové organizace. Metodika definuje takzvaných „šest nejlepších praktik softwarového vývoje“ [RUP].

#### 6.8.1.1. Iterativní vývoj

Dnešní složitost softwarových produktů neumožňuje jít cestou definování celého problému (včetně jeho kontextu), pak navrhnout celé řešení, vytvořit software a nakonec produkt testovat.

Možným východiskem je iterativní přístup, který pomocí postupných kroků, které se několikrát dokola opakují během životního cyklu tvorby software.

Význam iterativního přístupu spočívá ve schopnosti včasného odhalení rizik v každém stádiu projektu. Každá iterace končí spustitelnou verzí (buildem), přičemž časté ověřování stavu projektu zajišťuje, že projekt probíhá v daném časovém rozvrhu.

#### **6.8.1.2. Správa požadavků**

Rational Unified Process popisuje, jak zajistit, organizovat a zdokumentovat požadovanou funkcionalitu a nutná omezení, mapuje a dokumentuje výhody i nevýhody variant a provedená rozhodnutí. Tímto přístupem snadno zachycuje a sladuje obchodní požadavky.

#### **6.8.1.3. Komponentová architektura**

RUP popisuje, jak navrhnout flexibilní architekturu založenou na komponentách, přičemž komponenty chápe jako netriviální moduly (subsystémy), které podporují jasně definovanou funkci.

#### **6.8.1.4. Vizuální modelování**

Pomocí vizuální abstrakce (reprezentované UML – Unified Modeling Language), pomáhá RUP spojit různé aspekty tvorby softwaru. Například zjistit, zda jsou prvky systému kompatibilní a udržovat konzistenci mezi návrhem a jeho implementací.

#### **6.8.1.5. Řízení kvality**

RUP pomáhá při plánování, návrhu, implementaci, spuštění a vyhodnocení. Hodnocení kvality je zabudováno do procesu, do každé z činností, které se týkají všech zúčastněných, a používá objektivní měřítko a kritéria (tzn. není oddělenou činností).

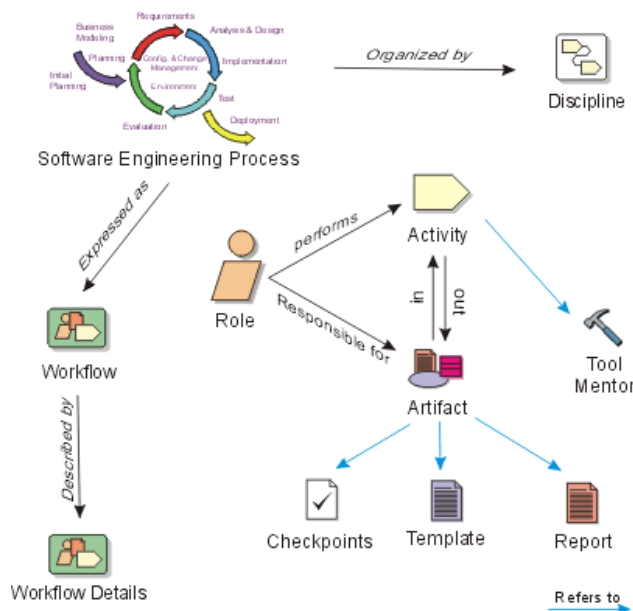
### 6.8.1.6. Řízení změn

RUP popisuje, jak kontrolovat, sledovat a monitorovat změny a umožnit úspěšný iterativní vývoj. Poskytuje návod, jak vytvořit bezpečné pracovní prostředí pro každého vývojáře tím, že bude izolován od změn provedených v jiných pracovních prostředích, a také tím, že tyto změny všech softwarových produktů (modelů, kódů, dokumentů atd.) budou řízeny.

Těchto „šest nejlepších praktik softwarového vývoje“ zvyšuje produktivitu týmu, protože každému jeho členovi poskytuje bázi potřebných znalostí, a to pomocí návodů, šablon a manuálů pro všechny kritické činnosti vývoje. Všichni členové týmu používají společný jazyk a mají stejný pohled na proces vývoje software.

### 6.8.2. Metamodel

Softwarový vývoj stojí podle RUP na čtyřech základních pilířích - pracovníci (Workers), činnosti (Activities), pracovní procesy (Workflows) a výstupy (Artefacts). Tím poskytuje flexibilní rámec pro metodický popis organizační struktury a pracovních procesů a postupů v dané organizaci.



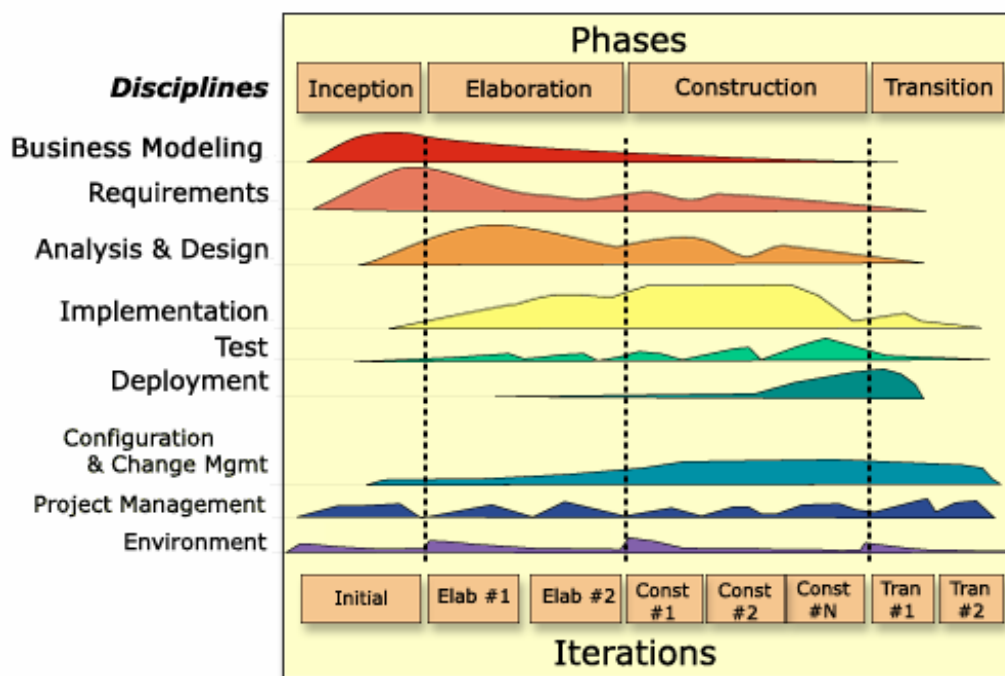
Obrázek 33: Metamodel RUP [RUP]



### 6.8.3. Základní struktura procesů

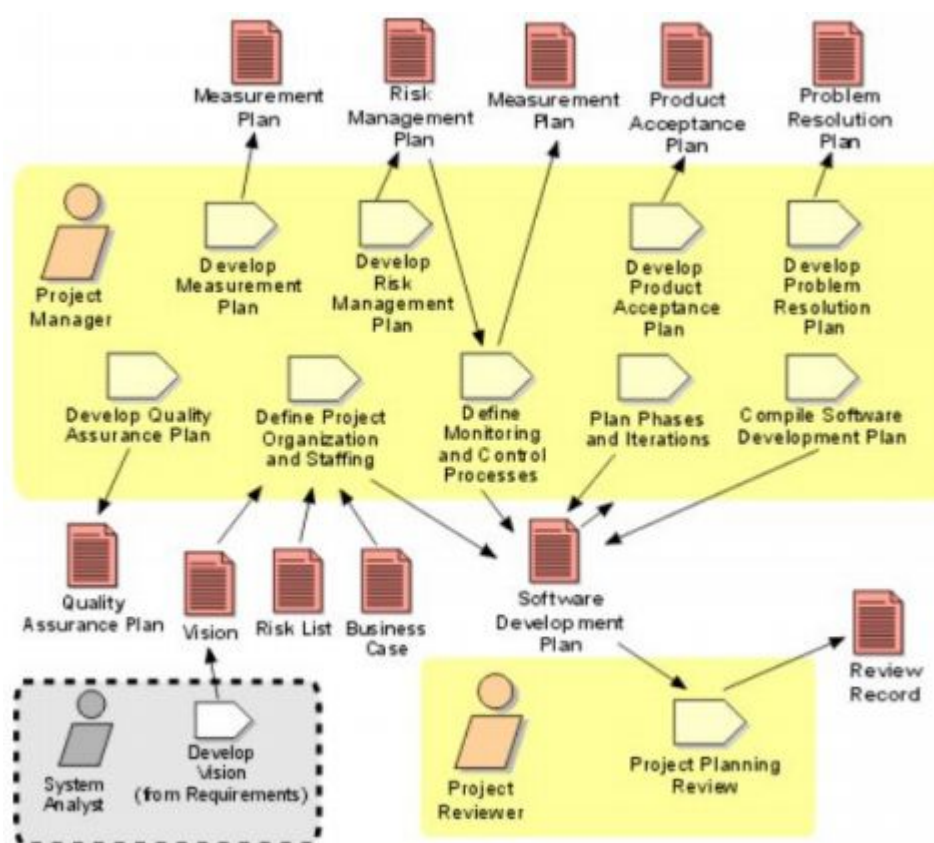
Metodika RUP pokrývá životní cyklus dodávky jednoho projektu a to od jeho počátku až po nasazení do produkčního prostředí. Metodika nepokrývá řešení záruky, servis ani programme management.

Pomocí konceptů a guidelines velmi přehledně vysvětluje oblasti softwarového vývoje.



Obrázek 34: Přehled procesů RUP [RUP]

#### 6.8.4. Ukázka procesu



Obrázek 35: Detail disciplíny Project Management [RUP]

RUP definuje základní postup při realizaci projektu na základě stanovení cíle projektu (jaké výstupy má projekt dodat), plánu projektu (tedy kdy má jaký výstup vzniknout), definice kompetencí (kdo je za jaký výstup odpovědný) a plánu zdrojů (jaké zdroje je vhodné a možné využít).

Dodržování předem stanovené metodiky umožňuje také průběžné hodnocení projektu. RUP stanovuje projektové milníky, které dopředu vytyčí, jaké výstupy mají být v dané době dokončeny. Snadno pak lze poznat, zda běží vše podle plánu, nebo dochází ke skluzu. Plnění termínů je totiž třeba mít neustále pod kontrolou. U většiny projektů jsou smluvně stanoveny termíny určitých výstupů a chce-li si dodavatel u klienta udržet pověst seriózního dodavatele, měl by je respektovat.

### 6.8.5. Metodika a programme management

Metodika RUP pokrývá životní cyklus dodávky jednoho projektu od Business modelování až po jeho nasazení do produkce. Metodika nepokrývá oblast programme managementu.

### 6.8.6. Metodika a realizace provozních projektů

Metodika RUP nepokrývá realizaci provozních projektů. Projekt pro ni končí jeho uzavřením po nasazení do produkce.

### 6.8.7. Vyhodnocení aplikovatelnosti metodiky

Z úhlu pohledu této práce, metodika RUP nepokrývá ani programme management ani procesy spjaté s realizací provozních projektů. Přesto při realizaci supportu jsou v RUP disciplíny softwarového vývoje, kde je možné se inspirovat (např. analýza, design atd.) při návrhu procesů KC.

DISCIPLÍNA/METODIKA	RUP
<i>Programme management</i>	<i>Nevhodná</i>
<b>Supportní projekty</b>	<b>N/A</b>
<i>Management provozních projektů</i>	<i>Nevhodná</i>
<i>Správa prostředí</i>	<i>Nutno upravit</i>
<i>Správa požadavků</i>	<i>Nutno upravit</i>
<i>Analýza a Design</i>	<i>Nutno upravit</i>
<i>Oprava chyb/Vývoj</i>	<i>Nutno upravit</i>
<i>Testování</i>	<i>Nutno upravit</i>
<i>Nasazení</i>	<i>Nutno upravit</i>

## 6.9. Porovnání metodik

Níže uvedený přehled shrnuje výsledky zjištěné v rámci analýzy. Autor každé zkoumané vlastnosti metodiky přiřadil hodnotu v rozpětí jedna až tři, kdy hodnotu jedna lze interpretovat jako „vlastnost z dané metodiky je vhodné použít pro potřeby této práce“ a hodnotu tři lze interpretovat jako „vlastnost z dané metodiky není vhodné použít pro potřeby této práce“.

Pro usnadnění výzkumu a jeho vyhodnocení autor každé zkoumané vlastnosti přiřadil stejnou váhu neboť se domnívá, že dopad z takové simplifikace na vyhodnocení metodik je zanedbatelný.

POROVNÁNÍ METODIK					
DISCIPLÍNA/METODIKA	ITIL	COBIT	COR- TEX	PMM	RUP
<b><i>Programme management</i></b>	3	3	2	3	3
Definice programme managementu	3	3	1	3	3
Obecný proces	3	3	1	3	3
Konkrétní procesy, aktivity	3	3	2	3	3
Výstupy	3	3	2	3	3
Doporuční pro implementaci (vytvoření KC)	3	3	2	3	3
<b>Provozní projekty</b>					
<b><i>Management provozních projektů</i></b>	3	3	2	3	3
Obecná definice procesu	2	3	2	3	3
Konkrétní definice procesu	3	3	3	3	3
Definice aktivit (činností, kroků) a jejich popis	2	3	2	3	3
Definice výstupů a jejich popis	2	3	2	3	3
Přítomnost šablon výstupů	3	3	2	3	3
Využitelnost procesu bez dodatečných úprav	3	3	2	3	3
Implementovatelnost procesu bez dodatečných úprav	3	3	2	3	3
<b><i>Správa prostředí</i></b>	3	3	2	3	2
Obecná definice procesu	2	3	2	3	2
Konkrétní definice procesu	2	3	3	3	2
Definice aktivit (činností, kroků) a jejich popis	2	3	2	3	2
Definice výstupů a jejich popis	2	3	2	3	2
Přítomnost šablon výstupů	2	3	3	3	2
Využitelnost procesu bez dodatečných úprav	3	3	3	3	3

Implementovatelnost procesu bez dodatečných úprav	3	3	3	3	3
<b><i>Správa požadavků</i></b>	2	3	2	2	2
Obecná definice procesu	2	3	2	3	1
Konkrétní definice procesu	2	3	2	2	2
Definice aktivit (činností, kroků) a jejich popis	2	3	2	2	2
Definice výstupů a jejich popis	2	3	2	2	2
Přítomnost šablon výstupů	2	3	2	2	2
Využitelnost procesu bez dodatečných úprav	3	3	3	3	3
Implementovatelnost procesu bez dodatečných úprav	3	3	3	3	3
<b><i>Analýza a Design</i></b>	2	3	2	2	2
Obecná definice procesu	2	3	2	2	1
Konkrétní definice procesu	2	3	2	2	1
Definice aktivit (činností, kroků) a jejich popis	2	3	2	2	1
Definice výstupů a jejich popis	2	3	2	2	2
Přítomnost šablon výstupů	2	3	2	2	2
Využitelnost procesu bez dodatečných úprav	3	3	3	3	2
Implementovatelnost procesu bez dodatečných úprav	3	3	3	3	2
<b><i>Oprava chyb/Vývoj</i></b>	2	3	2	3	2
Obecná definice procesu	2	3	2	3	1
Konkrétní definice procesu	2	3	2	2	2
Definice aktivit (činností, kroků) a jejich popis	2	3	3	3	2
Definice výstupů a jejich popis	2	3	2	3	2
Přítomnost šablon výstupů	2	3	2	3	2
Využitelnost procesu bez dodatečných úprav	3	3	3	3	3
Implementovatelnost procesu bez dodatečných úprav	3	3	3	3	3
<b><i>Testování</i></b>	2	3	2	3	2
Obecná definice procesu	2	3	3	3	2
Konkrétní definice procesu	2	3	3	3	2
Definice aktivit (činností, kroků) a jejich popis	2	3	3	3	2
Definice výstupů a jejich popis	2	3	2	2	2
Přítomnost šablon výstupů	2	3	2	2	2
Využitelnost procesu bez dodatečných úprav	3	3	3	3	3
Implementovatelnost procesu bez dodatečných úprav	3	3	3	3	3
<b><i>Nasazení</i></b>	2	3	2	3	2

Obecná definice procesu	2	3	2	3	2
Konkrétní definice procesu	2	3	2	3	2
Definice aktivit (činností, kroků) a jejich popis	2	3	2	2	2
Definice výstupů a jejich popis	2	3	2	2	2
Přítomnost šablon výstupů	2	3	2	2	2
Využitelnost procesu bez dodatečných úprav	3	3	3	3	3
Implementovatelnost procesu bez dodatečných úprav	3	3	3	3	3

*Pozn. - Komentář k tabulce:*

- 1 – Metodiku je vhodné použít
- 2 – Proces je nutné upravit
- 3 – Metodiku není vhodné použít

## 6.10. Vyhodnocení metodik

<b>Vyhodnocení metodik</b>	<b>ITIL</b>	<b>COBIT</b>	<b>CORTEX</b>	<b>PMM</b>	<b>RUP</b>
SUMA Provozní projekty	130	168	131	151	<b>125</b>
SUMA Programme management	18	18	<b>10</b>	18	18
Celkem	148	186	141	169	143

Na základě provedené analýzy autor konstatuje, že nejvhodnější metodikou pro vytváření kompetenčního centra je metodika CORTEX. Nejvhodnější metodikou pro design procesu provozních projektů je metodika Rational Unified Process.

## 6.11. Vyhodnocení metodik (zadání práce) – programme management

<b>DISCIPLÍNA/METODIKA</b>	<b>CORTEX</b>
<b>Programme management</b>	<b>Nutno upravit</b>
Definice programme managementu	Vhodná
Obecný process	Vhodná
Konkrétní procesy, aktivity	Nutno upravit
Výstupy	Nutno upravit
Doporučení pro implementaci (vytvoření KC)	Nutno upravit

Tato tabulka zobrazuje stav programme managementu metodiky CORTEX. Položky označené „nutno upravit“ jsou zadáním pro design jednotlivých procesů v rámci této práce. Velikost úprav závisí na konkrétní aplikaci projektu.

## 6.12. Vyhodnocení metodik (zadání práce) – Provozní projekty

DISCIPLÍNA/METODIKA	RUP
<i>Programme management</i>	<i>Nevhodná</i>
<b>Provozní projekty</b>	<b>N/A</b>
<i>Management provozních projektů</i>	<i>Nevhodná</i>
<i>Správa prostředí</i>	<i>Nutno upravit</i>
<i>Správa požadavků</i>	<i>Nutno upravit</i>
<i>Analýza a Design</i>	<i>Nutno upravit</i>
<i>Oprava chyb/Vývoj</i>	<i>Nutno upravit</i>
<i>Testování</i>	<i>Nutno upravit</i>
<i>Nasazení</i>	<i>Nutno upravit</i>

Tato tabulka zobrazuje stav procesů pokrývajících realizaci provozních projektů dle metodiky Rational Unified Process. Položky označené „nutno upravit“ jsou zadáním pro design jednotlivých procesů v rámci této práce.

## 7. Aplikace programme managementu

Obecně jsou doporučovány čtyři základní kroky pro založení programme managementu (dle metodiky CORTEX) [COR]:

1. Definice konceptu.
2. Příprava případové studie.
3. Schválení programme managementu.
4. Vytvoření programme managementu.

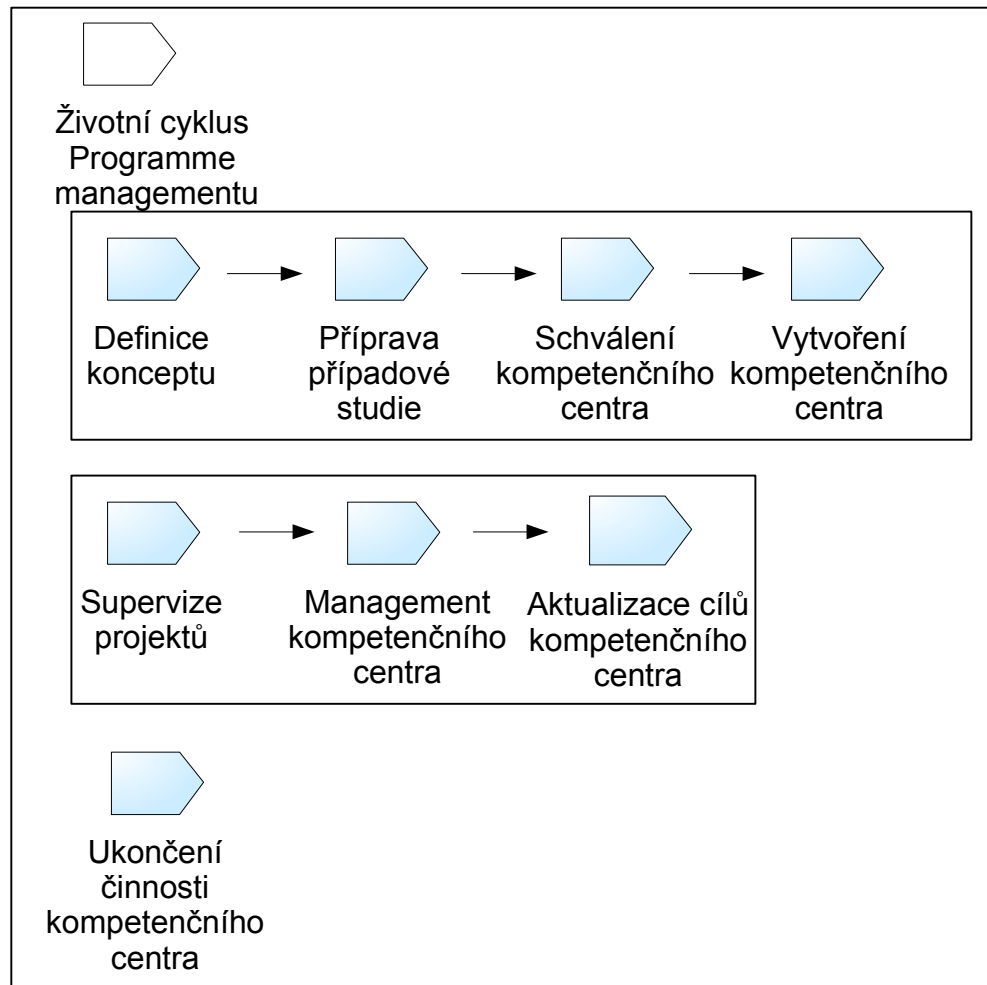
Je-li programme schválen a nastaven, je třeba zajistit jeho realizaci pomocí následujících procesů:

1. Řízení cílů programme managementu.
2. Koordinace aktivit programme managementu.
3. Supervize projektů.

Realizace výše uvedených skupin procesů lze vyřešit pomocí:

- Aktualizace vize programme managementu.
- „Trackování“ přínosů.
- Podpora realizace přínosů.
- Monitoring SLA.
- Motivace.
- Řízení externích vztahů.
- Řízení problémů.
- Řízení očekávání zákazníka.
- Řízení projektů.

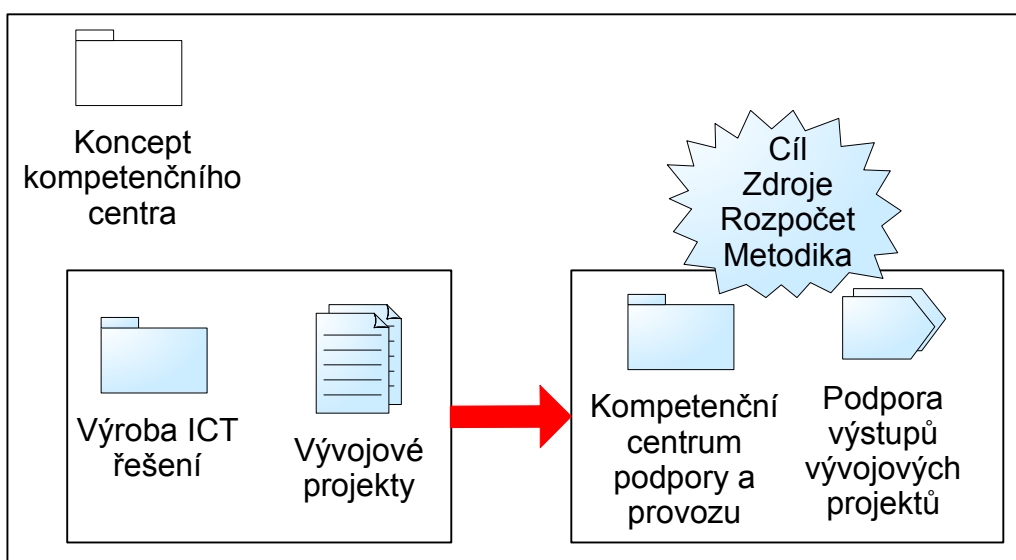




**Obrázek 36: Životní cyklus programme managementu**

## 8. Návrh kompetenčního centra provozních projektů

Pod pojmem kompetenční centrum lze chápat dedikovanou logickou divizi (nikoli organizační), která má svůj cíl, rozpočet, zdroje a metodiku fungování. Na konceptuální úrovni, jsou si kompetenční centra podobná. Rozdíl je v konkrétní části metodiky – realizační procesy.

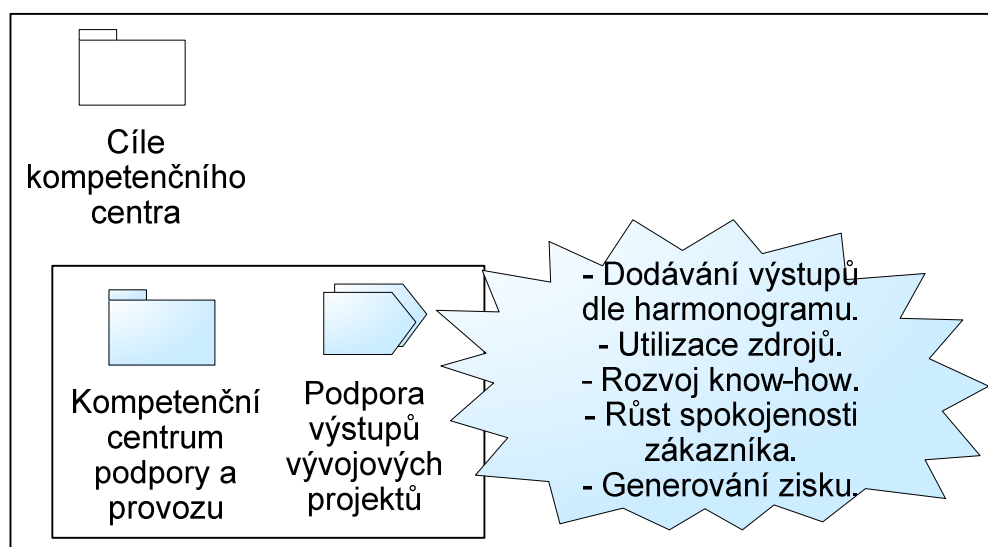


**Obrázek 37: Konceptuální pohled na kompetenční centrum**

Je zřejmé, že kompetenční centrum provozních projektů bude v rámci supportu realizovat jiné procesy, činnosti a výstupy než-li kompetenční centrum pro např. řízení projektů nebo realizaci staveb na klíč. Autor této práce se domnívá, že níže uvedené procesy jsou dostatečně univerzální pro všechna kompetenční centra vyjma procesů realizačních. Není cílem této práce navrhnout procesy na tak vysoké, teoretické bázi, aby byly použitelné pro jakékoli kompetenční centrum. Toto zobecnění je možné provést v rámci dalšího rozšíření této práce.

## 8.1. Cíle

Kompetenční centrum má svůj cíl, tj. zabezpečit vytvoření a dodávku kvalitního produktu dle stanoveného harmonogramu. KC musí být nastaveno tak, aby maximálně vytěžovalo (ve smyslu utilizace) pracovníky KC, dále rozvíjelo jejich schopnosti a dovednosti. Je třeba zohlednit i podnikatelský faktor úspěchu, tj. zisk. V neposlední řadě je udržení nebo zvyšování spokojenosti zákazníka s kvalitou služeb.



**Obrázek 38: Cíle kompetenčního centra**

## 8.2. Hardware a Software

Kompetenční centrum provozních projektů potřebuje ke svému fungování speciální výrobní linku, prostředek, pomocí kterého připravuje a realizuje opravy a nové verze. Není cílem této práce definovat konfigurace HW a verze SW neboť toto považuje autor této práce za minoritní záležitost.

Pro úspěšnou realizaci provozních projektů je třeba zřídit dvě testovací prostředí – jedno u klienta a jedno v kompetenčním centru. Je to jistá pojistka pro odhalení i minimálních chybiček v opravách nebo verzích. Tyto chybičky jsou způsobeny rozdílem mezi produkčním prostředím, testovacím prostředím zákazníka a testovacím prostředím kompetenčního centra. Citlivost údajů lze chápat ze dvou úhlů – jednak ve smyslu zveřejnění a úniku informací, jednak ve smyslu citlivosti na správnost a přesnost vstupů.

Typy software v kompetenčním centru:

- Kancelářský software (std. MSOffice, lze použít Open office nebo jiný klon open source).
- Software pro správu verzí (CVS, v případě zvýšených nároků na kvalitu a možnosti verzování např. IBM Rational ClearCase případně filosofii Unified Change Managementu (UCM).
- Software pro vývoj (Komerční nebo open source prostředí pro vývoj v Java, komerční prostředí pro vývoj Oracle).
- Software pro testování – Komerční nebo Open sourcové prostředí pro správu defektů, komerční prostředí pro realizaci manuálních a automatizovaných testů, statistiku nad testy – PVCS Tracker nebo Merkury Test Director, Rational Test Suite.

Nedílnou součástí konceptu je stanovení procesu předání patchů a verzí klientovi. Osvědčený způsob pro předání malých patchů je email, pro velkých FTP. Verze je nejlepší předávat na CD nebo jiném mediu.

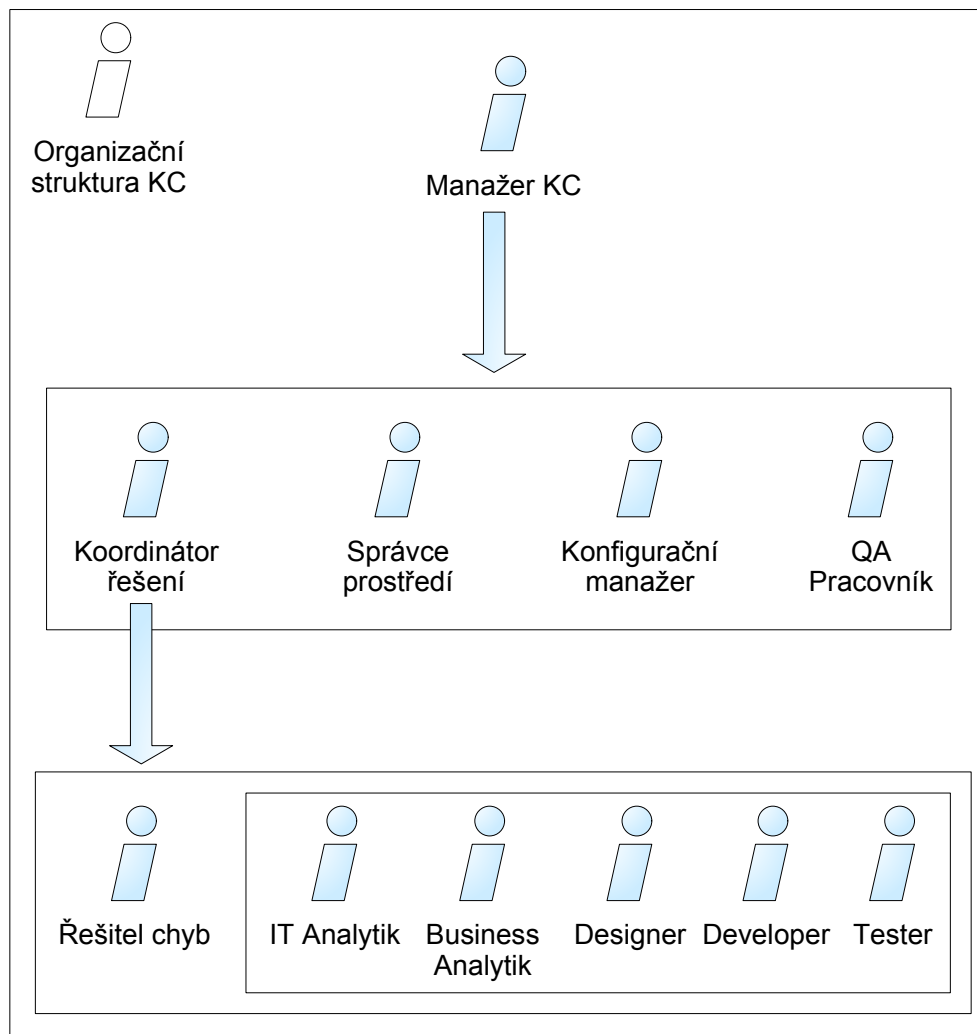
Vzhledem k obecnému růstu požadavků na zabezpečení komunikace autor doporučuje použít v případě emailové komunikace techniku digitálního podpisu („public key authentication“, protokol „SMIME“), a bezpečné „FTP“ – „SFTP“.

### **8.3. Organizační struktura**

V kompetenčním centru pro zajištění běhu provozních projektů je třeba vytvořit a obsadit následující role. Popis rolí obsahuje pouze jejich klíčové zodpovědnosti:

- Manažer KC
  - o Kontakt se zákazníkem.
  - o Reporting vedení divize.
  - o Alokace zdrojů.
  - o Řízení podpory první a druhé úrovně.
  - o Řízení Koordinátorů řešení.
  - o Metodika provozu.
  - o Dohled nad kvalitou výstupů (ve spolupráci s QA).
- Koordinátor řešení
  - o Odhad pracnosti opravy chyby/ vývoje nové funkcionality.
  - o Řízení oprav jednotlivých chyb.
  - o Příprava podkladů pro reporting.
  - o Alokace řešitelů, testerů.
- Správce prostředí
  - o Správa pracovišť, prostředí.
- Konfigurační manažer
  - o Konfigurační management.
- QA pracovník
  - o Kontrola výstupů.
  - o Kontrola procesů.

- Doporučení nápravných opatření.
- Řešitel chyb
  - Analýza požadavků.
  - Design požadavků.
  - Vývoj požadavků.
  - Otestování nové funkcionality.
  - Oprava chyb.
  - Testování chyb.
  - Příprava patchů, případně nové verze.
  - Výroba release notes.



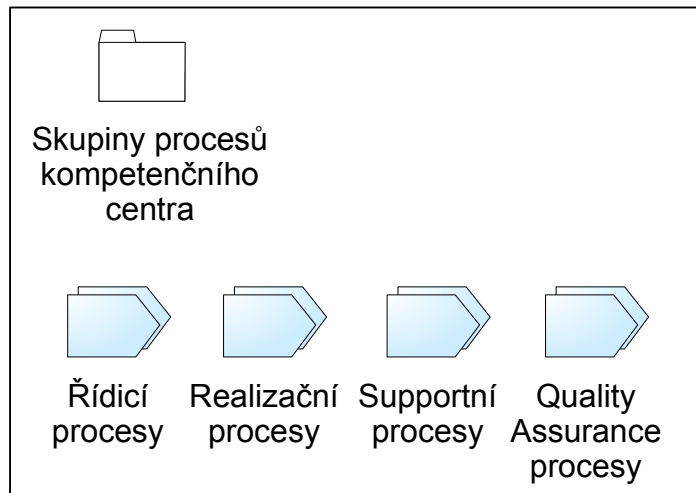
**Obrázek 39: Organizační struktura v KC pro vybrané role<sup>1</sup>**

## 8.4. Procesy

Procesy v kompetenčním centru lze rozdělit do čtyřech základních skupin:

- Řídící procesy.
- Realizační procesy.

- Supportní (podpůrné) procesy.
- QA procesy.



**Obrázek 40: Čtyři základní skupiny procesů kompetenčního centra**

Na základě výsledků analýzy použitelnosti metodik (viz. kapitola 6) autor této práce došel k závěru, že je nezbytné všechny procesy vyprojektovat s použitím vstupů metodik Rational Unified Process a CORTEX.

#### **8.4.1. Návrh procesu Poskytování podpory**

Jedním z cílů kompetenčního centra je zabezpečit v co nejkratším čase support první i druhé linie. To v praxi znamená:

- Řešení drobných problémů koncových uživatelů.
- Zabezpečení provozu systémů v produkci.
- Schopnost opravy kritických chyb v produkčním prostředí.

---

<sup>1</sup> Mezi osobou a rolí funguje vazba m:n, neboť jedna role může být obsazena několika osobami a naopak jedna osoba může být obsazena v několika rolích.



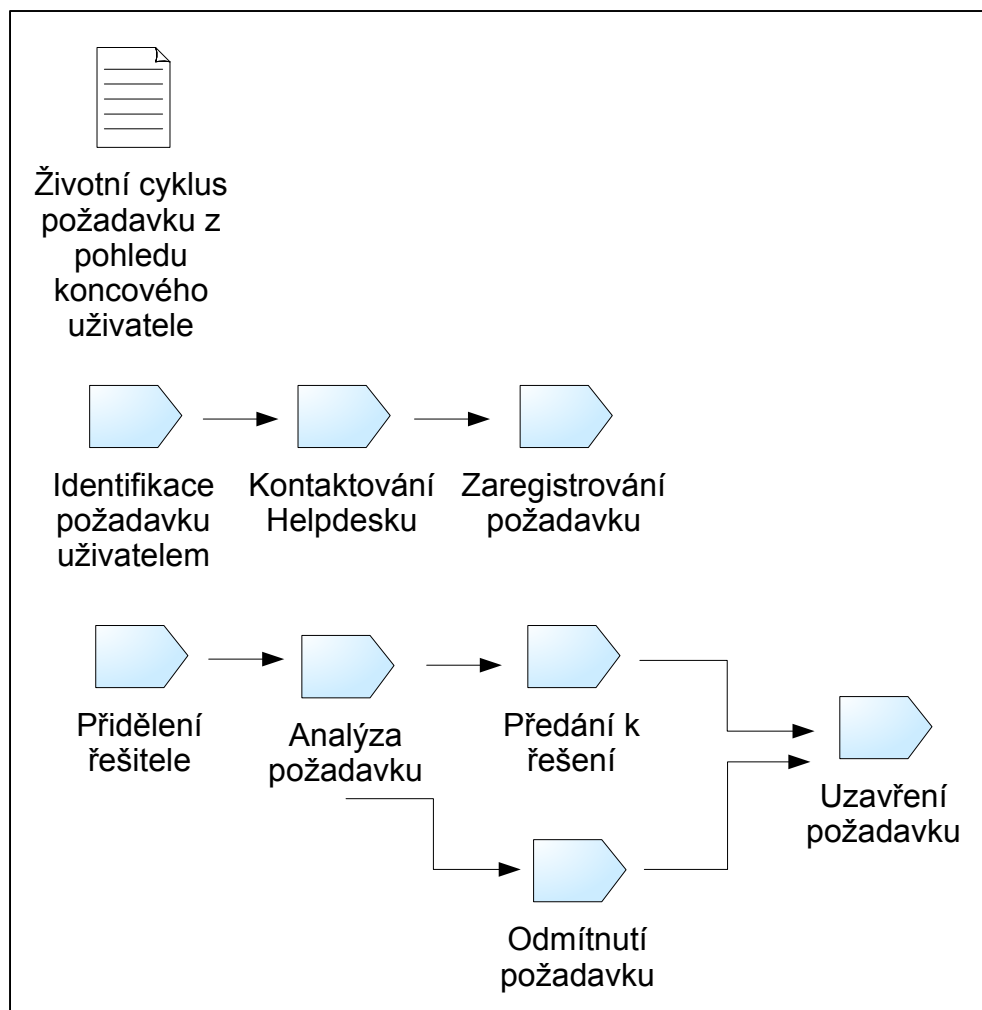
- Drobný rozvoj nové funkcionality.

Kompetenční centrum musí umět evidovat příchozí problémy a řídit jejich životní cyklus. Toto může být řešeno buď přímo na úrovni kompetenčního centra pomocí operátora, komunikačních kanálů a informačního systému, který podporuje správu problémů. Druhou variantou je řešit problematiku pomocí služeb Helpdesku.

Jelikož je problematika Helpdesku hodně obsáhlá, nicméně velmi dobře zachycená např. v metodice ITIL [BARTLET 2003], nepovažuje autor této práce za přínosné se jí obsáhleji zabývat.

Je-li přijat požadavek (nezávisle na typu komunikačního kanálu, kterým může být email, telefonát nebo dedikovaná aplikace, do které má přístup i zákazník), musí být zaregistrován. Dle typu požadavku operátor rozhodne (na základě manuálu), zda je schopen požadavek vyřešit sám nebo koho má přidělit (a pro kritické případy i kontaktovat) jako řešitele.

Řešitel provede analýzu požadavku a rozhodne, zda je možné požadavek vyřešit (a jakým způsobem), nebo zda je to požadavek neoprávněný (např. plynoucí z neznalosti IS koncovými uživateli nebo z nesdílení informací o plánovaném rozvoji atd.)



**Obrázek 41: Stavy požadavku z pohledu jejich zadavatele**

Evidenci požadavků na jednom místě, může v jakoukoli denní dobu zadavatel zkontrolovat stav řešení svých požadavků.

Životní cyklus požadavku z pohledu řešitele je:

- Nový – požadavek byl zaregistrován.
- Řešený – požadavek byl přidělen řešiteli a je v procesu řešení.

- Vyřešený – Požadavek je vyřešen a předán zákazníkovi (zadavateli) k vyjádření.
- Akceptovaný – Zadavatel otestoval řešení požadavku a odsouhlasil jeho nasazení do produkčního prostředí.
- Uzavřený – Zadavatel je spokojen s vyřešením požadavku a souhlasí s jeho uzavřením.

Je samozřejmě přípustné, aby řešení požadavku bylo jeho zadavatelem odmítnuto. To způsobí posun ze stavu Vyřešený zpět do stavu Řešený.

Je možné, aby požadavek ještě nabyl následujících stavů:

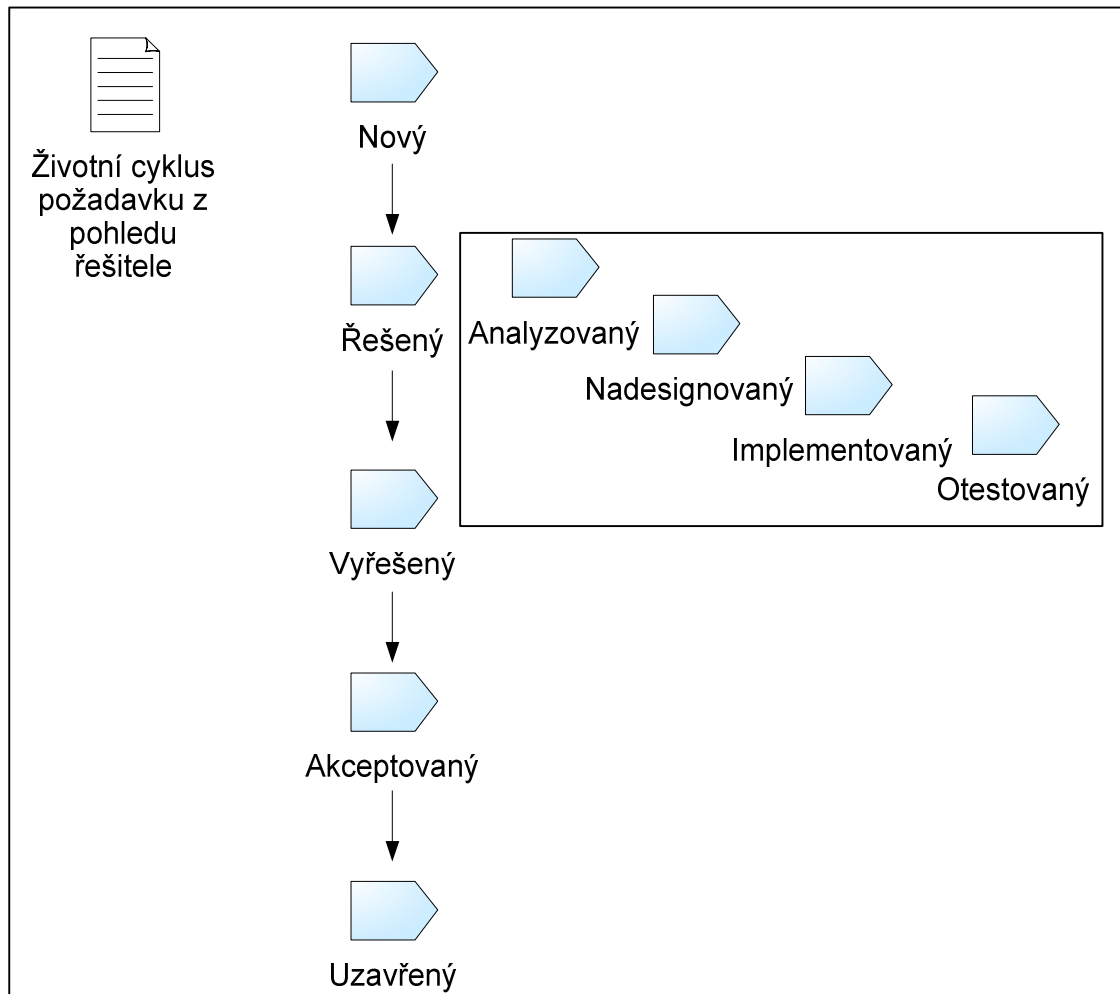
- Odmítnutý – Požadavek byl analyzován a odmítnut jako neoprávněný.
- Pozastavený – Řešení požadavku bylo na žádost zadavatele pozastaveno v čase. Tento stav je také použitelný pro případ, kdy řešitel čeká na další vstupní informace od zadavatele.

Z pohledu managementu a řešitelů má požadavek rozdílné stavy. Rozdílnost stavů může být na první pohled matoucí, ve skutečnosti je naopak pro zadavatele zpřehledňující, neboť zabraňuje zahlcení nadbytečnými informacemi. Je v zájmu kompetenčního centra zbytečně nemást koncové uživatele.

Z pohledu IT se jedná o následující stavy:

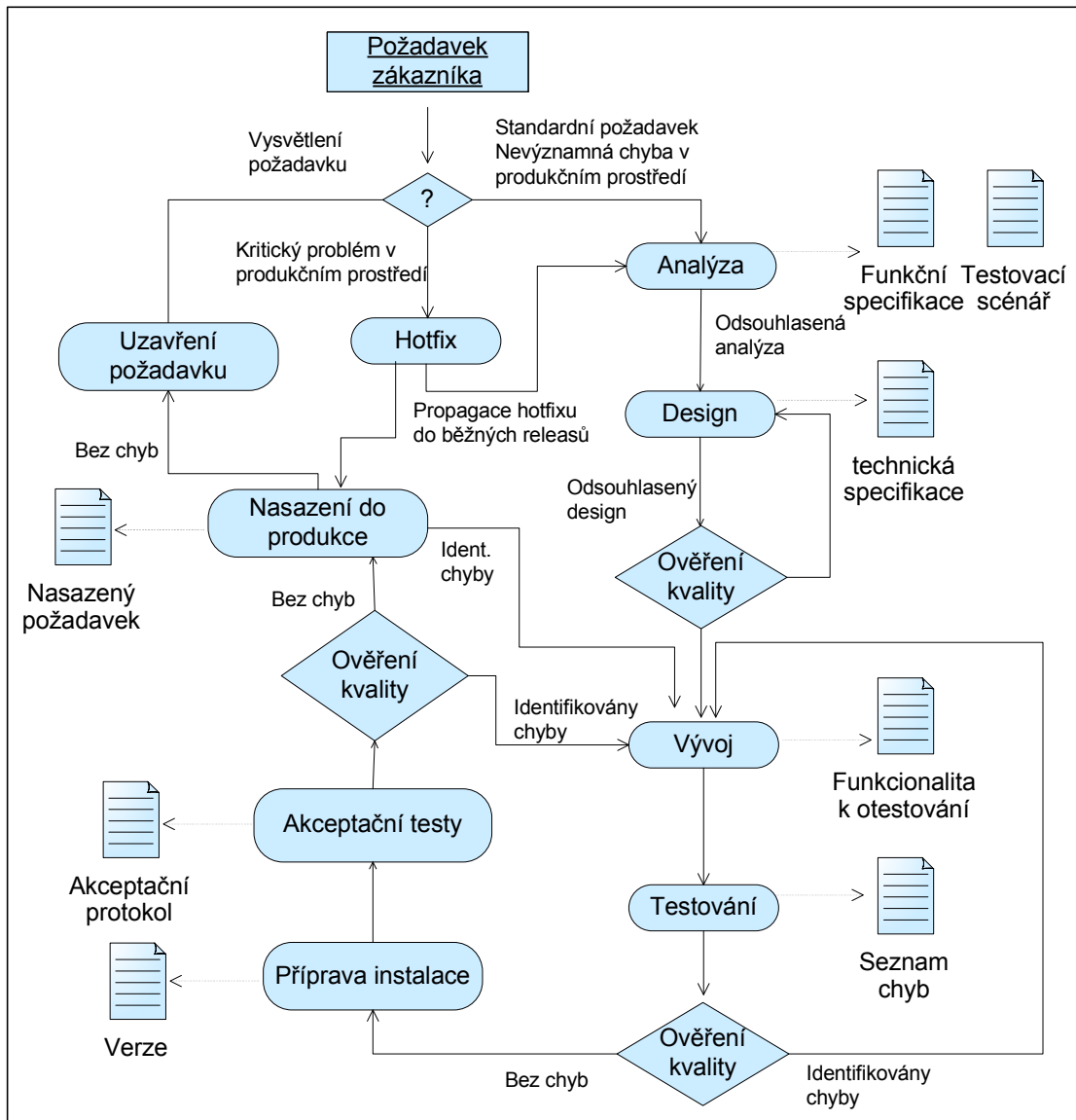
- Nový – Požadavek byl zaregistrován operátorem do aplikace a předán k řešení.
- Řešený – Požadavek je v procesu řešení. Typicky nabývá těchto podstavů.
  - Analyzovaný.
  - Nedesignovaný.
  - Implementovaný.

- Otestovaný.
- Vyřešený – Požadavek je vyřešen a předán zadavateli k akceptaci.
- Akceptovaný – Zadavatel akceptuje řešení.
- Uzavřený – Po nasazení do produkce je požadavek uzavřen.



**Obrázek 42: Stavy požadavku z pohledu řešitele**

Dekompozice procesu poskytování podpory je zachycena na následujícím schématu:



Obrázek 43: Schéma procesu “Poskytování podpory”

## 9. Autorův návrh řídicích procesů KC

Při návrhu řídicích procesů autor stanovil klíčové aktivity řízení a jejich výstupy a to jak pro interní, tak i externí účely (pro standardizaci komunikace se zákazníkem).

Klíčové aktivity programme managera kompetenčního centra:

- Řízení problémů (incidentů).
- Utilizace zdrojů.
- Řízení očekávání zákazníka.
- Ověření kvality.
- Reportování.
- Účtování.

Detailní popis těchto klíčových aktivit je uveden v následujících podkapitolách.

### 9.1. Řízení problémů (incidentů)

Programme manager nebo jiná, jím dedikovaná role zabezpečuje řízení jednotlivých problémů. To v praxi znamená zajistit jejich evidenci a transfer na koordinátory řešení. Ti odhadnou pracnost a stanoví nejvhodnějšího řešitele. Manager musí zjistit vytížení řešitele a na základě jeho dostupnosti stanovit vhodný termín vyřešení. Pak kontaktuje zákazníka a dohodne s ním, zda je termín a návrh řešení vyhovující. Po odsouhlasení postupu se zákazníkem je požadavek zařazen k realizaci. Při realizaci požadavku případné problémy a potenciální skluzy v řešení ihned komunikuje se zákazníkem.

Po svém vyřešení na straně řešitele, je požadavek předem dohodnutou formou předán klientovi k testům – je nezbytné udržovat s klientem plán nasazení funkcionality k testům, stejně tak i plán nasazení do produkce. Na základě těchto plánů je manager schopen efektivněji plánovat lidské zdroje, neboť pracuje s informací, že k nějakému datu bude potřeba mít pracovníka dostupného na řešení problémů při testech nebo při rolloutu do produkce. Zatím neexistuje koncept, který by zabezpečil, aby testovací prostředí, produkční prostředí klienta a testovací prostředí dodavatele bylo stejné (což může být předmět prakticky zaměřené doktorské práce).

V praxi zpravidla zákazník kombinuje svůj strach a pohodlí, a proto vyžaduje při každém nasazení změn do produkce, buď přímo osobní účast dodavatele nebo alespoň jeho přítomnost na telefonu. Autor této práce se domnívá, že je otázkou profesionality poskytnout zákazníkovi maximální možnou součinnost a pohodlí, neboť dobře opečovávaný a spokojený zákazník je základem dlouhodobé a produktivní spolupráce.

## **9.2. Využití zdrojů**

Utilizace zdrojů (lidské, peněžní, kapitálové) je klíčovou činností manažera. V oblasti lidských zdrojů je cílem dlouhodobě pracovníky dostatečně vytěžovat, nepřetěžovat a nikdy se nedostat do stavu, kdy manažer se o sobě domnívá, že má mnoho práce a jeho podřízení nemají co dělat. Tato situace je často k vidění u menších českých společností a typu manažerů, kteří extrémně rychle dosáhli vrcholu bez odpovídajících zkušeností. Největším problémem tohoto stavu je neschopnost delegovat práci a neschopnost stanovit prioritu činností. Tito manažeři stále věří ve věty „Co si neudělám sám, tak nemám.“ nebo „Co si neudělám sám, je chybně.“ Diskuse na téma manažerských stylů vedení je velmi zajímavá, nicméně autor ji považuje za významně přesahující rozsah této práce.

Také téma motivace podřízených k maximálnímu výkonu je zajímavé a široké, ale dle názoru autora je také mimo téma této doktorské práce. I toto téma může být předmětem prakticky orientované doktorské práce.

Z ekonomického pohledu autor této práce doporučuje zohlednit následující paradigma – je velmi rozumné mít odděleny zdroje - pracovníky servisu, kteří jsou dedikováni na servisní zásahy u klienta od standardních členů vývoje. Je to z toho důvodu, že standardní vývoj nebo oprava je plánovatelnou činností. Oprava náhlého problému u zákazníka je ze své podstaty činností neplánovatelnou.

Pro větší vývoj funkcionality je proto vhodnější si vypůjčovat pracovníky z klasických vývojových týmů (existuje-li taková možnost). Tento tým lze doplnit i servisními pracovníky, nicméně při jejich plánování je vždy třeba mít na paměti, že jejich primární účel práce je jiný (jak je uvedeno již výše, bohužel neplánovatelný – emergenci problému nelze u zákazníka předem naplánovat).

Pro efektivní rozvoj talentů a schopností jednotlivých členů týmu je třeba zabezpečit jejich kontinuální růst. Na úrovni kompetenčního centra, nebo spíše celé společnosti, je nezbytné definovat kariérní dráhy pro jednotlivé profese, jejich úrovně, podmínky postupu, a to samozřejmě navázat systém odměňování a benefitů atd. Rozvoj a profesní růst týmů a jednotlivců je téma, které přesahuje rámec této práce. Toto téma může být předmětem prakticky orientované doktorské práce.

### **9.3. Řízení očekávání zákazníka**

Umění pozitivně řídit očekávání zákazníka není exaktní disciplína s přesně stanovenými pravidly a postupy. Je to velmi široká a specifická oblast, neboť pro ni neexistuje mnoho obecných či konkrétních doporučení. Jinými slovy, všichni z ranku manažerů vědí, čeho chtějí dosáhnout (spokojený, usměvavý zákazník, který si objedná v dalším roce služby ve stejném nebo větším rozsahu), nicméně pro tento cíl dosud neexistuje univerzální postup, jak jej dosáhnout.

Existuje mnoho literatury zaměřené na marketing, inovace zboží [viz PORTER 1986] atd., ale ta bohužel nepokrývá uvedenou problematiku v oblasti služeb. Nicméně autor této práce na základě zkušenosti z realizace ICT projektů formuluje následující základní doporučení pro řízení očekávání zákazníka:



- Akceptování zákaznických přání a požadavků až po jejich analýze dopadu – cílem tohoto kroku je zabránit nespelnitelným slibům nebo chybným odhadům v oblasti času, úsilí a rozpočtu.
- Důsledné odsouhlasení postupu, priorit, kroků, obsahu výstupů.
- Dodržování dohodnutých termínů a kvality výstupů.
- Vědomě neříkat nepravdu nebo zastírat problémy.
  - o Včetně kapacitních, termínových a rozpočtových problémů.
  - o Zákazníkovi se NELŽE.
- Proaktivní hledání řešení pro případ problémů.
  - o Výsledek odpovídající tzv. „win-win“ řešení.
- Hledání synergických efektů.
- Hledání směrů rozvoje aplikací (nabízet zákazníkovi novou, pro něj prospěšnou funkcionalitu).
  - o Dodávání komplexních a kvalitních řešení.
  - o Obecně proaktivně jednat se zákazníkem na principu vlastních návrhů a jejich oponentury, doplnění.

## 9.4. Ověřování kvality

Programme manager garantuje kvalitu vstupů předaných zákazníkovi – a to jak z úhlu pohledu obsahu, tak i formy. Samozřejmě, že to neznamená, aby v praxi zkoumal kvalitu a efektivitu zdrojového kódu – na to zpravidla nemá dostatečnou znalost a ani čas. Programme manager, jak evokuje název jeho pozice, je manažer, nikoli technik.

Je-li potřeba zabezpečit ověření kvality z pohledu obsahu, musí programme manažer umět nalézt buď přímo v KC, ale raději v celé společnosti odborníka, který je toho schopen a jej si vyžádat přes manažera zdrojů a nechat ho kvalitu zkontrolovat.

Výhoda revize obsahu pracovníkem mimo KC je evidentní – mentální nezatíženost odborníka řešením revidovaných problémů ani fyzická participace na jejich řešení. Tento nadhled je obrovsky cenný, neboť může odhalit neefektivní části kódu, neefektivní funkcionalitu, nesrozumitelnost funkčních specifikací, dokumentace atd.

Garance kvality Programme managerem je převážně chápána v kontextu úplnosti a formální náležitosti výstupů – tj. obsahuje-li předávaná verze aplikace zákazníkovi všechny aspekty korporátní kultury, tj. jak má verze vypadat, jaké jsou její náležitosti, jmenné konvence, doprovodná dokumentace, předávací protokol, na čem má být předána, v kolika kopiích atd. Manažer zodpovídá i za věcný výstup, tj. že součástí předávky je předávací protokol, přehled změn, výsledky testů a jiné výstupy, které jsou buď definovány na úrovni procesů společnosti nebo procesů a výstupů z jednotlivých činností kompetenčního centra.

## **9.5. Auditování**

V rámci ověřování kvality výstupů, ale i procesů a vyhodnocení celkové aplikace metodiky v každodenní praxi, je nezbytné plánovat a provádět pravidelný audit nad každým servisním projektem (soulad procesů, výstupů) i nad aplikací procesů v KC.

Výsledkem auditu jsou zjištěné nebo nezjištěné nesrovnalosti při porovnávání kvality výstupů s jejich předpisem, nedostatky v dodržování procesů (interní metodiky) a výstupů jednotlivých projektů stejně tak i KC. Při zjištění nedostatků je definován plán a kompetentní osoby zodpovídající za jejich odstranění. Tento výstup je klíčovým vstupem pro zlepšování kvality služeb jak jednotlivých projektů, tak KC jako celku.

Auditování je činnost, která je mnohými členy týmu chápána jako „něco obtěžujícího, navíc, někdo nás zase přišel poučit“ atd. Je plně v kompetenci programme managera toto nepochopení odstranit. Auditování je naopak motivační proces, který se snaží pracovníkům ulehčit práci a to dohledem nad kvalitou vstupů do jejich činností, procesu činností a kvality výstupů. Cílem auditu není represní, ale podpůrná činnost.

## **9.6. Management KC**

Cílem řízení kompetenčního centra je zabezpečení jeho zdravého růstu – tj. zabezpečení dodávky požadavků, sledování trendů, ověření ekonomických výsledků, ověření utilizace atd. V přesně definovaných cyklech by měl manager toto ověření provádět ve spolupráci s managerem kvality a svým nejbližším nadřízeným s cílem identifikovat silná místa k ještě většímu posílení, a slabá místa k jejich postupnému zlepšování.

### **9.6.1. Agenda a Zápis z jednání**

Pro každé jednání se zákazníkem je vhodné připravit formalizovanou osnovu jednání. Tato osnova v podobě agendy podléhá odsouhlasení stran. Z každého jednání vzniká zápis, který je třeba odsouhlasit a podepsat za zúčastněné strany.

Je dobrým zvykem jednat se zákazníkem i na méně formálních, kratších jednáních. Zde autor této práce doporučuje pouze připravit agendu v bodech a předem neodsouhlasovat. Jako vhodnou formu zápisu autor této práce vidí shrnutí klíčových závěrů formou odrážek do emailu.

### **9.6.2. Registr rizik**

Risk management je klíčovou disciplínou řízení s cílem eliminovat v maximální míře nejvýznamnější rizika (rizika s největším dopadem). Vhodnou pomůckou pro řízení rizik je používání přehledové tabulky nejčastěji nazývané Risk List [BOOCH 1999] nebo Risk Registr [COR]. V tomto přehledu jsou zachycena rizika projektů, jejich dopad, číselné ohodnocení, historie a také predikce vývoje v budoucnu. Registr rizik KC vznikne na základě kompilace risk registrů jednotlivých projektů a zahrnutím ostatních neprojektových a specifických rizik KC (nemocnost, převzetí dalšího projektu do servisu a z toho plynoucí organizační, rozpočtová a technická rizika atd.)

Risk managementu autor této práce věnoval podstatnou část své diplomové práce a proto na její odpovídající části tímto odkazuje [MAROUNEK 2002].

### **9.6.3. Přehled požadavků**

Přehled požadavků zachycuje všechny požadavky a stav jejich řešení. Zobrazuje informace o jednotlivých požadavcích, jejich prioritě, severitě, datu přijetí, plánu na řešení, řešitele a aktuální stav požadavku.

Na základě údajů z tohoto přehledu je možné zjistit, v jaké stavu se nachází provozní projekt a také generovat podklady pro indikátory – počet požadavků v čase, množství požadavků v jednotlivých kategoriích, ve stavu atd. Detailněji o kvalitě a indikátorech pojednává kapitola 11.8. Řízení kvality.

### **9.6.4. Plán projektu**

Pro každý provozní projekt je třeba vyhotovit vlastní plán projektu, ve kterém je řečeno, co je cílem projektu, kdo jej vede, složení realizačního týmu, týmu zákazníka, finanční ukazatele, milníky, kritéria úspěšnosti, modifikace pracovních procesů atd.

Plán projektu je klíčový dokument, který schvaluje strana zákazníka, neboť obsahuje náležitosti, které se jej přímo dotýkají a jeho spolupráce s ním - mj. definuje požadavky na spolupráci, frekvenci jednání, eskalační procedury atd.

### **9.6.5. Plán kompetenčního centra**

Plán kompetenčního centra je nadřazeným, sjednocujícím dokumentem pro plány jednotlivých projektů. Tento dokument zachycuje informace o vzniku KC, týmu, realizovaných projektech, kritériích úspěšnosti, klíčových procesech, modifikaci korporátních procesů atd.

Při pravidelných ověření kvality popisovaných v kapitole 11.8. Řízení kvality, je ověřen aktuální stav KC a jeho trend oproti kritériím z tohoto plánu.

## **9.7. Reportování**

Pro své nadřízené Manager KC vytváří přehledy, které jsou striktně interního charakteru. Cílem těchto reportů je pravidelně informovat nadřízeného o aktuální situaci a trendu vývoje KC a také eskalovat problémy.

Nadřízeným jsou také předkládány k nahlédnutí reporty, které jsou výstupem při ověřování kvality.

### **9.7.1. Hodnocení projektu interní**

Tento dokument stručně popisuje aktuální stav projektu, komentuje finanční ukazatele, klíčové problémy, rizika, definuje plán na další období. Dokument je připravován jednou měsíčně pro každý provozní projekt. Kompilací podstatných údajů z těchto hodnocení vznikne přehledová zpráva za celé KC.

Nedílnou součástí hodnocení stavu jsou aktuální a trendové indikátory, které slouží jak pro měření výkonnosti KC, tak i jeho jednotlivých provozních projektů, kvalitě pracovníků, kvalitě plánování atd. Detailněji o kvalitě a měření pojednává kapitola 11.8. Řízení kvality.

Dále každý provozní projekt vytváří přehled pro zákazníka.

### **9.7.2. Hodnocení projektu externí**

Po odstranění citlivých (interních) informací je hodnocení projektu připraveno také k distribuci zákazníkovi. Toto hodnocení by mělo být se zákazníkem podepisováno, jako formální odsouhlasení s aktuálním stavem a dalším plánem. Na základě jeho odsouhlasení a ještě např. předávacího, případně akceptačního protokolu je možné připravit fakturaci (která je také v kompetenci programme manažera nebo jím definované role).

Pro externí statistiky jsou sbírána data o počtu řešených chyb, počtu a stavu chyb, účetní a ekonomická data atd. Detailněji o kvalitě a měření pojednává kapitola 11.8. Řízení kvality.

## **9.8. Účtování**

Pro zachování účetní transparentnosti KC, programme manager musí mít přehled o účetních datech jednotlivých projektů – tj. o nákladech a výnosech. Kromě pohledu přes jednotlivé projekty musí vidět celkové výsledky v kontextu KC, tj. pohledem napříč přes všechny projekty.

### **9.8.1. Finanční stav projektu**

Výstupem účetní aktivity je tzv. Finanční stav projektu, ze kterého lze vidět historii, aktuální a predikované ukazatele projektu – náklady, výnosy, zisk, spotřebované náklady, plán atd. Finanční stav kompetenčního centra vznikne pohledem přes jednotlivé projekty, přičemž je třeba započíst i režijní aktivity, které nejsou přiřazovány na projekt (např. Školení atd.)

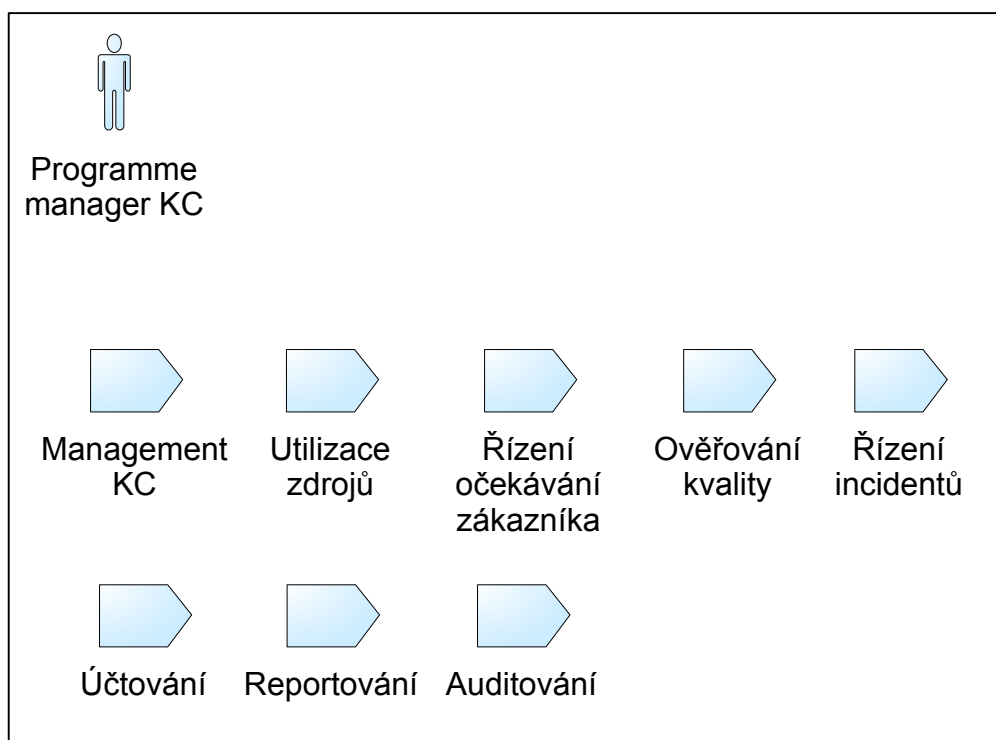
Na úrovni jednotlivých projektů koordinátor udržuje informaci, kolik zdrojů a času bylo plánováno na vyřešení požadavku a skutečný stav jejich čerpání. Porovnáním těchto údajů je možné stanovit kvalitu plánovacího procesu. Sběr a vyhodnocování dat je třeba podpořit vhodným IS pro to, aby tato činnost byla nákladově efektivní.

### **9.8.2. Vydané faktury**

Programme manager je také zodpovědný za fakturaci jednotlivých projektů. Je na jednotlivých koordinátorech, aby zajistili potřebné podklady pro jejich vystavení – podepsané hodnocení projektu, předávací, akceptační protokoly apod.

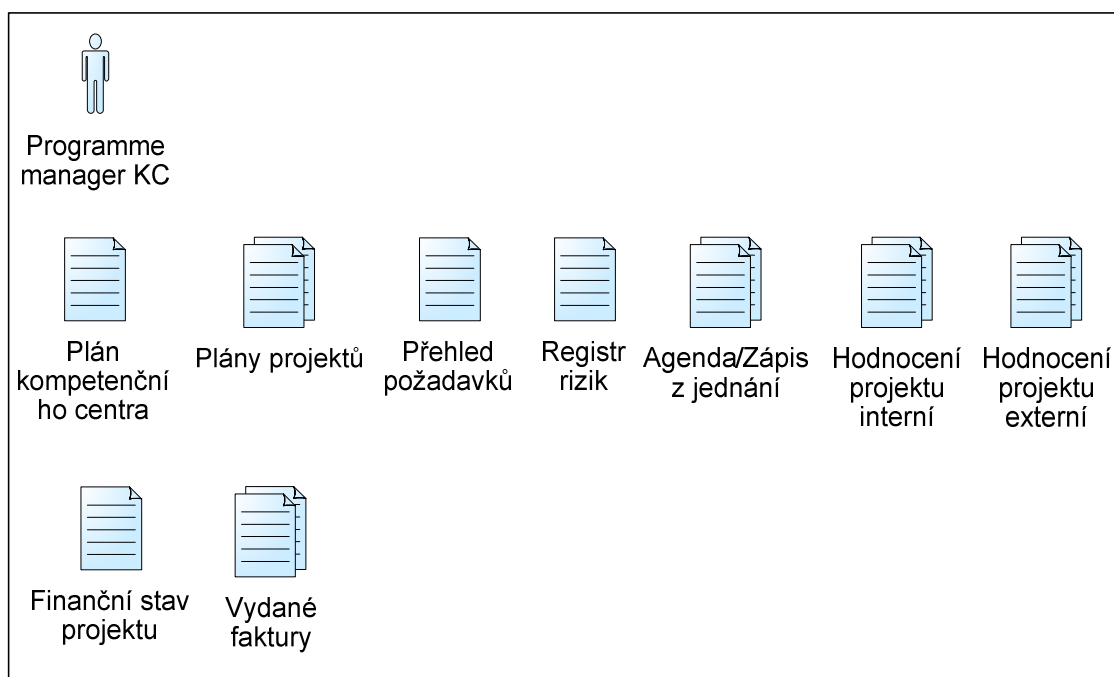
Záleží na nastavení korporátních procesů nebo KC, kdo bude vystavovat fakturu – zda fakturační oddělení, koordinátor řešení nebo přímo programme manažer. Důležité je, aby existovala v KC role, která je za vystavení zodpovědná (zkontroluje, že faktura byla správně vystavena).

Vzhledem k tomu, že částky za servisní podporu jsou částečně hrazeny paušálně a částečně hrazeny dle aktivit v jednotlivém měsíci, autor této práce se domnívá, že nejvhodnější je, aby faktury vystavoval koordinátor řešení a kontroloval programme manager. Programme manager také kontroluje zaplacení těchto faktur a stará se o celý případ.



**Obrázek 44: Klíčové aktivity programme manažera kompetenčního centra**

Nedílnou součástí aktivit jsou také jejich výstupy. Tyto výstupy jsou v souladu s klíčovými metodikami podporujícími programme management a také s normami ISO a TickIT.

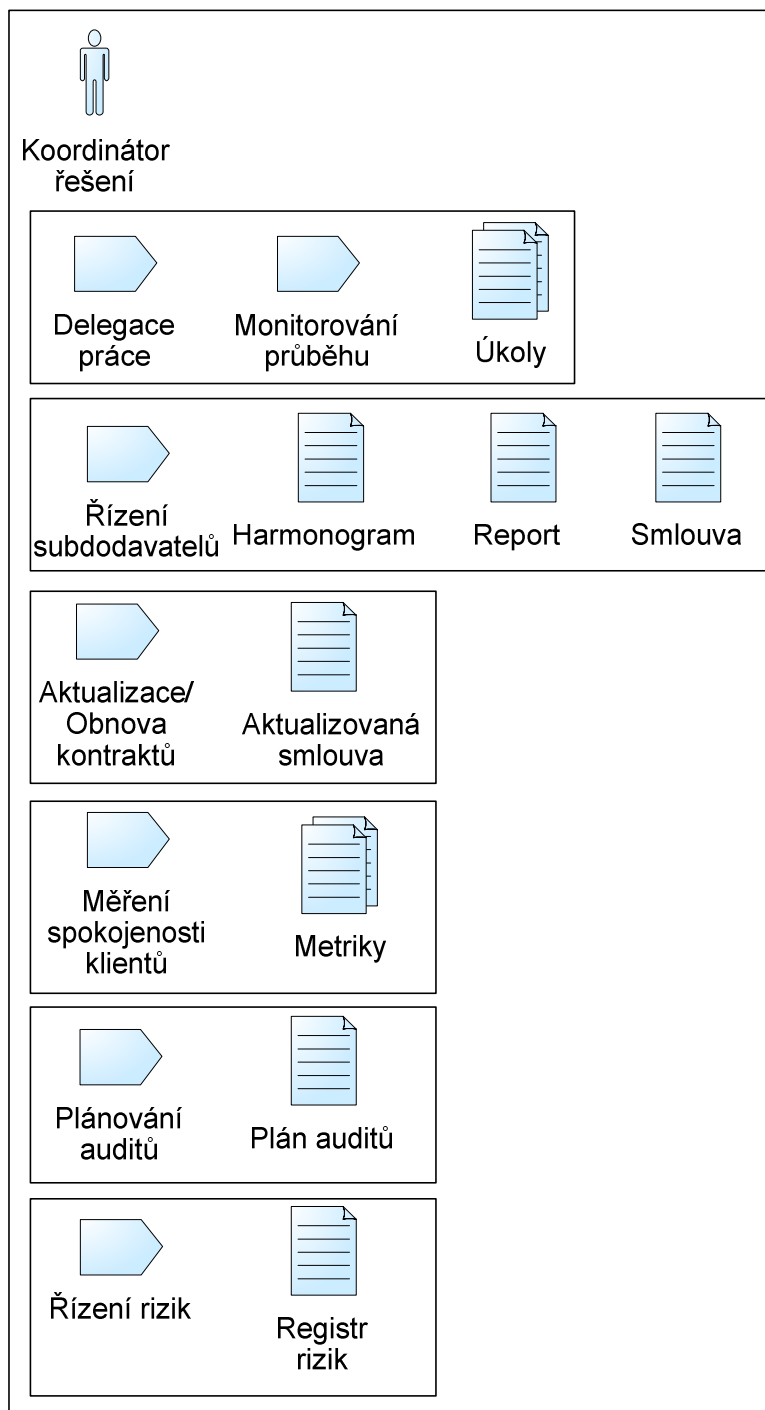


**Obrázek 45: Výstupy aktivit programme managera kompetenčního centra**

## 9.9. Specifické aktivity řízení provozních projektů

Aktivity zmíněné v předchozí kapitole jsou manažerského charakteru a dle velikosti kompetenčního centra mohou být vykonávány jak manažerem, tak vybrané z nich (týkající se spíše provozních procesů než-li fungování kompetenčního centra) i koordinátorem řešení. Aktivity zahrnuté v této kapitole primárně spadají do aktivit koordinátora řešení.





**Obrázek 46: Proces a výstupy řízení IT provozních projektů**

### **9.9.1. Delegation práce a monitorování průběhu realizace**

Koordinátor řešení je zodpovědný za naplánování a přidělení práce jednotlivým členům týmu. V případě, že má větší objem práce, než-li je kapacita jeho týmu, vznáší požadavek na programme managera. Ten jej řeší v souladu se standardním procesem.

Koordinátor je také zodpovědný za průběžnou kontrolu plnění úkolů a řešení odchylek mezi plánem a skutečností.

### **9.9.2. Řízení subdodavatelů**

Vytváření subdodavatelských aliancí bude čím dál více perspektivnější a pravděpodobně v budoucnu jediná cesta, jak dodávat složité projekty zákazníkovi ve stanoveném čase. Již dnes jsou některé IT projekty tak složité, že je nemožné je dodávat jako jeden celek jedním dodavatelem, ale daleko vhodnější, bezpečnější a nákladově efektivnější cestou je dodávat je v konsorciu subdodavatelů, kteří mezi sebe rozloží riziko plynoucí z dodávky. Velmi aktuální je tato problematika pro případ systémově integračních projektů nebo při totální obnově IS – toto téma je mimo rámec této práce.

Pro případ kompetenčního centra je zajímavou cestou úspor a zkvalitnění služeb využití outsourcingu. Detailnější rozpracování této problematiky je mimo rozsah této práce.

### **9.9.3. Obnova kontraktu**

Na základě předem definované, nejlépe roční bázi a plné spokojenosti zákazníka s dodávkou služeb probíhá aktualizace a prodlužování servisních kontraktů. Cílem této činnosti je korigovat cenu poplatků za služby, rozšiřovat spektrum nabízených služeb a aktualizovat SLA. Provozní projekty jsou realizovány za pevný paušální poplatek a služby nad rámec těchto paušálů na bázi spotřebovaného času a nákladů. Při pokusu převodu účtování služeb na tzv. „fix time, fix price“ projekty koordinátor zpravidla narazí na obtížnost, skoro i nemožnost definovat jasně výstupy, milníky atd.

#### **9.9.4. Měření spokojenosti zákazníka**

Na každém projektu je třeba zavést proces měření spokojenosti zákazníka jako kontrolní prvek a také jako nezbytnou zpětnou vazbu pro průběžné zlepšení služeb. Obecně je vhodné měřit jakoukoli veličinu na projektu, přičemž je nutné zohlednit ekonomický aspekt – přínos z měření musí být větší nebo roven nákladům spojeným se získáním údajů. Pro potřeby provozního projektu autor doporučuje měřit spokojenost zákazníka. Výsledek měření zachycuje objektivní informaci ohledně kvality dodávek v očích zákazníka – viz. kapitola 11.8. Řízení kvality.

#### **9.9.5. Plánování kontroly projektu**

Koordinátor řešení ve spolupráci s programme managerem je zodpovědný za plánování interních kontrol kvality projektu. Cílem těchto kontrol je zjistit odchylky oproti definovanému korporátnímu standardu, případně standardu kompetenčního centra. Četnost a hloubka kontrol je dána korporátním pravidlem nebo obecnými standardy např. ISO. Detailně o ověřování kvality pojednává kapitola 11.8. Řízení kvality.

#### **9.9.6. Risk management**

Risk management je klíčovou disciplínou řízení s cílem eliminovat v maximální míře nejvýznamnější rizika (rizika s největším dopadem). Risk managementu autor této práce věnoval podstatnou část své diplomové práce a proto na její odpovídající části tímto odkazuje [MAROUNEK 2002].

## 10. Autorův návrh realizačních procesů KC

### 10.1. Proces poskytování podpory - realizace

Proces poskytování podpory klientům KC autor rozdělil na dvě hlavní linie v souladu s metodikou ITIL [BARTLET 2003]:

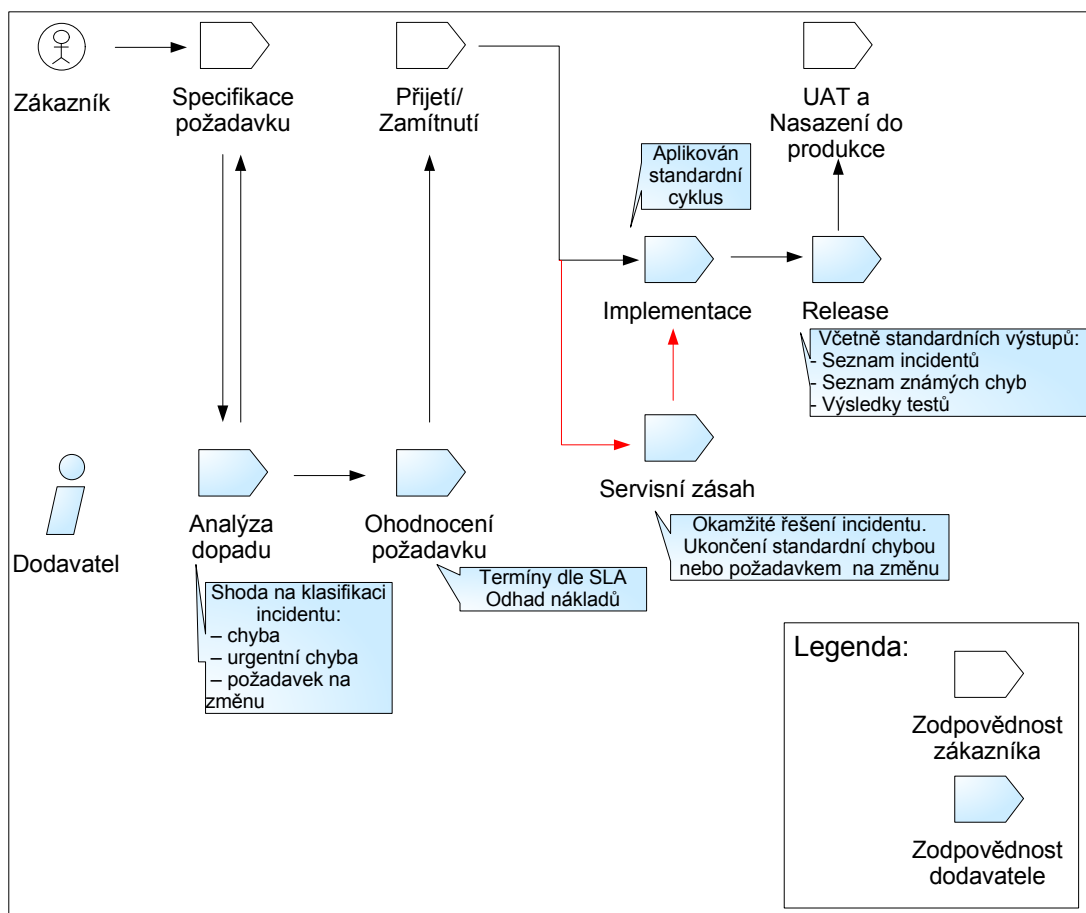
- Podpora první úrovně – Promptní řešení hlášených problémů na úrovni helpdesku – mj. součástí této úrovně jsou i servisní zásahy přímo u klienta s cílem řešit klíčové problémy.
- Podpora druhé úrovně – Plánovaný vývoj nové funkcionality.

Nezávisle na typu problému, a zvolené úrovni jeho podpory, je třeba důsledně dokumentovat všechny požadavky ze strany zákazníka a i stav jejich řešení. Formalizace zachycení těchto požadavků umožňuje snadné řízení a generování přehledů o aktuálním stavu. Perfektně zvládnutá dokumentace problémů včetně návrhu jejich řešení je dále využita jako zdroj pro knowledge base. (Blíže o knowledge base a CMDB, incident a problem managementu pojednává ITIL, [BARTLET 2003]). Informace o problému (přání) zákazníka a stavu jeho řešení jsou zachyceny v dokumentu „Požadavek zákazníka“.<sup>1</sup>

Dokument Požadavek zákazníka slouží jako nositel informací o problému zákazníka, aktuálním stavu řešení, popisu řešení, plánovaných kroků do budoucna a ekonomických odhadů. Pro snadnou externí i interní komunikaci, autor této práce navrhuje níže uvedený životní cyklus požadavku. Tento cyklus zahrnuje všechny nezbytné aktivity od fáze vzniku problému až po dodání vytvořené opravy zákazníkovi do jeho testovacího prostředí.

---

<sup>1</sup> Každý požadavek zákazníka je třeba zachytit písemnou formou, aby jej bylo možné řídit. Standardně dokument obsahuje identifikaci zákazníka, popis požadavku, prioritu, severitu, požadovaný termín řešení a podpis. Některé šablony v rozšířené verzi umožňují ještě zachytit analýzu a návrh řešení a stavy požadavků.



**Obrázek 47: Životní cyklus požadavku**

Popis procesů životního cyklu požadavku

Proces	Role	Popis
Specifikace požadavku	Zákazník	Zákazník specifikuje obchodní proces, procesní změny, funkční změny, nefunkční požadavky atd. Dodavatel pomáhá se specifikací, dopady do řešení, realizovatelností, doporučení varianty realizace atd.

Analýza dopadu	Řešitel	Řešitel zkoumá realizovatelnost požadavku s cílem stanovit dopady do řešení, termínu, rozpočtu. Dále identifikuje potenciálně rizikové oblasti a navrhuje cesty jejich eliminace. (Je-li požadavek zásadního dopadu, je vhodné realizovat studii proveditelnosti – podmínky, za kterých je vhodné studii dělat a jakým způsobem přesahuje rozsah této práce.
Ohodnocení požadavku	Řešitel	Po interním rozhodnutí řešitele o realizaci, nerealizaci požadavku a s přihlédnutím k rizikům a aktuální utilizaci zdrojů, je stanovena priorita, čas, rozpočet a lidské zdroje. Takto ohodnocený požadavek je předán zákazníkovi k posouzení.
Přijetí/Zamítnutí	Zákazník	Zákazník rozhodne, zda požadavek chce realizovat navrženou cestou či nikoli.
Servisní zásah	Řešitel	Tento proces je realizován v případě, že se u zákazníka vyskytl krizový stav, který je třeba urgentně vyřešit. Cílem tohoto procesu je neprodleně vyřešit problém u zákazníka a umožnit mu dále procesně fungovat. Nedílným výstupem této aktivity je dokumentace a předání problému k dořešení do standardního procesu.
Implementace	Řešitel	V rámci implementace jsou požadované funkčnosti/úpravy vyvinuty, otestovány, zdokumentovány a připraveny k předání formou patchů nebo nové verze.
Release	Řešitel	Cílem tohoto procesu je zkompilovat a slinkovat

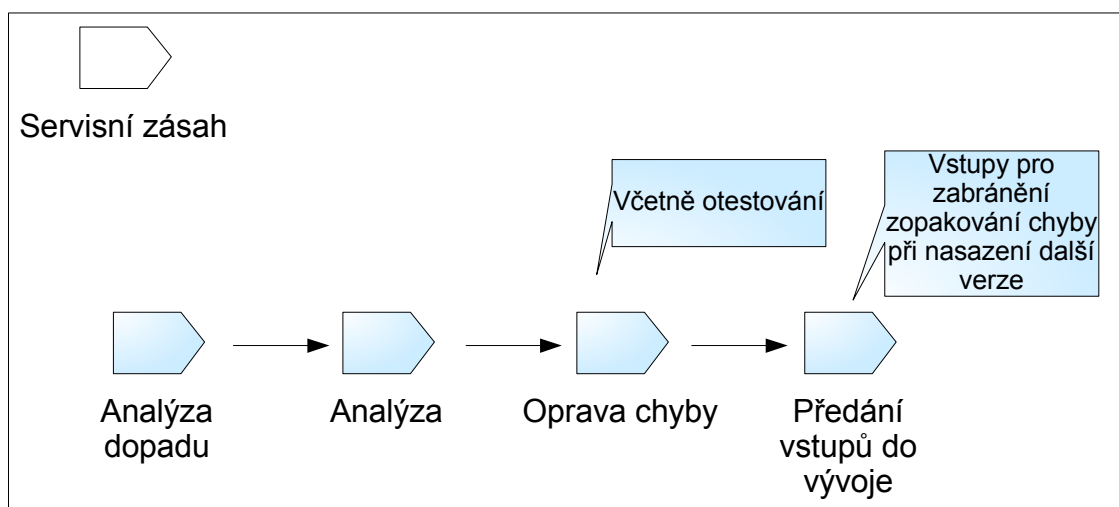
		zdrojové kódy a připravit tak novou verzi, která obsahuje všechny patche, opravy a novou funkcionalitu.
UAT a Nasazení do produkce	Zákazník	Interně otestovaný release včetně dokumentace zákazník otestuje ve svém testovacím prostředí (akceptační testy). V případě identifikace chyb, jsou nahlášeny a verze není dále použita. V případě bezproblémové instalace a testů, je release nasazen do provozu.

*Pozn. Detailní specifikace a návrh jednotlivých procesů je uveden dále v jednotlivých kapitolách.*

### **10.1.1. Provoz a podpora – první úroveň**

Výše uvedený proces ošetřuje neduh (z předešlých projektů), kterým jsou problémy plynoucí ze servisních zásahů přímo na produkčním prostředí (krizová situace, kdy například nelze odeslat clearingové informace do clearingového centra národní banky, bez kterých banka nemůže provést vypořádání transakcí – detaily vypořádání a bankovních IS jsou významně nad rámec této práce, a proto se jimi autor nebude nadále zabývat). Při těchto zásazích jsou zpravidla problémy jednorázově vyřešeny opravou na místě, což umožní zákazníkovi dále pracovat a přerušovaný proces dokončit. Pokud tato aktivita neskončí požadavkem na změnu nebo snížením priority řešení chyby, tak nastane v budoucnu velký problém, neboť není zabráněno jeho opětovnému výskytu.

Opětovný výskyt incidentu vzniká při vytváření nové verze, existuje-li nová funkcionality nebo velké množství patchů - některé opravy (právě z produkce) vypadnou. Aktualizace dokumentace a ukončení servisního zásahu předáním klíčové informace v podobě požadavku na změnu (Change Request), požadavku na zlepšení (Enhancement Request) nebo chyby, zamezí pospanému stavu.



**Obrázek 48: Provoz a podpora – první úroveň**

Popis procesu provoz a podpora první úrovně

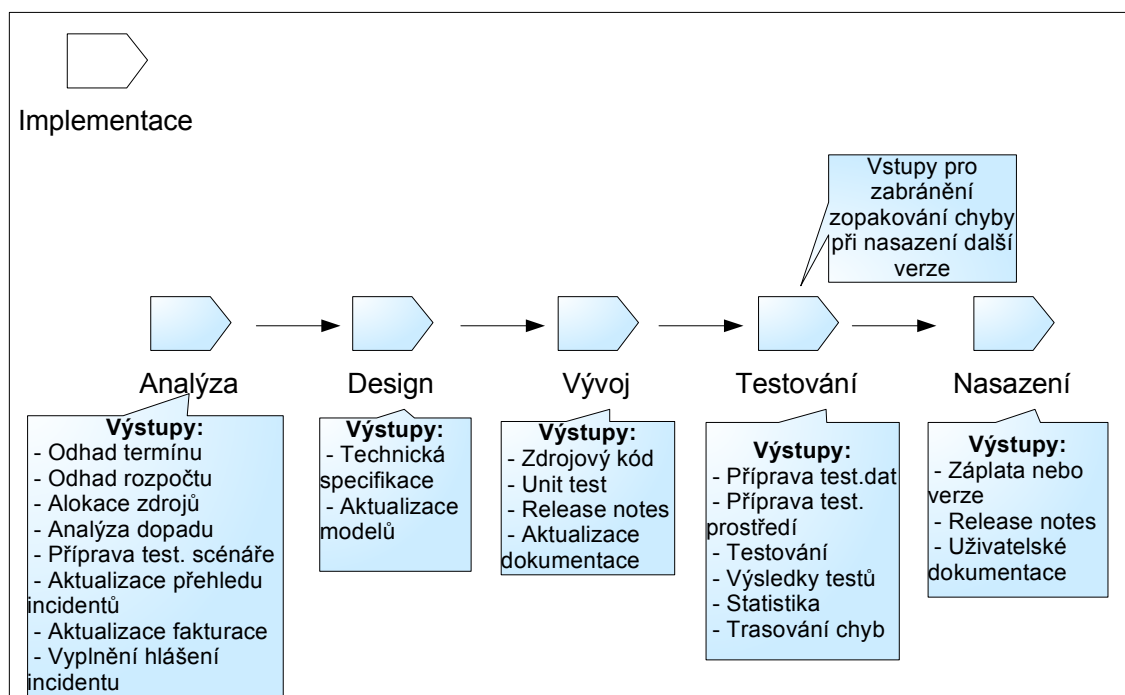
Proces	Popis
Analýza dopadu	Při analýze dopadu řešitel stanovuje dopad incidentu do řešení, termínu, rozpočtu. Identifikuje potenciálně rizikové oblasti a navrhuje cesty jejich eliminace. V případě urgentního problému je kladen důraz na dopad zákazníka a jeho podnikání s cílem zjistit, v které části IS může být chyba.
Analýza	Analýza zahrnuje dvě oblasti – hledání důvodu chyby a variant její opravy. Dále je akcentována snaha o nalezení takového dočasného řešení, které umožní zákazníkovi i nadále fungovat (byť například omezeně).



Oprava chyby	Cílem tohoto procesu je oprava chyby na místě přímo v produkčním prostředí zákazníka.
Předání informací do vývoje	Po opravě chyby a zprovoznění zákaznických procesů ve standardním módu, jsou předány klíčové informace v podobě CR, ER nebo chyby do developmentu. Tento transfer zabraňuje emergenci chyby v budoucnu (v další účtovací periodě).

### 10.1.2. Provoz a podpora – druhá úroveň

Cílem tohoto procesu je zákazníkovi poskytovat služby ve standardním, plánovaném módu. Tento proces v zásadě kopíruje životní cyklus tvorby software. Přitom má vlastní specifika – např. je nutné zohlednit stávající infrastrukturu, architekturu, aktualizovat existující dokumentaci atd.



Obrázek 49: Provoz a podpora – druhá úroveň

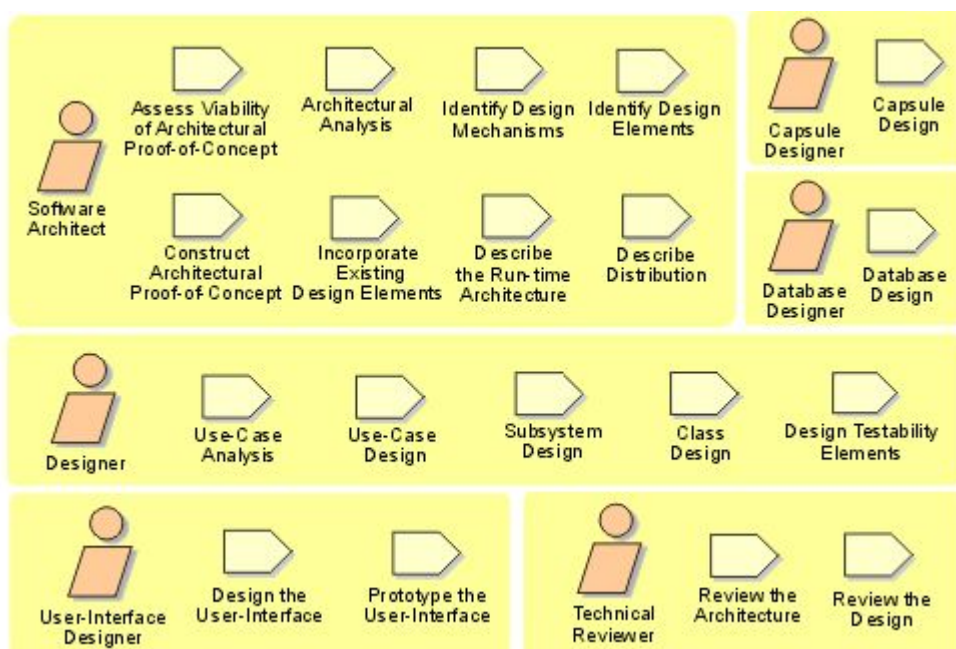
Popis procesu provoz a podpora druhé úrovně

Proces	Popis
Analýza	<p>Analýza spočívá v identifikaci požadavku uživatele a jeho standardizovaného zachycení. Součástí analýzy je návrh GUI, popsání scénářů, odhad času a rozpočtu.</p> <p>Po odsouhlasení analýzy ze strany zákazníka jsou alokovány zdroje a aktualizována projektová čísla.</p>
Design	<p>Navržení realizace požadavku formou komponentové architektury. Jsou vytvořeny odpovídající nové model(y), případně modifikovány stávající. V rámci designu jsou také revidovány dopady do celkové architektury řešení, infrastruktury HW a SW a bezpečnosti.</p>
Vývoj	<p>Požadavek je naprogramován, otestován na úrovni unit a integrován se zbytkem řešení.</p>
Testování	<p>Příprava testovacích scénářů probíhá již ve fázi analýzy.</p> <p>Cílem testování je ověřit soulad mezi očekáváním zákazníka a dodaným výstupem. V rámci KC je ověřeno chování funkcionality izolovaně i integrovaně. Dále je otestován výkon a bezpečnost.</p> <p>Výstupem testování je otestovaná verze, seznam chyb a release notes.</p>
Nasazení	<p>Na základě výstupu z testování je rozhodnuto o nasazení/nenasazení verze. V případě kladného rozhodnutí je zákazníkovi předána verze k nasazení do jeho vlastního testovacího prostředí. Formou akceptačních testů je rozhodnuto o nasazení/nenasazení verze do produkce.</p>

## 10.2. Analýza

### 10.2.1. Analýza a design dle metodiky RUP<sup>1</sup>

RUP definuje analýzu a design jako jednu z disciplín softwarového vývoje. Její klíčové role a aktivity znázorňuje níže uvedené schéma. Při detailnějším pohledu je vidět, že výše uvedený proces je vhodné použít pro analýzu a design nového systému, který je stavěn od základů. Tento proces a výstupy jsou ale nepoužitelné a zbytečně složité pro aplikaci nad již běžícím systémem.

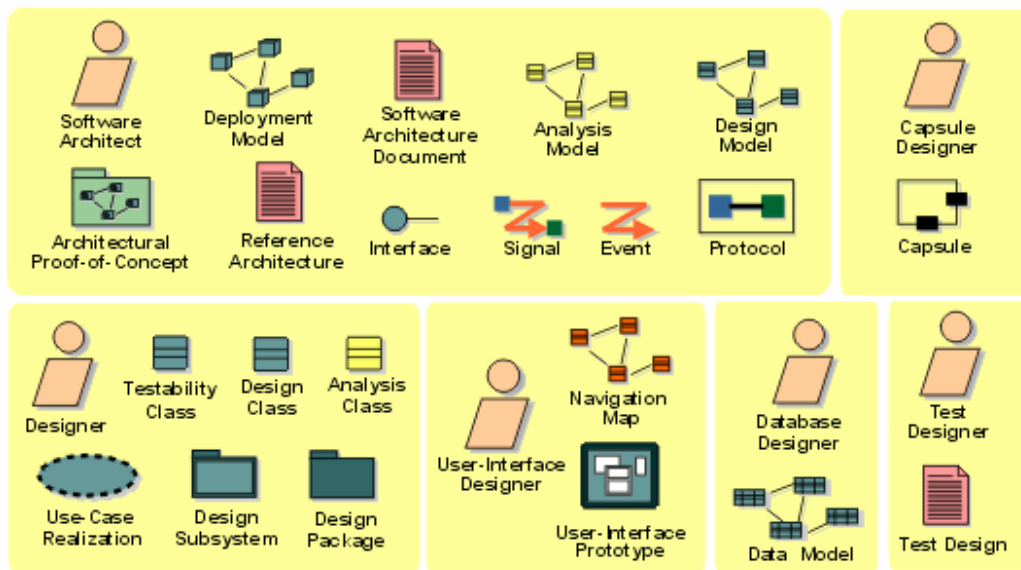


Obrázek 50: Role a aktivity disciplíny analýza a design dle RUP [RUP]

<sup>1</sup> Následující kapitoly striktně dodržují strukturu:

Disciplína dle metodiky RUP – role, aktivity, výstupy

Návrh disciplíny pro potřeby kompetenčního centra



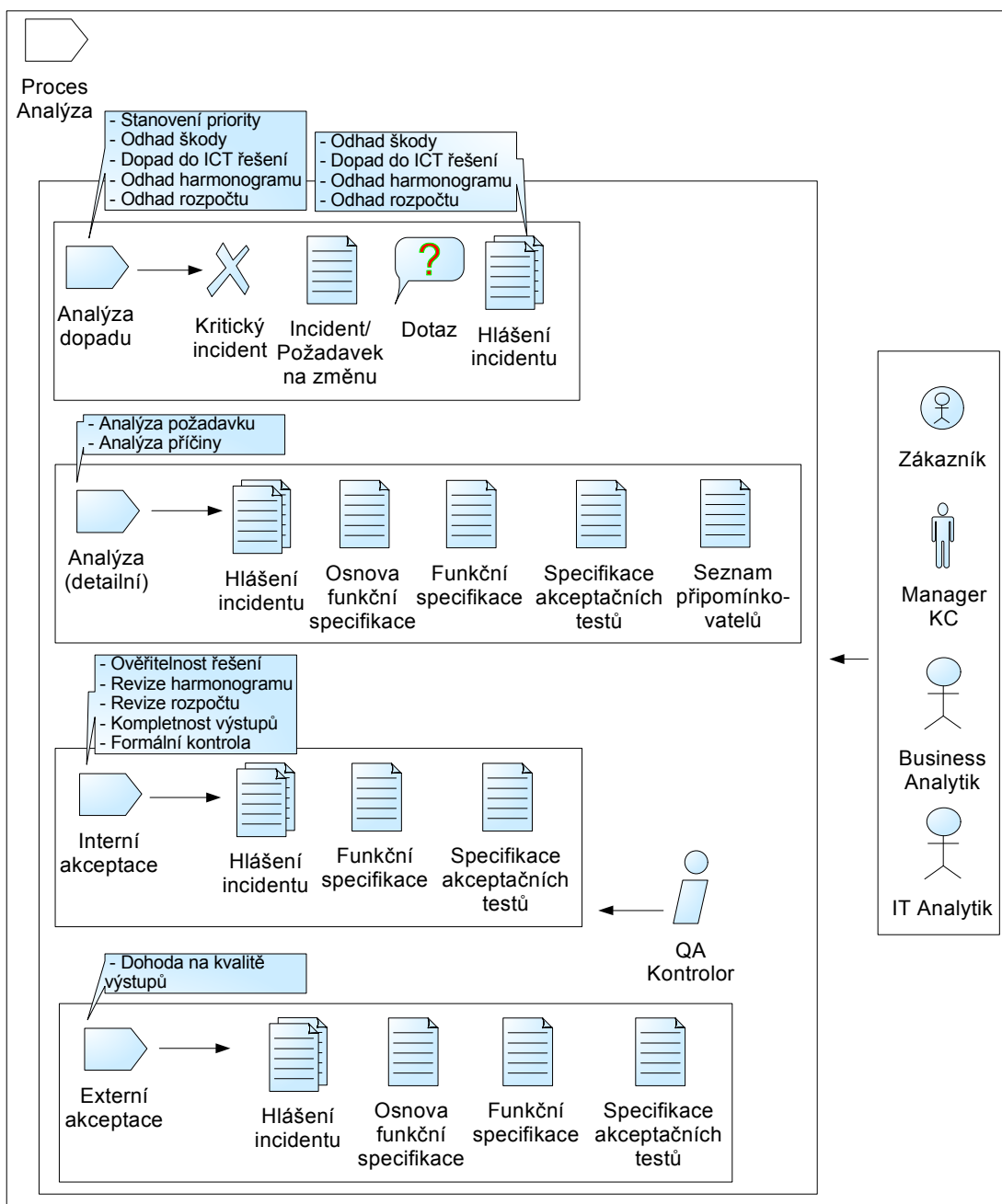
The roles involved and the artifacts produced in the Analysis & Design discipline.

### Obrázek 51: Role a výstupy disciplíny analýza a design dle RUP [RUP]

Pro detailní popis, vysvětlení disciplíny, rolí, aktivit, artefaktů atd. autor přímo odkazuje na metodiku Rational Unified Process [RUP]. Metodika RUP sama nedává doporučení pro použití vybrané podmnožiny aktivit a jejich výstupů pro potřeby provozních projektů. Na základě výsledku analýzy požadavků na realizaci provozních projektů a procesů metodiky RUP, vytvořil autor této práce vlastní návrh procesu, který představuje v následující kapitole.

#### 10.2.2. Návrh procesu analýza

Vzhledem ke komplexnosti provozních projektů, je třeba analýzu realizovat v podstatě dvakrát. V prvním kole se jedná o tzv. analýzu dopadu s cílem stanovit prioritu problému a případné škody (viz. dále). Dále také stanovit dopad do řešení, odhad času a rozpočtu na realizaci. Detailní analýza incidentu nebo požadavku, jako druhý krok, má za cíl zachytit a zanalyzovat požadavky zákazníka nebo zjistit příčinu chyby.



Obrázek 52: Návrh procesu analýza v KC

Analýzu realizuje business analytik – zpravidla pro případ nové, složitější funkcionality. Alternativně pro méně složité, náročné požadavky, opravu chyb může být připravena přímo programátorem, který tuto funkcionalitu zná. V druhém případě je vhodné neopomenout výstup zrevidovat s cílem zabezpečit kvalitu řešení.

#### **10.2.2.1. Analýza dopadu**

Analýza dopadu je prvním přirozeným krokem po přijetí a registraci incidentu. Přidělený analytik zjišťuje důležitost (prioritu, severitu) a podstatu incidentu, jeho dopady do produkce. Toto je klíčová činnost neboť zpoždění opravy v případě tzv. „hotline incidentu“ v produkčním prostředí může skončit vysokou pokutou jak pro zákazníka, tak i následným přenesením škod na kompetenční centrum zákazníkem. Zde je třeba připomenout, že poškození dobrého jména společnosti je citelnou a obtížně vyčíslitelnou škodou.

Na analýze dopadu aktivně spolupracuje zadavatel s následujícím výsledkem:

- Kritický produkční incident s vysokou prioritou, který je třeba bez odkladu vyřešit (např. Nedaří se navázat komunikaci s clearingovým centrem národní banky).
- Ne-kritický incident nebo požadavek na novou funkcionalitu, který je možné řešit v rámci standardního životního cyklu podpory (např. nedaří se vygenerovat report XY, který je používán velmi zřídka).
- Incident je dotazem, žádostí o detailnější informace (např. Jak vygenerovat report, který bude obsahovat požadované údaje, přičemž procedura není popsána v uživatelské dokumentaci).

Jako druhé kolo je analyzována podstata problému a odhadnuta pracnost a termín řešení. Je zřejmé, že jiný cyklus řešení má urgentní chyba v produkčním prostředí, kterou je třeba řešit ihned (zpravidla na místě přímo u zákazníka) a nekritická oprava, která je řešena přímo v KC.

### **10.2.2.2. Analýza požadavků zákazníka**

V tomto kroku jsou požadavky zákazníka sbírány a analyzovány v souladu s tradičními metodikami softwarového vývoje. Cílem analýzy je pochopit přání a očekávání zákazníka, stanovit hranici nové funkcionality požadavků a dopady do stávajícího řešení. Klíčovým vstupem této disciplíny je požadavek na změnu, požadavek zákazníka, helpdeskový telefonát, mail atd. Výstupem této činnosti je dokument funkční specifikace. Tento dokument obsahuje mj. standardní vizuální modely pro snazší pochopení a prezentaci řešeného problému. Formát výstupu závisí na typu a velikosti požadavku – může to být rozsáhlý dokument nebo několik odstavců v příloze požadavku zákazníka.

Tvorbu analytického výstupu nelze podcenit – jednak se jedná o pochopení a odsouhlasení požadavku zákazníka, jednak proto, že na základě této definice je připraven a odsouhlasen plán a rozpočet řešení. Oblast, která bývá v této chvíli nejvíce podceňována je úplnost zadání a zvolení takové míry detailu, aby mohla být definována akceptační procedura a akceptační kritéria.

Specifikace akceptačních testů je druhým velkým výstupem fáze analýzy. Právě v tomto dokumentu dochází k odsouhlasení procesu a kritérií akceptace. Cílem tohoto postupu je stanovit proces a podmínky, za kterých lze prohlásit, že řešení splňuje požadavky zákazníka.

Velmi často je těžké se zákazníkem na pravidlech akceptace předem dohodnout, neboť mu omezují prostor pro oblíbené „kličkování“, prodlužování akceptace, přidávání nových požadavků a měnění starých bez vlivu na rozpočet a termín. Z výše uvedeného vyplývá, že je nezbytné trvat a předem odsouhlasit kritéria akceptace.

### **10.2.2.3. Akceptace**

Uvedené výstupy je třeba odsouhlasit na dvou úrovních:

- Interně.
- Externě.

## **A. Interní akceptace**

Cílem interní akceptace je prověřit, že řešení je skutečně realizovatelné, nezahrnuje nereálné nebo neuskutečnitelné požadavky a očekávání (např. z výkonnostního úhlu pohledu) nebo konflikt s interními standardy dodavatele. Důležitý je také soulad s odsouhlaseným harmonogramem a rozpočtem. V neposlední řadě je ověřena kompletnost a formální správnost výstupu.

## **B. Externí akceptace**

Aby bylo možné přejít do druhého kroku – návrh řešení, je nutné se zákazníkem odsouhlasit funkční specifikaci. Již při přípravě dokumentu je odsouhlasena struktura a zodpovědné osoby, které mají pravomoc dokument připomínkovat a schválit. Tento krok zabrání prodloužení schvalovacího procesu. Důležitost také spočívá v zabezpečení, že všichni vedoucí závislých projektů neb aktivit realizovaných liniově, jsou včas a řádně informováni.

## **10.3. Design**

### **10.3.1. Design dle metodiky RUP**

V pojetí metodiky RUP je design součástí disciplíny Analýza [RUP]. Ačkoli je RUP uznávanou metodikou, autor této práce rozporuje spojení těchto disciplín. Z povahy věci, cílem analýzy je zformulovat a zachytit požadavky zákazníka, cílem designu je navrhnout jejich řešení pomocí odpovídajících metod.

Spojitost disciplín nelze vidět ani pohledem přes role, neboť analýzu zpravidla realizuje jinak zdatný, znalý člověk, než-li design. Neduhem pak bývá, že analýza je moc technická a zákazník ji odsouhlasí, i když ji nerozumí. Pro programátora je design zase moc vágní, přičemž je nucen mnoho informací dodatečně zjišťovat.



Nákladově efektivní hledisko nelze samozřejmě podcenit, ale na druhou stranu jej neberme jako jediné. Z krátkodobého hlediska je dobře a úplně popsané řešení pravděpodobně dražší, nicméně ze střednědobého nebo dlouhodobého úhlu pohledu, kdy dochází k přirozené migraci zdrojů, ztráty nepsaného know-how, je toto cestou jak naopak velmi ušetřit.

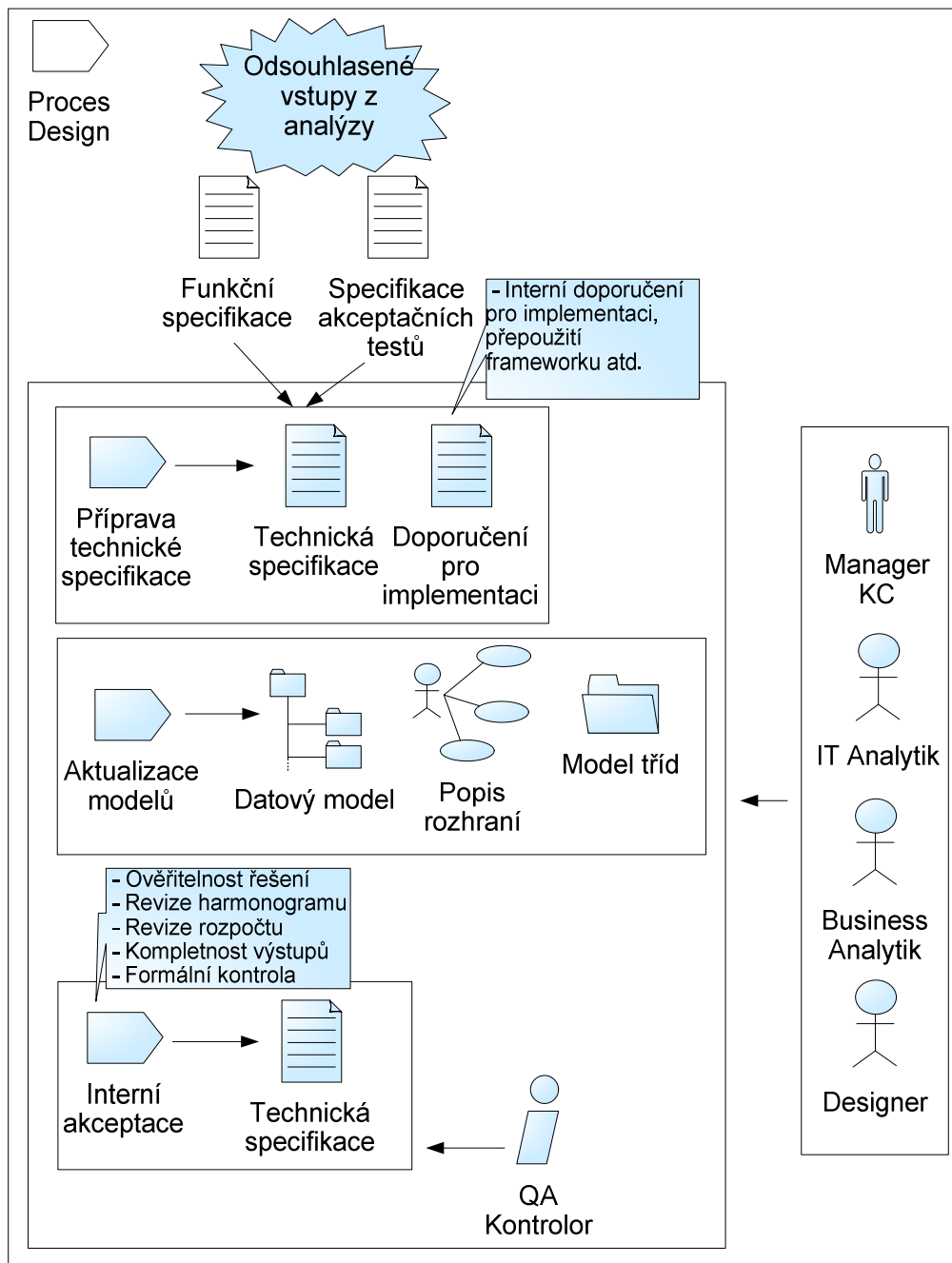
### **10.3.2. Návrh procesu design pro KC**

Po odsouhlasení analytických výstupů zákazníkem, jsou tyto výstupy dále detailněji rozpracovány s cílem připravit zadání pro programátora. Při této činnosti jsou aktualizovány architektonické modely - datový model, model tříd, popisů rozhraní, atd. Tyto činnosti je třeba realizovat až po formálním odsouhlasení analýzy, jinak hrozí, že budou předělávány. Udržet integritu a kontinuitu modelů oproti řešení, pokud nejsou používány odpovídající nástroje na řízení verzí a změn, je v praxi složité.

Vstupem do této fáze je tedy odsouhlasená funkční specifikace zákazníkem. Výstupem je technická specifikace. Ta může mít mnoho podob (zpravidla modely a dokument), nicméně vždy musí obsahovat:

- Detailní popis, jak má být funkcionality vyvinuta.
- Návrh na nové využití komponent.
- Relevantní odkazy do funkční specifikace.
- Odkazy na doplňkové informace, kontaktní osoby atd.

Design typicky navrhuje designér, kterým bývá velmi zkušený programátor.



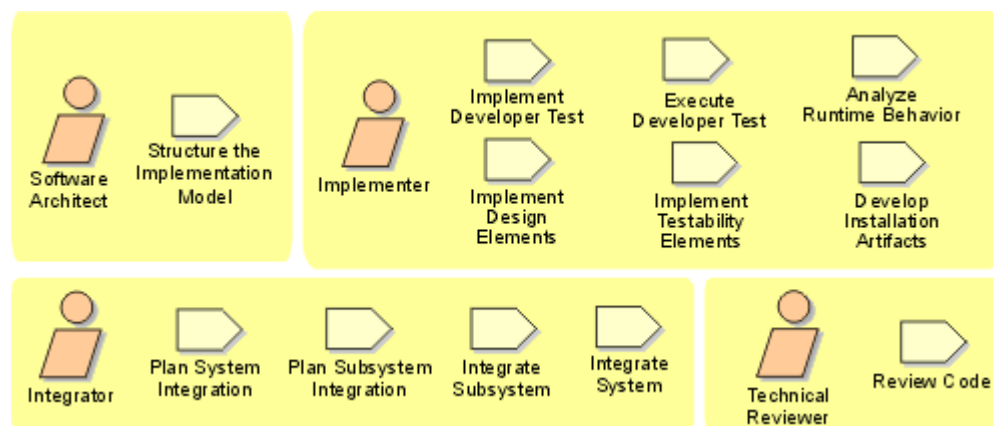
**Obrázek 53: Návrh procesu design v KC**

### 10.3.2.1. Interní Akceptace

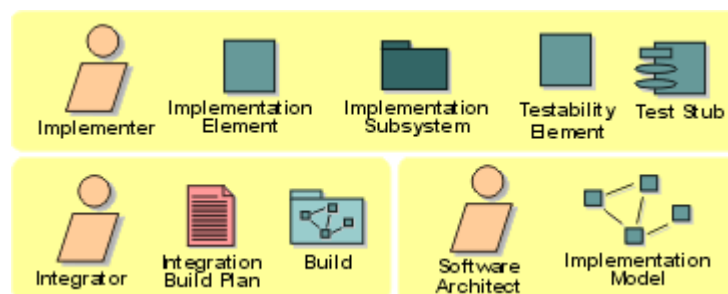
Ve fázi designu nelze interní akceptaci podcenit. Externí akceptace může být vynechána, neboť zpravidla na straně zákazníka nebývá dostatečná znalost (záleží na formě hostingu, outsourcingu atd.). Interní akceptace výstupu má za cíl ověřit, že požadavky zákazníka budou naplněny a že neexistují bílá místa nebo mezery mezi očekávanou funkcionalitou a plánovanou implementací. Revize harmonogramu je také součástí akceptace.

## 10.4. Implementace

### 10.4.1. Implementace dle metodiky RUP



Obrázek 54: Role a aktivity disciplíny implementace dle RUP [RUP]



Obrázek 55: Role a výstupy disciplíny implementace dle RUP [RUP]

Pro detailní popis, vysvětlení disciplíny, rolí, aktivit, artefaktů atd. autor přímo odkazuje na metodiku Rational Unified Process [RUP]. Metodika RUP sama nedává doporučení pro použití vybrané podmnožiny aktivit a jejich výstupů pro potřeby provozních projektů. Na základě výsledku analýzy požadavků na realizaci provozních projektů a procesů metodiky RUP, vytvořil autor této práce vlastní návrh procesu, který představuje v následující kapitole.

#### **10.4.2. Návrh procesu implementace pro KC**

Cílem fáze implementace je vytvořit jednotlivé komponenty tak, aby byly splněny požadavky uživatele (definované, analyzované a navržené ve funkční a technické specifikaci.)

Vstupem jsou schválené specifikace - technická a funkční.

Typickým výstupem implementace jsou:

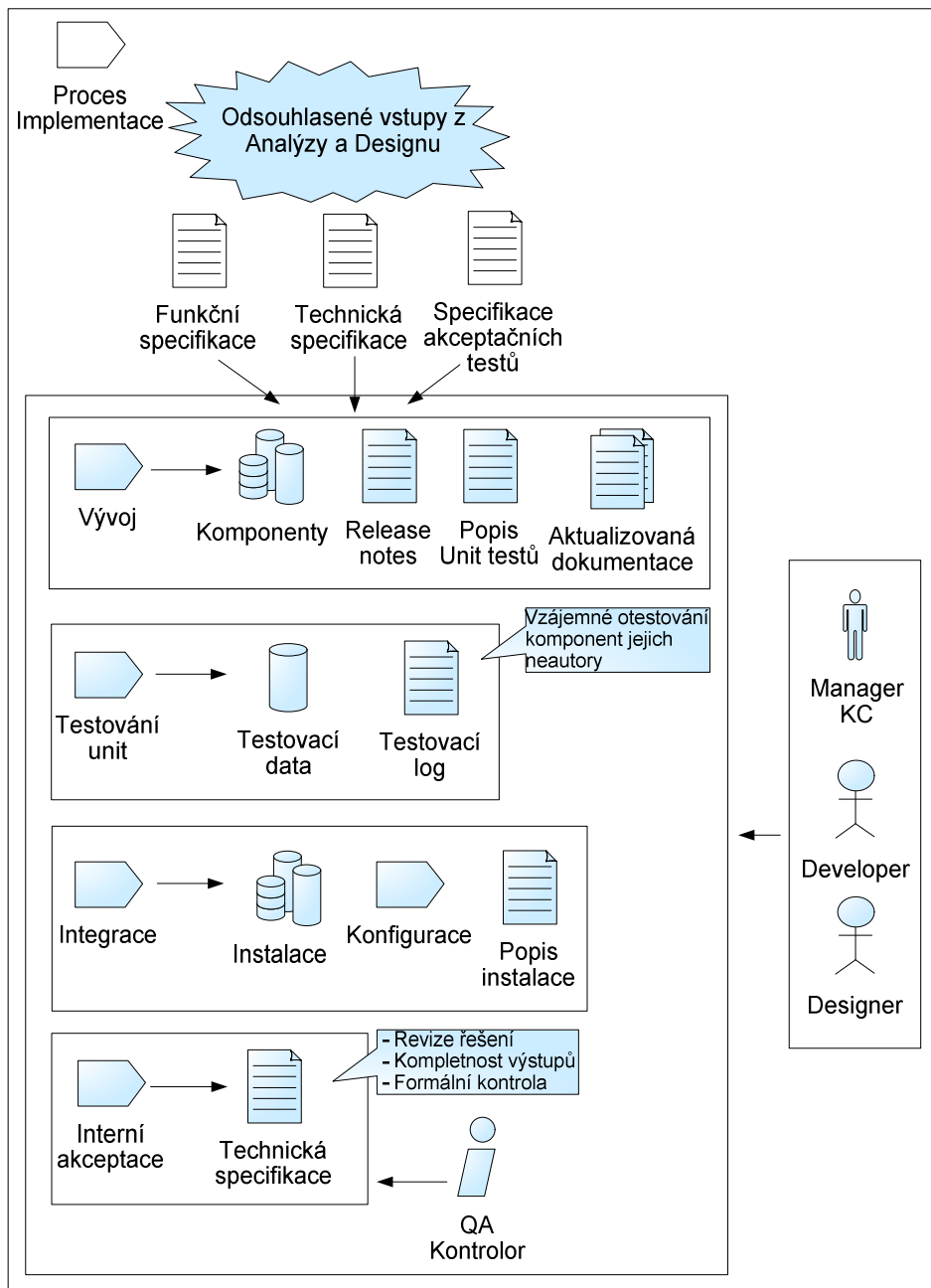
- Zdrojový kód (verzovaný v odpovídajícím nástroji).
- Release notes, které zachycují změny a nezbytné vstupy pro instalaci a konfiguraci.
- Popis Unit testů – co a jak má být testováno.
- Aktualizace dokumentace IS.

Vývoj je typicky realizován programátorem za pomoci IDE a v optimálním případě i frameworku aplikace. Součástí implementace jsou také úvodní fáze testování realizované přímo programátorem.

#### **10.4.2.1. Testování unit**

Nedílnou součástí vývoje je testování komponent (unit). Cílem tohoto testování je ověřit chování jednotlivých částí modulů oproti funkční a technické specifikaci. Vstupem této činnosti je naprogramovaná funkcionalita, funkční a technická specifikace, testovací procedury, data a další. Výstupem testovacího procesu je testovací log, zachycující výsledky testů dle testovací specifikace, seznam chyb a otestovaná funkcionalita na úrovni unit.

Tato aktivita je typicky realizována autorem komponenty nebo vývojářem nepodílejícím se na vývoji komponenty (vzájemné otestování komponentem).



Obrázek 56: Návrh procesu implementace v KC

### 10.4.2.2. Integrace

Cílem tohoto kroku je integrovat jednotlivé komponenty v kompaktní, funkční celek mezi sebou – to v praxi znamená nainstalovat všechny komponenty a ověřit funkčnosti jejich rozhraní, komunikační kanály, konfiguraci atd.

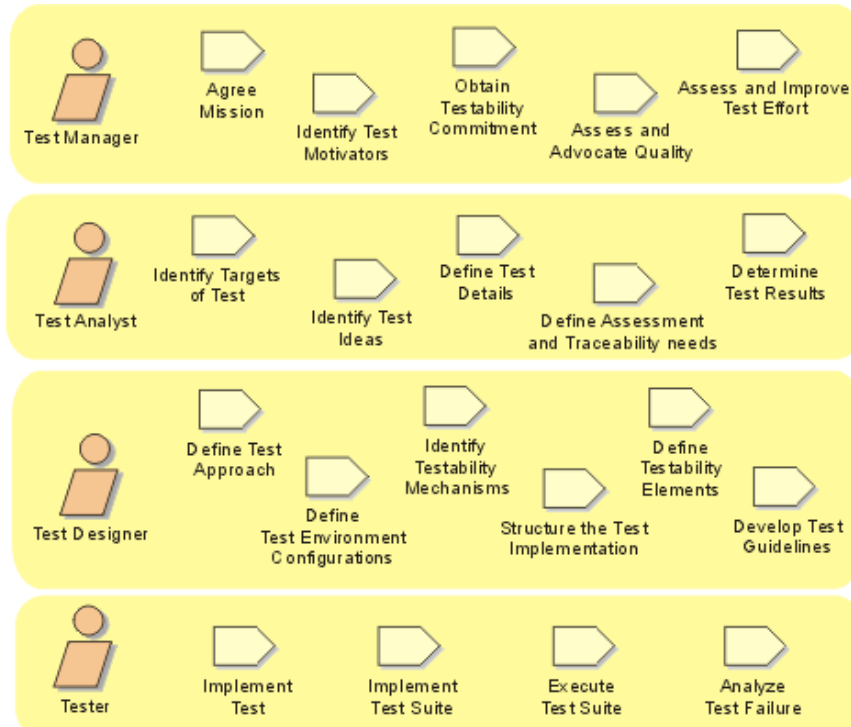
Vstupem jsou otestované komponenty na úrovni unit v různé podobě – vložené procedury v RDBMS, třídy programovacího jazyka nebo speciální služba třetí strany (např. SMTP server).

Výstupem je „kompilace“ samostatné instalace kompletního systému, vlastní instalace, konfigurace a ověření připravenosti pro testování. Výstupem je také postup instalace v podobě psaných instrukcí.

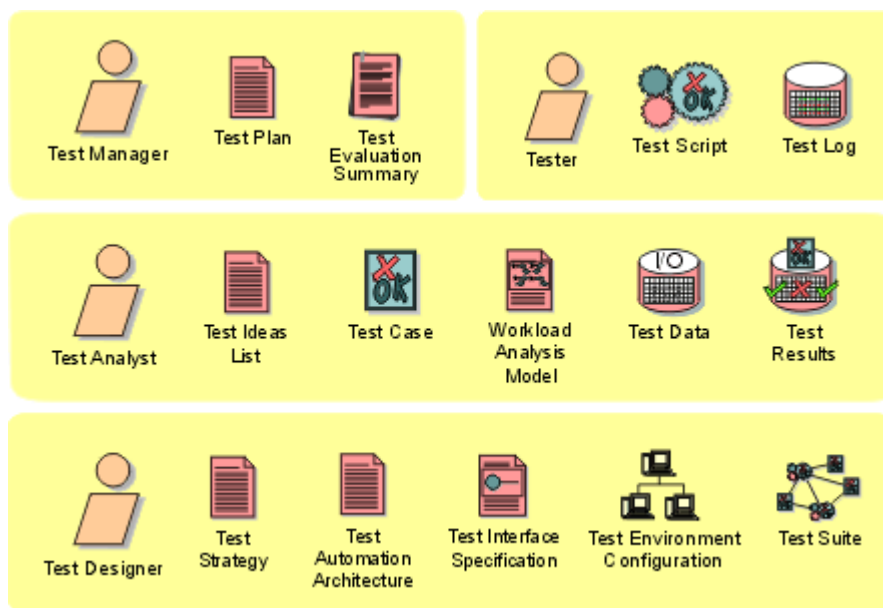
Tuto činnost obvykle vykonává zkušený vývojář.

## 10.5. Testování

### 10.5.1. Testování dle metodiky RUP



Obrázek 57: Role a aktivity disciplíny testování dle RUP [RUP]

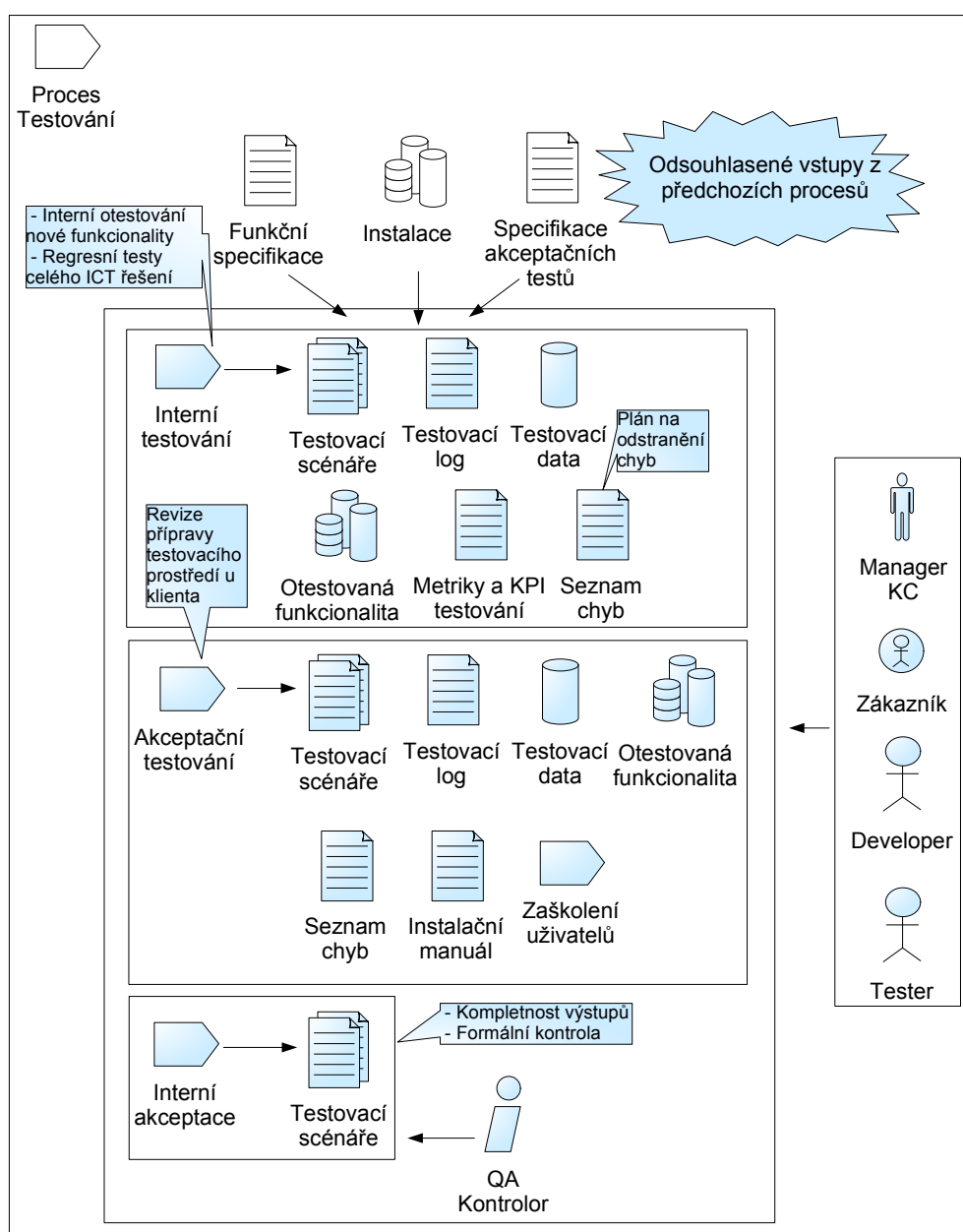


**Obrázek 58: Role a aktivity disciplíny testování dle RUP [RUP]**

Pro detailní popis, vysvětlení disciplíny, rolí, aktivit, artefaktů atd. autor přímo odkazuje na metodiku Rational Unified Process [RUP]. Metodika RUP sama nedává doporučení pro použití vybrané podmnožiny aktivit a jejich výstupů pro potřeby provozních projektů. Na základě výsledku analýzy požadavků na realizaci provozních projektů a procesů metodiky RUP, vytvořil autor této práce vlastní návrh procesu, který představuje v následující kapitole.



## 10.5.2. Návrh procesu testování pro KC



Obrázek 59: Návrh procesu testování v KC

Cílem testování je ověřit, že funkcionality splňuje požadavky uživatele oproti funkční a technické specifikaci. To je realizovatelné ve dvou krocích – nejprve interním otestováním a následně externí akceptací zákazníkem.

#### **10.5.2.1. Interní testování**

Interní testování slouží jako prostředek ověření kvality dodané funkcionality od programátorů.

Vstupem do této fáze je nainstalovaná a zkonfigurovaná aplikace (funkcionality), funkční specifikace a akceptační kritéria.

Výstupem je testovací scénář, otestovaná funkcionality, test log, seznam chyb včetně plánu na jejich odstranění a rozhodnutí, zda je možné funkcionality předat do externí akceptace či zda má být nejprve opravena, znovu otestována a pak teprve předána k akceptaci.

Při návrhu testovacího scénáře je nutné vzít v potaz testování chybných a hraničních hodnot v případech, kde to má smysl.

Testovací log přehledně zachycuje výsledky testů.

Tuto aktivitu typicky realizuje tester, analytik se znalosti business problematiky nebo vývojář, který není autorem funkcionality (pro menší, zřetelně ohraničenou funkcionality). Na rozdíl od vývoje je kladen také zvýšený důraz na tzv. regresní testy. Cílem regresních testů je ověřit, že přírůstek funkcionality přidaný do aplikace negativně neovlivňuje předchozí funkcionality systému. Po každém větším zásahu do aplikace je nutné přetestovat klíčovou funkcionality celé aplikace. Je na rozhodnutí manažera kompetenčního centra a odborníka na testování, zda je vhodnější aplikaci testovat manuálně, nebo automatizovaně pomocí testovacích robotů. Vzhledem ke zřejmým výhodám a nevýhodám obou variant, nelze jednoznačně rozhodnout, která varianta je vhodnější. Zde autor této práce vidí námět na další rozvoj – Jakým způsobem efektivně modifikovat testovací strategie, případně testované IS tak, aby bylo možné je plně testovat automatizovaně.

Návrh metrik efektivnosti testování (a vývoje) je nedílnou součástí této strategie.

### 10.5.2.2. Akceptační testy

Proces akceptace je druhou oblastí testování. Oproti interním testům je v této fázi nutná přítomnost zákazníka, neboť on je oprávněn rozhodnout, zda je aplikace způsobilá k provozu či nikoli.

Nezbytným předpokladem začátku akceptačních testů je připravenost infrastruktury u zákazníka – např. nainstalované servery, síťové prvky, prioritizace provozu atd. Instalaci může provádět zákazník, dodavatel nebo pověřená třetí strana – záleží na dohodě. Instalační manuál musí být popsán do takové míry, že není důležité, kdo podle něj bude aplikaci instalovat. V případě instalace aplikací/systémů třetích stran, typicky tuto činnost vykonává zákazník.

Nezbytným vstupem akceptace je existence odpovídajících dokumentů a software. Akceptaci dokumentů lze chápat jako jejich revizi z faktického a formálního úhlu pohledu. Jako podklad slouží funkční specifikace a odsouhlasené zápisy z jednání. Zákazník v definovaném čase provede revizi a buď dokumenty odsouhlasí nebo k nim zformuluje a předá své připomínky. V dohodnutém čase dodavatel připomínky zapracuje a nastává druhá akceptace. Počet akceptačních koleček je uzavřen teprve tehdy, existuje – li shoda mezi zákazníkem a dodavatelem nad obsahem a kvalitou dokumentů.

Akceptace softwarů a jeho komponent je závislá na UAT testech – podle plánu testů, specifikace testů, dohodnutých v rámci analytické fáze. Forma akceptace je různá – v jednoduchém případě stačí přetestovat ohraničený kus funkcionality. V případě velkých zásahů do aplikace je třeba stanovit plán testů a otestovat celou aplikaci.

Důležitou součástí UAT je školení. Řádně vyškolení uživatelé jsou důležití, pro zabránění generování chyb plynoucích z nepochopení nebo neznalosti aplikace.

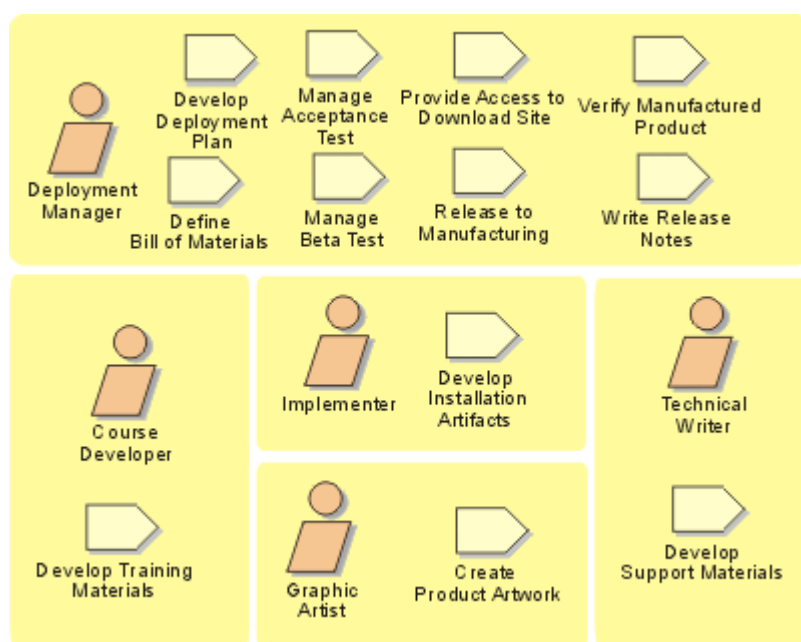
Akceptační testy realizuje personál zákazníka ve spolupráci s pracovníky kompetenčního centra dodavatele. Výsledky testů jsou řádně zdokumentovány. Hlášené chyby, závady a nesrovnalosti proti zadání jsou předávány formalizovaným způsobem.

Proces akceptace je popsán v plánu akceptace – ten obsahuje specifikaci testů, způsob hlášení problémů a zodpovědné osoby obou stran, které mohou stanovovat priority a incidentů. Frekvence oprav a nasazování nových verzí může být zmíněna taktéž.

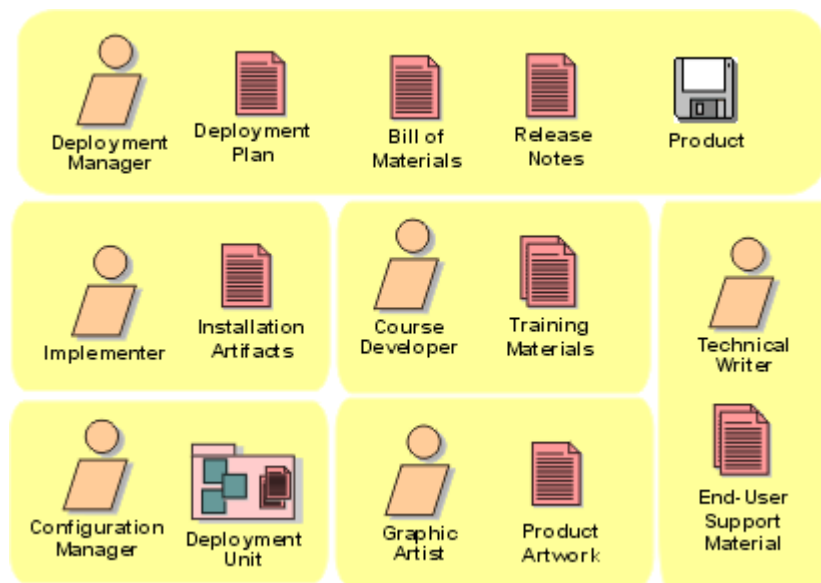
Výstupem této fáze je Protokol o akceptaci pro větší dílo, pro menší opravy stačí pouze podepsání vyřešení požadavku odpovědným pracovníkem. Výstupem je funkcionality v testovacím prostředí klienta připravená na rollout, pilotní provoz a nasazení do produkce.

## 10.6. Nasazení

### 10.6.1. Nasazení dle metodiky RUP



**Obrázek 60: Role a aktivity disciplíny nasazení dle RUP [RUP]**



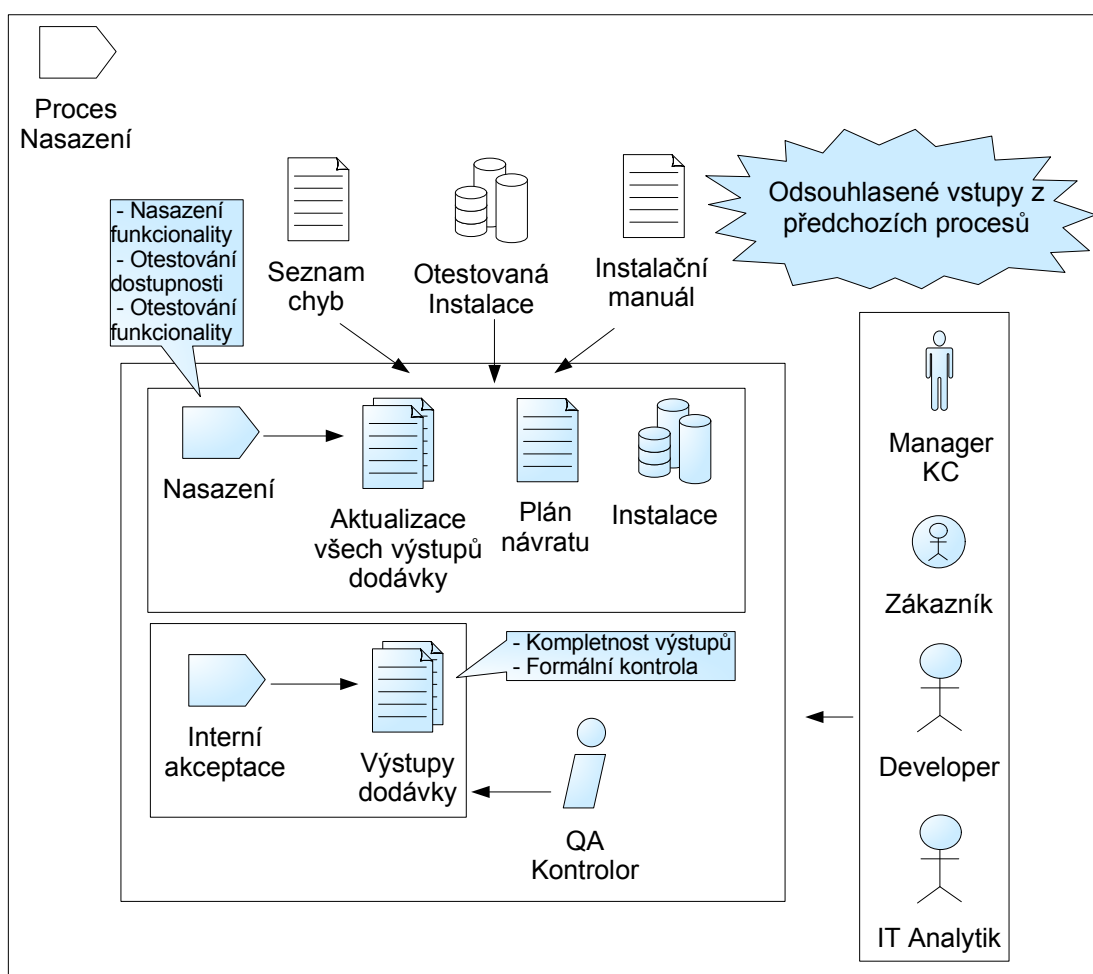
**Obrázek 61: Role a výstupy disciplíny nasazení dle RUP [RUP]**

Pro detailní popis, vysvětlení disciplíny, rolí, aktivit, artefaktů atd. autor přímo odkazuje na metodiku Rational Unified Process [RUP]. Metodika RUP sama nedává doporučení pro použití vybrané podmnožiny aktivit a jejich výstupů pro potřeby provozních projektů. Na základě výsledku analýzy požadavků na realizaci provozních projektů a procesů metodiky RUP, vytvořil autor této práce vlastní návrh procesu, který představuje v následující kapitole.

### **10.6.2. Návrh procesu nasazení pro KC**

V této fázi zodpovědnost plně přebírá zákazník. Pracovníci kompetenčního centra mohou na vyžádání podpořit zákazníka. Na základě výstupu z akceptačního testování je rozhodnuto o nasazení/nenasazení verze.

V případě kladného rozhodnutí zákazník s pomocí dodavatele nasadí verzi do produkčního prostředí. V závislosti na realizované změně a typu IS je otestována funkčnost v produkci (otestování dostupnosti, otestování pomocí testování transakce atd.) Součástí nasazení nové verze do produkčního prostředí je vytvoření strategie návratu zpět – tzv. Plán návratu. Cílem tohoto plánu je zachytit kroky vedoucí k návratu systému do stavu před aplikací změny. Plán je realizován v případě, že změna vyvolá takový dopad, že je ohroženo fungování instituce.



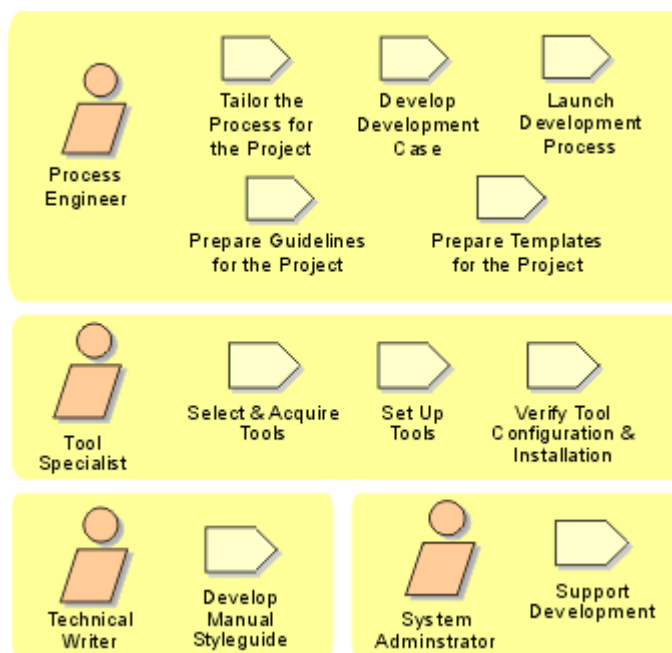
**Obrázek 62: Návrh procesu nasazení v KC**

Nasazování změn až po úspěšné akceptaci v testovacím prostředí u klienta významně snižuje riziko výskytu problémů v produkci vázaných na rozdíly mezi prostředím klienta a dodavatele. Otestováním změn v testovacím prostředí u klienta je riziko minimalizováno. Optimální cesta je samozřejmě udržet všechna prostředí konzistentní vůči sobě, což mnohdy z povahy podnikání, bezpečnosti, částečného rozvoje aplikací vlastní silou zákazníka atd. nejde. To, jak udržet konzistentní prostředí vůči sobě je významně nad rámec této práce - zde autor této práce vidí námět na její další rozvoj.

## 11. Autorův návrh podpůrných procesů KC

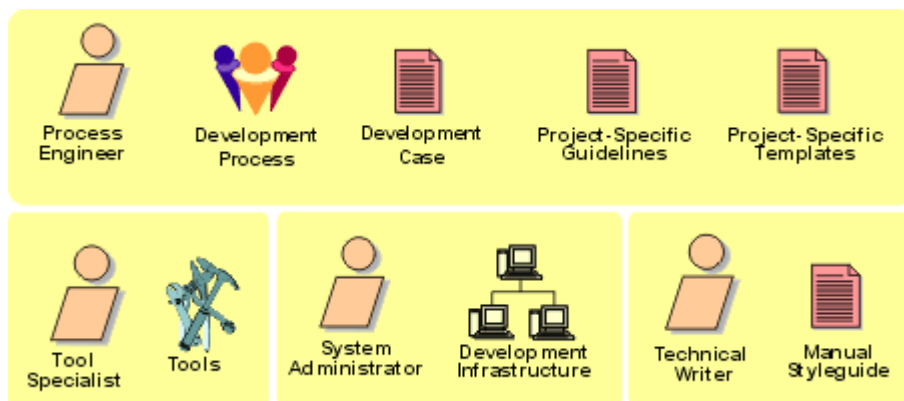
Bezchybnou realizaci obchodních procesů usnadňují podpůrné procesy. Tato skupina procesů (v kontextu této práce jsou realizační procesy synonymem pro procesy obchodní) zahrnuje procesy, které nejsou čistě obchodní a nepřinášejí přidanou hodnotu přímo pro zákazníka – nicméně bez nich obchodní procesy kompetenčního centra nemohou být realizovány. Cílem této skupiny procesů je podpořit rutinní, každodenní aktivity realizované v KC.

### 11.1. Příprava a správa prostředí dle metodiky RUP



Obrázek 63: Role a aktivity disciplíny správa prostředí dle RUP [RUP]





**Obrázek 64: Role a výstupy disciplíny správa prostředí dle RUP [RUP]**

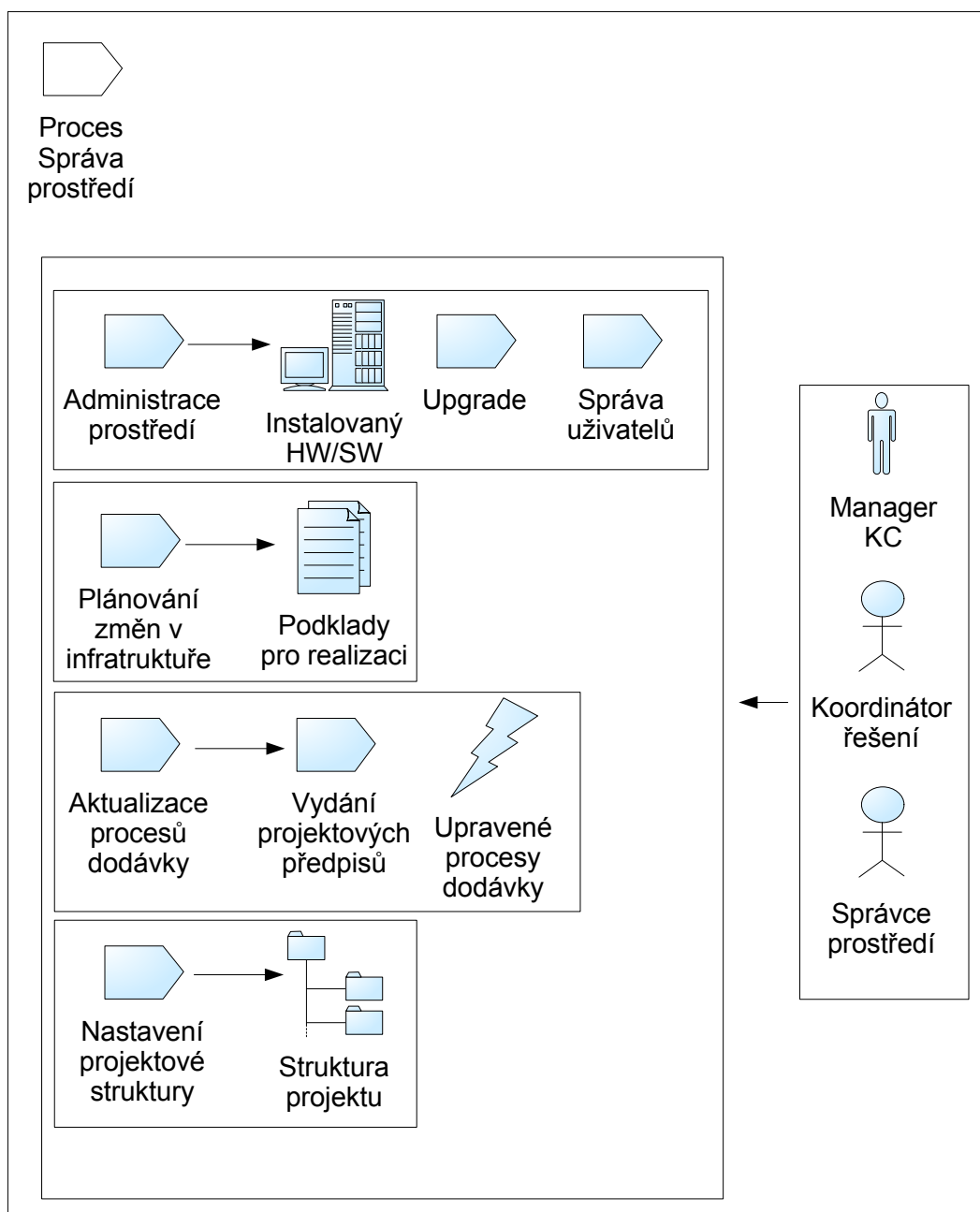
Pro detailní popis, vysvětlení disciplíny, rolí, aktivit, artefaktů atd. autor přímo odkazuje na metodiku Rational Unified Process [RUP]. Metodika RUP sama nedává doporučení pro použití vybrané podmnožiny aktivit a jejich výstupů pro potřeby provozních projektů. Na základě výsledku analýzy požadavků na realizaci provozních projektů a procesů metodiky RUP, představuje autor této práce vlastní návrh procesu.

## 11.2. Návrh procesu správa prostředí pro KC

Proces příprava a správa prostředí zachycuje činnosti pokrývající instalaci a konfiguraci hardware a software jednotlivých vývojářů, případně upgrade jednotlivých komponent. Dále zahrnuje nastavení projektové adresářové struktury, vytvoření projektově specifických guidelines (např. politiku verzování).

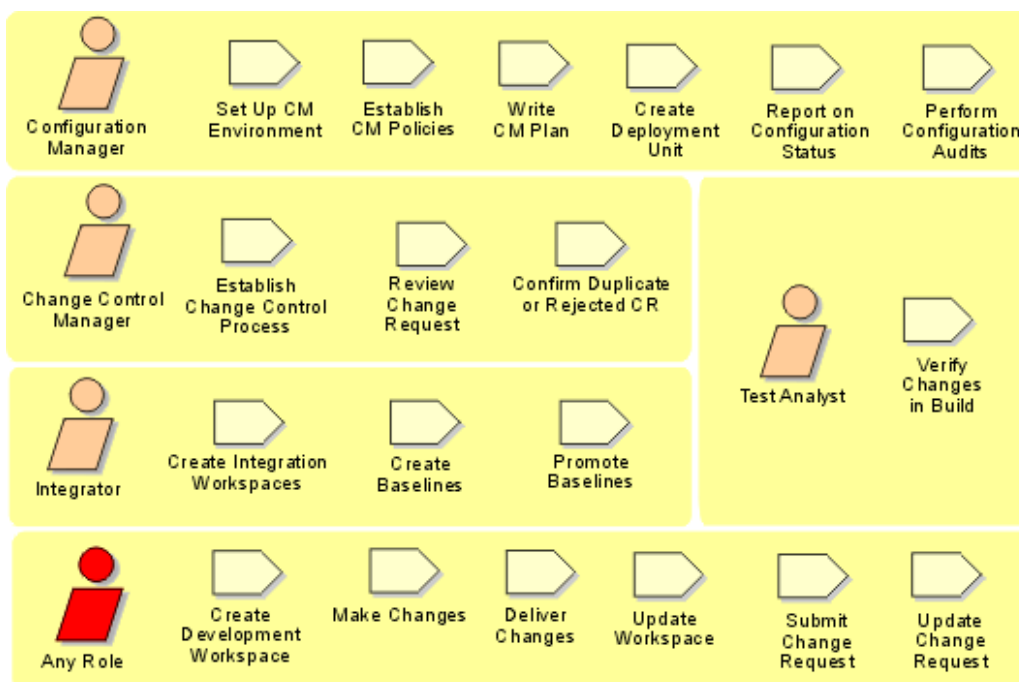
Autor této práce si dovoluje proces převzít a rozšířit jej navíc o potřebu plánování změn a posílení infrastruktury, upgrade atd. Dále jej rozšiřuje o administraci prostředí, která převážně znamená nastavení přístupových práv k jednotlivým částem infrastruktury, vytvoření vlastního prostoru pro vývoj software developerům (aby si vzájemně nemodifikovali zdrojový kód) a aktivní přidělování diskových kapacit a ostatních zdrojů (např. licencí software).

Tuto činnost typicky vykonává správce prostředí, kterým je zkušený administrátor a pracovník se znalostí sítí.

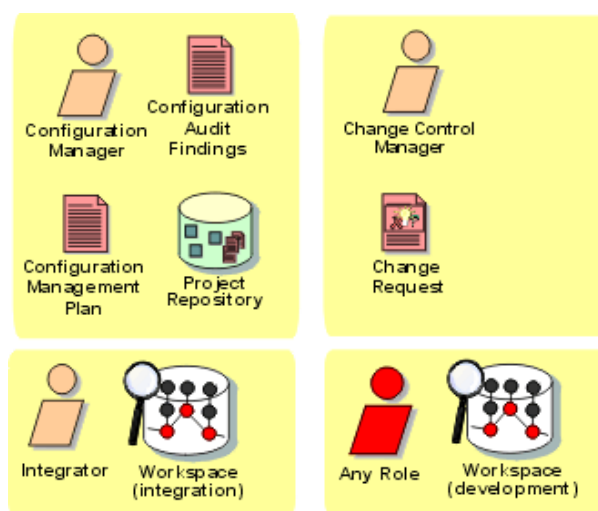


**Obrázek 65: Návrh procesu správa prostředí v KC**

### 11.3. Řízení konfigurací dle metodiky RUP



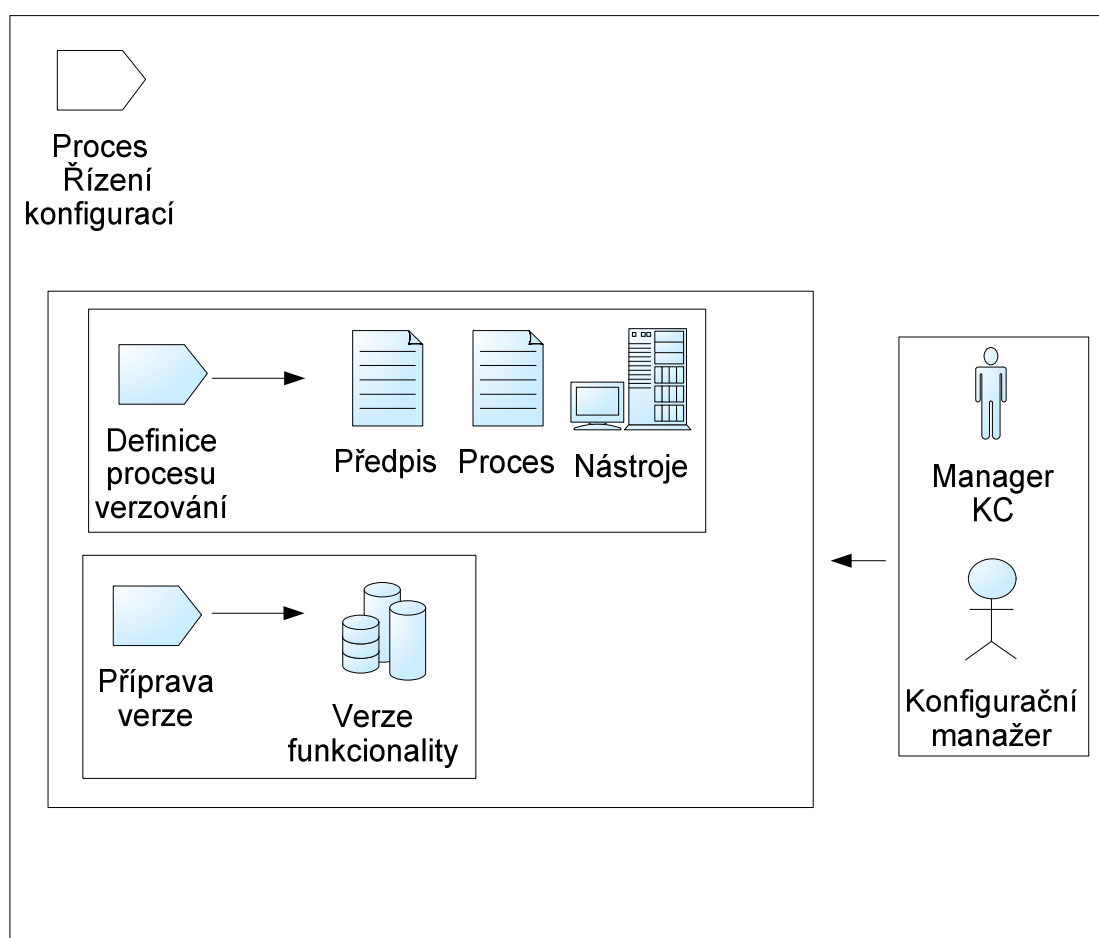
Obrázek 66: Role a aktivity disciplíny řízení konfigurací dle RUP [RUP]



Obrázek 67: Role a výstupy disciplíny řízení konfigurací dle RUP [RUP]

Pro detailní popis, vysvětlení disciplíny, rolí, aktivit, artefaktů atd. autor přímo odkazuje na metodiku Rational Unified Process [RUP]. Metodika RUP sama nedává doporučení pro použití vybrané podmnožiny aktivit a jejich výstupů pro potřeby provozních projektů. Na základě výsledku analýzy požadavků na realizaci provozních projektů a procesů metodiky RUP, představuje autor této práce vlastní návrh procesu.

## 11.4. Návrh procesu řízení konfigurací pro KC



Obrázek 68: Návrh procesu řízení konfigurací v KC

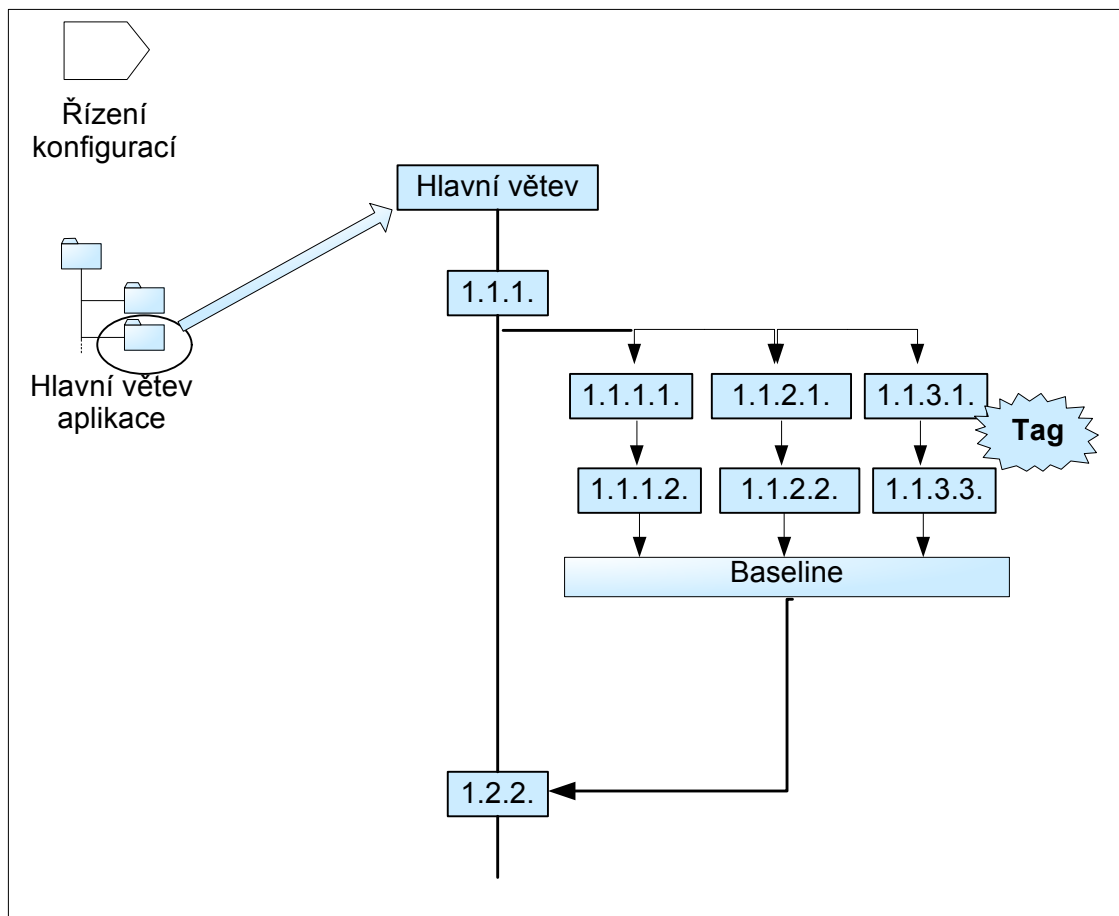
Pro potřeby kompetenčního centra je proces řízení konfigurací dle RUP zbytečně složitý. V rámci tohoto procesu je třeba nastavit pouze verzování. To zahrnuje předpis, jak nakládat se zdrojovým kódem a jakým postupem a kdy vytvářet release (patch).

Při softwarovém vývoji je nezbytné udržet zpětnou kompatibilitu zdrojového kódu. V případě IS běžícího v produkci je problém komplexnější, jelikož IS v produkci a nová verze, která je připravována již nemusí být shodné. Kompatibilitu zdrojového kódu lze udržet pomocí tzv. „větvení“, kterou nástroje pro správu verzí běžně umožňují.

Pro každou opravu nebo novou verzi je vytvořena na „hlavní vývojové větvi“ „vedlejší větev“. Tím je zabezpečeno, že zdrojový kód vztahující se k dané změně je separován od zbytku aplikace. Při tvorbě nové verze nebo záplaty, jsou vybrané „vedlejší větve“ spojeny do hlavní a následně je vše zkompileováno v jeden funkční celek. Takto připravená verze je na větvích označena tzv. „baseline“.

Výhoda větvení umožňuje do různých verzí vkládat různou množinu funkcionality v různých verzích a tu identifikovat pomocí „baseline“. V případě, že by všechna funkcionalita byla pouze v jedné hlavní větvi, nebylo by možné jednoduše záplatovat, případně upgradovat IS vybranou skupinou nebo pouze jednou funkcionalitou.

Ačkoli autor této práce není vývojář, považuje problematiku „větvení“ a jejich „spojování“ za klíčovou při vývoji a hlavně při provozu IS. Důsledný proces v oblasti řízení konfigurací totiž zabraňuje takovým excesům jako je dodávka polotovarové funkcionality, případně dodání verze, ve které funkcionalita chybí. Není řídkým jevem v praxi, že funkcionalita dodaná v první verzi, se „záhadně“ ztratí z verze např. 3.1. Toto je způsobeno právě nepořádkem v procesu řízení konfigurací.



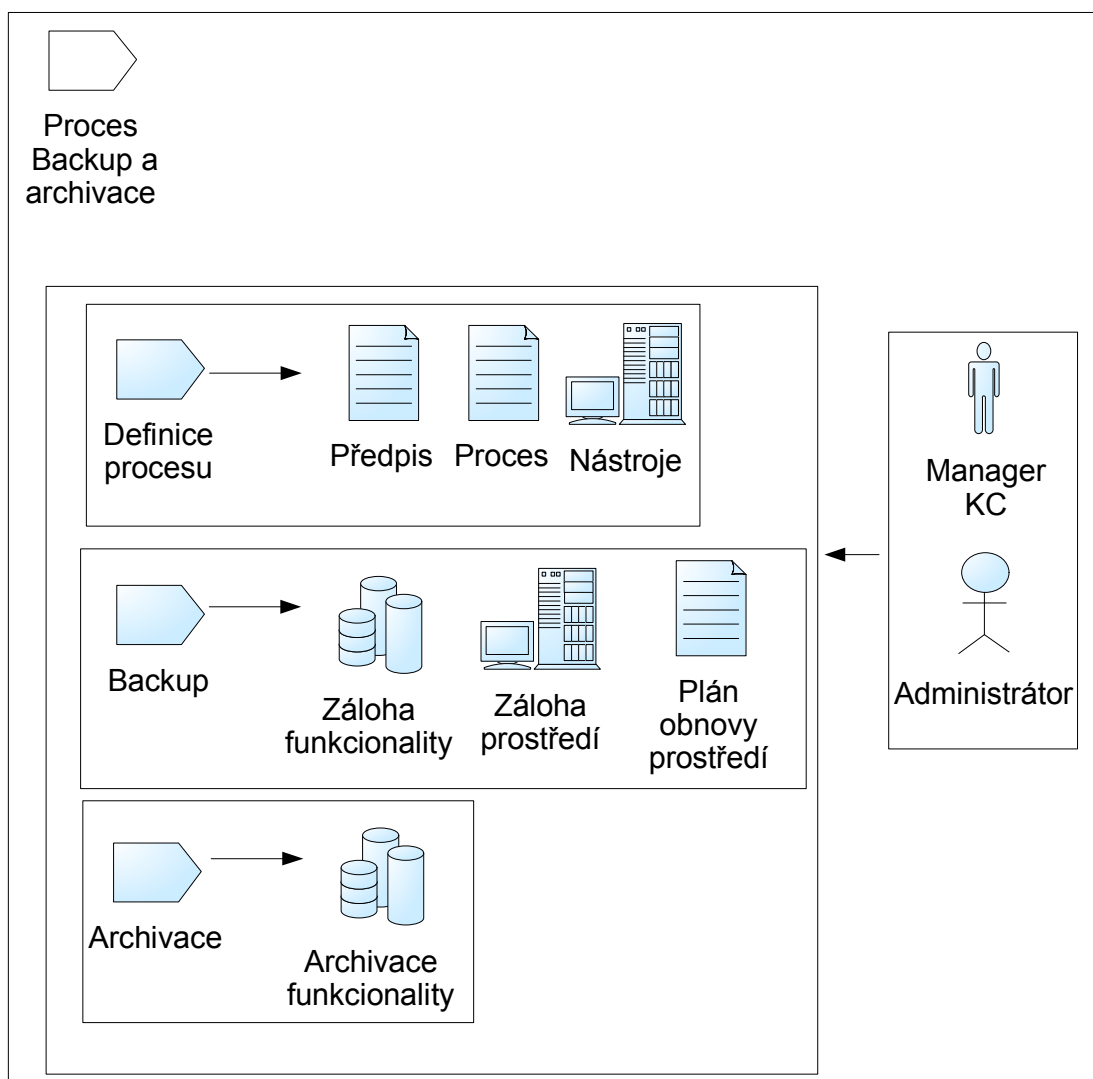
Obrázek 69: Řízení konfigurací

## 11.5. Backup a archivace dle metodiky RUP

Procesy spjaté se zálohováním a archivací nejsou řešeny metodikou Rational Unified Process.

## 11.6. Návrh procesu backup a archivace pro KC

Nedostatkem metodiky RUP je absence procesů spjatých se zálohováním a archivací. Cílem tohoto procesu je záloha klíčových dat v pravidelných intervalech v rámci KC. Nedílnou součástí procesu je také, jakým způsobem data evidovat, uchovávat a obnovovat.



**Obrázek 70: Backup a archivace**

Detailní návrh procesů nepovažuje autor této práce za nezbytný neboť je poplatný nástrojům archivace. Standardních nástrojů, podporujících backup a archivaci je na dnešním trhu celá řada.

Pravidelné zálohy vytváří administrátor.

## 11.7. Řízení kvality dle metodiky RUP

Průběžné ověřování kvality je jednou ze šesti dobrých praktik<sup>1</sup>, na kterých je postavena metodika RUP. Tato filosofie je implementována do každé disciplíny formou indikátorů a kontrolních milníků. Metodika tedy nedefinuje proces, aktivity, výstupy a role, tak jak tomu je u ostatních procesu softwarového vývoje.

Absence tohoto procesu významně snižuje použitelnost metodiky z pohledu quality assurance.

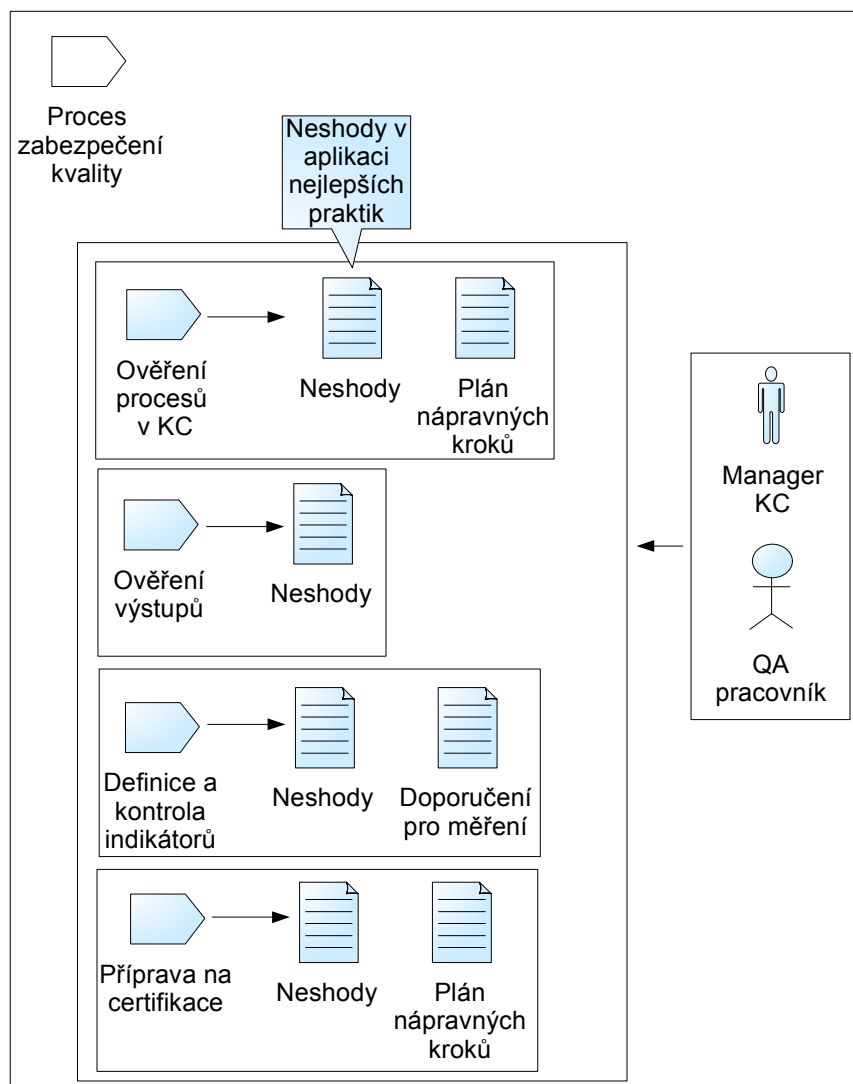
---

<sup>1</sup> Šest dobrých praktik dle metodiky RUP [8]:

- Iterativní vývoj.
- Správa požadavků.
- Komponentová architektura.
- Vizuelní modelování.
- Řízení kvality.
- Řízení změn.



## 11.8. Návrh procesu řízení kvality pro KC



Obrázek 71: Řízení kvality v KC

Nedostatkem metodiky RUP je absence přesně definovaných procesů spjatých se zabezpečením kvality. Cílem řízení kvality je formální i věcné ověření následujících náležitostí:

### **11.8.1. Ověření procesů z pohledu kvality**

Tato kontrola ověřuje shodu v použití jednotlivých disciplín v kompetenčním centru oproti jejich předpisu. Dále prověřuje úpravu jednotlivých kroků a výstupů z pohledu kvality. Cílem je identifikace a napravení neshod.

### **11.8.2. Ověření výstupů z pohledu kvality**

Cílem tohoto ověření je kontrola úplnosti výstupů, jejich formální stránky (layout, logo, šablona atd.) a samozřejmě i věcné (obsah, srozumitelnost, dodržení standardu disciplíny atd.) Výsledkem je identifikace případné nekvality výstupu a její náprava.

### **11.8.3. Definice a kontrola použití indikátorů**

V rámci revize je posouzena vhodnost a úplnost použitých indikátorů. Cílem je doporučit takové indikátory, které umožňují měřit kvalitu postupu, výstupu a použitého procesu (viz. následující kapitola).

### **11.8.4. Indikátory**

Pro potřeby kompetenčního centra je vhodné využít následující indikátory (definované a ověřené autorem této práce)<sup>1</sup>:

#### **11.8.4.1. Interní indikátory**

Interní indikátory slouží k měření kvality poskytování služeb a fungování KC jako celku. Výstup není předáván zákazníkovi a primárně slouží k ověření kvality, případně indikuje místa, které je třeba analyzovat a zlepšit:

- Plánování (čas) – poměr plánovaného času na opravu ku skutečně strávenému.

---

<sup>1</sup> Interní a externí indikátory definované v této kapitole byly použity při pilotním ověření návrhu metodiky na vybraných provozních projektech. Hodnoty uvedené v grafech jsou modifikovány z důvodu publikace v této práci.

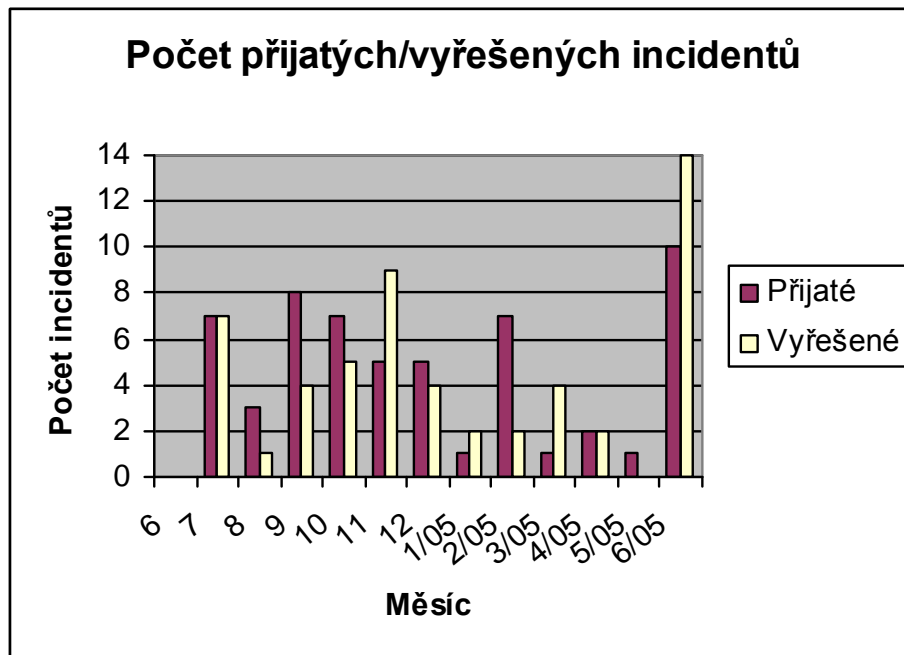
- Plánování (rozpočet) – poměr plánovaného rozpočtu na opravu ku skutečně strávenému.
- Plánování (termín) – dodržení plánu (ano/ne).
- Plánování (marže) – dodržení ziskovosti KC (ano/ne).

#### **11.8.4.2. Externí indikátory**

Externí indikátory slouží pro měření kvality poskytnutých služeb přímo zákazníkovi. Výsledky měření jsou zákazníkovi předávány v pravidelných intervalech a odsouhlasovány. Na základě těchto indikátorů jsou v praxi definována SLA. Níže uvedené indikátory je vhodné vidět ze dvou úhlů pohledu – jednorázově, kdy je zřejmé, jakým způsobem funguje KC diskrétně, ale také za nějakou jednotku, ze které lze vyčíst kvalitu poskytování služeb spojitě.

#### **11.8.4.3. Příklad vhodných indikátorů pro KC**

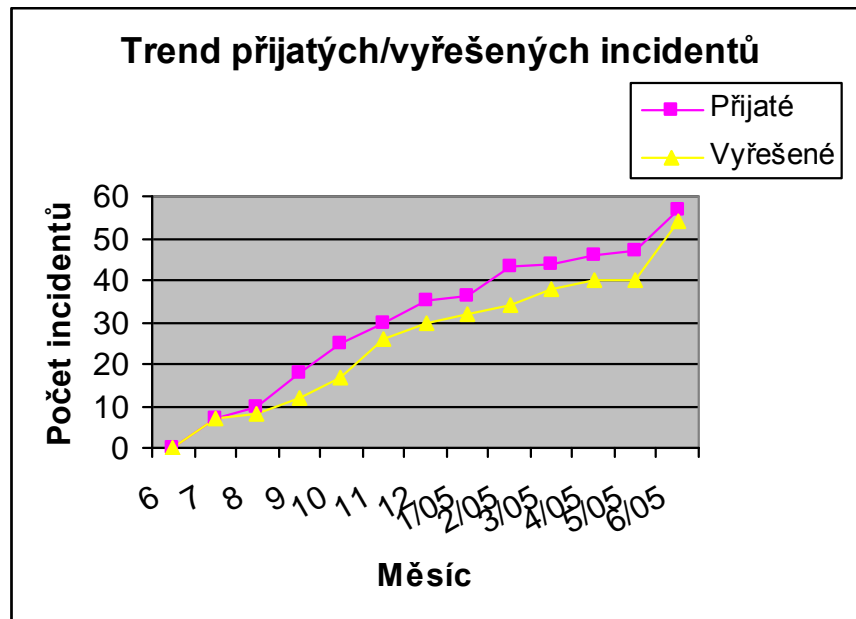
- **Počet přijatých, vyřešených chyb po jednotlivých měsících**
  - o Indikátor zachycuje poměr mezi počtem nahlášených a počtem vyřešených chyb za měsíc. Cílem (z pohledu KC) je vyřešit za měsíc co nejvíce chyb. Tento indikátor může být rozšířen o pohledy přes priority, respektive dopady do procesů zákazníka (dobrou praktikou v rámci IT odvětví jsou kategorie A, B, C – chyba typu A má za následek absolutní nefunkčnost aplikace, je kritická a business procesy zákazníka není možné dále vykonávat. Chyba typu B je středně kritická, způsobuje částečnou nefunkčnost aplikace a business procesy zákazníka je možné vykonávat s nějakým omezením. Chyba typu C je nekritická, nemá vliv na kvalitu business procesů. Chyby typu C jsou typicky náměty na změnu ergonomie řešení).



**Obrázek 72: Počet přijatých, vyřešených incidentů měsíčně na jeden produkt (zdroj: KC podpory bankovních IS)**

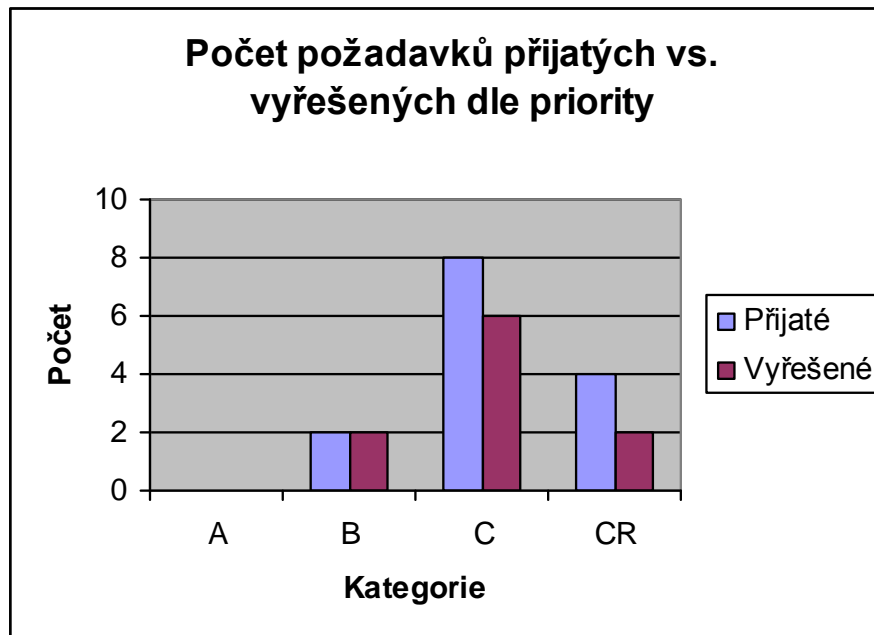
- **Trend přijatých, vyřešených incidentů po jednotlivých měsících**

- o Indikátor zachycuje trend počtu přijatých a řešených incidentů za nějakou periodu. Cílem je, aby trend byl v čase klesající – neboť tak deklaruje zvýšení kvality dodané aplikace i jejích záplat (v čase je identifikováno stále méně chyb). Je-li trend právě opačný, s velkou pravděpodobností mohou být příčinou následující oblasti - zdrojový kód aplikace je „nekvalitní“, proces release managementu vykazuje nedostatky, proces testování je podceněn, procesy designu a jeho ověření kvality jsou nefunkční.



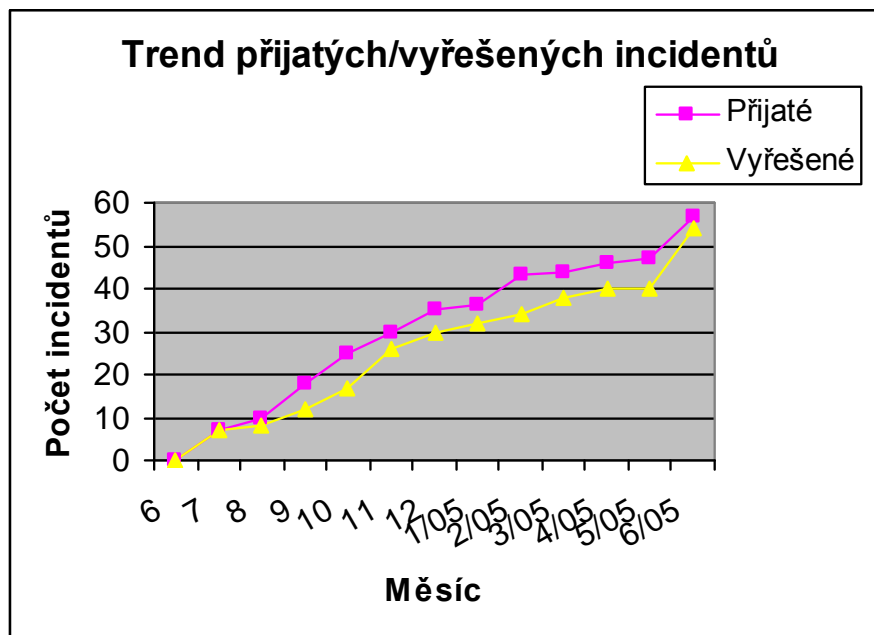
**Obrázek 73: Trend přijatých, řešených chyb  
(zdroj. KC podpory bankovních IS)**

- **Rozložení požadavků zákazníka**
  - o Indikátor ukazuje, jaký je poměr mezi jednotlivými typy hlášených incidentů – kolik z nich je typu A, B, C a kolik požadavků na změnu. Snahou KC je udržet rozvrstvení takovým způsobem, že chyby typu A se nevyskytují vůbec, B omezeně (jednotky za měsíc), C omezeně. Rozvrstvení podléhá kvalitě dodávky pro zákazníka. Požadavků na změnu samozřejmě neomezeně (neboť jsou zdrojem další, plánované práce a přínosů pro pracovníky KC). Na druhou stranu velký poměr požadavků na změnu může za určitých okolností indikovat nekvalitně odvedené procesy spjaté se sběrem požadavků uživatelů a jejich analýzou.



**Obrázek 74: Počet hlášených/vyřešených požadavků  
(zdroj: KC podpory bankovních IS)**

- **Spokojenost zákazníka**
  - Indikátor slouží k měření spokojenosti zákazníka s výsledkem dodaného řešení. Je-li dodána nová funkcionality nebo oprava chyby, která funguje v souladu s očekáváním zákazníka a navíc nezavleče do řešení další chybu, je výsledek označen jako YES. Když toto neplatí, je označen jako NO. Hodnota indikuje spokojenost zákazníka s kvalitou/ nekvalitou dodaných výstupů.
- **Trend Spokojenosti zákazníka po jednotlivých měsících**
  - Tento indikátor zachycuje kumulované hodnoty přijatých a vyřešených incidentů za nějaké období. Cílem je maximálně snížit počet přijatých incidentů za měsíc a naopak maximálně zvýšit počet vyřešených incidentů při udržení spokojenosti zákazníka na maximální hodnotě.



**Obrázek 75: Trend spokojenosti zákazníka  
(zdroj: KC podpory bankovních IS)**

#### 11.8.4.4. Hlídaná místa na procesech s důrazem na kvalitu

Jelikož kompetenční centrum zdědilo neduhy jednotlivých projektů, při designu a nasazení jednotlivých disciplín autor vytypoval klíčová místa procesů, které je potřeba nad rámec standardního přístupu ke kvalitě dohledovat:

Disciplína	Dohledované místo	Poznámka
Testování (Ve fázi analýza požadavků)	Vznik písemně zachycených testovacích scénářů	Absence testovacích scénářů neumožňuje funkcionalitu testovat. Nemožnost regresních testů v budoucnu
Testování	Písemné zachycení výsledku testů	Není dokladováno, že

		testy vůbec proběhly a s jakým výsledkem
Testování	Výsledky testů jsou součástí release notes	Dokladování zákazníkovi, že dodaný výstup byl testován
Uživatelské akceptační testy	Součástí verze předané zákazníkovi jsou release notes	Identifikace dodané funkcionality, jejího testování a seznam známých chyb
Uživatelské akceptační testy	Odpovídající části dokumentace a modelů jsou aktualizovány	Podstatný podklad pro budoucí provoz a rozvoj aplikace/IS

Dohledem těchto míst dojde ke zvýšení kvality dodávek. Pro provozní projekty KC, které jsou provozovány v čase, absence dokumentace a testovacích scénářů v krajní situaci znamená nemožnost aplikaci provozovat, udržovat a rozvíjet, případně testovat změny. Zpětné vyrobění dokumentace má za následek dopad do rozpočtové efektivity, zdrojů a času jednotlivých projektů.

### 11.8.5. Doporučení pro implementaci procesů

Na základě vlastní zkušenosti autor vidí dvě možné varianty implementace navržených procesů:

- Metoda „velkého třesku.“
- Postupné nasazení po jednotlivých disciplínách.



#### **11.8.5.1. Metoda velkého třesku**

Tato metoda spočívá v implementaci všech disciplín najednou. Od určitého data vznikne kompetenční centrum, jsou do něj převedeny jednotlivé projekty a každý projekt funguje dle dohodnutých principů (dle návrhu jednotlivých procesů). Před milníkem spuštění předchází práce, která se zabývá překlápěním jednoho stylu fungování na druhý, mapováním výstupů, identifikací kritických oblastí atd.

Na základě své zkušenosti, autor tuto cestu nedoporučuje. Pokud by ji přeci jen nějaký subjekt chtěl následovat, necht' velký třesk použije na jeden pilotní projekt, na kterém odzkouší fungování a identifikuje rizika přechodu. Touto cestou maximálně ohrozí fungování podpory jednoho projektu, ale zásadně neohrozí fungování všech zbylých projektů.

Autor velký třesk doporučuje pouze pro případ nových projektů, které za sebou dosud nemají historii.

#### **11.8.5.2. Metoda postupného nasazování jednotlivých disciplín**

Tato metoda je bezpečnější variantou přechodu. Ve své podstatě znamená identifikaci slabých míst jednotlivých projektů, jejich prioritizace a navržení způsobu odstranění. Pořadí nasazování disciplín je určeno na základě priorit. V prvním kroku je vždy ustanoveno kompetenční centrum a aplikovány principy programme managementu a projektového řízení. Následně jsou postupně nasazovány jednotlivé disciplíny. Je-li disciplína ukotvena, vstřebána a funkční, je nasazena další.

Při implementaci KC, je první vhodnou disciplínou nasazení testování. Po jejím ukotvení je jako druhá, nejvíce vhodná disciplína release managementu a následně procesy QA. Výběr disciplín reflektuje v první řadě nekvalitní výstupy z testování a v druhé řadě nekvalitu jednotlivých verzí.

## 12. Přehled navržených procesů

Tato kapitola shrnuje seznam navržených procesů kompetenčního centra autorem této práce. Dále zachycuje, pro které procesy byla použita metodika Rational Unified Process jako vstupní informace.

<b>Skupina procesů / Proces</b>	<b>Zdrojová metodika</b>	<b>Je součástí KC</b>	<b>Pozn.</b>
<b>Řídící procesy</b>	Není	Ano	Není: Zkoumané metodiky nedoporučují, jak daný proces navrhnout a jak jej implementovat, případně jej vůbec nepokrývají.
Řízení problémů (incidentů)	Není	Ano	
Utilizace zdrojů	Není	Ano	
Řízení očekávání zákazníka	Není	Ano	
Ověřování kvality	Není	Ano	
Reportování	Není	Ano	
Účtování	Není	Ano	
<b>Realizační procesy</b>			
Proces poskytování podpory	Není	Ano	
Proces poskytování podpory první úrovně	Není	Ano	
Proces poskytování podpory druhé	Není	Ano	

úroveň			
Životní cyklus požadavku	Není	Ano	
Analýza	RUP	Ano	RUP: Proces byl přeprojektován na základě zjednodušení vstupů z této metodiky a je s ní kompatibilní. U vybraných procesů nebyl vstup z RUP k dispozici anebo bylo vhodnější jej nepoužít.
Design	RUP	Ano	
Implementace	RUP	Ano	
Testování	RUP	Ano	
Nasazení	RUP	Ano	
<b>Podpůrné procesy</b>			
Příprava a správa prostředí	RUP	Ano	
Řízení konfigurací	RUP	Ano	
Backup a archivace	Není	Ano	
<b>QA procesy</b>			
Proces řízení kvality	Není	Ano	
Definice interních indikátorů	Není	Ano	
Definice externích indikátorů	Není	Ano	

## **13. Využití konceptu kompetenčního centra mimo projekty ICT**

Návrh kompetenčního centra pro řízení ICT projektů pokrývá procesy uvedené v kapitole 12 – Přehled pokrytých procesů, přičemž respektuje cíl této práce. Navržené procesy lze libovolně modifikovat pro jiný typ projektů, než-li pouze pro projekty IT charakteru (např. z oblasti stavebnictví, zemědělství, logistiky atd.). Tyto projekty se vyznačují primárně jinou cílovou náplní, nicméně jejich způsob realizace pomocí prostředků ICT může často nastat.

V případě modifikace procesů pro projekty mimo oblast IT, konstituční a řídicí procesy kompetenčního centra zůstanou stejné. Realizační, podpůrné procesy a částečně i proces kvality je třeba přizpůsobit cíli a náplni centra.

Při redesignu procesů autor doporučuje dodržet analogický postup této práce (identifikace rozdílů mezi procesy KC a metodikou KC). Strukturu procesů, jejich výstupy a použitou notaci lze znovu použít.

## 14. Závěr

Cílem této disertační práce je navrhnout metodiku, která podpoří realizaci několika vzájemně provázaných ICT provozních projektů. Na základě tzv. dobrých praktik v odvětví, principů projektového a programového řízení, vlastních znalostí a zkušeností, vytvořil autor návrh metodiky kompetenčního centra ICT provozních projektů jako virtuálního útvaru společnosti, které má za cíl dodat takový typ projektů.

Jelikož problematiku IT projektů a řízení několika vzájemně provázaných projektů dohromady řeší světoznámé metodiky [ITIL, COBIT, PMM, RUP, COR], které jsou uznávané v IT odvětví, autor zvolil cestu definice kritérií pro vhodnost použití vybraných metodik a to jak z úhlu pohledu vytváření kompetenčního centra, tak pro realizaci jednotlivých projektových disciplín. Analýzou uvedených metodik bylo zjištěno, že žádná z nich vyčerpávajícím způsobem nepokrývá ani vytváření a realizaci kompetenčního centra, ani specifické úpravy disciplín vývoje a dodávky software pro potřeby provozních projektů.

Není velký rozdíl mezi chápáním programme managementu klíčovými metodikami IT, rozdíl je v akcentu na klíčové části programme managementu, jak programme management implementovat a doporučení oblastí vhodných pro outsourcing. Autor ukazuje, jak použít principy programme managementu pro vytvoření kompetenčního centra.

Výsledkem analýzy je zjištění, že žádná z uvedených metodik nepokrývá ani procesy spjaté s kompetenčním centrem ani s jeho obchodními procesy. Autor této práce formulovat následující cíle:

- Na základě analýzy vhodnosti použití jednotlivých metodik definovat provozní procesy (řídící, realizační, podpůrné a QA procesy) s použitím vstupů z vybrané metodiky.
- Navržení procesu vytváření a provozu kompetenčního centra na základě principu programme managementu.

- Definice indikátorů vhodných pro sběr a vyhodnocení v rámci kompetenčního centra.
- Definice oblastí, které je třeba po uvedení konceptu kompetenčního centra dohledovat, aby nebyla ohrožena kvalita dodávky projektů.
- Nastínění směru dalšího rozvoje disertační práce a jejich výsledků.

Autor disertační práce naplnil výše uvedené cíle následujícím způsobem:

- Analýzou vybraných metodik vyhodnotil metodiku CORTEX jako nejvhodnější pro vytváření a běh kompetenčního centra. Tento předpis upravil o vlastní design konkrétních procesů a aktivit, identifikaci jejich výstupů a autorský návrh doporučí, jakým způsobem vytvářet kompetenční centrum.

DISCIPLÍNA/METODIKA	CORTEX
<b><i>Programme management</i></b>	<b><i>Nutno upravit</i></b>
Definice programme managementu	Vhodná
Obecný process	Vhodná
Konkrétní procesy, aktivity	Nutno upravit
Výstupy	Nutno upravit
Doporučení pro implementaci (vytvoření KC)	Nutno upravit

- Analýzou vybraných metodik vyhodnotil metodiku RUP jako nejvhodnější vstup pro návrh jednotlivých procesů kompetenčního centra provozních projektů. Vybrané procesy metodiky autor přepracoval nebo kompletně nově navrhl. Autor také identifikoval a vyprojektoval procesy metodikami dosud nepokryté.

DISCIPLÍNA/METODIKA	RUP
<i>Programme management</i>	<i>Nevhodná</i>
<b>Provozní projekty</b>	<b>N/A</b>
<i>Management provozních projektů</i>	<i>Nevhodná</i>
<i>Správa prostředí</i>	<i>Nutno upravit</i>
<i>Správa požadavků</i>	<i>Nutno upravit</i>
<i>Analýza a Design</i>	<i>Nutno upravit</i>
<i>Oprava chyb/Vývoj</i>	<i>Nutno upravit</i>
<i>Testování</i>	<i>Nutno upravit</i>
<i>Nasazení</i>	<i>Nutno upravit</i>

- Autor navrhl následující procesy KC:
  - o Řídící procesy.
    - § Řízení problémů (incidentů).
    - § Utilizace zdrojů.
    - § Řízení očekávání zákazníka.
    - § Ověřování kvality.
    - § Reportování.
    - § Účtování.
  - o Realizační procesy.
    - § Proces poskytování podpory.
    - § Životní cyklus požadavku.
    - § Proces poskytování podpory první úrovně.
    - § Proces poskytování podpory druhé úrovně.
    - § Analýza a Design.
    - § Implementace.
    - § Testování.
    - § Nasazení.
  - o Podpůrné procesy.

- § Příprava a správa prostředí.
  - § Řízení konfigurací.
  - § Backup a archivace.
- QA procesy.
  - § Proces řízení kvality.
  - § Definice interních metrik.
  - § Definice externích metrik.
- Identifikace míst, které je třeba dohledovat při přechodu od jednotlivých provozních projektů po jejich dodávku kompetenčním centrem (autorův návrh). Zavedený proces na sběr a vyhodnocování metrik, stejně tak jako zpřísnění procesu testování, podstatně zlepšují kvalitu a splnitelnost termínu dodávek.
- Autorův návrh indikátorů pro vyhodnocení kvality služeb poskytovaných KC
  - Počet přijatých, vyřešených incidentů po jednotlivých měsících.
  - Trend přijatých, vyřešených incidentů po jednotlivých měsících.
  - Rozložení požadavků zákazníka.
  - Spokojenost zákazníka.
  - Trend spokojenosti zákazníka po jednotlivých měsících.
- Autorův návrh na další rozvoj práce:
  - Návrh a modifikace procesů podporujících globalizaci konceptu kompetenčního centra (sdílení zdrojů po celém světě).
  - Identifikace oblastí, které je vhodné v rámci KC outsourcovat a zlepšit tak kvalitu a efektivitu poskytovaných služeb.
  - Modifikace procesů s cílem jejich univerzalizaci pro libovolné kompetenční centrum.
  - Vytvoření konceptu, metodiky s cílem zabezpečit shodnost následujících prostředí - testovací prostředí KC, testovací prostředí klienta, produkční prostředí klienta a záložní prostředí klienta.



- Motivace k maximálnímu rozvoji, coaching podřízených – nové manažerské styly 21.století.
- Profesní rozvoj jednotlivce v týmu, sociologie týmu.
- Konkurence nebo subdodavatel? Jak subdodavatelské aliance ovlivní dodávání ICT služeb a projektů.
- Realizace studie proveditelnosti – kdy ji realizovat, výstup.
- Testovací strategie bankovních aplikací v ČR – problematika manuálního a automatizovaného testování.

Výsledkem této práce je také zjištění, že idea kompetenčního centra může být použita na dodávku libovolných, spjatých projektů a to i mimo oblast IT. Pro takový případ konstituční procesy a procesy řízení zůstanou stejné. Rozdíl je v realizační, podpůrných a částečně procesech kvality, které musí být přizpůsobeny náplni centra.

Autor spatřuje využitelnost této práce v následujících oblastech:

<b>Subjekt</b>	<b>Využitelnost</b>
Společnosti realizující supportní IS projekty	<ul style="list-style-type: none"> <li>- Zlepšení realizace pomocí implementace disciplín</li> <li>- Implementace procesu QA</li> <li>- Implementace metrik</li> <li>- Ustavení kompetenčního centra</li> </ul>
Kdokoli	<ul style="list-style-type: none"> <li>- Vysvětlení ideje kompetenčního centra</li> <li>- Vysvětlení procesů konstituce a fungování kompetenčního centra</li> </ul>
Všichni uživatelé ICT	<ul style="list-style-type: none"> <li>- Ustavení kompetenčního centra pro realizaci projektů</li> </ul>
Zákaznická strana	<ul style="list-style-type: none"> <li>- Využití konceptu jako závazné politiky pro dodavatele IT služeb</li> </ul>
Kdokoli	<ul style="list-style-type: none"> <li>- Přetvoření konceptu kompetenčního centra a IS provozních procesů na komerční metodiku</li> </ul>
IBM Rational	<ul style="list-style-type: none"> <li>- Rozšíření metodiky RUP o procesy spjaté s KC a realizací provozních projektů</li> </ul>

Autor této práce se domnívá, že návrh metodiky kompetenčního centra lze využít převážně pro realizaci provozních projektů v bankovních ústavech. Pokud budou vybrané procesy modifikovány, lze je takto využít i mimo bankovní sféru.

Procesy kompetenčního centra lze také využít jako vstup pro rozšíření standardů a metodik realizace a řízení ICT projektů, případně z nich vytvořit samostatnou metodiku.

## 15. Zkratky a vysvětlivky

Zkratka/Pojem	Vysvětlení
Aktivita	Aktivita představuje elementární stavební prvek procesu. Je často popsána jednotlivými kroky, rolí, která ji vykonává, vstupy a výstupy, dobou trvání, náklady, prioritou a vstupní podmínkou realizace. Řetěžením aktivit dohromady vzniká proces.
Akceptační testy	Cílem akceptačních testů je ověření dodávaného řešení oproti požadavkům zákazníkem. Akceptace softwaru a jeho komponent ve vztahu ke kompetenčnímu centru je různá – v jednoduchém případě stačí přetestovat ohraničený kus funkcionality. V případě velkých zásahů do aplikace je třeba stanovit plán testů a přetestovat klíčové scénáře v aplikaci, nebo otestovat aplikaci celou.
Analýza dopadu	Analýza dopadu je prvním přirozeným krokem po přijetí a registraci incidentu. Přidělený analytik zjišťuje důležitost (prioritu, severitu) a podstatu incidentu, jeho dopady do produkce.
Baseline	Baseline označuje použitou verzi několika komponent v jedné verzi řešení. Je to referenční hranice, na kterou se odkazují další verze komponent.
BU	Business Unit, obchodní jednotka.
CMMi	Capability Maturity Model Integrated, model pro měření a ohodnocení kvality procesů.
COBIT	Control Objectives for Information and Related Technology, metodika pro audit ICT.
CVS	Control Version System, nástroj pro řízení verzování.
Datový model	Datový model slouží pro uložení informací o reálných objektech. Zachycuje objekty v podobě tříd, jejich vlastnosti (atributy), metody a vzájemné vztahy.

Design procesu	Cílem této aktivity je zachytit posloupnost jednotlivých aktivit, rolí, návazností, vstupů a výstupů v rámci procesu. Při redesignu procesu dochází k revizi a optimalizaci jednotlivých složek procesu.
FTP	File Transfer Protocol, protokol pro přenos souborů.
Funkční specifikace	Funkční specifikace zahrnují slovní popis a schémata funkčních i nefunkčních požadavků zadavatele na ICT řešení. Struktura funkční specifikace může být různá dle komplexity řešeného problému viz. [RUP].
Helpdesk	Helpdesk, Service desk je centralizované, unikátní místo, na kterém se schází všechny požadavky a problémy uživatelů. Výhodou centralizace je to, že operátor má přehled o všech požadavcích, problémech a jejich řešení.
ICT	Information and Communication Technology, informační a komunikační technologie.
IDE	Integrated Development Environment, nástroj pro vývoj IS.
Implementace procesu	Cílem implementace procesu je realizovat design procesu – transformovat navržený proces v nějakém nástroji v život. Výsledkem implementace je fungující proces v praxi. V některém případě mohou být součástí implementace vstupy pro další případnou optimalizaci procesu.
IS/IT	Information System/Information Technology, informační systémy, informační technologie.
Iterativní vývoj	Iterativní přístup pomocí postupných kroků, které se několikrát dokola během životního cyklu tvorby software opakují, vedou postupně, v iteracích, k vytvoření celkového ICT řešení. Význam iterativního přístupu spočívá ve schopnosti včasného odhalení rizik v každém stádiu projektu. Každá iterace končí spustitelnou verzí (buildem), přičemž časté ověřování stavu projektu zajišťuje, že projekt probíhá v daném časovém rozvrhu.

ITIL	Information Technology Infrastructure Library, metodika provozu.
KPA's	Key Process Area, klíčový proces.
KC	Kompetenční centrum.
Kompetenční centrum	Pod pojmem kompetenční centrum lze chápat dedikovanou logickou divizi (nikoli organizační), která má svůj cíl, rozpočet, zdroje a metodiku fungování. Na konceptuální úrovni, jsou si kompetenční centra podobná. Rozdíl je v konkrétní části metodiky – realizační procesy.
Kompetenční centrum IS projektů	Pod pojmem kompetenční centrum IS projektů lze chápat dedikovanou logickou divizi (nikoli organizační), která má svůj cíl, rozpočet, zdroje a metodiku fungování. Cílem tohoto centra je efektivně realizovat provoz dodaných ICT řešení u různých zákazníků (kvalita, termín, rozpočet).
LCMG	LogicaCMG, s.r.o.
Metamodel metodiky	Metamodel zachycuje model modelu a jazyk, kterým je vyjádřen. V kontextu této práce metamodel metodiky zachycuje strukturu modelu metodiky, její objekty a vzájemnou provázanost těchto objektů.
Metodika	Metodika zachycuje nejlepší praktiky v daném odvětví. Ty lze dekomponovat na přístupy a doporučení, etapy, procesy, činnosti, pravidla, role, výstupy a šablony výstupů. Metodika definuje role, načasování aktivit, způsob realizace aktivit, vstupy a výstupy.
Plán návratu	Cílem plánu návratu je stanovit takové kroky, zodpovědné role, načasování a rozhodovací kritéria, které usnadní rozhodování v případě problému, zda problém řešit nebo vrátit prostředí do stavu před uvedením změny. Kroky, kterými je prostředí navraceno zpět, jsou také součástí plánu návratu.
Programme management	Programme management je portfolio projektů a aktivit, které jsou koordinovány a řízeny dohromady takovým způsobem, aby

	dosahovali svých cílů a přínosů jak jednotlivě, tak i dohromady a to v podobě synergického efektu na strategické úrovni.
PMM	Project Management Metodology, Metodika řízení projektů.
Podpůrný proces	Podpůrné procesy jsou takové procesy, které umožňují realizaci realizačním procesům. V případě kompetenčního centra IS projektů je realizačním procesem návrh řešení, ale podpůrným je například archivace, správa prostředí atd.
Popis instalace	Množina kroků, které vedou k nainstalování a základní konfiguraci dodaného ICT řešení.
Popis rozhraní	Formalizovaný popis, model, rozhraní na jednotlivé ICT řešení.
Proces	<p>Proces je definován jako množina navazujících činností s jasně stanoveným cílem. Je aktivován na základě definovaných podnětů, přičemž v jeho rámci jsou transformovány vstupy na výstupy.</p> <p>Proces je opakovatelná strukturovaná transformace vstupů na výstupy za účelem dosažení společného cíle. Proces může být rozložen na jednodušší procesy (podprocesy) nebo činnosti. Proces má vždy jasně určený podnět, který jej spouští, začátek a konec.</p> <p>Procesní popis by měl popisovat všechny atributy podle této definice, tj. cíl procesu, vstupy, výstupy, činnosti provádějící dílčí transformační kroky a jejich posloupnost, podněty spouštějící proces a role odpovědné za vykonání dílčích činností. Podle potřeby je možné doplnit další kvalitativní atributy procesu, např. požadovanou mezní dobu pro dokončení činnosti či čekání na následující činnost, popis položek vstupního formuláře, zdroj informací, očekávané znalosti a zkušenosti zapojených rolí apod.</p>
Projekt	Dle normy ISO 10006 „Směrnice jakosti v managementu projektu“ je projekt jedinečný proces sestávající z řady koordinovaných a řízených činností s daty zahájení a ukončení, prováděný pro dosažení předem stanoveného cíle, který vyhovuje specifickým

	požadavkům, včetně omezení.
Provozní projekt	<p>Provozní projekt je řízený proces s cílem rutinně provozovat, opravovat chyby a provádět upgrade verze ICT řešení.</p> <p>Klíčovými charakteristikami provozního projektu jsou:</p> <ul style="list-style-type: none"> <li>• Přesné cíle – provoz, rozvoj ICT řešení.</li> <li>• Dané termíny – milníky, SLA.</li> <li>• Daný paušál za provoz ICT řešení.</li> <li>• Pohyblivá cena za rozvoj ICT řešení.</li> </ul>
Řídící proces	Řídící proces umožňuje dohledovat provoz nebo činnost. Typicky tuto činnost realizuje manager.
RDBMS	<p>Relational Database Management System, Relační databázový systém.</p> <p>RDBMS uchovává data v databázi ve formě řádků a sloupců. Řádek odpovídá záznamu, sloupce odpovídají atributům.</p>
Realizační proces	Realizační proces umožňuje realizovat daný výstup v rámci projektu nebo KC. Je to produkční proces, pro které byly primárně projekt nebo KC ustanoveni.
Risk management	Risk management je klíčovou disciplínou řízení s cílem eliminovat v maximální míře nejvýznamnější rizika (rizika s největším dopadem). Risk managementu autor této práce věnoval podstatnou část své diplomové práce a proto na její odpovídající části tímto odkazuje [MAROUNEK 2002].
QA	Quality Assurance, řízení kvality.
Quality Assurance	Cílem QA aktivit je zabezpečit odpovídající kvalitu jednotlivých výstupů. K tomu se používají kritéria akceptace, milníky, review atd.
Role	Role zahrnuje soubor znalostí a kompetencí, které je třeba znát proto, aby nějaký člověk mohl danou roli vykonávat. Typicky funguje vazba mezi rolí a člověkem M:N.
RUP	Rational Unified Process, metodika vývoje software.

sFTP	Secure File Transfer Protocol, zabezpečený protokol pro přenos dat.
SI	System Integration, systémová integrace.
SLA	Service Level Agreement, dohoda o úrovni poskytování služeb.
SMIME	Secure / Multipurpose Internet Mail Extensions, zabezpečený protokol pro výměnu zpráv.
Specifikace funkčních testů	Specifikace funkčních testů zachycuje seznam testovacích scénářů, posloupnost jejich provádění, akceptační kritéria a načasování testů.
Technická specifikace	Technická specifikace zachycuje návrh řešení požadavků uživatele. Vstupem je funkční specifikace. Výstupem je sada modelů, jejich specifikace a doporučení pro implementaci.
Testovací data	Vybraný vzorek dat, na kterých bude testována funkcionální a výkon ICT řešení.
Testovací log	Výsledek testů – termín, testovací scénáře, tester, výsledek testu, hlášené chyby.
Testovací scénář	Testovací scénář zachycuje kroky, vstupní údaje, kontrolní údaje s cílem otestovat danou funkcionální.
UCM	Unified Change Management, metodika řízení změn a řízení konfigurací.



## 16. Seznam použitých zdrojů

### 16.1. Publikace

[ANDREWS 1987] ANDREWS, K.R. *The Concept of Corporate Strategy*. 3. vyd. McGraw-Hill: Irwin, 1987. 152 s. ISBN: 0256183295.

[AXELROD 1984] AXELROD, R. *The evolution of cooperation*. 1. vyd. Basic Books, 1984. 223 s. ISBN 0-465-0212-2.

[BARNATT 1996] BARNATT CH. *Management strategy and information technology*. 1. vyd. London: Thomson Business Press, 1996. 180 s. ISBN: 0-412-7495-05

[BEBR 1998] BÉBR, R. *Manažerské informační systémy I – Externí zdroje informací*. 1. vyd. Praha: VŠE, 1998. ISBN 80-7097-885-8.

[BEBR 2005] BÉBR, R., et al. *Informační systémy pro podporu manažerské práce*. 1. vyd. Praha: Professional Publishing, 2005. ISBN 80-86419-79-7.

[BOOCH 1999] BOOCH, I., et al. *The unified software development process : The complete guide to the Unified Process from the original designers*. 2.ed. Reading : Addison-Wesley Publ. Co., 1999. 463 p. ISBN 0-201-57169-2.

[CANTOR 1998] CANTOR, M. *Object-oriented project management with UML*. 1.ed. New York : Wiley, 1998. 386 p. ISBN 0-471-25303-0.

[COAD 1991] COAD, P., YOURDON, E. *Object Oriented Analysis*. 1.ed. Yourdon Press: Prentice-Hall, 1991. 233 s. ISBN 0-13-629981-4.

[DOLANSKY 1996] DOLANSKÝ, V., et al. *Projektový management*. Praha: Grada, 1996. 372s. ISBN 80-7169-287-5.

[DOUCEK 2004] DOUCEK, P. *Řízení projektů informačních systémů*. 2. vyd. Praha: Professional Publishing, 2006. 188 s. ISBN 80-86946-17-7.

[FEUERLICHT 2002] FEUERLICHT, G., VOŘÍŠEK, J. *Impact of the Service Model for Delivering Enterprise Applications*, Proceedings of "BITWorld 2002" conference, ESPOL, Guayaquil, Ecuador, 2002, ISBN 0905304403.

[GALA 2006] GÁLA, L., et al. *Podniková informatika*. 1.vyd. Praha: Grada Publishing 2006. 482 s. ISBN 80-247-1278-4.

[GULDENTOPS 2000] GULDENTOPS, E., et al. *COBIT: IT Governance 3ed*. 3.ed. IT Governance institute, 2000. 156 p. ISBN 1-893209-17-2.

[HAREN 2005] HAREN, V. *ITSMF – Best practise: Foundations of IT Service Management based on ITIL*. 1.ed. Netherland, 2005. 234p. ISBN 9077212582.

[KOONTZ 1993] Koontz, H., Wehrich, H. *Management*. 5.vyd. Praha: Victoria Publishing, 1993. 530 s. ISBN 80-85605-45-7.

[LINHART 2002] LINHART, J. *Slovník cizích slov pro nové století*. 1.vyd. Litvínov: Dialog, 2002. 412 s. ISBN 80-85843-61-7.

[MAROUNEK 2002] MAROUNEK, P. *Internet a konkurenceschopnost: Využití internetu v podnikových informačních systémech (diplomová práce)*. KIT, ČZU, 2002. 69 s.

[MAROUNEK 2004/1] MAROUNEK, P. *Je interní IT útvar úzkým hrdlem velkých bank?*. Praha, 2004. In: Systémová integrace – sborník 2004. ISSN 1210-9479.

[MAROUNEK 2004/2] MAROUNEK, P. *Aplikace Programme Managementu v praxi*. Praha 07.12.2004. In: ROSICKÝ, Antonín, MILDEOVÁ, Stanislava (ed.). Systémové přístupy 2004. Praha : Oeconomica, 2004, s. 185–191. ISBN 80-245-0828-1.

[MAROUNEK 2005] MAROUNEK, P. *Praktická aplikace Programme Managementu*. Praha 24.2.2005. In: HRONOVÁ, Stanislava (ed.). Sborník prací účastníků vědeckého semináře doktorského studia 2005. Praha : Oeconomica, 2005, s. 87–98. ISBN 80-245-0885-0.

[MAROUNEK 2006] MAROUNEK, P. *Úvod do Rational Unified Process (RUP) z perspektivy řízení projektů*. Praha 31.5.2006. In: HOSPES, Jan, a kol. (ed.). ICTM 2006 Řízení informačních a komunikačních technologií. Praha : VUT, 2006, s. 122-128. ISBN 80-01-03498-4.

- [POL 1998] POL, M. *Structured testing of information system*. 2.ed. Kluwer: Deventer, 1998. 104 s. ISBN 90-267-2910-3.
- [PORTER 1986] PORTER, M. *Cases in Competitive Strategy*. 2.ed. London : Collier Macmillan, 1983. 541 p. ISBN 0-02-925410-8.
- [POUR 1996] POUR, J. *Aplikační software*. 1.vyd. Praha: VŠE, 1996 (Skriptum). 217 s. ISBN 80-707-9943-9.
- [POUR 1997] Pour, J. et al. *Architektury informačních systémů v průmyslových a obchodních podnicích*. 1.vyd. Praha: Ekopress, 1997. 301 s. ISBN 80-861-1902-5.
- [ROSENAU 2000] Rosenau, M.D. *Řízení projektů : příprava a plánování, zahájení, výběr lidí a jejich řízení, kontrola a změny, vyhodnocení a ukončení*. 1.vyd. Praha: Computer Press, 2000. 344 s. ISBN 80-7226-218-1
- [REPA 1999] Řepa, V. *Analýza a návrh informačních systémů*. 1.vyd. Praha: Ekopress, 1999. 403 s. ISBN 80-86119-13-0.
- [ROYCE 1998] ROYCE, W. *Software Project Management*. 1.ed. Reading: Addison-Wesley Publ. Co., 1998. 448 p. ISBN 0-201-30958-0.
- [SCHEER 1999] Scheer, A.W. *ARIS: od podnikových procesů k aplikačním systémům*. 1.vyd. Brno: Comsoft ČR, 1999. 275 s. ISSN 80-238-4719-8.
- [TIETZE 1992] Tietze, P. *Strukturální analýza : Úvod do projektu řízení*. 1.vyd. Praha: Grada, 1992. 224 s. ISBN 80-854-2445-2.
- [UČEŇ 2001] UČEŇ, P., a kol. *Metriky v informatice : Jak objektivně zjistit přínosy IS*. 1.vyd. Praha: Grada Publishing, spol. s r.o., 2001. 140s. ISBN 80-247-0080-8.
- [VORISEK 1996] Voříšek, J. *Informační technologie a systémová integrace*. 1.vyd. Praha: VŠE, 1996. 198 s. ISBN 80-707-9895-5.

[VORISEK 1997] Voříšek, J. *Strategické řízení informačního systému a systémová integrace*. 1.vyd. Praha: Management Press, 1997. 323 s. 80-859-4340-9.

[VORISEK 2000] Voříšek, J. *Nová dimenze systémové integrace : integrace podnikových procesů a znalostí*, Praha 2000. In: Systémová integrace '2000. Sborník mezinárodní konference 2000. Praha : VŠE, 2000, s. 195-206. ISBN 80-245-0041-8.

[VORISEK 2001/1] Voříšek, J. *Model "SPSR" : model řízení podnikové informatiky*. Demanovská Dolina 2001. In: Sborník mezinárodní konference Systémová integrácia 2001. Demanovská Dolina: TU Žilina, 2001. s. 5-18. ISBN 8-7100-880-X.

[VORISEK 2001/2] Voříšek, J., Dunn, D. *Management of Business Informatics: Opportunities, Threats, Solutions*. Praha 2001. In: Proceedings of "Systems Integration 2001" conference 2001. Praha : VŠE, 2001. ISBN 80-245-0169-4.

[VORISEK 2003] Voříšek J., et al. *Aplikační služby IS/ICT formou ASP : Proč a jak pronajímat informačnické služby*. 1.vyd. Praha: Grada Publishing, 2003. 213 s. ISBN 80-247-0620-2.

[VRANA 2005] Vrana, I. Et al. *Zásady a postupy zavádění podnikových informačních systémů*. 1.vyd. Praha: Grada, 2005. 188 s. ISBN 80-247-1103-6.

## **16.2. Elektronické zdroje a Internet:**

[COR] CORTEX [CD ROM]. Version Baseline 20.1. London: LogicaCMG, plc., 13.1.2004.

[BARTLET 2003] BARTLET, J., et al. ITIL : The key to managing IT services: Service delivery [CD ROM]. Version 2.0. Crown: OGC, 2003.

[PMM] Project Management Metodology [CD ROM]. Verze 2003.1.7.2. Praha: Unicorn Holding a.s., 2003.

[RUP] Rational Unified Process [CD ROM]. Version 2003.06.00. Unknown: IBM and Rational Software, 2003.

[COBIT] COBIT [CD ROM]. Version 3.0. Unknown: IT Governance Institute, 2000.

[ITIL] ITIL: ICT Infrastructure Management [CD ROM]. Unknown. Norwich: OGC, 2002.

[PMBOK] PMBOK GUIDE 1.3. [CD ROM]. Version 1.3. USA: Project Management Institute, 2004.

## 17. Příloha č. 1 – Seznam obrázků

Obrázek 1: Model globální strategie [dle VORISEK 1997] .....	25
Obrázek 2: Oblasti řízení ICT [POUR 1996].....	27
Obrázek 3: Model SPSPR [VORISEK 2001] .....	28
Obrázek 4: Řepův životní cyklus IS projektů [REPA 1999] .....	31
Obrázek 5: Řízení projektu tunelovým způsobem [COAD 1991].....	32
Obrázek 6: Životní cyklus vodopád .....	33
Obrázek 7: Životní cyklus PMBOK [PMBOK].....	34
Obrázek 8: Spirálový model [ROYCE 1998] .....	35
Obrázek 9: Iterativní vývoj [RUP].....	36
Obrázek 10: Klíčové aktivity použité při vytváření této disertační práce.....	43
Obrázek 11: Vysvětlení notace použité ve schématech diagramů .....	44
Obrázek 12: Programme management [BOOCH 1999] .....	46
Obrázek 13: IT jako poskytovatel služeb [ITIL].....	54
Obrázek 14: Členění funkce řízení služeb [BARTLET 2003].....	56
Obrázek 15: Ukázka procesu ITIL - Řešení incidentů na operativní úrovni .....	58
Obrázek 16: Dlouhodobé (strategické) cíle [BARTLET 2003].....	60
Obrázek 17: Střednědobé (taktické) cíle [BARTLET 2003] .....	61
Obrázek 18: Krátkodobé (operační) cíle IT útvaru [BARTLET 2003] .....	62
Obrázek 19: Řízení incidentů.....	64
Obrázek 20: Přehled COBIT [GULDENTOPS 2000].....	67
Obrázek 21: Čtyři domény COBIT [GULDENTOPS 2000].....	68
Obrázek 22: Seznam procesů v doméně „Plánování a Organizace“ [GULDENTOPS 2000] .....	68
Obrázek 23: Seznam procesů v doméně „Dodávka a Podpora“ [GULDENTOPS 2000] .....	69
Obrázek 24: Seznam procesů v doméně „Akvizice a implementace“ [GULDENTOPS 2000] .....	70

Obrázek 25: Seznam procesů v doméně „Monitorování“ [GULDENTOPS 2000].....	70
Obrázek 26: Řízení IT [GULDENTOPS 2000].....	71
Obrázek 27: Struktura CORTEX [COR].....	75
Obrázek 28: Detail procesu Obchod (Win Business) [COR].....	76
Obrázek 29: Typický rozvoj systému v produkčním prostředí [MAROUNEK 2004/1]	79
Obrázek 30: Struktura PMM [PMM].....	80
Obrázek 31: Fáze Realizace dle metodiky PMM a RUP [ROYCE 1998, RUP].....	83
Obrázek 32: Životní cyklus projektu dle PMM [PMM].....	86
Obrázek 33: Metamodel RUP [RUP].....	89
Obrázek 34: Přehled procesů RUP [RUP].....	90
Obrázek 35: Detail disciplíny Project Management [RUP].....	91
Obrázek 36: Životní cyklus programme managementu.....	98
Obrázek 37: Konceptuální pohled na kompetenční centrum.....	99
Obrázek 38: Cíle kompetenčního centra.....	100
Obrázek 39: Organizační struktura v KC pro vybrané role.....	104
Obrázek 40: Čtyři základní skupiny procesů kompetenčního centra.....	105
Obrázek 41: Stavby požadavku z pohledu jejich zadavatele.....	107
Obrázek 42: Stavby požadavku z pohledu řešitele.....	109
Obrázek 43: Schéma procesu “Poskytování podpory”.....	110
Obrázek 44: Klíčové aktivity programme manažera kompetenčního centra.....	120
Obrázek 45: Výstupy aktivit programme manažera kompetenčního centra.....	121
Obrázek 46: Proces a výstupy řízení IT provozních projektů.....	122
Obrázek 47: Životní cyklus požadavku.....	126
Obrázek 48: Provoz a podpora – první úroveň.....	129
Obrázek 49: Provoz a podpora – druhá úroveň.....	130
Obrázek 50: Role a aktivity disciplíny analýza a design dle RUP [RUP].....	132
Obrázek 51: Role a výstupy disciplíny analýza a design dle RUP [RUP].....	133
Obrázek 52: Návrh procesu analýza v KC.....	134
Obrázek 53: Návrh procesu design v KC.....	139
Obrázek 54: Role a aktivity disciplíny implementace dle RUP [RUP].....	140

Obrázek 55: Role a výstupy disciplíny implementace dle RUP [RUP].....	140
Obrázek 56: Návrh procesu implementace v KC.....	143
Obrázek 57: Role a aktivity disciplíny testování dle RUP [RUP] .....	144
Obrázek 58: Role a aktivity disciplíny testování dle RUP [RUP] .....	145
Obrázek 59: Návrh procesu testování v KC.....	146
Obrázek 60: Role a aktivity disciplíny nasazení dle RUP [RUP].....	149
Obrázek 61: Role a výstupy disciplíny nasazení dle RUP [RUP] .....	150
Obrázek 62: Návrh procesu nasazení v KC .....	151
Obrázek 63: Role a aktivity disciplíny správa prostředí dle RUP [RUP] .....	153
Obrázek 64: Role a výstupy disciplíny správa prostředí dle RUP [RUP].....	154
Obrázek 65: Návrh procesu správa prostředí v KC .....	155
Obrázek 66: Role a aktivity disciplíny řízení konfigurací dle RUP [RUP] .....	156
Obrázek 67: Role a výstupy disciplíny řízení konfigurací dle RUP [RUP].....	156
Obrázek 68: Návrh procesu řízení konfigurací v KC .....	157
Obrázek 69: Řízení konfigurací .....	159
Obrázek 70: Backup a archivace.....	160
Obrázek 71: Řízení kvality v KC .....	162
Obrázek 72: Počet přijatých, vyřešených incidentů měsíčně na jeden produkt (zdroj: KC podpory bankovních IS).....	165
Obrázek 73: Trend přijatých, řešených chyb .....	166
(zdroj. KC podpory bankovních IS).....	166
Obrázek 74: Počet hlášených/vyřešených požadavků (zdroj: KC podpory bankovních IS).....	167
Obrázek 75: Trend spokojenosti zákazníka (zdroj: KC podpory bankovních IS) .....	168