

**Metodologie řízení projektů**  
**Project management methodology**

**Bakalářská práce**

**Petr Smetana**

**Vedoucí práce: PhDr. Milan Novák, Ph.D.**

**Jihočeská univerzita v Českých Budějovicích**

**Pedagogická fakulta**

**Katedra informatiky**

**2009**

## **Prohlášení**

Prohlašuji, že svoji bakalářskou práci jsem vypracoval/-a samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách.

V Českých Budějovicích dne .....

podpis:

Petr Smetana

## **Anotace**

Metodologie řízení projektů se zabývá studiem způsobů řešení problémů a hledání odpovědí v rámci řízení projektu a to nejen v IT. Správných postupů vedoucích k úspěšnému vyřešení projektu může být více. Jsou závislé na konkrétní metodice a její implementaci.

Tato práce bude zkoumat metodiky pro řízení vývoje softwaru. Teoretická část bude dále obsahovat pojednání o optimálním řízení projektů, nasazení projektových zdrojů a problematice kritických míst v plánování a řízení. Praktická část bude věnována vývoji aplikace na evidenci a správu projektů pro vývoj softwaru. Bude obsahovat Ganttův Diagram, řízení lidských zdrojů a řízení zakázek. Přínos práce by měl být v prozkoumání současných metodik pro řízení softwarových projektů a poté i v koncepci aplikace, která na základě řešerše vznikne.

## **Abstract**

Project management methodology deals with studies of ways to solve problems and search of the answers in the domain of the project management not even in IT. There can be more correct ways leading to the successful solution of the project. They are dependent on particular methodics and its implementation.

This paper is going to explore methodics for software development management. Theoretical part is going to contain treatise on optimal the project management, setting of project resources and problematics of critical points in scheduling and management. Practical part is going to be devoted to the development of application for evidence and management of projects for software development. It's going to contain Gantt's diagram, management of human resources and management of orders. The benefit of this paper should be in the research of contemporary project management methodics and than also in the concept of the application, which is going to be created on the base of the research.



# Obsah

|          |                                               |           |
|----------|-----------------------------------------------|-----------|
| <b>1</b> | <b>ÚVOD.....</b>                              | <b>8</b>  |
| <b>2</b> | <b>METODOLOGIE ŘÍZENÍ PROJEKTŮ V IT .....</b> | <b>9</b>  |
| 2.1      | ZDŮVODNĚNÍ, VÝZNAM A POTŘEBA STUDIE.....      | 9         |
| 2.2      | CÍLE PRÁCE A VÝZKUMNÉ OTÁZKY .....            | 9         |
| 2.2.1    | <i>Cíle práce .....</i>                       | <i>9</i>  |
| 2.2.2    | <i>Výzkumné otázky .....</i>                  | <i>9</i>  |
| 2.3      | HISTORIE A PERSPEKTIVY METODOLOGIÍ V IT.....  | 10        |
| 2.3.1    | <i>Historie.....</i>                          | <i>10</i> |
| 2.3.2    | <i>Perspektivy.....</i>                       | <i>11</i> |
| 2.4      | VYMEZENÍ POJMŮ .....                          | 12        |
| 2.4.1    | <i>Projekt.....</i>                           | <i>12</i> |
| 2.4.2    | <i>Řízení projektu .....</i>                  | <i>14</i> |
| 2.4.3    | <i>Oblasti poznatků.....</i>                  | <i>14</i> |
| 2.4.4    | <i>Systémový přístup.....</i>                 | <i>17</i> |
| 2.4.5    | <i>Životní cyklus projektu .....</i>          | <i>17</i> |
| 2.4.6    | <i>Životní cyklus vývoje systému.....</i>     | <i>18</i> |
| 2.5      | ŘÍZENÍ .....                                  | 21        |
| 2.5.1    | <i>Řízení integrace projektu .....</i>        | <i>21</i> |
| 2.5.2    | <i>Řízení rozsahu projektu.....</i>           | <i>23</i> |
| 2.5.3    | <i>Řízení času v projektu .....</i>           | <i>25</i> |
| 2.5.4    | <i>Řízení nákladů na projekt.....</i>         | <i>28</i> |
| 2.5.5    | <i>Řízení kvality projektů.....</i>           | <i>29</i> |
| 2.5.6    | <i>Řízení lidských zdrojů v projektu.....</i> | <i>30</i> |
| 2.5.7    | <i>Řízení komunikace v projektu.....</i>      | <i>32</i> |
| 2.5.8    | <i>Řízení rizik v projektu.....</i>           | <i>33</i> |
| 2.5.9    | <i>Řízení obstarávání v projektu.....</i>     | <i>36</i> |
| 2.6      | PRACOVNÍ POSTUPY .....                        | 38        |
| 2.6.1    | <i>Metodika Booch .....</i>                   | <i>38</i> |
| 2.6.2    | <i>Metodika Objectory.....</i>                | <i>39</i> |
| 2.6.3    | <i>Metodika RUP .....</i>                     | <i>40</i> |
| 2.6.4    | <i>Metodika OMT .....</i>                     | <i>43</i> |
| 2.7      | JAZYK UML .....                               | 48        |
| 2.7.1    | <i>Objekty a UML .....</i>                    | <i>49</i> |

|          |                                                      |           |
|----------|------------------------------------------------------|-----------|
| 2.7.2    | <i>Aspekty modelu UML</i> .....                      | 49        |
| 2.7.3    | <i>Struktura UML</i> .....                           | 50        |
| 2.7.4    | <i>Společné mechanismy</i> .....                     | 53        |
| 2.7.5    | <i>Architektura</i> .....                            | 57        |
| 2.8      | ANKETA O POČÍTAČOVÉ PODPOŘE PROJEKTOVÉHO ŘÍZENÍ..... | 59        |
| 2.8.1    | <i>Výsledky dotazníkového šetření</i> .....          | 60        |
| <b>3</b> | <b>METODICKÝ POSTUP</b> .....                        | <b>73</b> |
| 3.1      | SPECIFIKACE SOFTWAREOVÉHO SYSTÉMU PRO ŘÍZENÍ .....   | 73        |
| 3.1.1    | <i>Správa lidských zdrojů</i> .....                  | 74        |
| 3.1.2    | <i>Správa projektů</i> .....                         | 75        |
| 3.1.3    | <i>Statistika</i> .....                              | 76        |
| 3.2      | DISKUZE .....                                        | 77        |
| <b>4</b> | <b>ZÁVĚR</b> .....                                   | <b>81</b> |
| <b>5</b> | <b>PŘÍLOHA</b> .....                                 | <b>86</b> |

# **1 Úvod**

Dnešní rychlá doba nám přináší nepřehledné množství podnětů, o kterých je třeba více, či méně přemýšlet. Stejně tak je to s projekty, které je nutno na základě posbíraných informací analyzovat a v případě návratnosti investice realizovat. Metodologie řízení projektů je souborem procesů umožňujících zvyšovat efektivitu práce skrze techniky a nástroje, které procesy nabízí. Správných postupů vedoucích k úspěšnému vyřešení projektu může být více. Jsou závislé na konkrétní metodice a její implementaci.

Tato práce zkoumá metodologii řízení v IT. Teoretická část bude dále obsahovat zásady pro optimální řízení projektů, nasazení projektových zdrojů a problematice kritických míst v plánování a řízení. Praktická část bude věnována rešerši počítačové podpory projektového řízení a vývoji aplikace na evidenci a správu projektů pro vývoj softwaru. Ta bude obsahovat Ganttův diagram, řízení lidských zdrojů a řízení zakázek. Přínos práce by měl být v prozkoumání současných postupů při řízení softwarových projektů, jejich optimalizace a poté i v koncepci aplikace, která na základě rešerše vznikne.



## **2 Metodologie řízení projektů v IT**

### **2.1 Zdůvodnění, význam a potřeba studie**

Nalezení optimálního způsobu řízení zaručuje subjektu, který ho bude používat, konkurenční výhodu a to hned v několika směrech. Především však jde o zvýšení efektivity práce, tudíž o minimalizaci nákladů a snížení doby trvání projektu. V současné společnosti opírající se o fungování monetárního systému to znamená maximalizaci zisku. Na principy řízení projektů lze hledět jako na obecná doporučení, která určují jak se chovat v pozici projektového manažera. Úspěšné řízení projektů neznamena ve finále jen finanční úspory, ale také růst projektového týmu v rámci získaných zkušeností.

### **2.2 Cíle práce a výzkumné otázky**

#### **2.2.1 Cíle práce**

Cílem práce je na základě literárního rozboru a výsledků rešerše stanovit obecné principy řízení projektů, které budou zakomponovány jako funkcionality do nové aplikace pro správu a řízení projektů.

#### **2.2.2 Výzkumné otázky**

Jak zajistit optimální softwarovou podporu pro řízení projektu?

- Jaké jsou nejčastěji uváděné nástroje pro řízení rozsahu?
- Jaké jsou nejčastěji uváděné nástroje pro řízení času?
- Jaké jsou nejčastěji uváděné nástroje pro řízení nákladů?
- Jaké jsou nejčastěji uváděné nástroje pro řízení kvality?
- Jaké jsou nejčastěji uváděné nástroje pro řízení lidských zdrojů?
- Jaké jsou nejčastěji uváděné nástroje pro řízení komunikace?
- Jaké jsou nejčastěji uváděné nástroje pro řízení obstarávání?

- Jaké jsou nejčastěji uváděné nástroje pro řízení integrace?

## 2.3 Historie a perspektivy metodologií v IT

### 2.3.1 Historie

V šedesátých letech se projevovala softwarová krize velkým prodlužováním a prodražováním projektů, nízkou kvalitou výsledných produktů, problematickou údržbou a nízkou produktivitou práce programátorů. Hledání jednoduchého a účinného řešení na existující problémy vedlo k zavedení strukturovaného programování. V sedmdesátých letech dochází k dalšímu rozvoji metodologií. Je kladen větší důraz na úvodní etapy životního cyklu softwaru. Osmdesátá léta přináší vývoj nových programovacích jazyků a paradigmat. Objevují se první CASE (Computer Aided Software Engineering) nástroje. V devadesátých letech se spolu s příchodem objektově-orientovaných technologií, internetových projektů a širokým rozšířením výpočetní techniky začaly ukazovat slabé stránky v té době zavedených strukturovaných metodik (PRINCE, PROMT). Za největší nevýhodu strukturovaného přístupu se dá považovat jeho malá flexibilita a neschopnost vyrovnat se s dynamikou reálného světa, ve kterém se softwarové projekty realizují. (17)

Stráví-li projektový tým poměrně velké množství času snahou popsat systém, který má být vytvořen, aniž by při tom skutečně něco reálného vzniklo, zákonitě velmi rychle narazí na limity schopnosti abstrakce a představivosti těch, kdo systém chtějí. S ohledem na selhávání strukturovaných přístupů byla snaha o jejich náhradu modernějšími a flexibilnějšími přístupy. Za určité nástupce strukturovaných metodik je možno považovat první vlnu tzv. objektových metodik, mezi které lze zařadit např. Object Modeling Technique, metodu Booch, Object-Oriented Software Engineering, metodu Fusion a další. Tato první vlna byla charakteristická zaměřením na objektově orientovanou

analýzu a návrh. Autoři vytvořili vlastní grafickou notaci pro vizuální modelování a určili postupy vedoucí ke specifikaci systému. Málokterá metodika v této době pokrývala celý životní cyklus tvorby softwaru. Roztříštěnost objektových metodik v oblasti vizuálního modelování sjednotila specifikace Unified Modeling Language (UML). Koncem devadesátých let minulého století se objevilo několik komplexních metodik řízení softwarového vývoje, které se snažily obsáhnout celý životní cyklus projektu. Příkladem může být Unified Process, metodika OPEN/OML, AllFusion Process Library nebo MS Solutions Framework. (16)

Uvedené metodiky je možno uvažovat jako metodiky moderní či tradiční. Jsou charakteristické především tím, že se snaží pokrýt celý životní cyklus projektu a jsou založeny spíše na iterativním než kaskádovém životním cyklu. Pro tvorbu softwaru používají vizuálního modelování, automatizaci a podpůrné nástroje. Přístup k vývoji softwarových aplikací je vysoce organizovaný a systematický. Tyto metodiky využívají objektově-orientovanou terminologii a lze je upravit pro různé typy projektů.

### **2.3.2 Perspektivy**

Na počátku dvacátého prvního století se objevil nový fenomén na poli metodik vývoje softwaru, agilní metodiky. Za výchozí dokument těchto metodik je možno pokládat Manifest pro agilní softwarový vývoj od Kenta Becka či Alistaira Cockburna. Podle nich je možno agilní metodiky nastínit jako metodiky snažící se o lepší vývoj softwaru.

Dosahují toho především upřednostňováním jednotlivců a komunikace v týmu namísto procesů a nástrojů. Preferují funkční software před komplexními specifikacemi. Dávají přednost spolupráci se zákazníky před dlouhými dohady nad smlouvami a upřednostňují rychlou odezvu na změny před slepým plněním plánů. Agilní metodiky se dívají na stávající metodiky

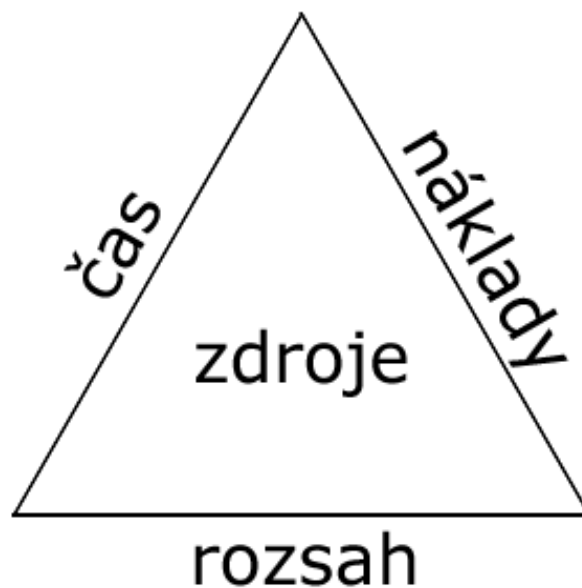
jako např. Unified Process jako na těžkotonážní. Oproti tomu samy sebe označují jako lehkotonážní. Uvedená terminologie je založena především na skutečnosti, že stávající metodiky jsou příliš orientovány na dodržování velkého množství přednastavených postupů a pravidel, což vede k zatěžování vývojového týmu a vzdaluje ho od vytvoření vysoce kvalitního produktu splňujícího požadavky na náklady, čas, kvalitu a rozsah. (11, str. 10-12)

Cílem agilních metodik je polidštit vývoj softwaru. Chtějí se ve větším měřítku soustředit na dosažení primárního cíle a méně se zabývat formalismem. Mezi agilní metodiky patří SCRUM, Crystal Light či extrémní programování (XP). Popularita agilních metodik, zejména extrémního programování, v posledních letech markantně vzrostla, agilní metodiky se staly fenoménem, který nelze přehlédnout. Proto jsou nově do tradičních metodik jejich autory implementovány i principy agilního vývoje. Což svědčí o skutečnosti kompatibility obou přístupů.

## **2.4 Vymezení pojmů**

### **2.4.1 Projekt**

Projekt je časově omezená pracovní činnost, jejímž cílem je vytvoření jedinečného produktu, služby, nebo dosažení jiného výsledku. Každý projekt je jistým způsobem omezen, a to svým rozsahem (Jaké práce budou v rámci projektu provedeny?), časem (Jak dlouho bude projekt trvat?) a náklady (Jaký je finanční rozpočet projektu?). Úspěšné řízení projektu se odvíjí od nalezení balance mezi těmito třemi veličinami a zároveň uspokojení zadavatele. Odpovědí může též být takzvaný magický trojúhelník Harolda Kerznera. (5, str. 21-24)



*Obr. 1 Kerznerův trojúhelník*

Projekty existují v různých formách a velikostech, většinou však je možno projekt definovat pomocí následujících atributů či základních vlastností.

- Projekt má jednoznačně určený cíl. (jedinečný výrobek, služba nebo jiný výsledek)
- Projekt je dočasný, má pevně stanovený začátek i konec.
- Řešení projektu se postupně vypracovává v rámci dekompozice konkrétních problémů, které v projektu mohou nastat.
- Projekt potřebuje určité zdroje, často z různých oblastí. (lidi, software, hardware, jiná aktiva) Zdroje jsou vždy omezené, proto je nutno se s nimi naučit účelně hospodařit.
- Projekt by měl mít hlavního zákazníka nebo zadavatele. Ten většinou přímo určuje směr vývoje projektu a uvolňuje na něj finanční zdroje.
- Součástí projektu je neurčitost. Každý projekt je jedinečný a někdy je

obtížné jasně nadefinovat jeho cíle, odhadnout trvání a potřebné náklady. Jednou z příčin jsou vnější faktory, například krach některé z dodavatelských firem, onemocnění člena projektového týmu. Tato neurčitost je jedním z hlavních příčin, proč je řízení projektů tak náročné. (1, str. 33-57)

#### **2.4.2 Řízení projektu**

Řízení projektu se sestává z veškerých poznatků, dovedností, nástrojů a technik uplatněných na aktivity projektu takovým způsobem, aby byly splněny i požadavky na projekt. Nejde tedy pouze o splnění konkrétních cílů projektu ohledně požadovaného rozsahu času, nákladů a kvality, ale současně by mělo být splněno očekávání osob, které jsou do projektu zapojeny nebo které jsou jeho aktivitami ovlivněny. Klíčovými prvky obecného rámce řízení projektů jsou účastníci projektu (zadavatel, manažer projektu, projektový tým, podpůrný personál, dodavatelé, oponenti), oblasti poznatků, nástroje a techniky pro řízení a konečně přínosy úspěšných projektů pro podnik jako celek. (6, str. 16)

#### **2.4.3 Oblasti poznatků**

Oblasti poznatků pro řízení projektu jsou sbírkou nejdůležitějších schopností či kvalifikace, které by si měl každý projektový manažer vypěstovat. Spadá do nich řízení rozsahu, času, nákladů, kvality, lidských zdrojů, komunikací, rizik, obstarávání a integrace projektu. Pro každou oblast poznatků existují různé nástroje a techniky, které jsou jejich základním stavebním kamenem.

- **Řízení rozsahu** projektu představuje definici a řízení všech prací, nutných k úspěšnému dokončení projektu.

*Nástroje a techniky:* stanovení rozsahu projektu, struktura rozpisu prací, pracovní příkazy, plán řízení rozsahu, analýza požadavků, řízení změn rozsahu

- **Řízení času** je odhad doby potřebné k dokončení prací, návrh časového plánu jako takový a zajištění včasného dokončení projektu.

*Nástroje a techniky:* Ganttovy diagramy, síťové diagramy projektu, analýza kritické cesty, metoda PERT, plánování kritického řetězu, crashing, rychlé sledování, revize milníků

- **Řízení nákladů** v projektu obsahuje přípravu rozpočtu na projekt a jeho dynamické řízení.

*Nástroje a techniky:* čistá současná hodnota, návratovost investic, analýzy doby návratnosti, řízení portfolia projektů, odhady nákladů, plán řízení nákladů, finanční software

- **Řízení kvality** určuje, že projekt naplní funkční předpoklady, pro které byl určen.

*Nástroje a techniky:* six sigma, diagramy řízení kvality, Paretovy diagramy, Ishikawovy diagramy, audity kvality, modely úspěšnosti, statistické metody

- **Řízení lidských zdrojů** určuje efektivní nasazení pracovních sil v rámci projektu.

*Nástroje a techniky:* motivační techniky, emfatické naslouchání, týmové smlouvy, matice přidělení odpovědností, histogramy zdrojů, vyrovnávání zdrojů, cvičení při budování týmu

- **Řízení komunikací** představuje vytváření, shromažďování, distribuci a archivaci informací o projektu.

*Nástroje a techniky:* plán řízení komunikací, řízení konfliktů, výběr komunikačních médií, komunikační infrastruktura, zpráva o stavu, virtuální komunikace, šablony, weby projektů

- **Řízení rizik** se skládá z rozpoznávání rizik, jejich analýzy a reakcí, které jsou riziky podmíněny.

*Nástroje a techniky:* plán řízení rizik, matice pravděpodobností a důsledků, hodnocení rizik, simulace Monte Carlo, sledování deseti nejrizikovějších položek

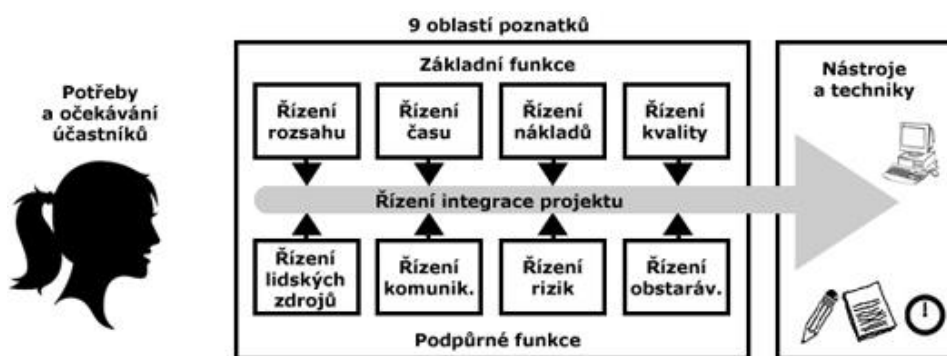
- **Řízení obstarávání** zaopatrjuje zboží a služby z vnějších organizací.

*Nástroje a techniky:* analýza „make-or-buy“, smlouvy, žádosti o návrh nebo o nabídku, výběr zdroje, vyjednávání o smlouvě, elektronické obstarávání

- **Řízení integrace** projektu je průřezovou oblastí ovlivňující se navzájem se všemi ostatními oblastmi poznatků

*Nástroje a techniky:* metody výběru projektu, metodologie řízení projektu, analýza účastníků, charta projektu, plán řízení projektu, software pro řízení projektů, výbor pro řízení změn, řízení konfigurace, kontrolní porady projektu, systémy pro autorizaci práce. (1, str. 40-45; 2, str. 8-12;8)

Pro lepší pochopení je přiložen obr. 2.



Obr. 2 základní rámec řízení projektů

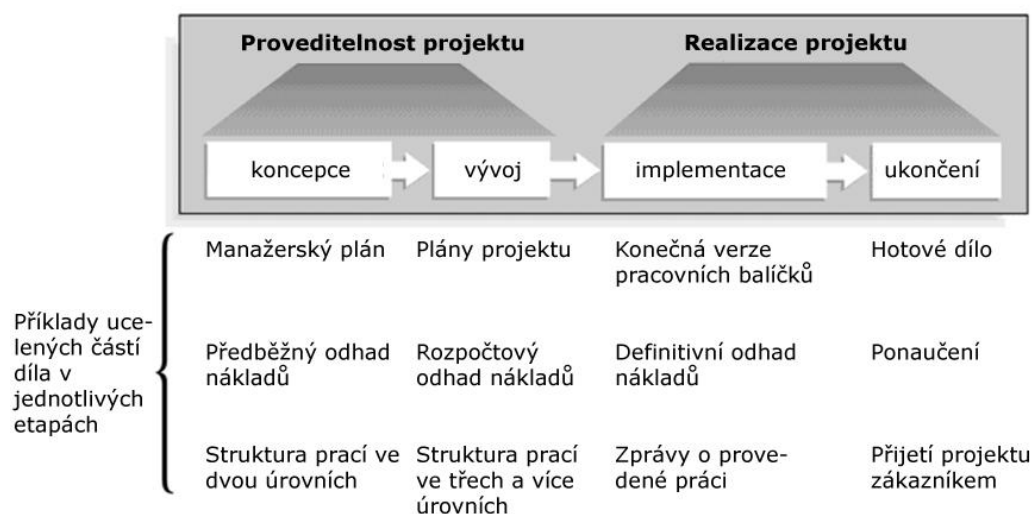


#### **2.4.4 Systémový přístup**

Tento pojem se poprvé objevuje někdy v padesátých letech. Označuje ucelený, analytický přístup k řešení sofistikovaných problémů. Jeho součástí je systémová filosofie, analýza systému a řízení systémů. Systémová filosofie je obecný model, ve kterém je možno přemýšlet o objektech jako o systémech. Systémem se pak rozumí obecná množina navzájem komunikujících komponent pracujících v určitém prostředí, sloužících k určitému účelu. Dalším důležitým pojmem je systémová analýza. Což je metoda řešení problémů, v níž je definován rozsah systému, který je dále rozdělen do dílčích komponent, v nichž je dále identifikován a posouzeny jeho problémy, příležitosti a potřeby. (2, str. 12)

#### **2.4.5 Životní cyklus projektu**

Projekty jsou v zásadě řešeny v rámci nějakého systému a jejich součástí je neurčitost, je vhodné rozdělit každý z nich do několika etap. Termín životní cyklus projektu pak označuje souhrn všech etap projektu a stanovuje, jaké práce budou v rámci každé etapy provedeny, jaké ucelené části díla budou v rámci projektu vytvořeny, kdy a kdo se bude na jednotlivých etapách účastnit a jak bude vedení kontrolovat a schvalovat práce provedené v těchto etapách. Etapy je možno v zásadě obecně rozdělit na úvodní koncepci, vývoj, implementaci a ukončení projektu. První dvě se označují společným pojmem proveditelnost projektu, zbylé dvě pak realizace projektu. Etapy jsou závislé na oboru, či konkrétním projektu. (1, str. 86-93)



Obr. 3 fáze klasického životního cyklu řízení projektů

## 2.4.6 Životní cyklus vývoje systému

Tímto termínem je označován obecný rámec pro popis jednotlivých etap vývoje informačních systémů. K rozšířeným modelům životního cyklu patří model vodopádu, spirálový model, inkrementální model, prototypový model, a model rychlého vývoje aplikací RAD. Tyto modely jsou příklady takzvaného prediktivního životního cyklu. To znamená, že rozsah projektu je možno předem jasně definovat a současně je možno předvídat časový plán i náklady na projekt. (1, str. 94-96)

Naproti tomuto způsobu vývoje stojí adaptivní (agilní) vývoj softwaru, ve kterém se vývoj softwaru postupně přizpůsobuje aktuálním podmínkám, protože na začátku nelze přesně stanovit cílové požadavky. Nejvýraznějším znakem tohoto přístupu je, že projekty jsou v něm řízeny cílem a jsou orientované na komponenty. Splnění cílových dat se sleduje pomocí časových cyklů. Požadavky jsou sestavovány iterativní metodou, vývoj je řízen riziky a je tolerantní ke změnám. Musí přijímat rizika a řešit je, nikoliv je potlačovat.

Nejrozšířenější zástupci tohoto modelu jsou extrémní programování a skrumáž.  
(11)

#### **2.4.6.1 Model vodopádu**

V životním cyklu modelu vodopádu jsou jasně definované, lineární etapy vývoje systému a jeho podpory. Jednotlivé etapy jsou řazeny za sebou. Tento životní cyklus předpokládá, že již definované požadavky na systém zůstanou neměnné. Problém tohoto modelu představuje uživatel, který není schopný na začátku přesně specifikovat požadavky. Tento model neodpovídá vývoji reálných projektů. (3, str. 60-63)

#### **2.4.6.2 Spirálový model**

Spirálový model je považován za vylepšeného nástupce modelu vodopádu, konkrétně pak na základě zkušeností s řešením velkých státních zakázek na vývoj softwaru. Zavádí prototypování a klade značný důraz na analýzu rizik. Tento model připouští skutečnost, že se většina softwaru vyvíjí nikoliv lineárně, ale právě ve spirálovitých iteracích vždy na vyšším stupni zvládnuté problematiky. (1, str. 90)

#### **2.4.6.3 Inkrementální model**

Tento model nabízí progresivní možnosti rychlého vývoje provozního softwaru. Vývoj v rámci nových verzí má aditivní charakter, co se funkcionality týče. Na základě specifikace systému jsou stanoveny ucelené části systému, které se uživateli odevzdávají postupně. (9, str. 108)

#### **2.4.6.4 Prototypový model**

Při použití tohoto modelu se namísto výsledného softwaru nejprve vytvářejí makety, na kterých se upřesní uživatelské požadavky kladené na cílový provozní software. Tento model požaduje výrazné zapojení uživatelů a vývojáři generují na základě makety funkční požadavky a specifikace fyzického návrhu. Hotové makety mohou být použity v dalších projektech a proto je účelné je archivovat. (1, str. 91)

#### **2.4.6.5 Model RAD**

Model rychlého vývoje aplikací používá zvláštní přístup, ve kterém vývojáři pracují s neustále se vyvíjejícím prototypem. Je zde patrná i velmi úzká vazba s uživateli, díky které je výsledný systém možno vyvinout dosti rychle a bez ohrožení kvality. Při vývoji se využívají vhodné nástroje pro rychlý vývoj aplikací jako je CASE (Computer Aided Software Engineering), JRP (Point Requirement Planning) a JAD (Joint Application Design), které usnadňují generování kódu a rychlé prototypování. (9, str. 109)

#### **2.4.6.6 Extrémní programování**

Tento model životního cyklu je reakcí na potřeby lidí při vývoji softwaru v rychle měnícím se prostředí. Specifickou vlastností životního cyklu extrémního programování je že vývojáři programují ve dvojicích. Je tím posílen synergický efekt, zvýšena produktivita práce a k uživateli se velmi záhy dostává hmatatelný výsledek. Tento způsob programování posiluje týmovou spolupráci a utvrzuje kolektivní vlastnictví zdrojového kódu. Nevýhodou je nízká možnost použití pro velké projekty (tým nad 20 lidí), vyžaduje zapojení zástupce uživatelů na plný úvazek a předpokládá vysoce

disciplinované vývojáře. V případě že je nutno na začátku projektu stanovit závazný časový plán a náklady je tento způsob považován za nevhodný. (11)

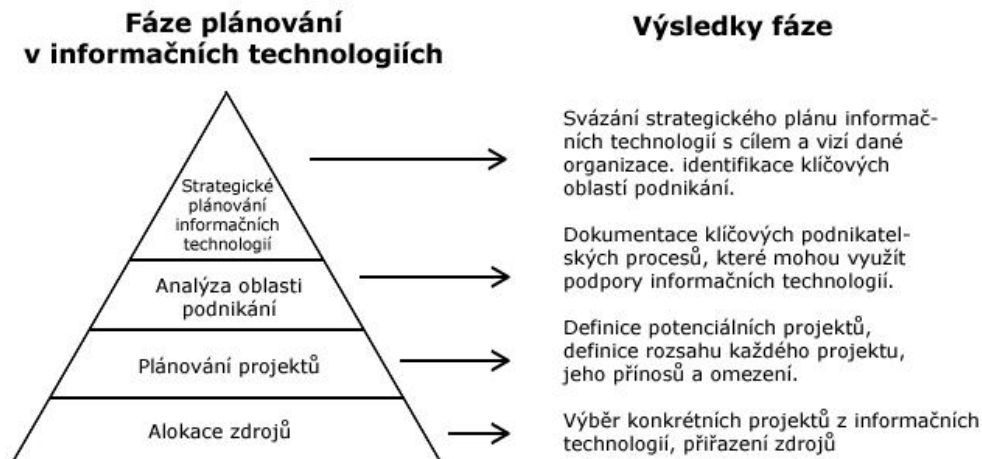
#### **2.4.6.7 Skrumáž**

Skrumáž (scrum) je proces částečně se podobající ostatním modelům životního cyklu postaveným na iterativním vývoji a reakci na měnící se požadavky. Jednotlivým iteracím se zde říká běhy, které běžně trvají 30 dní. Projektový tým má každý den krátkou poradu nazývanou se skrumáž. Na ní se rozhoduje, co se bude tento den dělat. Jednotliví členové pak mohou konzultovat své dosažené výsledky a problémy s manažerem projektu. Tento přístup vyžaduje silnou pozici projektového manažera, který musí koordinovat práce jednotlivých menších týmů. (1, str. 92)

## **2.5 Řízení**

### **2.5.1 Řízení integrace projektu**

Řízení integrace projektu představuje koordinaci všech ostatních oblastí poznatků pro řízení projektu v rámci jeho životního cyklu. Dá se též mluvit o řízení rozhraní, což je proces identifikace a řízení styčných bodů mezi různými elementy projektu. Nejčastěji jde o komunikaci mezi účastníky projektu. Tato integrace zajišťuje, že vše bude probíhat dle plánu a díky tomu bude projekt úspěšně dokončen.



*Obr. 4 Proces plánování při výběru projektů z informačních technologií*

V řízení integrace je zapojeno sedm hlavních procesů.

- **Vytvoření charty projektu**, která formálně potvrzuje vznik projektu, je vytvořena společně všemi účastníky projektu.
- **Návrh předběžného rozsahu projektu**, určuje rámcový seznam požadavků na rozsah projektu a vede k jeho předběžnému určení. Participují se na něm pak především cíloví uživatelé produktu, služby nebo jiného výsledku vzniklého z projektu. (1, str. 167-172)
- **Vytvoření plánu projektu** koordinujícího veškeré plánování v projektu a tvořící konzistentní dokument (plán řízení projektu).
- **Směrování a řízení výkonu projektu** je samo o sobě vykonáváním plánu řízení a provádění činností v něm popsanych. Výsledkem jsou pak ucelené části díla, požadované změny, informace o stavu prací, zavedení požadovaných změn, opravná a preventivní opatření a oprava defektů. (8)
- **Sledování a kontrola projektu** zajišťuje monitoring a kontrolu prací

s přihlédnutím na naplňování cílů projektu. Účelem sledování jsou pak doporučení ohledně preventivních a nápravných opatření.

- **Provádění integrovaného řízení změn** lze chápat jako koordinaci změn ovlivňující podobu ucelených částí projektu a procesních aktivit organizace. Meritem těchto procesů jsou schválená nápravná a preventivní opatření, schválené či zamítnuté požadavky změn, schválené a potvrzené opravy vad, kompletní části projektu a aktualizované verze plánu řízení a stanovení rozsahu projektu. (2, str. 14-17)
- **Uzavření projektu** je dokončením všech procesů projektu a jeho formální uzavření. Výsledkem těchto aktivit jsou finální verze produktu, služby či jiného výsledku, úpravy procesních aktivit organizace a smluvní postupy pro ukončení.

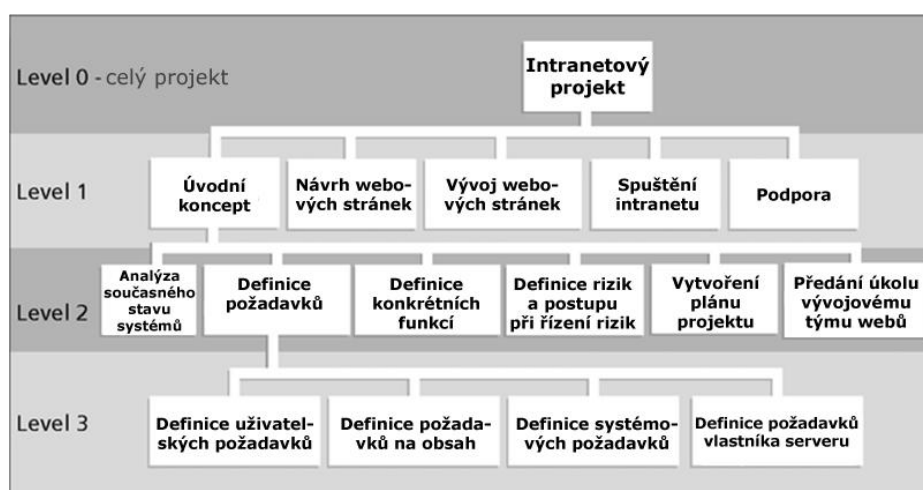
## 2.5.2 Řízení rozsahu projektu

Řízení rozsahu projektu je jednou z nejpodstatnějších a zároveň nejobtížnějších aspektů řízení. Slovním obratem rozsah projektu jsou míněny veškeré práce, zapojené do vytvoření produktů očekávaných v rámci projektu a zároveň procesy vedoucí k jejich vzniku. Tyto procesy definují a kontrolují práce, které do projektu budou, či nebudou zahrnuty. Taktéž zajišťují shodné chápání produktů, služeb a procesů potřebných k jejich vzniku, mezi účastníky projektu a projektovým týmem.

- **Plánování rozsahu** je rozhodnutí, jakým způsobem se bude definovat rozsah projektu, jak bude ověřován, kontrolován a jak bude vytvořena struktura rozpisu prací. Hlavním výstupem procesu plánování je plán řízení projektu vytvořený projektovým týmem.
- **Definice rozsahu** představuje revizi charty projektu a předběžného

stanovení rozsahu projektu, které byly vytvořeny během zahájení projektu. V procesech plánování dochází často k rozvoji požadavků, je tudíž nutno do charty doplňovat další informace. Hlavním výstupem tohoto procesu je stanovení rozsahu projektu, požadované změny a aktualizovaný plán řízení rozsahu projektu. (1, str. 202-205)

- **Struktura rozpisu prací** je rozdělením hlavní ucelené části projektu do ještě menších, lépe zvládnutelných komponent. Výstupem tohoto procesu pak je samotná struktura rozpisu prací, slovník struktury prací, směrný rozsah projektu, požadované změny v projektu, aktualizovaný plán řízení projektu a aktualizované stanovení změn rozsahu projektu. Pro stanovení rozpisu prací se používá např. metoda mind mapping. (8, str. 58-60)



*Obr. 5 Struktura prací v intranetovém projektu, uspořádaná podle fází*

- **Ověření rozsahu** je považováno za formální přijetí rozsahu projektu. V rámci tohoto procesu musí všechny zainteresované strany zkontrolovat a poté akceptovat seznam ucelených částí projektu. Pokud se v projektu objeví nesrovnalosti, případně nejsou-li ucelené části pro zadavatele či zákazníka přijatelné, vyžádají si standardně v projektu změny, které se následně projeví v doporučení následných nápravných

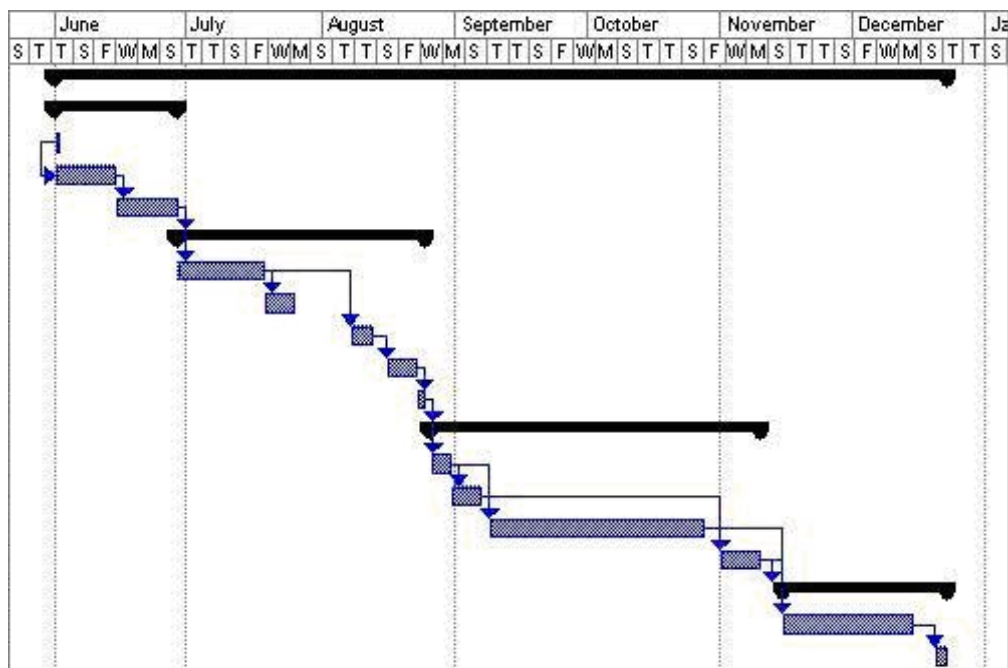


opatření. Výstupy tohoto procesu tudíž jsou schválené ucelené části projektu, požadované změny a nápravná opatření. (5, str. 109-115)

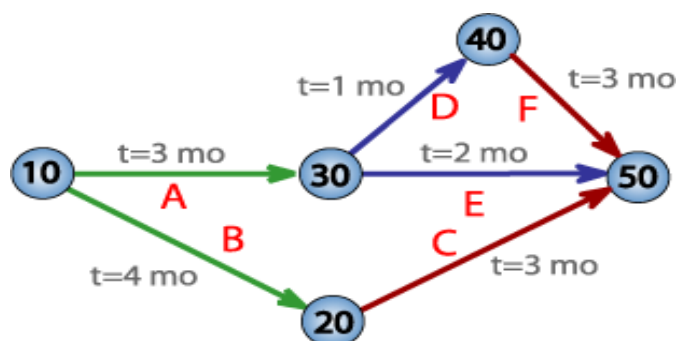
- **Kontrola rozsahu** představuje kontrolu změn rozsahu projektu, které ovšem bývají v řadě projektů z informačních technologií významným problémem. Hlavní částí kontroly rozsahu je identifikace, posuzování a implementace změn v rozsahu projektu, které nabíhají až s postupem řešení. Změny v rozsahu mají často vliv na to, zda je projektový tým schopen splnit časový rámec řešení a náklady na projekt. Proto je nutno zvážit, jaké budou mít změny v rozsahu přínosy a za jakou cenu budou zavedeny. Výsledky tohoto procesu jsou požadované změny, doporučená nápravná opatření a aktualizovaná verze stanovení rozsahu projektu, struktury a slovníku rozpisu prací, směrného rozsahu, plánu řízení projektu a procesních aktiv organizace. (1, str. 217-223; 10)

### 2.5.3 Řízení času v projektu

Nedodržení časového rámce bývá nejčastějším problémem při řízení projektů, je proto důležité používat správné metody řízení času v projektu a zajistit si jejich pomocí přijatelné výsledky. Řízení času v projektu je souborem veškerých procesů, nezbytných ke včasnému dokončení projektu. Dosáhnout včasné dokončení projektu ovšem není v žádném případě jednoduché. Nápomocné mohou být níže uvedené techniky, případně nástroje ve formě síťových a Ganttových diagramů.



Obr. 6 ukázka Ganttova diagramu vytvořeném v aplikaci MS Project 2003



Obr. 7 Síťový diagram PERT

V řízení času je zapojeno celkem šest hlavních procesů.

- **Definice aktivit** je soupisem konkrétních aktivit, které musí účastníci projektu a členové projektového týmu splnit, aby byly dokončeny všechny ucelené části díla. Aktivita je pak element práce, který je standardně definován ve struktuře rozpisu prací očekávanou dobou trvání, náklady a požadovanými zdroji. Výstupem tohoto procesu je

seznam aktivit s jejich atributy, seznam milníků a požadované změny.  
(13)

- **Proces řazení aktivit** obsahuje rozpoznání a dokumentaci relací mezi aktivitami v projektu. Výstupem této činnosti může být síťový diagram časového plánu projektu, požadované změny a aktualizované verze seznamu aktivit a jejich atributů. (2, str. 20-21)
- **Odhad zdrojů na aktivity** si lze představit jako odhadování objemu potřebných prostředků pro včasné dokončení činnosti. Do prostředků se zahrnují lidé, materiál a zařízení, které bude projektový tým potřebovat k řešení aktivit projektu. Výstupy tohoto procesu je struktura rozdělení zdrojů, požadavky aktivit na zdroje, požadované změny a aktualizované verze atributů aktivit. (4, str. 227)
- **Odhad trvání aktivit** je stanovením odhadu doby práce potřebné k dokončení jednotlivých aktivit a následně i projektu jako celku. Výstup obsahuje odhady dob trvání aktivit a aktualizované verze atributů aktivit. (4, str. 32)
- **Návrh časového plánu** je analýzou posloupnosti aktivit, odhadů dob trvání aktivit a požadavků na zdroje. Cílem je vytvoření časového plánu projektu, modelových dat o časovém plánu, směrného časového plánu, požadovaných změn a aktualizovaných verzí požadavků na zdroje, atributů aktivit, kalendáře zdrojů a plánu řízení projektu. (1, str. 245-263)
- **Řízení časového plánu** určuje kontrolu a řízení změn v časovém plánu projektu. Jeho výstupem je měření pracovního výkonu, požadované změny, doporučená nápravná opatření a aktualizovaná verze modelových dat o časovém plánu, směrného časového plánu, procesních aktiv organizace, seznamů aktivit a jejich atributů a plánu řízení projektu. (2, str. 23-25)

#### 2.5.4 Řízení nákladů na projekt

Tímto výrazem jsou označovány všechny procesy zajišťující dokončení projektu v rámci schváleného rozpočtu. Každý projekt musí mít jasné přiřazení přesné odhady času i nákladů. Odhady většinou záleží na zkušenostech projektového manažera, jehož úlohou je uspokojit účastníky projektu a současně se stále snažit o snižování a kontrolu nákladů. Do řízení nákladů na projekt jsou zapojeny následující tři procesy.

- **Odhad nákladů** znamená vytvoření přibližných nebo odhadovaných údajů o nákladech na zdroje, které jsou potřeba k dokončení projektu. Hlavními výstupy z procesu odhadování nákladů jsou odhady nákladů na aktivity a podpůrné detaily, dále pak požadované změny a aktualizovaná verze plánu řízení nákladů. Plán řízení nákladů v rámci celkového plánu řízení projektu je vhodné vytvořit pod procesy řízení integrace projektu. (4; 1, str. 285-301)
- **Rozpočet nákladů** představuje rozdělení odhadu celkových nákladů na jednotlivé pracovní položky, které pak tvoří směrný plán pro hodnocení efektivity. Hlavními výstupy z procesu rozpočtování jsou směrné náklady, požadavky na financování projektu, požadované změny a aktualizovaná verze plánu řízení nákladů. (4, str. 32)
- **Řízení nákladů** není ničím jiným než řízením změn v rozpočtu projektu. Jako hlavní výstupy tohoto procesu je možno uvést měření odvedené práce, předpovědi o dokončení projektu, požadované změny, doporučená nápravná opatření a aktualizované verze plánu řízení projektu, jehož součástí je i plán řízení nákladů, odhadu nákladů směrných nákladů a procesních aktiv organizace. (13)

### 2.5.5 Řízení kvality projektů

Úkolem řízení kvality v projektu je zajistit, že projekt uspokojí potřeby, kvůli kterým bylo zahájeno jeho řešení. Kvalita je v projektu dána klíčovými účastníky, především hlavním zákazníkem projektu. Kvalitu je tudíž nutno klást na stejnou úroveň jako rozsah projektu, čas a náklady. Jestliže účastníci projektu nejsou spokojeni s kvalitou řízení projektu nebo s jeho výslednými produkty, musí projektový tým nakonec upravit rozsah projektu, čas nebo náklady, aby tak potřeby a očekávání účastníků uspokojil. Splnit jen písemně stanovené specifikace rozsahu, času a nákladů nestačí. Má-li projektový tým dosáhnout uspokojení účastníků, musí si s nimi vybudovat dobré, fungující vztahy a pochopit, jaká jsou jejich vyslovená i nevyslovená přání a potřeby. Součástí řízení kvality projektu jsou tři procesy.

- **Plánování kvality** je vymezením norem kvality v projektu a stanovuje způsob jejich naplnění. Důležitou součástí je začlenění norem kvality do návrhu projektu. Mezi takovéto normy kvality u projektů z informačních technologií může patřit to, že v něm je nutno počítat s dalším růstem systému, vhodná doba odezvy systému, nebo podávání vždy přesných konzistentních informací. Standardy kvality mohou platit také pro služby z informačních technologií. Takto je možno například stanovit, jak dlouho smí trvat odpověď z technické podpory, nebo za jak dlouho je nutné dodat náhradní hardwarovou součástku. Hlavními výstupy z procesu plánování kvality jsou plán řízení kvality, metriky kvality, seznam požadavků na kvalitu, plán zlepšování procesů, aktualizovaná verze plánu řízení projektu a směrná kvalita. (1, str. 330 - 332)
- **Zajišťování kvality** je pravidelné posuzování celkové výkonnosti projektu a kontrolování, jestli projekt uspokojí odpovídající normy kvality. Součástí tohoto procesu je také převzetí odpovědnosti za kvalitu v celém životním cyklu projektu. Výstupem pak jsou

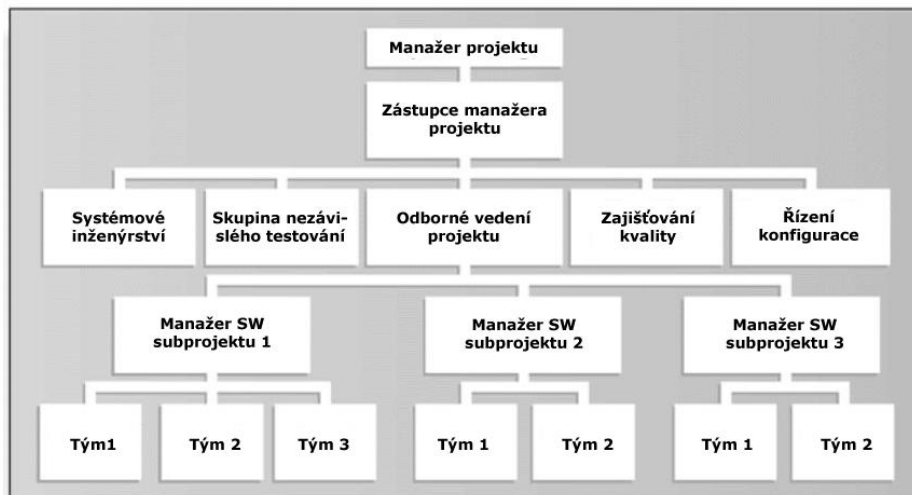
požadované změny, doporučená nápravná opatření a aktualizované verze procesních aktiv organizace a plánu řízení projektu. (13)

- **Kontrola kvality** znamená sledování konkrétních výsledků projektu, při kterém se ověří, jestli odpovídají příslušným normám kvality, a identifikují se možnosti zlepšení celkové kvality. Tento proces je často spojen s různými technikami řízení kvality, jako jsou Paretovy diagramy, diagramy kontroly kvality a statistické vzorky. Hlavními výstupy procesu kontroly kvality jsou měření kontroly kvality, ověřené a doporučené opravy závad, doporučená nápravná opatření a preventivní opatření, požadované změny, validované verze ucelených částí projektu, aktualizované verze směrné kvality, procesních aktiv organizace a plánu řízení projektu. (2, str. 31-32)

#### 2.5.6 Řízení lidských zdrojů v projektu

Součástí řízení lidských zdrojů v projektech jsou veškeré procesy, se kterými se dosáhne co nejefektivnějšího využití osob zapojených do řešení projektu. Do řízení lidských zdrojů jsou zapojeni všichni účastníci projektu a to zadavatelé, zákazníci, členové projektového týmu, podpůrní zaměstnanci, dodavatelé přispívající k řešení projektu a další. V řízení lidských zdrojů jsou rozeznávány následující čtyři nejdůležitější procesy.

- **Plánování lidských zdrojů** znamená identifikovat a dokumentovat veškeré role pracovníků v projektu, jejich povinnosti a vztahy nadřízenosti. Klíčovými výstupy tohoto procesu jsou role a povinnosti, organizační diagram projektu a plán řízení zaměstnanců. (5, str. 154)



*Obr. 8 ukázka organizačního diagramu*

- **Sestavení projektového týmu** vychází ze shromáždění veškerého personálu, který je přiřazen k řešení projektu a jenž na něm bude pracovat. Nejdůležitějšími výstupy z tohoto procesu jsou přiřazení úkolů zaměstnancům, informace o dostupnosti zdrojů a aktualizované verze plánu řízení zaměstnanců (1, str. 400-406)
- **Rozvoj projektového týmu** představuje budování a rozvíjení dovedností celé skupiny i jednotlivců, které povedou ke zlepšení efektivity projektu. Schopnosti budování a rozvoje týmu jsou pro mnohé z projektových manažerů poměrně velkým úskalím. Výsledkem tohoto procesu je hodnocení výkonnosti týmu. (6, str. 213-215)
- **Řízení projektového týmu** zahrnuje činnosti jako sledování pracovního výkonu jednotlivých členů týmu, motivace členů týmu, poskytování včasné zpětné vazby v rámci poznámek a připomínek, řešení problémů a konfliktů a koordinace změn. Při těchto činnostech se dá opět sledovat zvýšení efektivity projektu jako celku. Mezi výsledky tohoto procesu patří požadované změny, doporučená nápravná opatření a preventivní opatření, aktualizované verze procesních aktiv

v organizaci a aktualizovaný plán řízení projektu. (1, str. 411-413)

### 2.5.7 Řízení komunikace v projektu

Mnozí odborníci se shodují na tom, že největší hrozbou pro splnění jakéhokoli projektu, především pak projektu z oboru informačních technologií, jsou nedostatečné či nevhodné komunikace. Studie společnosti Standish Group z roku 2001 uvádí, že čtyřmi nejdůležitějšími faktory úspěchu projektu v informačních technologiích je podpora firemního vedení, zapojení uživatelů, zkušený projektový manažer a jasně stanovené podnikatelské cíle. Všechny tyto faktory přímo závisí na dobrých komunikačních schopnostech projektového manažera i týmu, především pak při komunikaci se zaměstnanci mimo obor informačních technologií. (1, str. 424-426)

Cílem řízení komunikací v projektu je zajistit včasné a náležité generování, shromažďování, distribuci, ukládání a likvidaci informací o projektu. Do řízení komunikací spadají celkem čtyři nejdůležitější procesy:

- **Plánování komunikace** znamená stanovení okruhu informací a komunikačních potřeb pro sdělení pro účastníky projektu. Stanoveno je kdo bude potřebovat jaké informace, kdy je bude potřebovat a jakým způsobem mu bude tato informace předána. Výstupem tohoto procesu je plán řízení komunikací. (13)
- **Distribuce informací** je proces, při kterém se všechny potřebné informace dostanou včas k jednotlivým účastníkům projektu. Hlavními výstupy tohoto procesu jsou aktualizované verze posledních aktiv organizace a požadované změny. (2, str. 36)
- **Oznamování efektivity** zahrnuje shromažďování a třídění dat o efektivitě, mezi které patří stavové zprávy, měření postupu a předpovídání. Výsledkem tohoto procesu jsou zprávy o výkonu, předpovědi, požadované změny, doporučená nápravná opatření a



aktualizovaná verze procesních aktivit organizace. (2, str. 36)

- **Řízení účastníků** znamená takové řízení komunikací, které uspokojí potřeby a očekávání účastníků v projektu a povede k vyšetření problému. Jako výstupy tohoto procesu je možno jmenovat vyřešené problémy, schválené požadavky na změny a nápravná opatření a aktualizované verze procesních aktivit organizace a plánu řízení projektu. (1, str. 438-451)

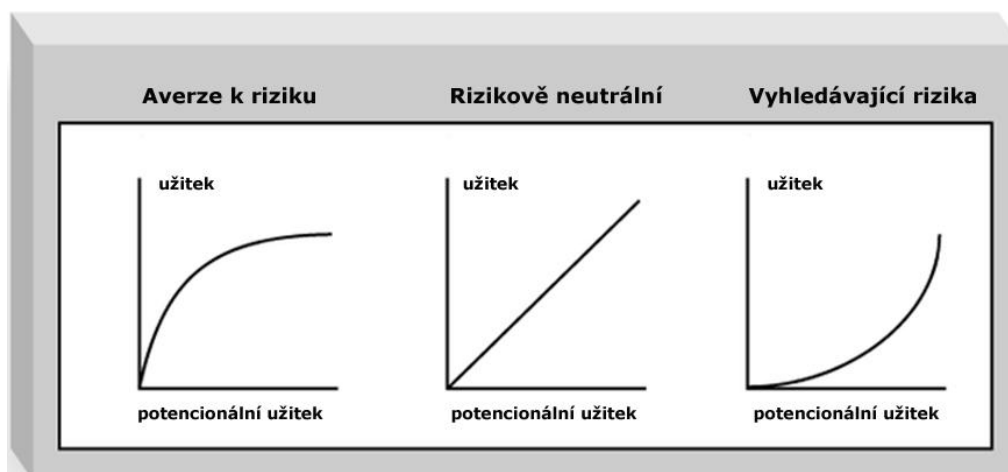
### 2.5.8 Řízení rizik v projektu

Řízení rizik v projektu je z jedné částečně exaktní vědou a částečně uměním, které zahrnuje identifikaci rizik jejich analýzu a reakci na ně po celou dobu chodu projektu, přičemž je nutno se řídit těmi nejlepšími zájmy pro naplnění cílů projektu. Řízení rizik je jedním z často přehlížených aspektů při řízení projektů, ale při správném uplatnění může naopak velmi výrazně přispět ke zvýšení míry konečné úspěšnosti projektů. Správné řízení rizik může mít pozitivní dopady pro výběr projektů, stanovení jejich rozsahu i do vytvoření realistického časového plánu a odhadu nákladů. Účastníci mohou díky řízení rizik lépe pochopit povahu projektu, do definování silných a slabých stránek projektu jsou zapojeni i členové projektového týmu a zároveň je možno takto integrovat ostatní oblasti poznatků při řízení projektů. (1, str. 464-469)

Dobré řízení rizik v projektech probíhá často zcela bez povšimnutí, stejně jako dobré krizové řízení. Při krizovém řízení vzniká přímé nebezpečí pro úspěch projektu. Krize tak na sebe upoutá velmi výraznou pozornost celého projektového týmu. Vyřešení krize je podstatně lépe viditelné než úspěšné řízení rizik.

Na cíl řízení rizik v projektu je možno pohlížet jako na minimalizaci potenciálních negativních rizik za současné maximalizace potenciálních pozitivních rizik. Rizika, která již projektový tým identifikoval a analyzoval, se

někdy označují pojmem známá rizika. Tato známá rizika je možno řídit proaktivně. Neznámá rizika, která nebyla dosud identifikována, ani analyzována není možno řídit. Je velmi důležité pokusit se všechna rizika v projektu předem identifikovat a řídit je. Součástí řízení rizik je celkem šest hlavních procesů. (5, str. 161-163)



*Obr. 9 Funkce užitku rizika při různých preferencích rizik*

- **Plánování řízení rizik** je rozhodování, jak postupovat při aktivitách řízení rizik v projektu a jak je plánovat. Při tomto plánování provádí projektový tým revize stanovení rozsahu projektu, plánu řízení projektu, faktorů podnikového prostředí a organizačních aktiv organizace, a tím pádem rozebírá a analyzuje aktivity řízení rizik v konkrétním projektu. Hlavním výstupem tohoto procesu je plán řízení rizik. (1, str. 469-474)
- **Identifikace rizik** je stanovení, která rizika mohou pravděpodobně ovlivnit průběh projektu, a dokumentování charakteristických vlastností každého z nich. Hlavním výstupem z tohoto procesu je základní verze registru rizik. (2, str. 39)
- **Kvalitativní analýza rizik** představuje kvalitativní rozbor hrozících

rizik a seřazení závažnosti podle jejich pravděpodobnosti a dopadů případného vzniku. Po identifikaci rizik může projektový tým ohodnotit rizika a aktualizovat informace v registru rizik pomocí několika různých nástrojů a technik. Nejdůležitějším výstupem tohoto procesu je aktualizovaná verze registru rizik. (1, str. 480-486)

- **Kvantitativní analýza rizik** obsahuje číselné odhady dopadů rizik na cíle projektu. Také v rámci toho procesu je prováděna aktualizace do registru rizik. (2, str. 40)
- **Plánování reakcí na rizika** znamená provedení takových kroků, které zlepší příležitosti a sníží hrozby vůči splnění cílů projektu. V tomto procesu využívá projektový tým výsledků z předcházejících procesů řízení rizik a může vytvořit vhodné strategie reakcí na rizika, které často vedou k aktualizaci registru rizik a plánu řízení rizik a také k vytvoření smluvních ujednání souvisejících s riziky. (1, str. 493-496)
- **Sledování a kontrola rizik** není nic jiného než monitoring identifikovaných a zbytkových rizik, identifikace rizik nových, vykonávání plánů pro reakci na rizika a vyhodnocování efektivity strategií rizik během celé doby života projektu. Podstatnými výsledky tohoto projektu a organizačních aktiv organizace. (6, str. 278)

Pro každou oblast poznatků existují specifická rizika, které je dobré hlídat.

- **Integrace** – neodpovídající plánování, nevhodná alokace zdrojů, nedostatečné řízení integrace, neprovedení revize po dokončení projektu
- **Rozsah** – nedostatečná definice rozsahu nebo pracovních balíčků, neúplné definice
- **Čas** – chyby při odhadech času nebo dostupnosti zdrojů, chyby při stanovení kritické cesty, nesprávná alokace časových rezerv a jejich řízení, příliš brzké uvedení konkurenčních produktů na trh.

- **Náklady** – chyby při odhadech, neodpovídající produktivita práce, náklady, změny nebo mimořádné události
- **Kvalita** – nesprávné postoje ke kvalitě, nedostatečná kvalita návrhu, materiálu nebo řemeslného provedení, neodpovídající program zajišťování kvality
- **Lidské zdroje** – nesprávné řízení konfliktů, nevhodná organizace projektu a nesprávné definování povinností, chybějící vůdčí postavení
- **Komunikace** – nedbalé plánování nebo nedbalá komunikace, nedostatečná komunikace, nedostatečné konzultace s klíčovými účastníky
- **Rizika** – ignorování rizik, nejasná analýza rizik, nedostatečné řízení pojištění
- **Obstarávání** – nesplnitelné podmínky či klauzule ve smlouvách, nedobré vztahy s dodavateli

### 2.5.9 Řízení obstarávání v projektu

Termín obstarávání míní výběr a nákup zboží nebo služeb vnějšího zdroje. Tento pojem se nejčastěji užívá ve státní a veřejné sféře. Mnohé ze soukromých společností používají jednoduše slovo nákup a profesionálové z oboru informačních technologií mají v oblibě slovo outsourcing. Úspěch mnoha projektů z informačních technologií, které využívají různé externí zdroje, je často postaven na dobrém řízení obstarávání v projektu. Výrazem řízení obstarávání v procesu jsou míněny všechny procesy, které zajišťují obstarání zboží a služeb, nezbytných k řešení projektu. Prostřednictvím jejich nákupu od vnějších organizací. Řízení obstarávání v projektu se skládá ze šesti hlavních procesů.

- **Plánování nákupů a obstarávání** je stanovením co, kdy a jak bude

potřeba obstarat. Při plánování obstarávání je nutno se rozhodnout, co bude třeba zadat formou outsourcingu, určit typ uzavírané smlouvy a popsat požadovaný rozsah prací pro potenciální prodávající. Proávající je přitom smluvní strana, dodavatel nebo poskytovatel, jenž dodává zboží či služby jiným organizacím. Výstupem z tohoto procesu je plán řízení obstarávání, smluvní pracovní příkazy, rozhodování typu make-or-buy a požadované změny do projektu, které mohou být důsledkem tohoto procesu. (1, str. 513-521)

- **Plánování smluv** je popisem požadavků na produkty nebo služby, které budou obstarávány, a identifikace potencionálních zdrojů nebo prodávajících. Mezi výstupy patří různé dokumenty pro obstarávání, jako jsou žádosti o návrh, kritéria hodnocení a aktualizované verze smluvních pracovních příkazů. (2, str. 41)
- **Vyžádání nabídek od dodavatelů** je proces, při kterém je nutno od potencionálních prodávajících získat potřebné informace, kalkulace, nabídky návrhy a podobně. Hlavním výstupem tohoto procesu je seznam kvalifikovaných dodavatelů, soubor dokumentů k obstarávání a konkrétní návrhy a nabídky. (2, str. 41)
- **Výběr dodavatele** je volbou mezi potenciálními prodávajícími a prochází se procesem jejich hodnocení a nakonec vyjednávání o přesné podobě smlouvy. Výstupem z tohoto procesu je množina vybraných dodavatelů, smlouvy, plán řízení smluv, informace o dostupnosti zdrojů, aktualizovaná verze plánu řízení obstarávání a požadované změny. (5, str. 213-214)
- **Administrativní zajištění smluv** představuje řízení vztahu s vybraným dodavatelem. Jako výstupy z tohoto procesu je třeba získat smluvní dokumentaci, požadované změny, doporučená nápravná opatření a aktualizované verze procesních aktiv organizace a plánu řízení

projektu. (1, str. 527-529)

- **Ukončení smlouvy** znamená dokončení a uhrazení jednotlivých smluv včetně vyřešení všech případných otevřených položek. Výsledkem tohoto procesu jsou ukončené smlouvy a aktualizované verze procesních aktiv organizace. (18)

## 2.6 Pracovní postupy

Při vývoji každého unikátního produktu či služby je vhodné postupovat dle zásad některého ze zvolených pracovních postupů (metodik). Pracovní postup definuje fáze a s nimi spojené techniky a procesy v rámci běhu projektu. Kromě souboru zavedených metodik, jsou ve značné míře rozšířeny metodiky vlastní. Bývají založené na principech řízení projektů a přizpůsobené pro prostředí konkrétní společnosti.

### 2.6.1 Metodika Booch

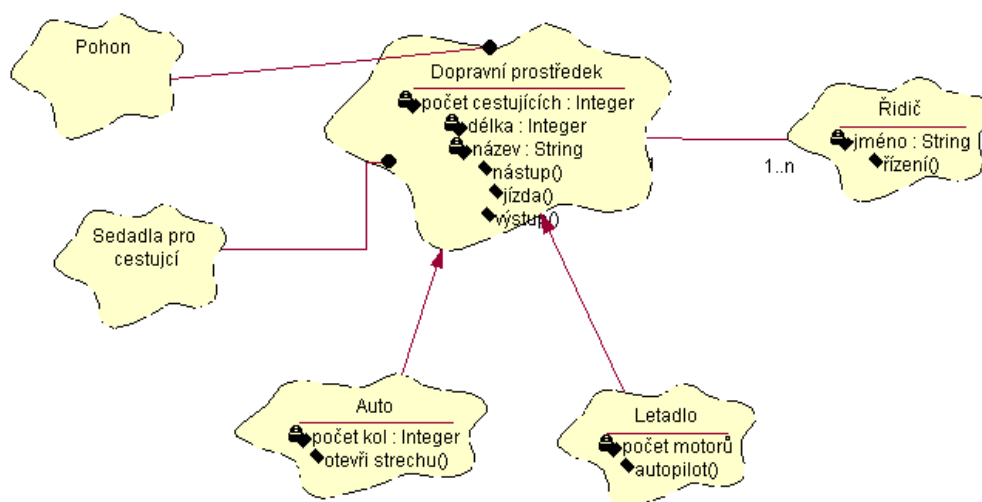
Metodika Booch od Grady Boocha z roku 1994 zasahuje do oblasti analýzy reality a analýzy požadavků, největší důraz je však kladen na design. Z přístupu je patrný silný vztah k programování a základy položené na inkrementálním modelu vývoje.

Booch skýtá více pohledů pro modelování systému, konkrétně pak model fyzický, logický, dynamický a statický. Logický model představuje množinu souvisejících tříd a objektů popisujících statický model. Tvorba kategorií tříd pro jejich seskupování je pak otázkou notace. Každá z kategorií je vlastním třídním diagramem. Vztahy mezi objekty pak zachycuje objektový diagram, jehož úkolem je i popis dynamického chování systému. Fyzický model obsahuje diagramy modulů a procesní diagramy. Výsledkem je model formace

tříd a objektů do modulů a vizualizace softwarové struktury v rámci nasazení na hardwarové komponenty. (16)

Metodika Booch popisuje i postup návrhu a vývoje informačního systému. V rámci toho obsahuje dva procesy. První je totožný s obecnou posloupností kroků při aktuálním vývoji objektově orientovaného softwaru a nazývá se makroproces. Do posloupnosti jeho technik patří sběr dat, analýza, design, implementace a poimplementační údržba. Druhým souborem technik je mikroproces popisující obecně uznávané kroky pro tvorbu diagramu tříd a objektů. Skládá se z určení tříd a objektů na určené úrovni abstrakce, zachycení jejich vlastností (sémantiky), tvorba vztahů mezi třídami a objekty a určení rozhraní a implementace vytvořených tříd a funkcí. (19, str. 31-39)

K hlavním podkladům této metodiky patří diagram tříd, objektový diagram, diagram modulů, diagram procesorů, stavový diagram, diagram interakcí.



Obr. 10 ukázka diagramu tříd v notaci Booch (16)

## 2.6.2 Metodika Objectory

Metodika Objectory od Ivara Jacobsona se poprvé objevila v roce 1992 v knize Objektově orientované softwarové inženýrství. Má za účel pokrýt celý

životní cyklus vývoje softwarové aplikace. Ve své době přišla tato metodika jako první s nástrojem use CASE, jehož meritem je takzvaný diagram užití. Tento nástroj zachycuje interakci mezi osobami nebo jinými objekty a systémem. Use CASE poté v rámci diagramu modeluje požadavky aktora na systém, je tedy posloupností akcí, které proběhnou, když systém komunikuje s aktorem za účelem poskytnutí vybraných dat. Z posloupnosti se poté vychází při tvorbě objektového modelu žádané oblasti. (12, str. 323)

Use CASE nástroje hrají dvě významné role. Jednak ilustrují požadavky na funkcionalitu systému a pak strukturují každý objektový/třídní model. Objektové modely robustních aplikací je vhodné dále rozdělit na samostatné use CASE diagramy menších celků a rozhraními je opět spojit ve funkční celek. Což je výhodné pro následnou dělbu práce. (16)

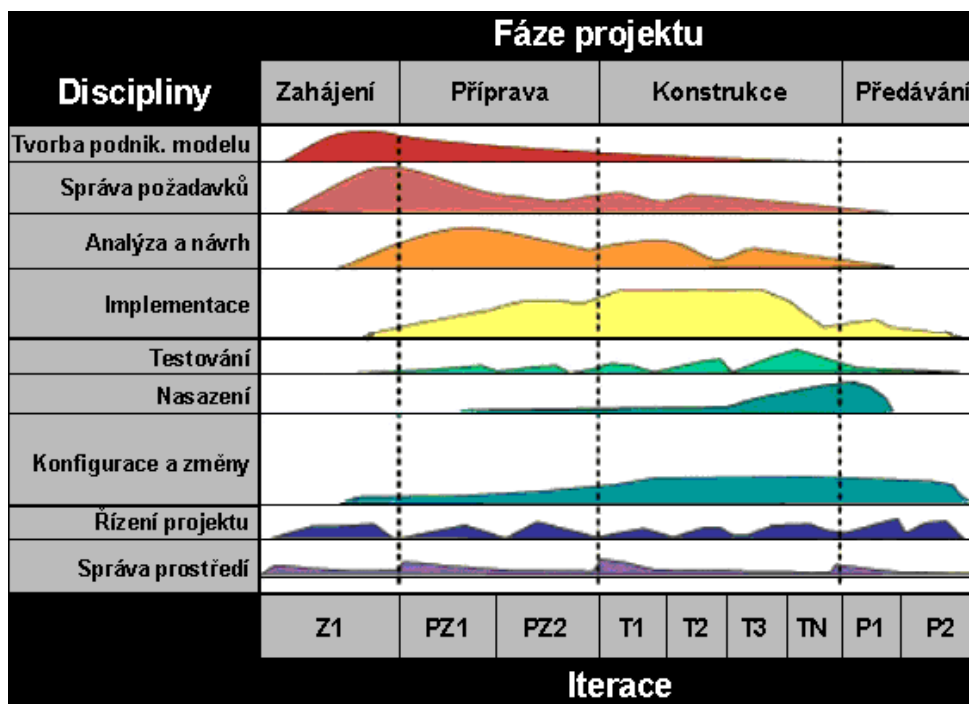
Pro mapování do objektových modelů se následně využívají dva typy nástrojů, a to objektové diagramy znázorňující jednotlivé objekty a aktory v jednotlivých use CASE a diagramy interakcí mající za úkol nastínění jak spolu objekty v use CASE spolupracují. Procesy dané v metodice Jacobson jsou totožné s metodami jiných metodik. Ne vždy jsou však prováděny jeden za druhým, ve většině případů dochází k jejich prolínání a kooperaci. Mezi procesy patří analýza požadavků, robustní analýza, design, implementace a testování. K hlavním podkladům této metodiky patří use CASE diagram, objektový diagram, diagram interakcí. (7)

### **2.6.3 Metodika RUP**

Metodika RUP je následovníkem metodiky Objectory. V literatuře se objevuje v roce 1995. V roce 1996 pak došlo k zásadnímu kroku, kdy tato metodika začala používat UML. Metodika RUP se však stále vyvíjí její poslední verze 5.5 je z roku 2002.

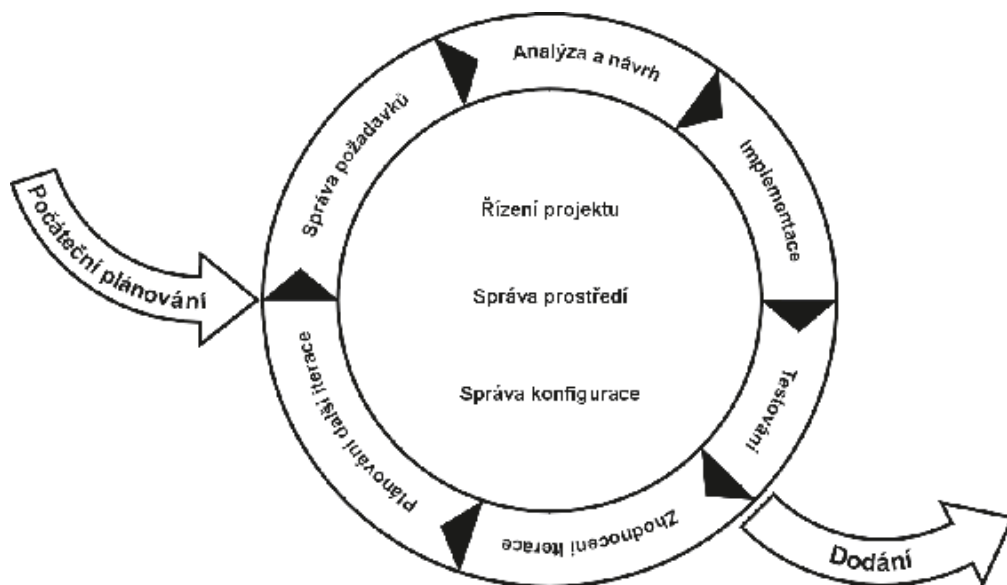
Hlavní filosofií metodiky RUP je šest obecných pravidel běžně používaných při vývoji aplikací.





Obr. 11 Schéma projektu dle metodiky RUP (14)

- **Iterativní vývoj** spočívá v separaci projektu na čtyři fáze, v každé z nich se nachází několik iterací vodopádového modelu. Každá iterace má za úkol vytvořit funkční celek. Tato metoda má mnohé výhody. Díky struktuře je možno objektivně posuzovat stav projektu. V rámci separace bloků dochází taktéž k rovnoměrnějšímu pracovnímu vytížení vývojového týmu, atd. (15)



Obr. 12 Iterativní vývoj (14)

- **Aktivní správa požadavků** u kaskádového modelu je sběr a dokumentace požadavků stanoven pouze na začátek projektu, jejich změna se v rámci dalšího vývoje většinou nepřipouští. RUP ovšem stojí za koncepcí aktivní správy požadavků, ta spočívá ve stálém kontaktu dodavatele se zadavatelem, který má možnost upřesňovat a upravovat požadavky při běhu projektu. (14)
- **Komponentová architektura** rozděluje netriviální součásti systému s určitou architekturou do jednotlivých bloků. Projekt lze díky tomu rozdělit mezi více vývojových týmů, změny se lépe zakomponují a tvorba systému je postupná v rámci funkčních bloků. RUP podporuje komponentové rozložení (COM a COBRA) (12, str. 351-352)
- **Vizuální modelování** je grafickým znázorněním systému a jeho fungování pomocí jazyka UML. Existují i nástroje, které umí na základě graficky definovaného třídního modelu vygenerovat kód a naopak. (software Rational Rose – C++) (7)

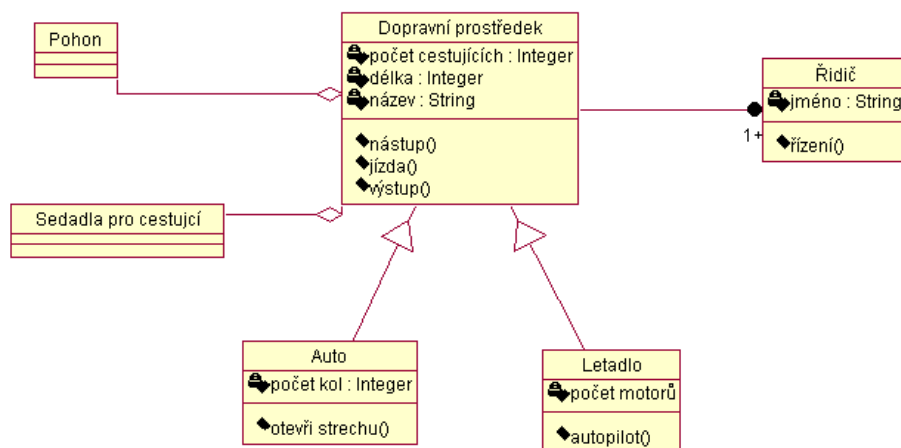
- **Ověření kvality** zaručuje splnění požadavků kladených na software. Ten by měl splňovat žádaná kvalitativní kritéria. V rámci testování zkoumáme funkcionalitu, spolehlivost a výkon.
- **Řízení změn** nabízí možnost efektivní implementace změn v rámci běžícího projektu. Jejich řízení ilustruje optimální přístup k zakomponování jak do kódu, tak do dokumentace. (12, str. 55-69)

#### 2.6.4 Metodika OMT

Metodiku OMT poprvé představil James Rumbaugh v knize Objektově orientované modelování a návrh v roce 1991. Z metodiky OMT pro její oblíbenost vychází mnoho metodik nových (např. Select perspective).

V rámci analýzy používá tři modely:

- **Objektový model** definuje třídy a jejich vztahy dohromady s atributy a metodami, ilustruje tím statickou strukturu systému. Jeho tvorba by měla obsahovat následující kroky. (16)
  - Vytvoření slovních popisů problému
  - Určení třídy objektů
  - Zrušení nepotřebných a chybných tříd
  - Připravení knihovny znalostí
  - Určení asociací mezi třídami
  - Zrušení chybných a nepotřebných asociací
  - Určení atributů tříd
  - Zrušení chybných a nepotřebných atributů
  - Určení vazeb dědičnosti
  - Opakovaná kontrola a určení nedostatků



Obr. 13 Ukázka diagramu tříd v OMT (16)

- **Dynamický model** vyjadřuje dynamiku objektů a všechny jejich stavy. Určuje chování objektů při běhu programu, toky zpráv a kontrolu mezi objekty. Pro tento účel obsahuje stavové diagramy pro všechny třídy, diagramy interakcí tříd, celkovou mapu událostí a diagram událostí. Jeho tvorba obsahuje tyto kroky. (16)
  - Určení use CASEs, příprava scénáře klasických interakčních sekvencí
  - Určení událostí mezi objekty a příprava mapy událostí ke každému scénáři
  - Tvorba diagramu událostí systému
  - Tvorba stavových diagramů v rámci třídy s dynamickým chováním
  - Kontrola konzistence a úplnosti událostí mezi stavovými diagramy

- **Funkční model** popisuje funkční souvislosti systému. Je z části stejný jako diagram datových toků. Stanovuje to, co systém dělá, nezabývá se však tím, jak. Tvorba funkčního modelu obsahuje tyto kroky. (16)
  - Určení vstupních a výstupních hodnot
  - Tvorba diagramů datových toků v rámci definice funkčních závislostí
  - Popis každé funkce
  - Určení omezení
  - Specifikace optimalizačních kritérií

Jednotlivé modely nastiňují vždy pouze některé vlastnosti systému. Obsahují totiž vztahy se zbývajících modely. Díky provázání jejich vztahů je metodika OMT velmi kompaktní. Objektový model ilustruje objektovou strukturu, na ní jsou napojeny operace z dynamického a funkčního modelu. Dynamický obsahuje akce závislé na hodnotách atributů v objektech. Akce mění hodnoty objektů a volají funkce. Funkce volané metodami a akcemi v objektovém a dynamickém modelu jsou vyjádřeny ve funkčním modelu. Funkce pracují s hodnotami dat z objektového modelu. (19, str. 78)

Mezi další diagramy OMT jsou model jednání a slovní scénáře. Definují požadavky, specifikují chování a určují hranice systému. Jde o verbální vyjádření scénáře v rámci každého use CASE. Primárně se sepíše seznam událostí, které se v daném use CASE probíhají. Události se číslovají a řadí podle toho, jak jdou za sebou. Paralelní události se číslovají ve více úrovních. Seznam je dobré přenést do tabulky, v níž se každé události přiřadí vstupy a výstupy. Taktéž je nutno uvést inicializační podmínku, která use CASE spouští, důsledky scénáře a možné chybové stavy. (7)

Dalším zásadním diagramem je diagram datových toků. Je založen na hierarchické struktuře. Nejvýš v hierarchii diagramů datových toků je diagram

kontextový. Obsahuje pouze aktéry, systém definovaný jediným procesem a datové toky mezi systémem a aktéry. Pod kontextovým diagramem je diagram nulté úrovně, v němž se systém rozkládá na několik procesů. Před jménem každého procesu v nulté úrovni je číslo. Další úrovně diagramů navazují na diagram nulté úrovně jako větve ve stromové struktuře. Pro označování procesů v nich obsažených se používá čísel s tečkovou notací. (16)

Mapa událostí, jejímž cílem je seznam všech událostí zpracovávaných či odesílaných třídami, je další z komponent. Podobá se objektovému modelu, obsahuje však jen třídy z diagramu interakcí. (3, str. 368)

Diagram událostí je popisem funkcionalit systému jako reakcí na vstupy. Jeho cílem je identifikace všech aktivit, které následně zakomponuje do orientovaného grafu. Aktivita jakožto uzly a událostičko orientované hrany. Tomuto diagramu náleží pouze jeden vstup a jeden výstup. Popis systému je v něm vyjádřen bez ohledu na objektovou strukturu, nebere v potaz ani hranice systému a popisuje i aktivity uživatele. Události se v diagramu mohou větvit a opět slévat. Činnost tímto vyjádřená se však v diagramech uvádět nemusí, neb má výhradně technický charakter. Větvením tedy vznikají paralelní události, které spolu mohou mít podmíněný vztah. Sled událostí lze na sebe vázat hierarchicky, což lze využít ve složitějších systémech. (7, str. 174-178)

V rámci systémového designu je vhodné provést následující rozhodnutí.

- Jak bude systém organizován do podsystémů?
- Jak bude určen souběžnost?
- Jak bude systém alokován v rámci procesorů a úloh?
- Jaká bude vybrána strategie pro implementaci datových úložišť?
- Jaké budou globální zdroje a mechanismus pro jejich zpřístupnění?
- Jaký bude přístup k implementaci řízení v programu
- Jaké budou okolní podmínky systému?

- Jaké budou priority zpracování?
- Jaký bude mezní provoz při startu, ukončení, či pádu systému? (12, str. 141)

Při objektovém designu je postup následující.

- Získání metod pro objektový model skrze ostatní modely
  - Nalezení operace pro procesy definované ve funkčním modelu
  - Určení operací pro události v dynamickém modelu v souvislosti s implementací řízení v programu
- Navržení algoritmů pro implementaci operací
  - Výběr algoritmů, minimalizujících náklady na implementaci operací
  - Výběr datových struktur příslušných algoritmů
  - Definice nových tříd a operací v rámci potřeby
  - Přiřazení odpovědností operacím s nejasným spojením s danou třídou.
- Optimalizace přístupu k datům
  - Přidání redundantních vztahů v rámci minimalizace nákladů na přístup k datům a maximalizace pohodlnosti přístupu
  - Uložení odvozených hodnot pro vyhnutí se počítání nových výpočtů
- Implementace řízení v programu dle postupu zvoleného v systémovém designu
- Přizpůsobení struktury tříd pro možnosti rozšíření dědičnosti
  - Přizpůsobení a přeuspořádání tříd a operací
  - Určení společného chování seskupených tříd
- Návrh implementace asociací
- Určení reprezentace atributů objektů
- Uspořádání tříd a asociací do modulů (16)

## 2.7 Jazyk UML

Jazyk UML je univerzální jazyk pro vizuální modelování systémů. Přestože je nejčastěji spojovaný s modelováním objektově orientovaných softwarových systémů, má mnohem širší využití, což vyplývá z jeho zabudovaných rozšiřovacích mechanismů. UML byl navržen, aby spojil nejlepší existující postupy modelovacích technik a softwarového inženýrství. Jako takový je explicitně navržen takovým způsobem, aby jej mohly implementovat nástroje CASE. Tato koncepce vychází z toho, že se rozsáhlé softwarové systémy většinou bez podpory nástrojů CASE neobejdou. Je nesmírně důležité si uvědomit, že jazyk UML nenabízí žádný druh metodiky modelování. Je však možno určité aspekty metodiky najít v každém z prvků, z nichž se model UML skládá. Samotný jazyk UML však poskytuje pouze vizuální syntaxi, kterou je možno využít při sestavování svých modelů. UML není vázán na žádnou specifickou metodiku nebo životní cyklus. Lze jej použít společně se všemi existujícími metodami. (3, str. 28-29)

Unifikace jazyka UML nevychází pouze z historických měřítek. Jazyk UML se snaží o unifikaci různých domén.

- **Vývojový cyklus.** UML nabízí vizuální syntaxi pro modelování během celého vývojového cyklu softwarového projektu – požadavky na analýzu počínaje a implementací konče.
- **Aplikační domény.** UML byl vytvořen jako nástroj pro modelování čehokoliv systému zasazenými do reálného času počínaje a podpůrnými systémy rozhodování konče.
- **Implementační jazyky a platformy.** UML je nezávislý na jakémkoliv programovacím jazyce a na jakékoliv platformě. Má podporu čistě objektově orientovaných programovacích jazyků, jako jsou java nebo



C#. Je ovšem velmi efektivní rovněž pro hybridní objektově orientované jazyky, jako je C++, i pro jazyky založené na objektech, například Visual Basic. Používá se rovněž pro modelování projektů vedených v neobjektových jazycích, například v jazyce C.

- **Vývojové procesy.** Díky univerzálnosti je UML možné použít v široké škále metodik.
- **Vlastní interní pojmy.** UML se o vnitřní jednotu a konzistenci snaží prostřednictvím malé množiny interních pojmů. Je pravda, že v této oblasti není vždy úspěšný. Lze však říci, že každý další pokus je mnohem dokonalejší než jeho předchůdci.

### 2.7.1 Objekty a UML

Základním předpokladem UML je skutečnost, že umožňuje modelování softwaru, stejně jako dalších systémů jako kolekce spolupracujících objektů. Přestože tato představa zřejmě zapadá do modelu objektově orientovaných softwarových systémů a programovacích jazyků, funguje stejně spolehlivě v obchodních a podnikatelských procesech a dalších aplikacích.

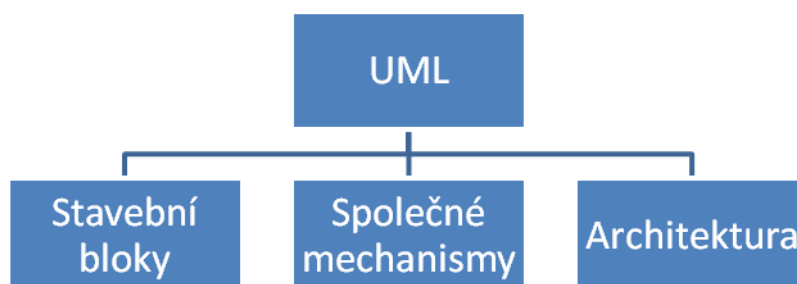
### 2.7.2 Aspekty modelu UML

- **Statická struktura** popisuje typy objektů pro modelování daného systému důležité, a také jak spolu tyto objekty souvisejí.
- **Dynamické chování** popisuje životní cyklus zmiňovaných objektů a způsob jejich vzájemné spolupráce s cílem dosáhnout požadované funkce navrhovaného systému. (21)

Oba výše zmíněné aspekty se v UML navzájem doplňují a jeden bez druhého není úplný.

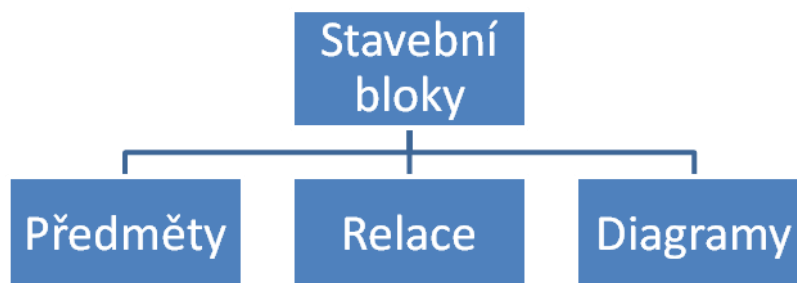
### 2.7.3 Struktura UML

Struktura UML se sestává ze stavebních bloků, společných mechanismů a architektury, které jsou popisovány dále.



*Obr 14. Struktura UML*

Stavební bloky jsou základními prvky modelu, jsou vyjádřeny předměty, relacemi a diagramy.



*Obr 15. Stavební Bloky UML*

### 2.7.3.1 Předměty

Předměty jsou samostatné prvky modelu. V UML jsou rozděleny do čtyř tříd.

- **Strukturní abstrakce** jsou podstatná jména modelu UML, jako jsou třídy, rozhraní, spolupráce, případ užití, aktivní třída, komponenta, či uzel.
- **Chování** jsou slovesa UML, patří do nich interakce a stavy.
- **Seskupení** jsou balíčky používané k seskupování významově souvisejících prvků modelu do soudržných jednotek
- **Poznámky** či anotace, které je možno k modelu připojit s úmyslem zachytit informaci sestavenou jen k tomuto účelu. (21)

### 2.7.3.2 Relace

Relace jsou pojítkem mezi předměty a vyjadřují vzájemnou souvislost předmětů. UML rozlišuje šest tříd relací.

- **Závislost** je relace, při které změna v určitém předmětu ovlivňuje význam závislého předmětu
- **Asociace** popisuje množinu spojení mezi objekty.
- **Agregace** určuje, že cílový prvek je součástí zdrojového prvku
- **Kompozice** je silnější forma agregace s více omezeními
- **Kontejner** stanovuje, že zdrojový prvek obsahuje cílový prvek
- **Zobecnění** znamená, že jeden prvek je specializací jiného prvku a lze jej nahradit prvkem univerzálnějším.
- **Realizace** je asociací mezi klasifikátory, kde jeden klasifikátor určuje dohodu, jejíž uskutečnění zaručuje druhý klasifikátor

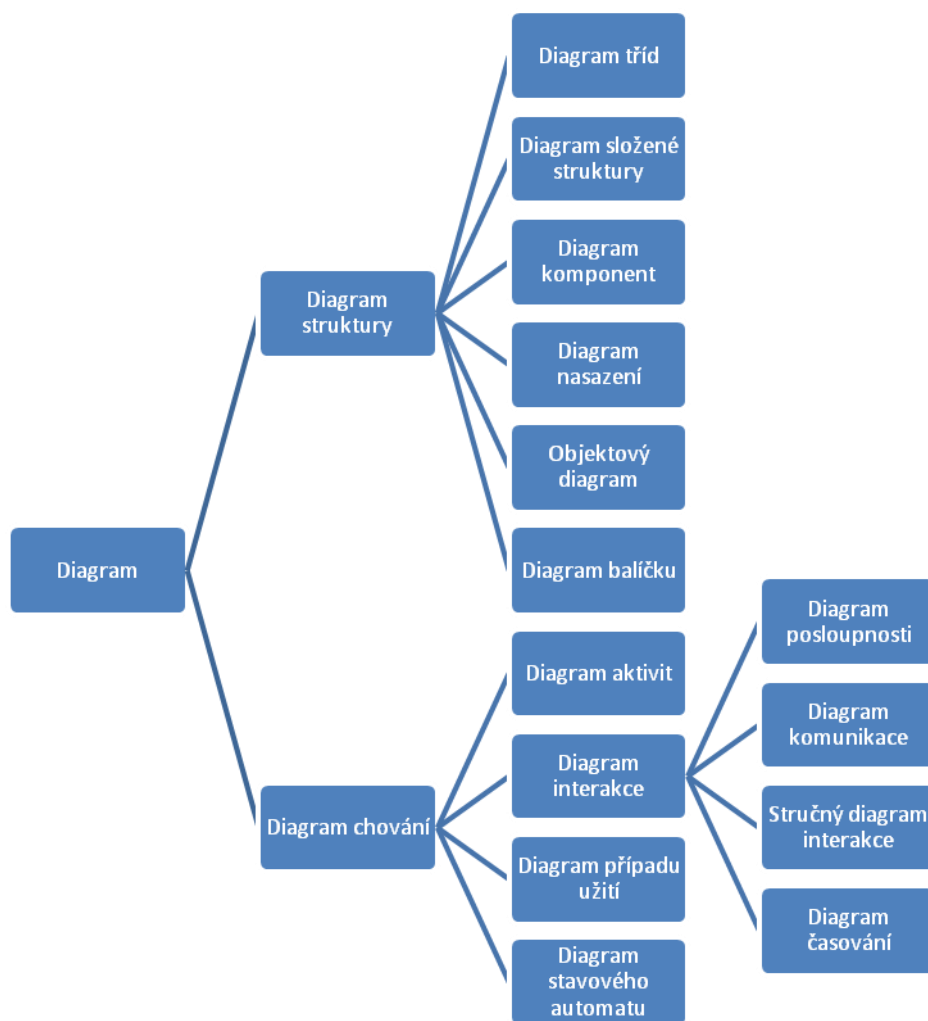
(21)

### 2.7.3.3 Diagramy

Ve všech nástrojích CASE založených na UML jsou každý nově vytvořený předmět nebo relace přidávány do vznikajícího modelu. Model je repositářem všech relací a předmětů vytvořených k tomu, aby popisovaly požadované chování softwarového systému, který je snaha navrhnout.

Diagramy jsou pohledy na model. V UML jich existuje celkem třináct různých druhů. Zmiňovanou skupinu diagramů lze rozdělit na ty, které modelují statickou strukturu a na ty, které modelují dynamickou strukturu. Statický model zachycuje předměty a strukturní asociace mezi nimi. Dynamický model pak zachycuje způsob, jakým na sebe jednotlivé předměty navzájem působí. (3, str. 36-37)

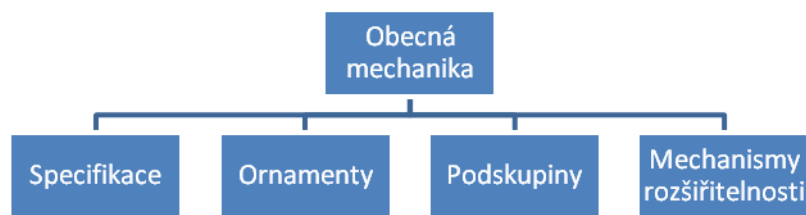
Neexistuje žádné pevně stanovené pořadí, ve kterém by se diagramy měly vytvářet. Většinou se však začíná diagramem use CASE, který definuje rozsah platnosti navrhovaného systému. Ve skutečnosti se většinou pracuje souběžně s několika různými diagramy zároveň. Každý z nich je dotvářen postupně, tak jak jsou odhalovány další detaily navrhovaného softwarového systému. Tím jsou diagramy nejen pohledem na model, ale rovněž prvořadým a základním mechanismem pro zadávání nových informací do existujícího modelu. (7)



Obr. 16 Diagramy v UML

#### 2.7.4 Společné mechanismy

Společné mechanismy jsou množinou obecných způsobů, kterými v UML dosáhneme určitého cíle. UML obsahuje čtyři společné mechanismy používané konzistentně v celém jazyku. Popisují čtyři strategie cesty k modelování objektů, jež jsou opakovaně používány v různých kontextech v rámci UML. K těmto strategiím patří specifikace, ornamenty, podskupiny a mechanismy rozšiřitelnosti. (14)



*Obr. 17 společné mechanismy UML*

#### 2.7.4.1 Specifikace

Modely UML mají alespoň dva rozměry. Grafický, který umožňuje vytvářet model prostřednictvím diagramů a symbolů, a textový, který se skládá ze specifikací různých prvků modelu. Specifikace jsou textovým popisem sémantiky jednotlivých prvků. Množina specifikací je jádrem modelu a utváří sémantický podklad, který udržuje celý model pohromadě a dává mu smysl. Různé diagramy jsou různými pohledy nebo obrazovými projekcemi tohoto podkladu. (3, str. 39)

Sémantický pohled je zpravidla udržován pomocí nástroje CASE, jenž obsahuje funkce pro zadávání, prohlížení a úpravy specifikace jednotlivých prvků modelu. (7)

UML má neobyčejnou dávku flexibility pro tvorbu modelů. V praxi se používají následující modely.

- **Proškrtané** vyjadřují, že určité prvky jsou sice v podkladu obsaženy, ale v daném diagramu jsou ukryty, většinou proto, aby byl pohled srozumitelnější
- **Neúplné** říkají, že určité prvky mohou v modelu chybět úplně
- **Nekonzistentní** určují, že model může obsahovat protimluvy (21)

Existence volných pravidel úplnosti a konzistence je důležitá, protože modely se postupně vyvíjejí a prodělávají mnoho změn. Přesto se klade důraz na tvorbu konzistentních modelů, které jsou natolik úplné, že umožňují tvorbu softwarového systému.

Při modelování v UML se většinou začíná tvorbou grafického modelu, který umožňuje vizualizaci systému. Později je k podkladu modelu přidávána sémantika. Má-li ovšem model být považován za užitečný nebo kompletní, musí být sémantika obsažena v podkladu modelu.

#### **2.7.4.2 Ornamenty**

Příjemnou vlastností UML je skutečnost, že každý prvek modelu je vytvářen jednoduchým symbolem, který lze obohatit řadou ornamentů, je-li prostřednictvím diagramu třeba zobrazit více informací. Tvorba velmi složitého modelu tedy může klidně začít pomocí několika základních symbolů s jedním nebo dvěma ornamenty. Takto je možno postupovat do té doby, dokud příslušný prvek neobsahuje dostatečný počet informací. (3, str. 41)

Ornamenty by měly být používány pouze v případě, kdy zvětšují srozumitelnost a čitelnost diagramu, případně tehdy, když zdůrazňují nějakou důležitou funkci modelu. V diagramu obvykle není třeba zobrazovat všechny podrobnosti. Mnohem důležitější je, aby diagram byl srozumitelný a snadno čitelný a znázorňoval cíle, jichž je nutno dosáhnout. (15)

#### **2.7.4.3 Podskupiny**

Podskupiny popisují různé způsoby vidění modelace světa v UML. Rozlišujeme dvě podskupiny.

- **Klasifikátory a instance.** Kde klasifikátor je abstraktním vyjádřením typu předmětu (bankovní účet) a instance je naproti tomu konkrétním

výskytem abstraktní představy (Janův bankovní účet). UML nabízí bohatou taxonomii třiceti tří klasifikátorů, pro ukázkou si můžeme uvést aktéra, třídu, komponentu, rozhraní, uzel, signál, případ užití, atd.

- **Rozhraní a implementace.** Při práci v UML vycházíme ze zásady oddělení toho, co předmět vykonává (rozhraní), od toho jak to vykonává (implementace). Rozhraní definuje dohodu, která zaručuje, čím se budou jednotlivé implementace řídit. Oddělení toho, co je slibováno, od skutečné implementace daného slibu je v UML zásadním pojmem. (3, str. 42-43)

#### 2.7.4.4 Mechanismy rozšiřitelnosti

Autoři UML si uvědomili, že návrh naprosto všeobjímajícího modelovacího jazyka, který by uspokojil potřeby všech uživatelů, je prostě nemožný. Z tohoto důvodu umístili do jazyka tři velice jednoduché mechanismy umožňující jeho rozšiřitelnost.

- **Omezení.** Omezující podmínka je textový řetězec uzavřený do složených závorek. Tento text specifikuje určitou podmínku či pravidlo týkající se prvku modelu, které musí být vyhodnoceno jako pravda. Lze říci, že tímto způsobem omezuje určité chování daného prvku. (21)
- **Stereotyp** zastupuje určité varianty v modelu existujícího prvku, který má sice stejnou podobu, ale používá se s jiným záměrem. Tím je pomocí stereotypů možno vytvářet nové prvky modelu založené na existujících prvcích. Název stereotypu stačí vložit do dvojitých lomených závorek a připojit k novému prvku. Každý model může obsahovat nanejvýš jeden stereotyp. (3, str. 153-175)

Jakýkoliv stereotyp může definovat sadu označených hodnot a omezujících podmínek, které platí pro stereotypní prvek. Ke stereotypu je možno přidružit rovněž samostatný symbol, barvu nebo texturu. Používání barev a textury však



není v UML doporučováno. Přesto je obvyklé, že je k novému stereotypu přiřazen nový symbol, což umožňuje rozšířit grafickou notaci řízeným způsobem.

Stereotypy tedy zavádějí nové prvky modelu s jiným významem, u nichž je nutno dále definovat jejich sémantiku. Dokumentaci sémantiky umožňují buď nástroje CASE, případně je možno vytvářet dokumentaci separátně a odkazovat se do ní.

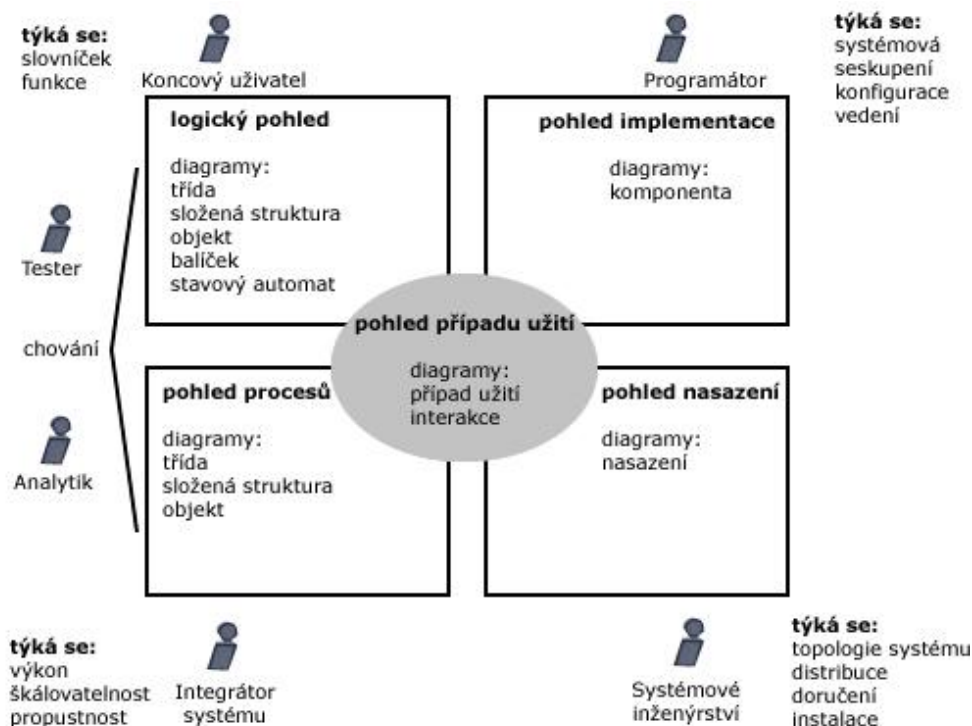
- **Označené hodnoty** umožňují rozšiřovat prvky o jejich vlastní hodnoty. Za vlastnost je považována jakákoliv hodnota připojená k prvku modelu. Většina prvků má mnoho předdefinovaných vlastností. Některé z nich je možno zobrazit v diagramech, jiné jsou zase součástí sémantického podkladu příslušného modelu. (21)

UML umožňuje rozšiřovat vlastnosti prvků pomocí označených hodnot. Označená hodnota je klíčové slovo s přidruženou hodnotou. Seznam hodnot prvku je množina čárkami oddělených dvojic značka=hodnota uzavřený do složených závorek.

### 2.7.5 Architektura

Architektura je organizační struktura systému včetně jeho rozkladu na součásti, jeho propojitelnosti, interakce, mechanismů a směrných zásad, která proniká do návrhu systému. Zachycuje strategické aspekty vyšší struktury systému. Na architekturu lze nahlížet mnoha způsoby, nejčastější členění je však dle pohledu 4+1, které ve své knize Kruchten 2 popisuje Phillipe Kruchten. Aby bylo možno zachytit všechny podstatné aspekty architektury daného systému, definuje UML čtyři různé pohledy na systém. Pohled logický, pohled procesů, pohled implementace a pohled nasazení. Všechny zmiňované pohledy jsou poté integrovány do pohledu případů užití. (3, str. 34-46)

- **Logický pohled** zachycuje slovník oblasti problému jako množinu tříd a objektů. Důraz je přitom kladen především na zobrazení způsobu, jakým objekty a třídy tvořící základ systému implementují jeho chování.
- **Pohled procesů** modeluje spustitelná vlákna a procesy jako aktivní třídy. Je to procesově orientovaná varianta logického pohledu, která obsahuje stejné artefakty.
- **Pohled implementace** modeluje soubory a komponenty, které utvářejí hotový kód systému. Slouží jednak ke znázornění závislostí mezi jednotlivými komponentami a také ke správě konfigurace množin vytvořených z těchto komponent. Umožňuje též definici verze systému.
- **Pohled nasazení** modeluje fyzické nasazení komponent na množinu fyzických výpočetních uzlů. Umožňuje modelování distribuce komponent na příslušné uzly distribuovaného systému.
- **Pohled případů užití** zachycuje základní požadavky kladené na příslušný systém jako množinu případů užití a utváří tak základ tvorby všech dalších pohledů. (21; 3, str. 46-48)



Obr. 18 Architektura UML

## 2.8 Anketa o počítačové podpoře projektového řízení

V průběhu dubna a května 2008 proběhl v rámci IMPA zatím ojedinělý průzkum. Účelem tohoto průzkumu bylo zmapovat prostřednictvím ankety způsob, jakým členové IMPA využívají informatiku k podpoře projektového řízení. Anketa byla přístupná online.

Průzkum uskutečnili členové odborné sekce Projekty ICT, přičemž prací v realizačním týmu se nejvíce účastnili pánové Müller, Vlach, Verner a Krebs. Celkem bylo zpracováno přes třicet odpovědí respondentů. Zastoupeny byly nejvíce komerční společnosti, dále potom neziskové organizace a jednotlivci zabývající se projektovým řízením. Respondenti byli z organizací všech velikostí: od malých, až po velké (více než 500 zaměstnanců).

### 2.8.1 Výsledky dotazníkového šetření

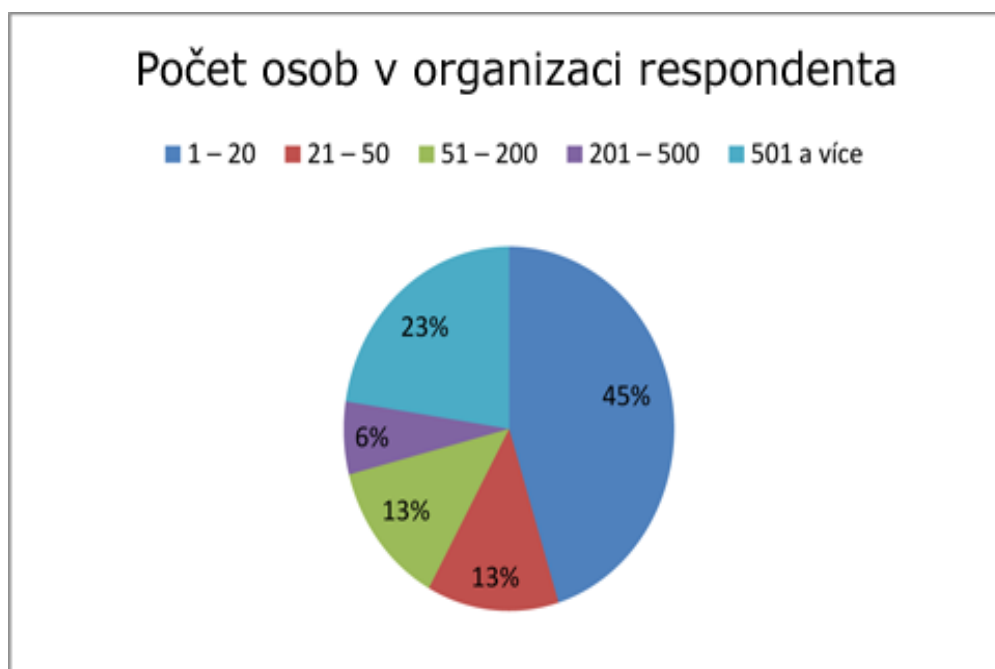
Graf 1



Zdroj: výzkum IMPA

Tento graf ilustruje rozložení respondentů v rámci jejich typu. Dvacet z respondentů je členy komerční společnosti, sedm se jich zabývá projektovým řízením individuálně a dva patří k neziskové organizaci.

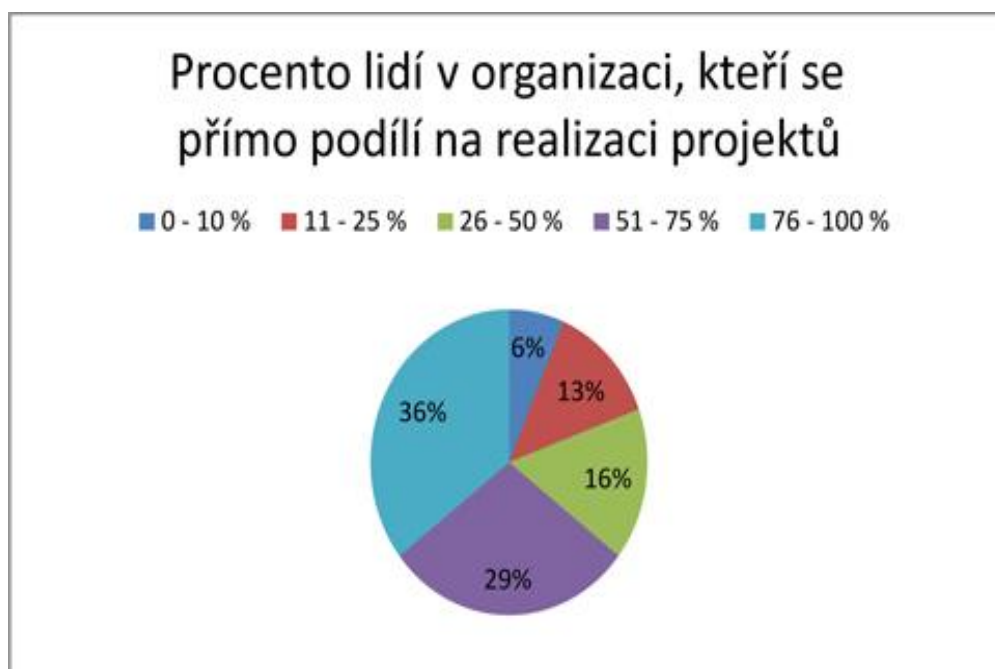
**Graf 2**



Zdroj: výzkum IMPA

Zde je zobrazena mohutnost domovské organizace, ze které respondent pochází. Třináct respondentů je z organizace čítající pouze 1-20 osob. Sedm respondentů je naopak z organizace čítající přes 500 osob. Po čtyřech respondentech jsou zastoupeny organizace čítající 21-50 a 51-200 osob. Poslední skupinu tvoří dva respondenti ze společností čítající 201-500 osob.

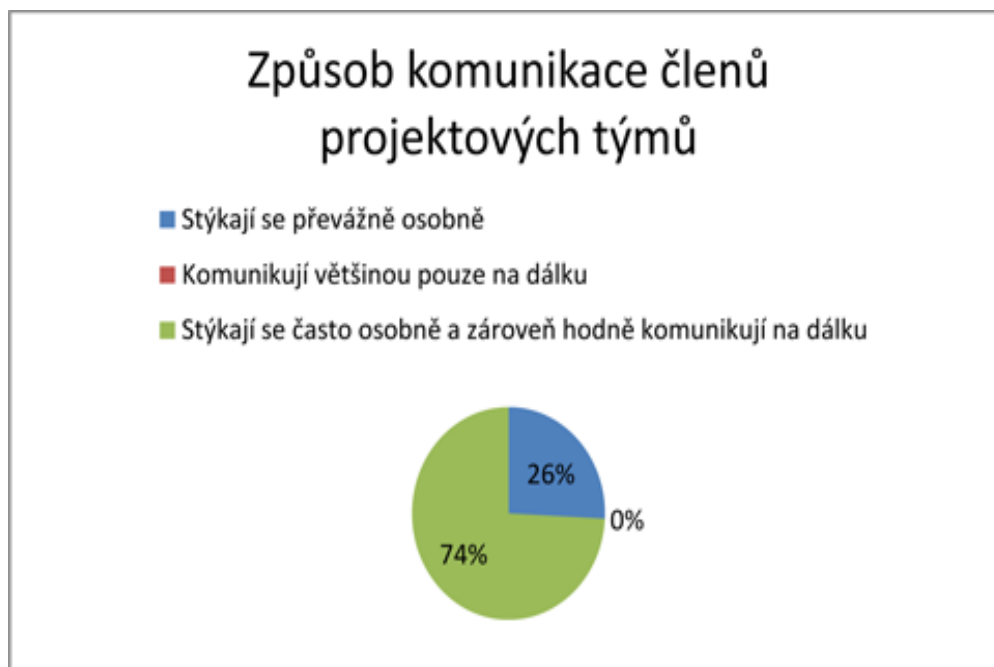
**Graf 3**



Zdroj: výzkum IMPA

Zde je nastíněno, jaké procento lidí se v organizacích respondentů podílí na realizaci projektů. Dvanáct respondentů uvádí účast osob v organizaci na realizaci projektů 76-100%. Devět respondentů uvádí 51-75% účast. Pět respondentů uvádí účast mezi 16-50%. Tři mezi 11-25% a jeden mezi 0-10%.

**Graf 4**



Zdroj: výzkum IMPA

Osm z respondentů tvrdí, že členové projektového týmu se stýkají převážně osobně. Dvacet dva respondentů tvrdí, že se členové týmu setkávají často osobně a zároveň hodně komunikují na dálku. Tým, který by spolu komunikoval většinou pouze na dálku, se v řadách respondentů nevyskytoval.

**Graf 5**



Zdroj: výzkum IMPA

Tento graf ilustruje zahraniční účast v projektech. Šestnáct ze třiceti respondentů se k ní vyjádřilo kladně.



**Graf 6**



Zdroj: výzkum IMPA

K využívání formální metodiky pro řízení projektů se hlásí dvacet tři z třiceti respondentů.

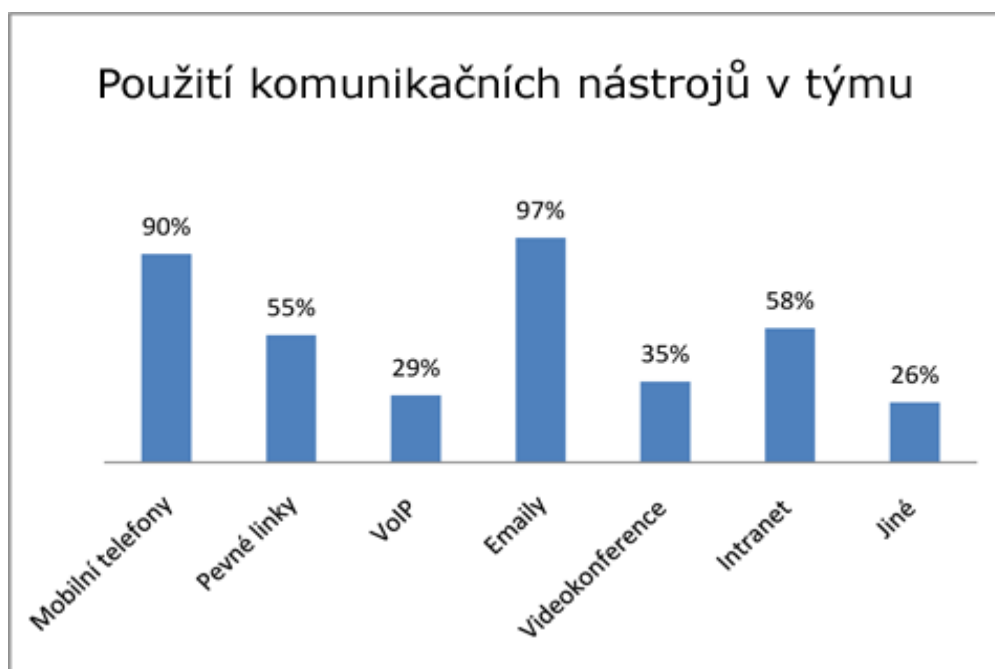
**Graf 7**



Zdroj: výzkum IMPA

Tento graf vypovídá o tom, že organizace dvaceti sedmi respondentů využívají vyhodnocování úspěšnosti projektů.

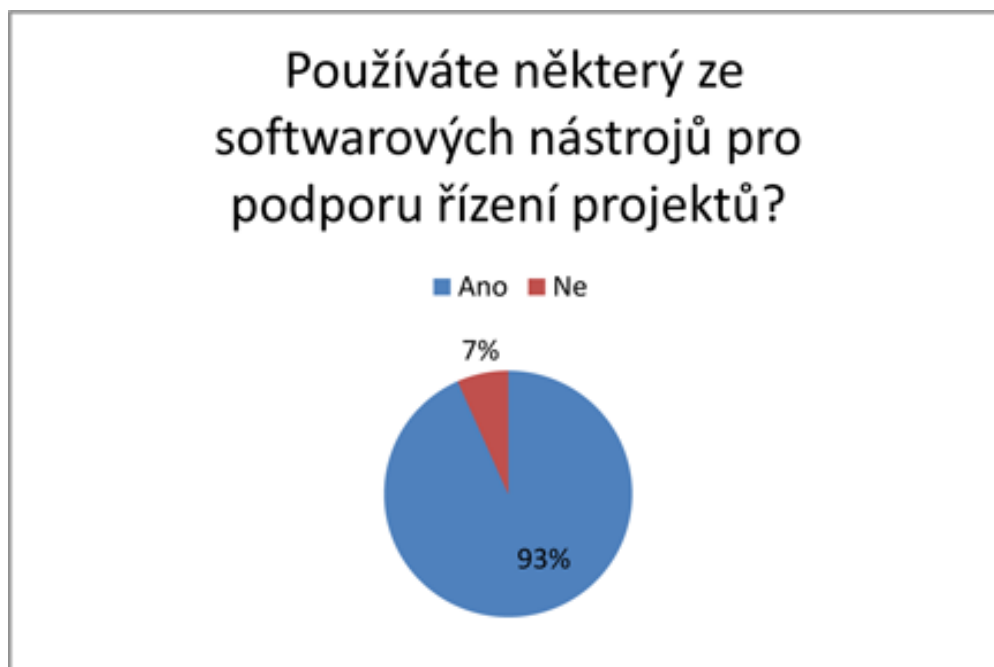
**Graf 8**



Zdroj: výzkum IMPA

Zde je výstup průzkumu komunikačních nástrojů, z něho je patrné největší rozšíření komunikace skrze email a mobilní telefon. V závěsu je i použití intranetových systémů.

**Graf 9**



Zdroj: výzkum IMPA

Všichni kromě jednoho respondenta uvádějí, že k řízení používají některý softwarový nástroj pro jeho podporu.

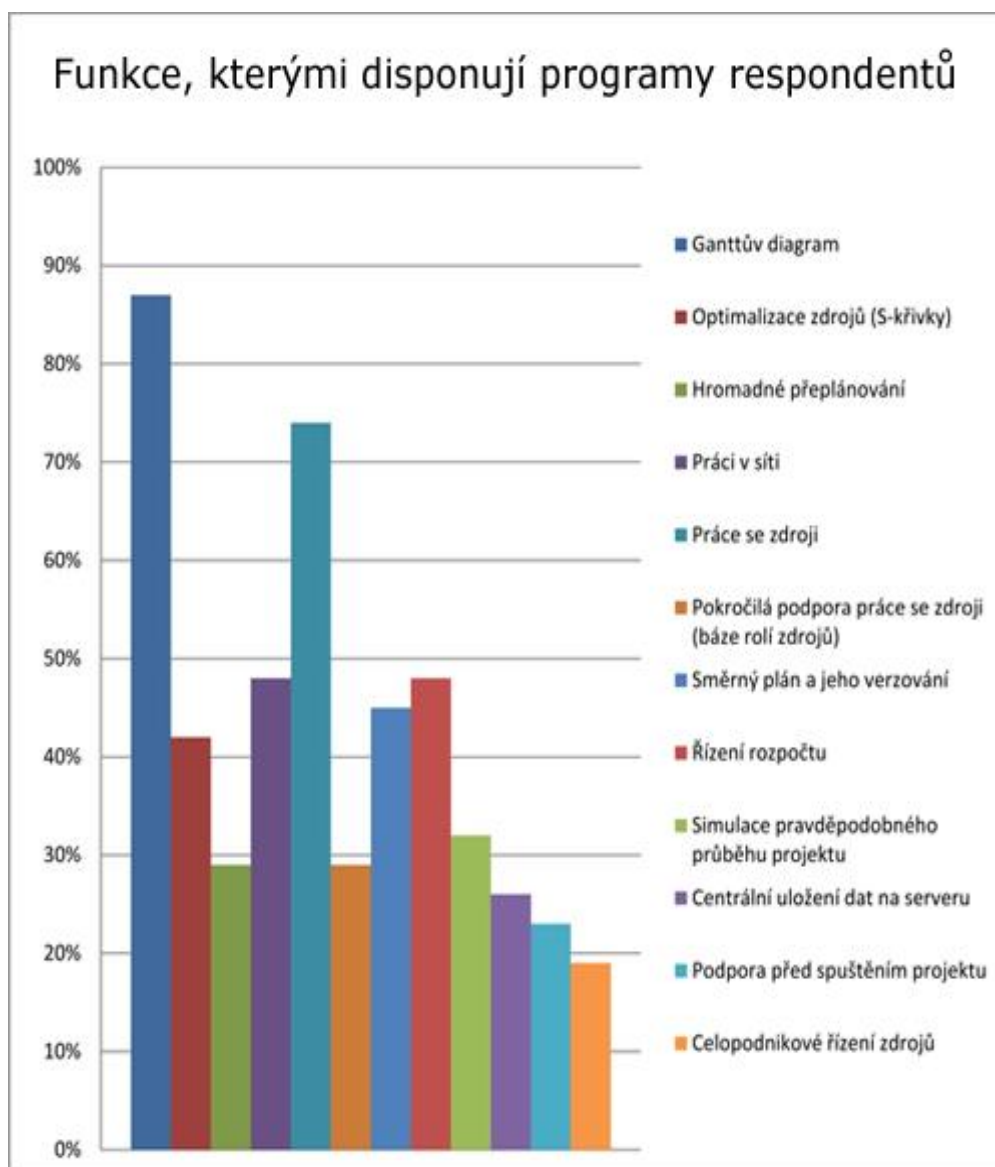
**Graf 10**



Zdroj: výzkum IMPA

V tomto směru průzkum ukazuje na nejrozšířenější softwarový produkt MS Project, daleko za ním pak následuje aplikace Primavera. Ostatní uváděné produkty se používají minoritně.

**Graf 11**



Zdroj: výzkum IMPA

Tento graf uvádí funkce, které jsou k dispozici v softwaru pro řízení projektu, který mají respondenti k dispozici. Jak je vidno nejpoužívanější funkcí je Ganttův diagram (87%), za ním následuje práce se zdroji (74%). Mezi

40-50% pak dominuje řízení rozpočtu, práce v síti, směrný plán a optimalizace zdrojů.

**Graf 12**



Zdroj: výzkum IMPA

Patnáct respondentů uvádí, že software, který využívají pro řízení, uspokojuje kompletně jejich potřeby. Třináct respondentů vyjádřilo nespokojenost s pokrytím potřeb.

**Graf 13**



Zdroj: výzkum IMPA

Na otázku, zda je softwarový nástroj pro podporu řízení projektů integrován s dalšími systémy odpovědělo kladně dvanáct respondentů, stejný počet odpověděl záporně.



### **3 Metodický postup**

Před analýzou v rámci tvorby softwarového systému pro řízení, který má být výstupem této práce, bylo nutné provést sběr dat. Soubor dat, který byl postoupen analýze v rámci literární rešerše, obsahoval nástroje a techniky z jednotlivých oblastí poznatků v řízení projektů, nástroje a techniky jednotlivých metodik, úvod do UML a anketu o počítačové podpoře projektového řízení z roku 2008 od mezinárodní organizace IPMA.

Po stanovení komponent na základě analýzy jsem započal vývoj modulární aplikace ve vývojovém prostředí PSPad. Při vývoji jsem využil své dosavadní poznatky z oblasti XHTML, PHP, CSS a javascriptu. Jako datový sklad aplikace jsem použil databázi MySQL.

Při návrhu aplikace jsem se nejdříve zabýval strukturou databáze, jednak s ohledem na optimalizaci datových toků v rámci rekurzivních dotazů, které jsou v aplikaci používány, a pak také samozřejmě vzhledem k vytýčeným výstupním datům v souladu s funkcionalitami, které měla aplikace obsahovat.

Po návrhu databáze, jsem vytvořil mainframe rozhraní aplikace, zajišťující připojení k databázi, autentifikaci uživatele podle přístupového jména a hesla. Na mainframe jsem začal dále vázat moduly pro správu uživatelů systému, správu lidských zdrojů, správu projektů a statistiky.

#### **3.1 Specifikace softwarového systému pro řízení**

Konstrukce aplikace PlanIT vychází ze zadání bakalářské práce a nabízí i mnohé další možnosti. PlanIT je multiplatformní řešení, při kterém nezáleží na tom, jaký operační systém uživatel používá. Aplikace je přístupná přes webový prohlížeč. Další specifickou doménou tohoto softwaru je možnost sdílení dat a to jak v rámci jednotlivých projektů, tak i jednotlivých úloh v Ganttově diagramu. Díky tomu, se vytváří nový rozměr pro sdílení

dokumentace a jiných souborů souvisejících s projektem. Velký důraz je kladen na ergonomii a intuitivní ovládání

### 3.1.1 Správa lidských zdrojů

Modul pro správu lidských zdrojů v projektu umožňuje editaci, mazání a tvorbu nových zaměstnanců, týmů, oddělení a firem. Dané položky jsou uspořádány ve stromové struktuře, pro následnou jednodušší manipulaci ve vytváření projektových týmů přímo u činností v Ganttově diagramu. Stromovou strukturu můžu intuitivně modelovat pomocí metod drag-and-drop.

- **Firma** obsahuje informace o počtu vnořených oddělení, týmů a pracovníků. Ze sumy hodinové mzdy pracovníků určuje, kolik stojí hodina této jednotky. Další informace je pak název, adresa, kontakt na zodpovědného pracovníka a popis, kam si personalisté mohou psát poznámky.
- **Oddělení** obsahuje informace o svém obsahu stejně jako firma, dále pak název a kontakt na zodpovědnou osobu.
- **Tým** obsahuje totéž co oddělení
- **Zaměstnanec** je položka obsahující stručné informace o zaměstnanci. Konkrétně jméno, příjmení, profese, adresa, telefon, email, hodinová mzda, foto a popis.

Většina společností s podnikovým intranetem, má již personalistickou databázi vytvořenou. Proto je vhodné tento modul provázat přímo na ni.

### 3.1.2 Správa projektů

Modul pro správu projektů je pro aplikaci stěžejním. Zajišťuje tvorbu Ganttova diagramu, přiřazování lidských zdrojů, předběžnou kalkulaci peněz, sdílení souborů a komunikaci.

- **Ganttův diagram** je efektivním nástrojem pro řízení času. Jeho ovládání je implementováno jako menu na horní liště. Ovládání umožňuje vytvořit nový projekt, vložit novou úlohu, listovat stávajícími projekty, posun zpět a vpřed po časové ose, vyhledávání úloh, přiblížení a oddálení časové osy, výpis sdílených souborů v projektu a tvorbu nového milníku. Všechny dialogy, jsou pro pohodlnější práci řešeny jako plovoucí okna. Pod horní lištou se nachází časová osa, která se mění s přiblížením. Pod časovou osou se nachází jednotlivé úlohy, vyjádřené dle trvání různě širokým obdélníkem s popiskem. Jeho šířku (dobu trvání) lze měnit pomocí metod drag-and-drop s tím, že se všechny činnosti, které mají tuto činnost jako rodiče, automaticky posouvají s ní. V obdélníku je dále uvedena délka trvání a náklady. Náklady jsou určeny kalkulací mzdy jednotlivých členů přiřazeného týmu a fixními náklady. U každé činnosti je znázorněn stav dokončení a to jak v procentech, tak graficky. Vztahy mezi činnostmi jsou vyjádřeny šipkami. Dole v projektu jsou zobrazovány milníky tyčící se nahoru k časové ose.
- **Úloha** je časově omezený celek, který může být vázaný na předchozí úlohy. Zodpovídá za ní garant, což může být projektový manažer vedoucí svůj tým. Má určené fixní náklady a rozpočet. Každá úloha má svůj reporting, v něm je možno zadávat stav postupu prací v procentech a psát krátké zprávy v rámci o vývoji dané úlohy. V každé úloze je možno vytvářet tým, který na ní bude pracovat. Při tvorbě týmu se nám ukáže známá stromová struktura z modulu správy lidských zdrojů, ze které můžeme metodou drag-and-drop přetahovat jednotlivé prvky

stromu. Poslední záložkou úloh je dokumentace, kde je prostor pro vzájemné sdílení materiálů souvisejících s úlohou.

- **Milník** značí událost k určitému datu a času. Předmětem milníku bývá většinou porada v rámci ukončení, či započetí úlohy. Pro detaily je v milníku uvedeno pole poznámky.
- **Šablony** jsou dokumenty vytvořené v sadě MS Office jako mustr pro jednotlivé nástroje, které se v řízení používají. Vyvolat se dají v dokumentaci k projektu, či úlohy. Šablony obsahují analýzu účastníků, analýzu ziskovosti, bod zvratu, chartu projektu, deník problémů, finanční analýzu, formulář požadavku změny, histogram zdrojů v projektu, matici pravděpodobnosti a dopadu, matici přidělení povinnosti, matici řízení očekávání, matici váženého rozhodování projektu, obchodní případ, odhad nákladů na projekt, organizační diagram, Paretův diagram, plán řízení komunikace, plán řízení projektu, plán zajišťování kvality, popis projektu, pracovní příkaz, předávací formulář, registr rizik, rozpis prací, seznam rizik, sledování rizik, stanovení rozsahu, týmovou smlouvu, vytvořenou hodnotu, žádost o návrh, zahajovací schůzku, závěrečnou dokumentaci, zprávu o milníku, zprávu o postupu prací a zprávu s poučením.

### 3.1.3 Statistika

Modul statistiky slouží k přehledu o stavu projektů. Podle trojúhelníku Harolda Kerznera jsou zde tři zásadní veličiny, ze kterých se vyvozuje optimální řízení projektu.

- **Sledování času** v rámci procentuálního reportingu plnění prací z jednotlivých úloh a procentuálnímu vyjádření stavu času k aktuálnímu datu vztaženému k době běhu celého projektu.

- **Sledování nákladů** je prováděno jak u celého projektu, tak u jednotlivých činností, které získáme po kliknutí na název projektu. Sleduje se, zda aktuální cena činnosti nepřesáhla rozpočet.
- **Sledování rozsahu** se děje pomocí dokumentace v projektu. U každého projektu i úlohy je výpis přidružených dokumentů. Směrný dokument pro sledování rozsahu je stanovení rozsahu. Rozsah se dá změnit např. formulářem požadavku změny. To už ale záleží na interní zavedené metodice.

V každém projektu je nutné mít také přehled o lidských zdrojích, proto je zde i výpis do projektu či úlohy začleněných pracovníků.

### 3.2 Diskuze

V této bakalářské práci jsem se snažil o zmapování metodologie řízení projektů v IT a vytvoření softwarového systému podle zadání vztaženému k zásadám uvedeným v odborné literatuře a na základě postřehů z ankety o počítačové podpoře projektového řízení.

Pro uvedení do problematiky byla nejdříve provedena analýza oblastí poznatků v projektovém řízení. Oblasti poznatků vyzdvihují a definují především Kathy Schwalbe a William R. Duncan. (1, 2) Každá oblast obsahuje nástroje a techniky, které ji umožňují řídit. Do stávajícího softwaru jsem se jich snažil zakomponovat co nejvíce, aby si každý potencionální uživatel mohl vybrat svou vlastní cestu. Při implementaci ve firmě by s největší pravděpodobností došlo ke změně šablon v rámci zavedené metodiky.

Následujících devět odstavců je věnováno odpovědím na výzkumné podotázky, které zaštiťuje otázka hlavní řešící optimální řízení projektu s ohledem na použité nástroje.

K řízení rozsahu se váže hned několik dokumentů, které jako optimální uvádí knihy od Steva McConnella a Aleny Svozilové. (4, 5) Patří mezi ně stanovení rozsahu projektu, rozpis prací, pracovní příkaz, plán řízení projektu a formulář požadavku změny. Pokud vezmeme v potaz definici, že řízení rozsahu projektu představuje definování a řízení všech prací, nutných k úspěšnému dokončení projektu, musíme připustit, že je částečně ilustrováno při tvorbě jednotlivých úloh Ganttova diagramu.

Řízení času má zajistit odhad doby potřebné k dokončení prací, návrh časového plánu jako takový a zajištění včasného dokončení projektu. Jeho realizace stojí na Ganttovu diagramu, ten vychází jako nejpoužívanější nástroj nejen z ankety o počítačové podpoře řízení projektů, stejně tak je na něj nahlíží Kathy Schwalbe a Milton D. Rosenau. (1, 6)

Řízení nákladů projektu obsahuje přípravu rozpočtu na projekt a jeho dynamické řízení. To je zajištěno jednak, kontrolou finančních toků implementovanou jak v modulu správy projektů, tak ve statistice. Pomocí kalkulace ceny jednotlivých úloh a hlídání rozpočtu. V šablonách jsou dále dokumenty o vytvořené hodnotě, odhad nákladů na projekt, analýzu ziskovosti, výpočet bodu zvratu, které jsou pro tento typ řízení v kombinaci s provedením v programu dle Tomáše Šubrta a Steva McConnella zásadní. (13, 4)

Řízení kvality stanovuje, že projekt naplní funkční předpoklady, pro které byl určen. Z dostupných nástrojů pro řízení kvality doporučovaných Jamesem Taylorem a Kathy Schwalbe. (8, 1) obsahuje software šablony pro Paretův diagram a plán zajišťování kvality. Kvalita je dána zákazníkem projektu a je většinou zahrnuta ve stanovení rozsahu.

Řízení lidských zdrojů určuje efektivní nasazení pracovních sil v rámci projektu. PlanIT řeší správu a přiřazování zdrojů v rámci modulů řízení projektů a řízení lidských zdrojů. Vytíženost je pak možno sledovat ve statistice. Podobné efektivní systémy jsou popisovány v knihách od Kathy

Schwalbe a Steva McConnella. (1, 4) Jako optimální šablony pro tento proces jsou pak v programu zakomponovány analýza účastníků, histogram zdrojů v projektu, matici přidělení povinností, organizační diagram a týmovou smlouvu.

Řízení komunikací je vytvářením shromažďování, distribuce a archivace informací o projektu. Komunikace se v PlanIT řídí zásadně v rámci aplikace a to možností sdílet mezi navzájem soubory, možností použití šablon. Konstrukce této funkcionality systému vyplývá z požadavku na maximální flexibilitu. Efektivní fungování systému řízení komunikací popisují Libor Gála, Jan Pour a Toman Prokop, z jejichž poznatků jsem při návrhu vycházel. (9)

Řízení rizik se skládá z rozpoznávání rizik, jejich analýzy a reakcí, které jsou riziky podmíněny. V aplikaci jsou pro řízení rizik zavedené základné šablony, které uvádí Alena Svozilová a Kathy Schwalbe. (5, 1) Patří mezi ně registr rizik, seznam rizik a sledování rizik.

Řízení obstarávání zaopatřuje zboží a služby z vnějších organizací. Dle Steva McConnella spočívá v analýze make-or-buy, smluv, žádosti o návrh nebo nabídku, výběru zdroje. (4)

Řízení integrace je průřezovou oblastí ovlivňující se navzájem se všemi ostatními oblastmi poznatků. Nástroje pro tuto oblast jsem vybíral dle doporučení Williama R. Duncana a Kathy Schwalbe. (2, 1) Mezi tyto nástroje patří charta projektu, plán řízení projektu, software pro řízení projektu, deník problémů, závěrečnou dokumentaci.

V rámci výsledků ankety o softwarové podpoře projektového řízení bych chtěl poukázat na určité faktory, které se v grafech objevují, a které jsem se snažil promítnout do koncepce aplikace. V grafu číslo 4 je uváděno, že nikdo z dotazovaných většinou nekomunikuje pouze na dálku, přesto v následujícím grafu je uváděno, že více než polovina projektů má zahraniční účast. Proto jsem se rozhodl vyvíjet aplikaci tak, aby byla dostupná všude

v rámci internetu přes webový prohlížeč, a aby objímala všechny aspekty komunikace v rámci tvorby projektu. Graf číslo 6 ilustruje, zda je v organizaci využívána nějaká formální metodika. Aplikaci jsem se snažil vyvíjet univerzálně, aby na ní šla navázat jakákoliv formální metodika, jejíž principy většinou tkví pouze v rozšíření dokumentace projektu, pročez je vyhrazen prostor prostřednictvím sdílení souborů. Graf číslo 7 vypovídá o tom, zda je v organizacích prováděno vyhodnocování projektu, v aplikaci je základ vyhodnocování položen v modulu statistika. Graf číslo 8 ukazuje využití komunikačních prostředků, poznatky byly využity v modulu pro správu lidských zdrojů. Grafy 9 a 10 vypovídají o tom, že většina organizací k řízení používá software pro podporu projektového řízení. Nejvíce zastoupen je MS Project. Tyto údaje však mohou být zkreslující. Výzkum pořádala organizace IMPA a na otázky odpovídali manažeři, kteří prošli jejím školením, ve kterém je pro řízení projektů upřednostňován především MS Project, ačkoliv trh nabízí mnohé další alternativy. Graf 11 ukazuje nejrozšířenější nástroje v softwaru pro řízení projektu, dle něj bylo taktéž přihlíženo k funkcionalitám. Graf číslo 12 ilustruje spokojenost uživatelů s funkcionalitou softwaru, spokojená je pouze polovina respondentů. Což svědčí o prostoru nepokrytých potřeb, které se v této oblasti ještě nacházejí. Poslední graf ukazuje tendenci propojovat systém řízení s firemním intranetem, či jinými nástroji používanými ve firmě. Což by v rámci kupříkladu správy lidských zdrojů, či připojení systému na elektronickou docházku neměl být problém.



## 4 Závěr

Metodologie řízení projektů je složitý proces, který vyžaduje velké množství znalostí a dovedností. V dnešní době není možno vždy vytvářet a řídit projekty jen intuitivním způsobem. Je potvrzeno praxí, že člověk nemá pro tuto problematiku dostatečné přirozené schopnosti. Pro zvýšení možnosti úspěchu je vhodné, aby projektový tým používal nějakou metodiku, která definuje principy, postupy a pravidla, které mají vliv na organizaci a efektivitu práce. Vývoj se musí v projektu koncentrovat na všechny aktivity potřebné pro dodání kvalitního produktu, či služby zákazníkovi. Na druhou stranu by proces vývoje neměl být zbytečně složitý.

Cílem této bakalářské práce bylo na základě literárního rozboru a výsledků ankety stanovit zásadní principy řízení projektu, které byly následně zakomponovány jako funkcionality do nové aplikace pro správu a řízení projektů PlanIT. Nejprve jsme pro všeobecný přehled vyšli ze stručného pojednání o historii a perspektivách metodologie v IT, kde jsme si nastínili rozdělení metodik do 3 kategorií: „strukturované, objektové a agilní“.

Než jsme se začali věnovat samotnému řízení, bylo nutno vymezit pojmy, které s metodologií řízení projektů v IT úzce souvisí. Kromě různých životních cyklů vývoje systému uvádím i seznam oblastí poznatků s nejpoužívanějšími nástroji a technikami pro řízení. Díky tomu jsem vyloučil v rámci aplikace efektivní použití modelu extrémního programování a modelu skrumáže, neboť v rámci jejich konstrukce se čas a náklady řídí velmi obtížně. Pro ostatní modely životních cyklů vývoje systému lze aplikaci použít bez problémů.

Řízením jsem se zabýval v rovině oblastí poznatků, ve všech devíti oblastech jsem analyzoval základní procesy a jejich výstupy. Získaná fakta jsem se posléze snažil zakomponovat do koncepce aplikace PlanIT stojící na pilířích trojúhelníku řízení projektů Harolda Kerzenra. Trojúhelník určuje

řízení zdrojů vztažené k času, nákladům a rozsahu. To, jak jsou zdroje řízeny, je určeno danou metodikou.

Další část se věnuje stručnému popisu 4 metodik používaných pro vývoj softwaru. Byla snaha o to aplikaci PlanIT tvořit tak, aby byla použitelná s co nejširším spektrem metodik. Díky tomu, že zmíněné metodiky většinou pouze rozšiřují obecnou metodologii řízení projektů, je jejich implementace řešená v rámci možnosti sdílení dokumentů, které jsou s metodikou spjaté.

Poslední část literární rešerše se věnovala stručné definici UML, jakožto nejrozšířenějšímu jazyku pro objektové modelování. Je v ní popsána jeho základní struktura a vyzdvihnuty nástroje CASE usnadňující jeho tvorbu. Jejichž výstupy je skrze aplikaci též možno sdílet.

Ačkoliv se objevují stále nové metodiky, je nutné si uvědomit, že je každý projekt specifický. Vyvstává tedy nutnost výběru optimální metodiky a většinou ještě její konfigurace. Ani sebelepší metodika přizpůsobená konkrétnímu projektu nemusí být zárukou úspěchu. Může však fungovat jako dobrý návod jak předcházet případným problémům. Z povahy všech aspektů reálného světa přesto nemůže garantovat úspěšné dokončení projektu. Bezpochyby je třeba stále mít na mysli zásadní faktory, které není možné formálně ošetřit. Patří mezi ně kupříkladu lidský faktor přinášející nejrůznější formy selhání. Z opačného pohledu vidíme skutečnosti, které je možné spolehlivě řídit a ošetřovat v rámci jasné komunikace, formální zprávy požadavků, předcházení rizikům, reálnému odhadu složitosti systému, důslednému testování, kontroly změn a dalších. Pokud jsou integrovány v přehledné formě v softwaru pro řízení, stávají se více přehlednými a výsledné řízení projektu nabývá na efektivitě.

## Reference

- [1] SCHWALBE, Kathy. *Řízení projektů v IT : Kompletní průvodce*. 1. vyd. Brno : Computer Press, 2007. 720 s. ISBN 978-80-251-1526-8.
- [2] WILLIAM R. , Duncan. *PMBOK*. USA : Project Management Institute, 1996. 45 s.
- [3] ARLOW, Jim, NEUSTADT, Ila. *UML2 a unifikovaný proces vývoje aplikací : Průvodce analýzou a návrhem objektově orientovaného softwaru*. Překlad Bogdan Kiszka. 2. aktualizované a doplněné vyd. Brno : Computer Press, 2007. 567 s. ISBN 978-80-251-1503-9.
- [4] MCCONNELL, Steve. *Odhadování softwarových projektů*. [s.l.] : Computer Press, 2006. 317 s. ISBN 80-251-1240-3.
- [5] SVOZILOVÁ, Alena. *Projektový management : Systémový přístup k řízení projektu*. Praha 7 : Grada, 2006. 360 s. Expert. ISBN 80-247-1501-5.
- [6] ROSENAU, Milton D. *Řízení projektů* . Eva Brumovská. 3. vyd. Brno : Computer Press, 2007. 344 s. Praxe manažera. Obsahuje rejstřík. ISBN 978-80-251-1506-0.
- [7] COCKBURN, Alistair. *Use Cases : Jak efektivně modelovat aplikace*. 1. vyd. Brno : Computer Press, 2005. 264 s. ISBN 80-251-0721-3.
- [8] TAYLOR, James. *Začínáme řídit projekty*. 1. vyd. Brno : Computer Press, a.s., 2007. 215 s. ISBN 978-80-251-1759-0.
- [9] GÁLA, Libor, POUR, Jan, PROKOP, Toman. *Podniková informatika : Počítačové aplikace v podnikové a mezipodnikové praxi*. Praha : Grada, 2006. 482 s. Management v informační společnosti. ISBN 80-247-1278-4.

- [10] WALKER, Alfred J., et al. *Moderní personální management : nejnovější trendy a technologie*. 1. vyd. Praha : Grada Publishing, 2003. 253 s. ISBN 8024704498.
- [11] KADLEC, Václav. *Agilní programování : Metodiky efektivního vývoje softwaru*. 1. vyd. Brno : Computer Press®, 2004. 278 s. ISBN 80-251-0342-0.
- [12] WIEGERS, Karl E. *Požadavky na software : Od zadání k architektuře aplikace*. [s.l.] : [s.n.], 2008. 448 s.
- [13] ŠUBRT, Tomáš. *Projektové řízení* [online]. 2003 [cit. 2009-04-20]. Dostupný z WWW: <[http://etext.czu.cz/sekce.php?titul\\_key=77&id=detail](http://etext.czu.cz/sekce.php?titul_key=77&id=detail)>.
- [14] ALDORF, Filip. *Metodika RUP* [online]. 2005 [cit. 2009-04-11]. Dostupný z WWW: <<http://objekty.vse.cz/Objekty/RUP>>.
- [15] JULINEK, Pavel. *Použití RUP pro malé SW projekty* [online]. 2.1.2008 [cit. 2009-04-20]. Dostupný z WWW: <[http://is.muni.cz/th/72639/fi\\_m/diplomova\\_prace.pdf](http://is.muni.cz/th/72639/fi_m/diplomova_prace.pdf)>.
- [16] BENEŠ, Michal. *Přehled OO notací a metodik* [online]. 16.6.2006 [cit. 2009-04-19]. Dostupný z WWW: <<http://objekty.vse.cz/Objekty/Metodiky/ANotace>>.
- [17] Unicorn. *Metodiky řízení softwarových projektů* [online]. 25.06.2007 [cit. 2009-04-05]. Dostupný z WWW: <<http://www.unicorn.eu/cz/press/clanek.php?id=15584>>.
- [18] ŠONKA , Ivan. *Project Management - Řízení projektu* [online]. 2007 [cit. 2009-04-19]. Dostupný z WWW: [www.rizeni-projektu.cz](http://www.rizeni-projektu.cz)
- [19] BUCHALCEVOVÁ, Alena. *Metodiky vývoje a údržby informačních systémů : Kategorizace, agilní metodiky, vzory pro návrh metodiky*. Mgr. Kateřina Rubášová. 1. vyd. Praha : © Grada Publishing,

- a.s., 2005. 164 s. Management v informační společnosti. ISBN 80-247-1075-7.
- [20] KANISOVÁ, Hana, MÜLLER, Miroslav. *UML srozumitelně*. 2. aktualiz. vyd. Praha : Computer Press, 2006. 176 s. ISBN 8025110834.
- [21] *UML® Resource Page* [online]. [cit. 2009-04-10]. Dostupný z WWW: <http://www.uml.org/>.

## **5 Příloha**

K bakalářské práci je přiložen datový nosič obsahující aplikaci PlanIT, dávkový soubor s SQL a krátký instalační manuál.