

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

## NÁSTROJ PRO KOLABORATIVNÍ PODPORU PŘEDPROJEKTOVÉ FÁZE PROJEKTŮ

BAKALÁŘSKÁ PRÁCE

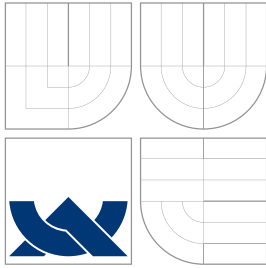
BACHELOR'S THESIS

AUTOR PRÁCE

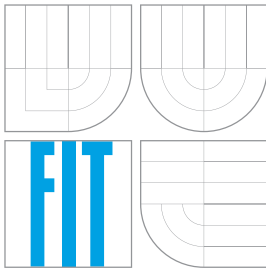
AUTHOR

MICHAL ŽÁK

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ  
FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# NÁSTROJ PRO KOLABORATIVNÍ PODPORU PŘEDPROJEKTOVÉ FÁZE PROJEKTŮ

A TOOL FOR COLLABORATIVE SUPPORT IN PRE-PROJECT PHASE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MICHAL ŽÁK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ŠÁRKA KVĚTOŇOVÁ, Ph.D.

BRNO 2012

## Abstrakt

Tato bakalářská práce se zabývá návrhem a realizací aplikace pro správu projektových příležitostí pro konkrétní potřeby společnosti IBM Česká republika. Zde jsou představeny základní metody a přístupy především k časovému a finančnímu plánování. Aplikace je vytvořena v programovacím jazyce C++ s použitím frameworku Qt od firmy Nokia. Nástroj využívá relační databázi SQLite.

## Abstract

The aim of this thesis is to design and realize an application for the management of project opportunities for the specific needs of IBM Czech Republic. The work presents the basic methods and approaches for particular time and financial planning. The application is developed in C++ using the Qt framework from Nokia. The tool uses the SQLite relational database.

## Klíčová slova

Projektový management, správa projektových příležitostí, projektové portfolio, databaze, předprojektová fáze, Qt, SQLite, C++, IBM

## Keywords

Project Management, Management of Project Opportunities, Project portfolio, Databases, Pre-project Phase, Qt, SQLite, C++, IBM

## Citace

Michal Žák: Nástroj pro kolaborativní podporu předprojektové fáze projektů, bakalářská práce, Brno, FIT VUT v Brně, 2012

# Nástroj pro kolaborativní podporu předprojektové fáze projektů

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Šárky Květoňové, Ph.D.

.....  
Michal Žák  
15. května 2012

## Poděkování

Děkuji Ing. Šárce Květoňové, Ph.D. za vedení při realizaci bakalářské práce a Ing. Jiřímu Slabému, Ph.D. za spolupráci na tvorbě nástroje pro správu projektových příležitostí.

© Michal Žák, 2012.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Základní terminologie</b>	<b>5</b>
2.1	Koncepce . . . . .	5
2.2	Trojimperativ . . . . .	6
2.3	SMART cíl . . . . .	8
2.4	Plánovat, proč a jak? . . . . .	8
2.5	Definice úspěšnosti . . . . .	9
<b>3</b>	<b>Životní fáze projektu</b>	<b>10</b>
3.1	Životní fáze produktu . . . . .	10
3.2	Předprojektová fáze . . . . .	11
3.3	Projektová fáze . . . . .	11
3.4	Poprojektová fáze . . . . .	12
<b>4</b>	<b>Předprojektová fáze</b>	<b>13</b>
4.1	Studie příležitosti . . . . .	13
4.1.1	Hierarchická struktura prací - WBS . . . . .	13
4.1.2	SWOT analýza . . . . .	14
4.2	Studie proveditelnosti . . . . .	15
4.2.1	Časové plánování . . . . .	16
4.2.2	Finanční plánování . . . . .	17

<b>5</b>	<b>Nástroj a jeho detailní analýza</b>	<b>20</b>
5.1	Srovnání stávajících produktů . . . . .	20
5.2	Obecné zadání . . . . .	22
5.3	Funkční požadavky . . . . .	22
5.4	Nefunkční požadavky . . . . .	24
5.5	Model případů užití . . . . .	24
5.6	Architektura . . . . .	25
<b>6</b>	<b>Implementace</b>	<b>27</b>
6.1	Databázová vrstva . . . . .	28
6.2	Logická a datová vrstva . . . . .	29
6.3	Prezentační vrstva . . . . .	30
6.4	Zhodnocení výsledků a návrh rozšíření . . . . .	31
<b>7</b>	<b>Závěr</b>	<b>33</b>
<b>A</b>	<b>Obsah CD</b>	<b>35</b>

# Kapitola 1

## Úvod

Pojem projektový management je v současnosti často používané slovní spojení. Je to značně obsáhlé téma, kde i jednotlivé podkapitoly mají potenciál mít svoji vlastní několika set stránkovou publikaci. Známe jeho přesný výklad? Máme alespoň tušení, co sousloví znamená a co se pod ním skrývá?

Tato práce se zabývá vysvětlením základních pojmů, které jsou používány v projektovém managementu. Pohled na problematiku je zde zvolen z časového hlediska jednotlivých životních fází. Z tohoto důvodu je postupováno chronologicky. Od okamžiku, kdy je na obzoru projektová příležitost, kdy zjišťujeme, zdali do takovéto příležitosti vstoupit, nebo ji s díky odmítnout. Přes fázi projektovou, zabývající se konkrétními vlastnostmi projektu a jeho realizací, až po fázi konečnou, kdy již můžeme zhodnotit úspěch a případně se poučit z chyb do budoucnosti. Tato poslední a důležitá část je pojmenována jako poprojektová fáze. Všeobecně jsou popsány hlavní části jednotlivých fází a jejich důležitost v celku. Stěžejním bodem je však fáze předprojektová, která je rozebrána s vyšší mírou detailu tak, aby ji bylo možné porozumět důkladně, bez nutnosti čtenáře hledat další důležité informace a doplňovat si jimi předložený text. Je tedy ponechána jako samostatná kapitola. Do této části je pak zahrnuta i praktická část práce. V závěrečné části práce jsou naznačeny možnosti dalšího rozšíření. Realizace probíhá ve spolupráci s firmou IBM Česká republika.

Již v dávné minulosti bychom mohli nalézt mnoho typů jednání s charakterem projektu. V době, kdy byly stavěny monumentální stavby pro faraony, byla tato činnost velice náročná pro naplánování a realizaci. V postavení první pyramidy je možné spatřovat potenciál pro projektové řízení. Samozřejmě jsou zde mnohé odlišnosti. Faraon nebyl takřka omezen v čase. Lidská práce byla rovněž téměř nevyčerpatelná. Co se týče prostředků, případně přírodních zdrojů, šlo jen o otázku přípravy a přepravy. V těchto okolnostech pak nalezneme rozdíl od dvacátého prvního století. Jedny z prvních principů a postupů se objevují ve 30. letech minulého století. Přibližně dalších třicet let trvalo, než si společnost uvědomila, že je potřeba vytvořit mezinárodní standardy. Převážně pro potřeby efektivní spolupráce a domluvy ve vznikajících mezinárodních týmech. V této době začaly vznikat nadnárodně uznávané organizace, jako jsou Společnost pro projektové řízení (IPMA - International Project Management Association) a Institut projektového řízení (PMI - Project Management Institute). Standardem se v takovémto případě nemyslí nejpřesněji technicky či matematicky definovaná norma. V této oblasti je nutné postihnout zcela odlišné aspekty.

Je logické, že standard musí být obecný a jedná se především o inspirativní doporučení, jaký postup je vhodný v daných situacích a geografických polohách použít.

Rychlost, jakým způsobem přicházejí v dnešních dnech nové produkty na trh, je ohromující. Aby zákazník dostával nové možnosti výběru, musí mu firma pružně dodávat kvalitní a cenově dostupné novinky. Aby byla schopna společnost dostát těmto tržním potřebám, je žádoucí používat moderní pojetí řízení. Do ústraní se tedy dostává liniové řízení podniku, kdy se firemní struktura podobá pyramidovému systému. Novým způsobem, zvláště pak pro větší firmy, je projektový management. Může se jednat o zavedení nové technologie výroby, vývoj softwarové aplikace nebo úspěšné dokončení jiných časově omezených cílů. To je jen jeden z mnoha důvodů, proč je výhodné používat takovéto řízení. Kromě dodávání řešení pro smluvní subjekty je to také způsob, jak řešit potřeby své vlastní organizace. Veškeré práce, pro které není již zhotovený propracovaný postup, lze řešit pomocí projektu.

Pro účely projektového řízení je k dispozici nespočet nástrojů, jak placených, tak volně dostupných. Práce poskytuje jejich srovnání v závislosti na tom, k jakému účelu byly vytvořeny. Je ovšem nutné rozlišovat použití takového softwaru od řízení projektu. Aplikace nemůže a nikdy ani nebude moci zajistit takzvané měkké faktory řízení, jako je například motivace projektového týmu, mezilidské vztahy a cíle. Lidský faktor je v tomto ohledu nezastupitelný. Samozřejmě je nezbytné znát projektové metody, avšak měli bychom říci, že je to spíš než metoda samotný způsob práce, způsob efektivní spolupráce.

Není nutné, aby čtenář disponoval značnými znalostmi z daného odvětví. Místy se však předpokládá základní povědomí. Cílem není seznámit čtenáře absolutně se všemi aspekty projektového cyklu. Takto složité téma se nedá zjednodušit pouze na jednu publikaci a pojmout to tedy jako triviální záležitost. Proto nejsou tak rozpracovány ostatní projektové fáze a zaměřuji se především na otevírající předprojektovou fázi. Nebudu zde věnovat přílišnou pozornost všem možným používaným praktikám. Existují celá odvětví na problematiku kupříkladu časových, rizikových nebo finančních otázek. Avšak vždy bude uvedeno alespoň pár základních myšlenek z daného odvětví.



## Kapitola 2

# Základní terminologie

Odborná literatura není ve výkladu základních pojmů naprosto jednotná a mnohé publikace pracují s odlišným výkladem základních pojmů. Z tohoto důvodu můžeme narazit na zcela jinak, avšak stále správně vysvětlené, následující termíny.

### 2.1 Koncepce

Existují tři koncepce, které jsou v současné době používány. Základním prvkem je projekt. Podle IPMA[1] (International Project Management Association) se **projekt** definuje jako proces, který je omezený náklady, zdroji a časem. Je realizovaný za účelem naplnění projektových cílů v dané kvalitě a odsouhlasenými požadavky.

Projekt má několik základních vlastností:

- Jednoznačně určený cíl - je důležité, aby byl přesně určen cíl, který může mít podobu výrobku, služby nebo jiného výsledku s předem danými vlastnostmi.
- Dočasný proces - vždy jde o záležitost s pevně stanoveným začátkem a koncem. Doporučená maximální délka by neměla přesáhnout 12 měsíců.
- Postupné vypracovávání - jedná se o situace, kdy se v průběhu projektu zpřesňují informace o stanoveném cíli. Výhodou je proto realizaci projektu provádět po sobě v krocích.
- Vyžadování různorodých zdrojů - pro splnění je zapotřebí mít k realizaci potřebné zdroje v daný čas. Zdroje, ať už lidské, či materiální, jsou vždy omezené. Vždy je zapotřebí s nimi manipulovat hospodárně.
- Snaha mít hlavního zákazníka či zadavatele - mnohdy je projekt řešen s několika účastníky. Pro nás je vždy nejlepší, aby byl stanoven jeden přední subjekt, který vystupuje v roli hlavního zadavatele. Ten pak určuje směr projektu a uvolňuje finanční prostředky.

- Neurčitost je součástí - nikdy nelze s určitostí vyloučit odchylku mezi plánem a skutečností. Těžké je hned ze začátku přesně definovat veškerá kritéria. Vždy se mohou vyskytnout i externí faktory, které mohou způsobit značné problémy.

Díky hlavním charakteristikám pak jednoduše můžeme rozlišit i další pojem - program. **Program**[1] je možné vnímat jako souhrn projektů s vyšším cílem. Hlavním rozdílem je pak strategický cíl organizace, který je definován s menší úrovní detailu. Plán programu není příliš podrobný. Dalším znakem je doba trvání, která značně přesahuje standardní dobu projektu a dosahuje i několika let. Projekt se do programu přidává pouze tehdy, když dopomáhá ke splnění cíle programu. Aktuální ukázkou programu jsou operační programy EU. Program i projekt mají podobné životní fáze a proces jejich řízení. Hlavní myšlenkou je tedy vymezení hranic a cílů, díky kterým si pak společnost může držet jednotnou filozofii.

**Poftfolio**[1] je soubor projektů nebo programů, které nemají společný cíl. Těm se pak přiřazuje prioritita. Do většího celku jsou včleněny za účelem řízení, kontroly, koordinace a optimalizace. Portfolia mohou být vytvářena podle potřeby. Mohou se tvořit tedy i na základě geografické polohy, oddělení nebo sdílení projektových zdrojů. Organizace musí vložit svoji strategii do klíčových indikátorů výkonnosti KPI (Key Performace Indicator) a ke každému přiřadit váhu. Důležitost projektů je pak posuzována právě v konfrontaci s těmito indikátory. Rozdíl proti předchozím dvěma případům je v životnosti. Portfolio přetrvává kontinuálně. Řešení je vhodné v případě, kdy ve společnosti různé projekty začínají soutěžit o potřebné společné zdroje. Řízení však s sebou nese značné změny v organizaci, nutnost sběru a práce s velkým množstvím informací.

Pro shrnutí si můžeme říci, jakým způsobem se rozpadá pomyslný strom. Jestliže je kořenem organizace, pak jako první uzel pod je portfolio. Poté pokračujeme dle pořadí dál na program a projekty. Ty poté můžeme detailněji popisovat pomocí etap, pod nimi pak souhrnovými činnostmi a v nejnižše položeném uzlu pomocí samotných činností.

Je možné si povšimnout stěžejních hledisek, která projekt provázejí. Čas je jedním z nejhlavnějších. Systém času je v projektu rozdělen do fází. **Fáze projektu**[1] je jasně oddělený časový úsek v celkové realizaci. Každá fáze má svoje přesně zadané rozmezí a jednotlivé cíle. Na základě splnění podmínek je možné přejít na další fázi. Příkladem může být přijmutí části projektu zákazníkem. V praxi je možné se setkat i s paralelně běžícími fázemi.

Oddělení fází je často spojeno s **milníky**[4]. Jedná se o událost, ve které se měří rozpracování produktů. Mohou sloužit i k jiným významným okamžikům, jako jsou rozhodnutí o změně varianty nebo ukončení projektu. Často se jedná o bod pro zpětnou kontrolu. V časové ose jsou bezrozměrné. Používají se k smluvním dohodám mezi zadavatelem a odběratelem.

## 2.2 Trojimperativ

Nejen čas je důležitým faktorem v procesu řízení. Trojimperativ ukazuje všechny tři hlavní složky dohromady. Rozsah projektu, čas a náklady. Rozsah projektu určuje, jaké práce musí být vykonány, čas pak vymezuje délku trvání projektu a časový plán. Náklady ukazují

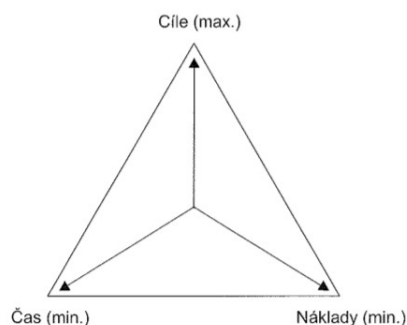
cenový odhad finančních prostředků, požadovaných na realizaci. Pro detailnější vysvětlení jsou jednotlivé složky trojimperativu popsány dále ve výčtu:

- Rozsah - stanovení objemu prací a finančních zdrojů, včetně kvalitativních charakteristik výsledného produktu (výstupu projektu). Rozhodující roli zde hraje především pečlivé vymezení cíle, standardně s respektováním zásad techniky SMART týkajících se jeho vhodného vymezení (podrobněji rozebráno v následující podkapitole).
- Čas - odhad délky trvání projektu dle vstupních požadavků (ze strany zákazníka) a nejrůznějších omezení. Výstupem je kvalitně zpracovaný časový plán projektu (zpravidla s využitím různých plánovacích nástrojů a technik).
- Náklady, neboli rozpočet - jednoznačně nejdůležitější faktor. Výdaje jsou většinou uváděny v konkrétní měně, nicméně častokrát jsou uvedeny počtem pracovních hodin, případně počtem pracovních hodin v různých kategoriích (programátor, systémový analytik, pomocný technik,...).

Mezi těmito značně konkurenčními složkami je nutné najít optimální kompromis. Přílišný důraz kladený například na technickou kvalitu ohrožuje časový plán a dokončení projektu ve stanoveném termínu.

Na obrázku 2.1 je způsob zobrazení trojimperativu, přičemž našim cílem je bod v trojúhelníkovém prostoru. Díky definici cílů jsme určili vzdálenost ostatních vrcholů. Dojde-li k situaci zvýšení rozsahu projektu, zpravidla se tato skutečnost projeví v prodloužení doby realizace a zvýšení finančních nároků na projekt (příp. obojím).

Tři rozměry mohou a nemusí být rozšířeny o čtvrtou dimenzi. Je samozřejmé, že i když jsou tři hlavní činitelé základem, nemusí tvořit úplný celek. Čtvrtým může být kvalita, nebo také míra uspokojení zákazníka, avšak to vše je možné zahrnout do definice cílů.



Obrázek 2.1: Trojimperativ (zdroj: [1])

Aktuální přístupy jsou v tomto ohledu jasné a striktní. Veškeré projekty mají být hotovy za minimum času a peněžních prostředků v co nejvyšší kvalitě.

## 2.3 SMART cíl

Definování cílů je stěžejní pro celý projekt. Musí se přesně určit podle SMART pravidel, o jakém výsledku se jedná. Nejasné určení vede k nepochopení spolupracujících stran. Proto nestačí uvést jako jeden z cílů přívětivé prostředí pro uživatele bez dalšího doplnění. Jak zadavatel, tak programátor si pod tímto zadáním mohou představovat diametrálně odlišný výsledek. Přesný a technický popis je výhodný pro obě strany i s ohledem pro případné budoucí spory.

Pravidla pro SMART cíl:

- S (specific) - řešení musí být přesně popsáno. Musí být jasné, co přesně má být výstupem a jak tohoto výstupu chceme dosáhnout.
- M (measurable) - cíl musí být možné konkrétně změřit. Je to důležité pro možnost určení, zdali jsme dosáhli požadovaného cíle či nikoliv.
- A (AGREED - akceptovaný) - všichni musí být informováni a musí souhlasit s přiměřeností cíle.
- R (realistic) - je reálné a realizovatelné za daných podmínek cíl uskutečnit.
- T (timed) - stanovení přesných termínů plnění.

## 2.4 Plánovat, proč a jak?

Přestože se nezkušenému člověku může zdát, že nejdůležitější částí projektového cyklu je samotná realizace, opak je pravdou. Plány jasně ukazují, jakým způsobem se bude dosahovat trojimperativu. Jde o určitou simulaci budoucnosti. Již v počátku lze odhalit neperspektivní projektovou příležitost, a tak nevkládat prostředky do její realizace.

Plán napomáhá komunikaci mezi účastníky a jejich vzájemnému porozumění. Jsou vodítkem pro sledování a vyhodnocování. Jejich výhodou je také předcházení problémům. Společnosti častokrát tvoří plány dlouhodobé, střednědobé a krátkodobé. V těchto případech má plán také strategickou úlohu.

Při tvorbě plánu, neboli odhadu, je vhodné zapojit konkrétní osoby, které na něm posléze budou pracovat. Tento krok značně zvyšuje motivaci k jejich dodržení. Zaměstnanec v dané oblasti také může díky svým předchozím zkušenostem lépe odhadovat realistické řešení, mimoto se pak jedná i o usnadnění práce projektového manažera.

Vlastnosti, které by podle [9] měl obecně plán splňovat:

1. Definuje všechny potřebné zdroje v daný čas.
2. Obsahuje finanční rozvahu s podrobným rozpisem.
3. Popisuje přesný postup prací s jasně definovaným časovým harmonogramem.

4. Vždy se snaží co nejlépe odhadovat budoucnost a obsahuje rezervu pro případné nepředvídatelné události.
5. Je realistický.

Případy, kdy jsou tvořeny plány pouze proto, aby byl uspokojen požadavek zadavatele nebo nadřízeného, jsou neefektivní. Vždy bychom měli mít na paměti, že plán je důležitý a jeho zhotovení bychom si měli sami přát. Na druhou stranu, plán by se neměl tvořit v situaci, kdy bychom se jeho tvorbou zabývali déle než problémy, které by vyvstaly bez jeho přítomnosti.

## 2.5 Definice úspěšnosti

Abychom mohli směřovat k úspěšnému řízení projektu, potažmo úspěšnému projektu, je zapotřebí si objasnit, jak poznáme, že jsme byli úspěšní. Hodnocení přísluší všem zainteresovaným stranám. Na samotné řízení je možné pohlédnout jako na podprojekt daného projektu. Přestože může být řízení sebelepší, nemusí se vždy cíl podařit. Je mnoho komplikací, kterým bohužel nelze zabránit. Projekt může být ukončen kvůli nedostatku financí, změně dlouhodobé strategie firmy nebo dokonce i kvůli nepředvídatelné přírodní katastrofě.

Praxe není učebnicí. I při dodržení všech parametrů trojimperativu nemusí být projekt úspěšný. Co s aplikací, která sice všechny dimenze splnila, ale je v reálném životě nepoužitelná? To samé však platí i opačně. Při nedodržení veškerých požadavků se přesto může jednat o úspěch. Používají se tzv. kritéria úspěchu projektu. Při jejich tvorbě je použita technika SMART. Pak je možné změřit naše úspěchy exaktně.

Při obecném pohledu je možné projekt hodnotit kladně v případě, že jsou splněny jeho očekávání. Další podmínkou je výsledná cena, kvalita a čas pro dodání produktu nebo služby na trh. Očekávaná návratnost, dopady na životní prostředí, ale i tzv. měkké faktory jsou taktéž možnými faktory úspěchu.

Klíčovým předpokladem úspěchu každého projektu je komunikace mezi zadavatelem a projektovým týmem, který jeho realizaci zajišťuje (především v úvodní fázi projektu, kdy dochází k ujasňování cílů projektu, včetně všech omezujících podmínek a kritérií na něj kladených). Nelze ji však podceňovat ani v dalších fázích realizace projektu (průběžné informování zadavatele projektu o průběhu samotné realizace, zejména pak v případě vzniku jakýchkoli odchylek či změn oproti plánu).

### Shrnutí

Každý projekt tvoří tři hlavní složky, a to: rozsah, čas a náklady. Je nutné najít mezi nimi co možná nejlepší vyvážení. Když poté definujeme cíle projektu, je nutné, aby byly specifické, měřitelné, akceptované, reálné a měly přesně stanovené termíny. Při tvorbě plánu projektu by nemělo dojít k situaci, kdy bude tvořen bez následného využití. Tím se vyhneme mnoha budoucím problémům. Plán nám pomáhá i při vyhodnocení úspěchu nebo neúspěchu. V následující kapitole jsou popsány životní fáze projektu v kontrastu s životními fázemi produktu.

## Kapitola 3

# Životní fáze projektu

Existuje nespočet náhledů na problematiku projektového řízení. Jeden z nejintuitivnějších je dle časového sledu událostí. V této části jsou popsány základní metody a postupy v návaznosti na časovou osu projektu.

### 3.1 Životní fáze produktu

Pro vyjasnění kontextu si musíme uvědomit, že je rozdíl mezi projektem a produktem. Projektový management se používá v obou případech. Tak, jako má projekt svůj životní cyklus, má i softwarový produkt svůj. Vývoj softwarového produktu je pouze jedna podmnožina projektů v oblasti informačních technologiích.

Vývojové modely (modely životního cyklu) podle [7]:

- Vodopádový model - nejstarší model vývoje. Etapy jsou seřazeny za sebou. Další etapa začíná po dokončení předcházející. Základním problémem je nutnost mít přesné požadavky již na začátku. V případě uživatele, který vznesl požadavek později, musíme jít prakticky od začátku. Hlavními etapami jsou požadavky, návrh, implementace, testování jednotek, testování systému a provoz s údržbou.
- Interaktivní model - snaha o odstranění problému ve vodopádovém modelu, kdy uživatel vidí spustitelnou verzi příliš pozdě. Proces vývoje je rozdělen do iterací. Začátkem je specifikace požadavků. Následují etapy návrhu, implementace a testování. Po každé iteraci má uživatel k dispozici spustitelnou verzi. Nevýhodou je možná horší struktura systému v důsledku příchozích požadavků.
- Inkrementální model - definují se ucelené části systému, a ty se pak dodávají zákazníkovi, přičemž se při každém vydání rozšiřuje jeho funkčnost.
- Spirálový model - založen na opakování etap ve spirále. Je možné jej vnímat jako vylepšení předchozích dvou, ale s tím rozdílem, že uživatel dostává prototyp a ne pouze software s omezenou funkčností. Prototyp se po schválení zahodí. Tím je možné odhalit chyby hned ze začátku a mít dobrou strukturu systému za cenu tvorby prototypů.

- Model rychlého vývoje aplikací - zvláštní přístup pro rychlý vývoj kvalitního softwaru. Pracuje se neustále s vyvíjeným prototypem. Použití generátorů kódu pro naplnění tržních potřeb s menším důrazem na kvalitu technického provedení.
- Agilní vývoj - z dnešního pohledu se jedná o Extrémního programování nebo Scrum metodologie. Hlavní důraz je kladen na člověka, komunikaci uvnitř týmu i mezi týmem a zákazníkem, rychlost a schopnost reagovat na změny ze strany zákazníka. Z dnešního pohledu nejvyžívanější možnost vývoje softwarového produktu.

Pro vývoj vybíráme takový model, který zohledňuje typ vyvíjeného softwaru a jeho složitost. Tento bod výběru však neřešíme pokud zakázku na vývoj softwaru nezískáme (rozhodování o tom, zda budeme projekt realizovat či nikoli, probíhá již v předprojektové fázi).

## 3.2 Předprojektová fáze

Jedná se o první setkání s projektovou příležitostí. Jejím hlavním úkolem je rozhodnout, zdali do projektu investovat. Pomoci nám v této fázi může zpracování tzv. studie příležitosti a studie proveditelnosti, kde posuzujeme rovněž možné náklady a přínosy plynoucí z realizace projektu. Ty mají za úkol odpovědět na základní otázky o dané projektové příležitosti.

Předprojektová fáze zde hraje tedy klíčovou roli, a jelikož je jedním z dílčích cílů podrobnější seznámení se s touto fází, je jí věnována samostatná kapitola.

## 3.3 Projektová fáze

Po přijetí rozhodnutí, že projektovou příležitost využijeme (budeme projekt realizovat), nastupuje řízení projektu jako takového. Tuto část je možné rozdělit do 4 dílčích částí:

V zahajovací části je pro nás úkolem sestavit projektový tým, distribuovat informace všem účastníkům, operace pro zajištění kvality, a v neposlední řadě i vyžádání si nabídek od případných dodavatelů. Výstupem je zadávací listina projektu.

Následuje část plánování, kdy má projektový tým k dispozici již přesný popis projektu a jeho úkolem je vytvořit výchozí plán.

Poté se může přikročit k samotné realizační části. Zde je úkolem sledovat a kontrolovat jednotlivé práce na projektu. Kontrola probíhá v oblasti rozsahu, dodržování časového plánu, nákladů a rizik. V případně nečekaných změn je nutné, abychom na ně pružně reagovali. Pokud znatelně mění výchozí plán, je nutné jej přepracovat. Nesmí se také zapomenout na podávání aktuálních zpráv o stavu projektu všem účastníkům projektu.

Poslední částí je pak ukončení, kdy dochází k fyzickému předání výsledků. Součástí této fáze je podepsání akceptačních protokolů zadavatelem projektu, veškerému finálnímu vypořádání a archivaci.

Přestože došlo k ukončení projektu, nejedná se o konec cyklu, protože samotná realizace přináší řadu nových znalostí a poznatků. K tomu je určena poslední fáze - poprojektová fáze.

### **3.4 Poprojektová fáze**

Po realizaci a ukončení je dobré udělat analýzu projektu a zjistit tak, kde nastala chyba, ze které je možné se poučit a neopakovat ji v budoucnu. Zhodnocení se dělá i tehdy, když projekt dopadl úspěšně. Hlavním cílem je identifikovat místo pro zlepšení, ne někoho potrestat. Na samotné ohlédnutí není zcela logicky kladen takový důraz. Dlouhodobější trendy některých ukazatelů se mohou projevit až s časovým odstupem, kdy se produkt již nějakou dobu používá. Proto se v určitých případech hodí počkat a provést revizi až po nějaké době. V této fázi je možné zhodnotit i kvalitu subdodavatelů, a tak s nimi případně do budoucna dále nespolupracovat.

#### **Shrnutí**

Vývoj produktu je pouze podmnožina projektů v oblasti informačních technologiích. Pro vývoj produktu je v dnešním světě nejvíce využíván agilní přístup. Podíváme-li se na životní fáze projektu, je možné je rozčlenit na tři základní části. Fáze předprojektová, kdy se rozhodujeme, zdali vstoupit do projektu. Fáze projektová ve které dochází k samotné realizaci projektu. Poslední je pak poprojektová fáze, kde provádíme závěrečné zhodnocení projektu (získání nových poznatků a poučení se z vlastních chyb do budoucna).



## Kapitola 4

# Předprojektová fáze

Na začátku je třeba se rozhodnout pro nebo proti dané projektové příležitosti. V našem případě se jedná o rozhodnutí pustit se dále do realizace, nebo ukončit vše v počátku. K tomuto účelu nám může pomoci zpracování dvou klíčových dokumentů, které nám zodpoví elementární otázky projektu. První z dokumentů je studie příležitosti a slouží jako hrubý přehled projektu. Pokud se organizace rozhodne pokračovat, následuje zpracování dokumentu s názvem studie proveditelnosti. Zde se podrobněji rozvádějí aspekty obsahu, způsobu dosažení, času, zdrojů a nákladů.

Pokud máme před sebou menší projekt, je pak přípravu možné shrnout pouze do jediného dokumentu, obsahující kombinaci obou studií.

### 4.1 Studie příležitosti

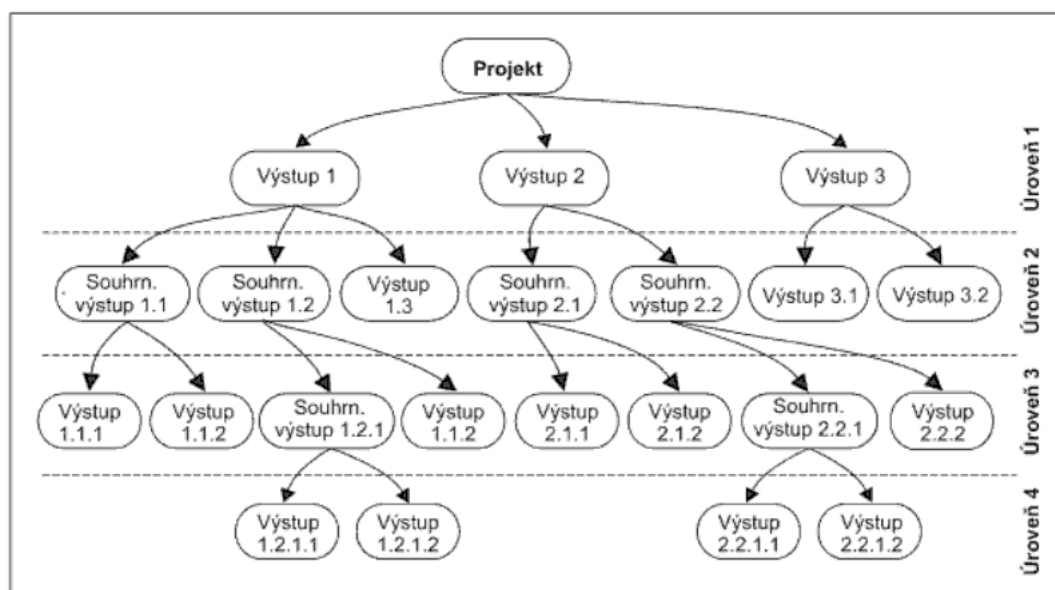
Výstupem studie je první specifikující dokument k danému projektu. Specifikace v úvodním stádiu není nijak zevrubná. Jedná se pouze o hrubý odhad. Využití zde nacházejí i agregované hodnoty s expertním odhadem. Cílem práce je odpovědět na základní otázku. Je smysluplné realizovat tento projekt? První rozhodující okamžik pro projektovou příležitost. Rozhodnutí o její neatraktivnosti znamená její konec a tím i ušetřené náklady již v počátku. Studie je dokument, který by sám o sobě neměl být nákladný. Její optimální rozsah je do 10 stran. V dokumentu uplatňujeme mimo jiné i analýzu nazvanou hierarchická struktura prací (WBS) a metodu analýzy silných a slabých stránek (SWOT).

#### 4.1.1 Hierarchická struktura prací - WBS

Abychom byli schopni cokoliv plánovat, je nezbytné znát rozsah a strukturu. Hierarchická struktura prací, neboli Work Breakdown Structure, slouží k přehlednému zobrazení všech činností. Ukázka dekompozice je znázorněna na obrázku 4.1. Obdobný princip bychom mohli aplikovat i v případě projektového portfolia nebo programů, které se rozpadají na úroveň jednotlivých projektů a podprojektů. V rámci dekompozice dochází k identifikaci všech potřebných kroků. Tedy i těch, které nejsou přímo požadavkem zákazníka, ale přesto

jsou nezbytné k realizaci. Při tvorbě je vhodné ke každému uzlu připojit tzv. kód WBS, neboli číslo, které může mít podobu číslování kapitol v této práci. Důvodem je zjednodušení komunikace ke konkrétním uzlům.

Realizace se pak provádí pouze s ohledem na nejnižší vrstvu. Nadřazené části jsou pak pouze souhrnem níže položených uzlů. Zanoření nejnižšího uzlu by nemělo přesáhnout čtvrtou úroveň. Uzel, tedy samostatná činnost, musí být natolik detailní, abychom ji byli schopni efektivně řídit. Kromě činnosti je možné použít taktéž pracovní balíky nebo úkoly. Nezávisle na názvu je důležité, aby byl dokument vždy aktuální, musí se do něj zanést jakákoliv změna v průběhu realizace. Pokud bychom potřebovali více jak čtvrtou úroveň, je lepší použít subprojekt. Subprojekt je část projektu, který je sám o sobě řízen jak projekt. Jeho výsledek je součástí nadřazeného projektu.



Obrázek 4.1: Hierarchická struktura prací (zdroj: [1])

#### 4.1.2 SWOT analýza

Tuto techniku je možné použít v různých fázích a s různými rozšířeními. Užití nachází ve tvorbě dlouhodobých plánů společnosti, analýz řešení problému nebo právě v předprojektové části na projekt samotný. Název je odvozen z prvních písmen hlavních parametrů:

- Strengths - silné stránky. Odpovědi jsou veškeré silné stránky našeho projektového týmu
- Weaknesses - slabé stránky. Seznam všech slabých stránek projektového týmu.
- Opportunities - příležitosti. Soupis příležitostí pro projektový tým.
- Threats - výčet hrozeb, které ohrožují projekt (mohou mít negativní dopad na jeho průběh).

Při nalézání jednotlivých odpovědí je dobré použít princip brainstormingu. Tato metoda, založená na spontánní diskuzi a hledání nových nápadů, zajišťuje komplexní pohled všech účastníků.

Pro kvalitní sestavení platí několik zásad. Zprv bychom se měli ujistit, zda všichni kolegové znají základní vlastnosti této analýzy. Komplexnost nám zajišťuje skupina, ve které probíhá diskuze. Pokud by byla tvořena pouze jednotlivcem, je nutné ji posléze nechat zkontrolovat. Analýzu z tohoto důvodu je vhodné nechat zpracovat všechny, a získané informace pak zpracovat do souhrného výsledku. Další zásadou je nesestavovat výsledek hned v prvním kole sezení. Bývá lepší, pokud tým získá určitý časový odstup a závěr se udělá až při další příležitosti. Posledním, avšak neméně důležitým bodem, je definice časového období, ve kterém mají kolegové uvažovat hrozby a příležitosti.

Formátem může být jednoduchý seznam číslovaný pod sebe nebo tabulka, která je zobrazena na obrázku 4.1. Příпустných rozšíření je několik. Jednotlivým bodům můžeme přiřadit váhu v podobě určené stupnice nebo procentuální pravděpodobnosti. Dalším způsobem je pak seřadit vše podle důležitosti.

	<b>Současnost</b>	<b>Budoucnost</b>
Pozitiva	silné stránky: 1. 2. atd.	příležitosti: 1. 2. atd.
Negativa	slabé stránky: 1. 2. atd.	hrozby: 1. 2. atd.

Tabulka 4.1: Vzorová tabulka SWOT analýzy (upraveno, zdroj: [1])

Analýza upozorňuje na rizika, které mohou nastat. Zároveň může zahrnovat i to, jakými silnými vlastnostmi si můžeme být jisti, a ty pak využít k odvracení hrozeb nebo získání konkurenční výhody.

## 4.2 Studie proveditelnosti

Při rozhodnutí pokračovat po zpracované studii příležitosti je nutné přistoupit k problematice důkladněji. Studie proveditelnosti má optimální rozsah do 25 stran. Primárně je třeba zpracovat časové a finanční otázky, dále pak i odhadované potřebné zdroje. V případě, že studie proveditelnosti obsahuje hierarchickou strukturu prací pouze jako návrh nebo zákazník udělal změnu, je tento návrh vypracován do koncového znění. Součástí by pak měla být i souhra závěrů ze studie příležitosti.

Existuje celá řada metod, zde si představíme ty nejnámější.

## 4.2.1 Časové plánování

Čas se nachází i jako jedna z dimenzí trojimperativu. Jeho plánování je zásadní. Je také velice dobře změřitelný. Projektoví manažeři uvádějí, že časový faktor je jedním z největších výzev v dnešní době.

### Orientované síťové grafy

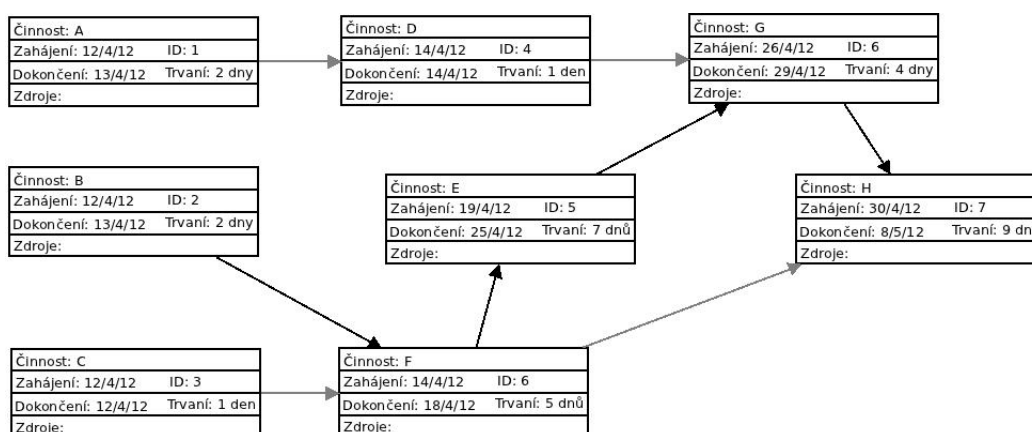
Pro správné seřazení již sestavených aktivit pomocí WBS je nutné dát těmto aktivitám časový rámec. Síťové grafy umožňují efektivně zobrazit kompletní aktivity projektu. Existuje mnoho technik, zde si ukážeme ty nejpoužívanější.

### Metoda uzlově orientovaného grafu - PDM

V metodě uzlově orientovaného grafu (PDM - Precedence diagramming metod) jsou aktivity zobrazeny pomocí obdélníků. Postupujeme zleva doprava. Aktivity na levé straně jsou zahajující a aktivita na pravé naopak značí ukončení projektu. Příkladem je uzlově orientovaný graf na obr 4.2. Každá z aktivit má přidělený potřebný počet dní na její zvládnutí.

Při tvorbě je zapotřebí určit typ závislosti mezi aktivitami:

- Ukončení-zahájení: nejprve je třeba dokončit předcházející aktivitu, aby bylo možné spustit následující aktivitu
- Zahájení-zahájení: následující aktivita je zahájena současně s předcházející aktivitou
- Ukončení-ukončení: následující aktivita je ukončena zároveň s předcházející aktivitou
- Zahájení-ukončení: Zřídka používaný vztah, kdy předcházející aktivita musí začít dříve, než skončí následující aktivita.



Obrázek 4.2: Síťový uzlově orientovaný graf (upraveno z: [10])

Metodu uzlově orientovaného grafu používá většina softwarových produktů. Graf na obrázku 4.2 je podobný výstupu z aplikace Microsoft Project.

## Metoda kritické cesty - CPM

Metodu (CMP - Critical path method) si představíme na grafu 4.2. Jedná se o odhad doby trvání projektu. Odhad provádíme tak, že sečteme veškeré možné cesty od začátku do konce a podle časové náročnosti rozhodujeme, kde leží kritická cesta. Ta je pak značena výraznější barvou.

$$\begin{array}{llll} \text{cesta1 :} & A + D + G + H = & 2 + 1 + 4 + 9 = & 16 \\ \text{cesta2 :} & B + F + E + G + H = & 2 + 5 + 7 + 4 + 9 = & 27 \\ \text{cesta4 :} & C + F + E + G + H = & 1 + 5 + 7 + 4 + 9 = & 26 \\ \text{cesta3 :} & B + F + H = & 1 + 5 + 9 = & 15 \end{array} \quad (4.1)$$

Kritická cesta je ta, která nám určuje nejdelší cestu od počátku do konce projektu. V našem případě se jedná o cestu (4.1), která je součtem trvání aktivit B, F, E, G, H. Pokud za jednotlivé aktivity dosadíme jejich trvání, je výsledek 27 dní. Zároveň tento údaj zobrazuje nejkratší možný termín dokončení projektu.

## Metoda PERT

Pro projekty s vysokou mírou nejistoty je do odhadu délky aktivit v síťovém grafu použita metoda PERT (Program Evaluation and Review Technigue). Pracuje s váženým průměrem, který je tvořen ze tří časových údajů. Jde o optimistický, nejpravděpodobnější a pesimistický odhad času potřebného na danou aktivitu. Vážený průměr je pak vypočítán pomocí vzorce

$$\text{Vážený průměr} = \frac{\text{optimistický odhad} + 4 \times \text{nejpravděpodobnější odhad} + \text{pesimistický odhad}}{6}$$

Tento odhad má v sobě zanesenu jistou míru zvýšeného rizika, a tak pomáhá k většímu dodržování stanovených termínů i přes možné komplikace při realizaci.

## Ganttův diagram

Ganttův diagram je standardní zobrazovací metoda v projektu, kde každá aktivita má uveden svůj začátek a konec. Zobrazení probíhá jako vodorovná úsečka v grafu, kde osa x reprezentuje čas a osa y jednotlivé aktivity. Délka čáry je odvíjí od času potřebného na dokončení úkolu. Obraz 4.3 znázorňuje Ganttův digram s milníky (v obrázku značeny tmavšími úsečkami), a také závislosti mezi jednotlivými úkoly (tenké šipky).

### 4.2.2 Finanční plánování

Plánování finančních výdajů je možné provádět několika způsoby. Mezi ty jednodušší patří odhad podle již realizovaného podobného projektu. Hlavní nevýhody tohoto typu odhadu



$$\text{ROE} = \frac{\text{zisk}}{\text{vlastní kapitál}} \times 100 = \text{výsledný ukazatel [\%]} \quad (4.3)$$

### Výnosnost aktiv - ROA

Výnosnost aktiv (ROA - Return on Assets) je další možností, jak zjistit výnosnost projektu. Podle [3] se tento ukazatel vypočítá jako součet zisku a úroků, který je dělen celkovým kapitálem tak, jak to zobrazuje vzorec (4.4).

$$\text{ROE} = \frac{\text{zisk} + \text{úroky}}{\text{celkový kapitál}} \times 100 = \text{výsledný ukazatel [\%]} \quad (4.4)$$

Kvalitně zpracovaná finanční rozvaha zahrnuje rovněž peněžní rezervu na nenadálé výdaje, a to hned v několika oblastech plánování. V rámci konkurenčního boje jsou sice tyto prostředky jako první použity ke snížení celkové ceny, není však rozumné jít do projektu s rezervou rovnající se nule.

### Shrnutí

Je evidentní, že předprojektová fáze je počátek, který by se v žádném případě neměl přeska-  
kovat. Je důležitou součástí projektového cyklu. Neměli bychom dokumenty vypracovávat  
pouze proto, aby byly napsány. Smyslem je předpovídat budoucnost a způsob, jakým se  
bude dosahovat vytyčeného cíle. Pokud bychom zjistili, že je to pro naši firmu neperspek-  
tivní, nemá smysl se do této projektové příležitosti pouštět.

Byly uvedeny základní metody v jednotlivých odvětvích. Ty, které se nejvíce použí-  
vají, a tedy které by měly být i součástí každé předprojektové fáze. Pro naše potřeby jsme  
tím získali pohled, jaké kroky se musí udělat pro projektovou příležitost. Všechny příleži-  
tosti se tak budou přehledně vkládat do aplikace, která je určena přesně k tomuto účelu  
a na míru potřebám firmy.

V následující kapitole je uvedena analýza, srovnání a další konkrétní informace o vý-  
voji aplikace.

## Kapitola 5

# Nástroj a jeho detailní analýza

V teoretické části práce byly přestaveny základní metody a přístupy zejména k časovému a finančnímu plánování. Získali jsme tak přehled, jaké kroky je třeba vykonat v souvislosti s každou novou projektovou příležitostí. V praktické části je cílem analyzovat a navrhnout efektivnější možnosti správy projektových příležitostí na míru potřebám firmy IBM Česká republika, spol. s.r.o.

Analýza probíhala v několika cyklech. Prvním krokem byla samotná účast a pozorování při používání stávajícího řešení. V dalším kole proběhly osobní schůzky s projektovými manažery a dotazování, jakým způsobem by se dala jejich práce zjednodušit a zefektivnit. Bylo zjištěno, jaká funkcionality je vhodná k zachování a naopak, které funkce jsou obtěžující. V těchto několika krocích se tak dospělo k zadání. Ze slovního popisu tak postupně krystalizovaly funkční i nefunkční požadavky, které jsou v této kapitole později podrobněji rozebrány.

Jedním z kroků, které předcházely samotné realizaci, bylo zjištění konkurenčních nástrojů pro možnost použití v podobném případě. Jejich kategorizace a srovnání je v následující části.

### 5.1 Srovnání stávajících produktů

Možností softwarové podpory je mnoho. Základní kategorizaci je možné provést na základě velikosti projektu. Pro nejnižší stupeň, a tedy na nejjednodušší projekty, je možné využít značné množství webových služeb (převážně za poplatek do dvaceti dolarů za licenci pro úplné používání). Existují i volně dostupné varianty bez omezení. U středně náročných projektů je možné využívat desktopové aplikace se specializací na projektový management. Zde se cena pohybuje kolem desítek dolarů. Pro ty největší projekty jsou pak vhodné speciální produkty, poskytující plnou počítačovou podporu. Tyto nástroje jsou již finančně náročnější, konkrétněji v řádech stovek dolarů.

Pro účely navrhované aplikace je třeba detailněji popsat zejména produkty nejvyšší kategorie. Ty můžeme rozlišit podle toho, zda se jedná o software určený k investičnímu plánování, nebo řízení konkrétního produktu.



Z pohledu využívání v reálném světě jsou největšími hráči v daném okruhu investičního plánování a řízení produktu dvě softwarová řešení. Každý z nich má zaměření do jiných částí projektového řízení.

První z nich je IBM Rational Focal Point od společnosti IBM. Tento nástroj nachází uplatnění v rozhodování a investičním plánování, kdy není jasné, který projekt je pro firmu nejvýhodnější.

Pro IBM Rational Focal Point je primárním cílem uživateli správně vyhodnotit, který projekt realizovat v závislosti na omezujících podmínkách. Tedy případy, kdy se máme rozhodnout mezi několika desítkami projektů, kdy každý z nich má jiné parametry v podobě ceny, délky projektu a potřebných zdrojů. Přičemž náš elementární záměr je vybrat ty projekty, se kterými vyčerpáme obnos peněz tím nejlepším možným způsobem a s největším užitekem pro naši společnost.

V obecném pohledu je možné říci, že tento nástroj není nezbytný. Každá společnost může tuto aplikaci nějakým způsobem nahradit. Otázka je, s jakým výsledkem. Rozhodnutí, kam investovat, je možné udělat vždy. Vyplývající otázka z této možnosti pak je, zdali byl správně vybrán právě ten nejlepší projekt.

Druhým softwarem v pořadí je pak Microsoft Project firmy Microsoft, který se soustřeďuje na již rozhodnutý konkrétní projekt a zajišťuje podporu pro jeho vedení a realizaci (fáze zejména operativního řízení projektu). Jeho hlavní myšlenkou je plánování a konkrétní řízení jednoho projektu. Případně se může jednat o řízení klidně i celého portfolia. Základní rozdíl je ve smyslu řízení, tedy již realizování daného projektu. Umožňuje detailní údržbu úkolů v projektu. Taktéž je zde prostor pro detailní rozbor zdrojů, časování a vzájemných závislostí. Zakomponována je možnost týmové komunikace a kooperace či reportingu.

Z pohledu četnosti využití se Microsoft Project používá více, protože je větší důraz na pružné řízení konkrétního projektu, příp. projektů v rámci projektového portfolia.

Tyto sofistikované nástroje jsou výborným řešením pro uvedené oblasti. Jejich celková komplexnost je v našem případě až zbytečně přílišná. Zejména proto, že výsledná aplikace má být zcela intuitivní, jednoduchá na ovládání a má disponovat pouze jasně specifikovanou funkcionalitou bez nadbytečných funkcí.

Stávající řešení správy projektových příležitostí, je dosti nekonformní. Firma využívá klasického programu Excel od firmy Microsoft. Základní operace, jako je přidání nové projektové příležitosti v podobě přidání nového řádku do tabulky, je sice funkční, avšak nedostatečné. Chybí zde jednoduché vyhledávání, které by umožnilo například rychle vyhledávat ceny za aktuální rok v přesně definovaném oddělení firmy. Tento požadavek by se tak musel řešit přes složité funkce a výsledek by bohužel stejně nemohl být zobrazen na vhodném místě. Práce se tak stává méně efektivní a přehled neúplný.

Všechny ukazatele a další požadavky, které byly zjištěny, jsou uvedeny níže ve funkcích i nefunkčních požadavcích.

## 5.2 Obecné zadání

Aplikace byla popsána obecným zadáním, které bylo tvořeno v několika vlnách s projektovými manažery. Je řada podmínek, kterým musí nástroj vyhovovat či které musí splňovat, aby jej bylo možné používat pro zamýšlený účel. Mezi hlavní požadavky patří následující:

- Aplikace musí umět přidávat, editovat a mazat projektové příležitosti v tabulkovém formátu. Je žádoucí, aby bylo vše možné realizovat bez nutnosti dalších mezikroků a bylo vše možné provádět z jediného tabulkového pohledu.
- Pro ostatní potřebné oblasti (zákazníci, zaměstnanci, zprostředkovatelé, obchodní značky, firemní sektory) taktéž možnost přidávání, editace a smazání jednotlivých řádků.
- Jednoduchá a efektivní možnost třídění podle jednotlivých sloupců (atributů) tabulky.
- Rychlá možnost vyhledání příslušné projektové příležitosti v celé tabulce. Vyhledávání musí fungovat i pro zadání více parametrů.
- Zobrazování aktuálních i specifických finančních ukazatelů, které budou mít schopnost se měnit podle vložených dat, v jednotlivých řádcích projektových příležitostí.
- Jednoduchá možnost importování stávajících originálních dat do nové aplikace bez potřeby složitého manuálního popisu dat.
- Nástroj musí být schopen vložené data exportovat ve stejném formátu, jako jej přijímá. Tento formát musí být možné zpětně importovat do využívaného softwarového řešení.
- Nástroj nesmí vyžadovat permanentní přístup k počítačové síti, a zároveň ke spuštěnému databázovému serveru.
- Cílovým operačním systémem je Linux.

Slovní popis tvořil podstatu pro vytvoření matice funkčních a nefunkčních požadavků, které jsou konkrétně sepsány v další podkapitole.

## 5.3 Funkční požadavky

Úkolem funkčních požadavků je, aby jasně specifikovaly, co a jak má produkt dělat. Finální složení funkčních požadavků kladených na aplikaci je rozepsáno na obrázku 5.1. Tabulka funkčních požadavků je rozdělena na dvě části, které obsahují kompletní shrnutí požadavků, které jsou na aplikaci kladeny.

Pro lepší identifikaci má každý požadavek přiřazen unikátní kód. Sloupec *rozpad* slouží k logickému sloučení jednotlivých požadavků dohromady. Samotné číslování ukazuje, jak jsou jednotlivé požadavky sdružovány do skupin (kupříkladu umožnění filtrování by mělo být schopné filtrovat již po napsání prvního znaku a to přímo na úvodní obrazovce).

MATICE FUNKČNÍCH POŽADAVKŮ					
Jméno projektu:		Nástroj pro správu opportunit			
Centrum:		FIT VUT			
Odpovědné osoby:		Michal Žák, Jiří Slabý			
Popis projektu:		Nástroj pro kolaborativní podporu předprojektové fáze projektů			
ID	Rozpad	Popis	Důležitost		
001	1.	Veškeré důležité hodnoty na jediné obrazovce, tak aby uživatel měl přehled o všech potřebných ziscích	V		
002	1.1	Viditelné všechny údaje o opportunity bez nutnosti použití horizontálního posuvníku	S		
003	1.2	Zobrazení pomocných statistických ukazatelů na úvodní obrazovce	V		
004	1.2.1	Automatické přepočítávání hodnot při změně dat souvisejících se zařazením projektové příležitosti	V		
005	1.2.2	Součet aktuálně zobrazených projektových příležitostí	V		
006	1.2.3	Součet aktuálního Q za Strech	V		
007	1.2.4	Součet aktuálního Q za Solid	V		
008	1.2.5	Součet aktuálního Q za Risk	V		
009	1.2.6	Jednotlivé součty za Q1, Q2, Q3 a Q4 rozdělené podle firemních značek	V		
011	2	Umožnění řazení dat v aktuálního zobrazení v libovolném sloupci	V		
012	3	Umožnění filtrování dat	V		
013	3.1	Okamžitá filtrace už po napsání prvního znaku. Pod každým názvem sloupce musí být k dispozici buňka, sloužící k možnosti napsání slova, podle kterého proběhne filtrace.	V		
014	3.2	Filtrace dle Q pomocí prokliku	S		
015	3.3	Umožnění selekce bez nutnosti uživatele otevřít další okno (formulář), aplikace musí umožnit okamžitou filtraci v úvodní obrazovce.	V		
016	4	Barevné rozlišení projektové příležitosti (po akci uživatele)	S		
017	5	Hlavní tabulka pro zobrazení všech opportunit	V		
018	5.1	Snaha o viditelnost informace v buňce	N		

Obrázek 5.1: Funkční požadavky - 1. část

Sloupec *popis* udává konkrétní znění požadavku. Poslední sloupec *důležitost* může nabývat třech hodnot, a to podle známky důležitosti u jednotlivých požadavků.

Hlavním požadavkem, který by měl být zmíněn za všechny ostatní, byla podmínka jednoduchosti z pohledu koncového uživatele tak, aby práce uživatele s aplikací nebyla rušena, kupříkladu nutností zadávat speciální formát dat. Hodnoty, které vyjadřují míru důležitosti v požadavcích, jsou následující:

- V - Vysoký důraz na implementaci.
- S - Střední důraz na implementaci.
- N - Nízký důraz na implementaci.

Na obrázku 5.2, je druhá část tabulky s funkčními požadavky aplikace.

MATICE FUNKČNÍCH POŽADAVKŮ					
Jméno projektu:		Nástroj pro správu opportunit			
Centrum:		FIT VUT			
Odpovědné osoby:		Michal Žák, Jiří Slabý			
Popis projektu:		Nástroj pro kolaborativní podporu předprojektové fáze projektů			
ID	Rozpad	Popis	Důležitost		
018	5.1	Snaha o viditelnost informace v buňce	N		
019	6	Nutnost jednoduchého exportu dat z aplikace do formátu CSV (Comma-separated values)	V		
020	7	Nutnost jednoduchého importu dat do aplikace	V		
021	8	Smysluplně přednastavené hodnoty (cena, importu, export)	S		

Obrázek 5.2: Funkční požadavky - 2. část

## 5.4 Nefunkční požadavky

Nefunkční požadavky mají za úkol představit vlastnosti a omezení systému, které by aplikace měla mít. Souhrn nefunkčních požadavků je popsán v tabulce 5.3. Systém tabulky je stejný jako u funkčních požadavků, a tedy jednotlivé sloupce plní stejný účel.

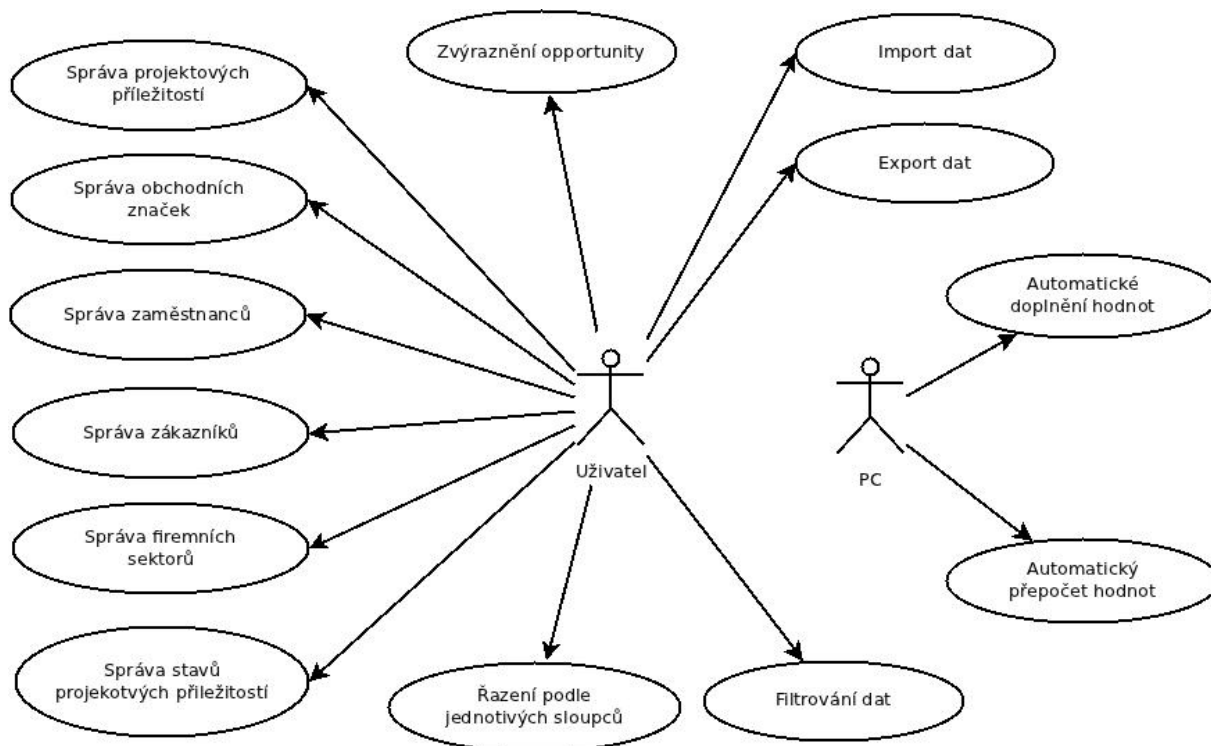
MATICE NEFUNKČNÍCH POŽADAVKŮ					
Jméno projektu:		Nástroj pro správu opportunit			
Centrum:		FIT VUT			
Odpovědné osoby:		Michal Žák, Jiří Slabý			
Popis projektu:		Nástroj pro kolaborativní podporu předprojektové fáze projektů			
ID	Rozpad	Popis	Důležitost		
001	1.	Bez nutnosti instalace	S		
002	2.	Cílová platforma Linux	V		
003	3.	Nesmí vyžadovat nainstalovaný databázový systém	V		
004	4.	Program by měl obsahovat co možná nejméně souborů	S		
005	5.	Bez nutnosti autentizace a autorizace	V		
006	6.	Snaha o samostatnost, bez nutnosti instalovat podpůrné programy	S		
007	7.	Data musí být ukládána okamžitě po vložení (pro případ náhlého ukončení)	V		
008	8.	Rychlá reakce na uživatelské příkazy	V		
009	9.	Optimalizace obrazovky s ohledem na promítání přes dataprojektor	S		

Obrázek 5.3: Nefunkční požadavky

Při vývoji aplikace jsou funkční i nefunkční požadavky základním kamenem pro tvorbu nástroje. V praxi není většinou modelový příklad, kdy toto zadání je naprosto přesné a neměnné, natolik běžný. I při vývoji nástroje pro správu projektových příležitostí došlo k několika změnám či nedorozumění ve výkladu konkrétního požadavku.

## 5.5 Model případů užití

K zobrazení funkční struktury aplikace je použit Use Case model, neboli model případů užití. Jeho primární určení je definovat chování nástroje z pohledu uživatele. V aplikaci pro



Obrázek 5.4: Model případů užití

správu projektových příležitostí jsou dvě kategorie uživatelů:

- Uživatel - řídicí pracovník zodpovědný za správu projektových příležitostí ve firmě.
- PC - systém, který plní automatické činnosti v předem určených stavech.

Tento model má pro grafické zjednodušení shrnuty události jako je vytvoření, editace nebo smazání do celkového názvu správa jednotlivých úseků. S každou uvedenou sekcí tak můžeme dělat všechny základní funkce.

## 5.6 Architektura

Architektura celého nástroje je postavena na několika vrstvách, přičemž vychází z předešlých funkčních i nefunkčních požadavků. Pro přehled uvedeme, jakým způsobem jsou postaveny a vzájemně propojeny.

Nejnižší vrstva z pohledu architektury je vrstva databázová, kde dochází k okamžitému uložení dat do jednotlivých tabulek. Nad databází můžeme najít vrstvu datovou a logickou. Jejím úkolem je vykonat příslušné úpravy a logicky zpracovat vložená data. Poslední je pak vrstva prezentační. V této části dochází k prezentaci dat uživateli. Všechny vrstvy jsou podrobně popsány v následující kapitole.

## Shrnutí

Analýza byla záležitostí několika kroků. Prvním bylo samotné pozorování, jakým způsobem se řeší stávající správa projektových příležitostí. Dále se konalo několik osobních schůzek, kde se tvořil obecný popis aplikace. Ten byl základem tvorby matice funkčních a nefunkčních požadavků. Z těchto požadavků se tvořil model případů užití. Architektura celé aplikace byla rozdělena do tří vrstev (databázová, logická a datová, prezentační).

Popis jednotlivých vrstev a jejich reálná implementace je uveden v následující kapitole. Ta se věnuje i souhrnnému pohledu na chování aplikace.

## Kapitola 6

# Implementace

Vše vychází z požadavků, které byly identifikovány ve firmě a jsou uvedeny v maticích funkčních (5.1, 5.2) a nefunkčních požadavků (5.3). V této části je uvedeno, s využitím jakých nástrojů byla aplikace realizována. Jsou rozebrány nefunkční požadavky, které je vhodné vysvětlit, jak byly promítnuty do realizace. Zároveň je detailně popsána struktura a funkce, jednotlivých vrstev aplikace.

Pro implementaci bylo použito multiplatformního frameworku Qt, který je vyvíjen firmou Nokia. Program je tedy tvořen ve vývojovém prostředí QT Creator v programovacím jazyce C++. Pro tuto kombinaci jsem se rozhodl hned z několika důvodů. Prvním z řady jsou rozsáhlé možnosti při tvorbě programu s grafickým uživatelským rozhraním. Dále pak schopnost budoucí přenositelnosti na další operační systémy je dalším faktorem. V neposlední řadě také kvůli výborně zpracované dokumentaci k celému frameworku. Podrobnější informace i zmíněná dokumentace je k nalezení pod níže uvedeným odkazem.<sup>1</sup>

Mezi nefunkčními požadavky je uvedeno bez nutnosti autentizace a autorizace uživatele, což je v souladu i se snahou o co nejmenší zásahy do uživatelského komfortu. Z toho důvodu nebylo implementováno žádné přihlašování uživatele. Zabezpečení v tomto ohledu není nutné. Jedná se o aplikaci, která je uložena na lokálním disku počítače cílového uživatele. Při spuštění tak okamžitě může pracovat.

Požadavky na samostatnost, a s tím spojený i požadavek na nevyžadování spuštěného databázového systému, byly vyřešeny prostřednictvím lokálního databázového řešení (popsáno později). Primárně byla vyvíjena aplikace pro operační systém linux, je ale později přenositelný do ostatních operačních systémů, a to s příslušným překladem na cílové platformě s přiložením potřebných knihoven. V případě prostředí Windows se jedná o knihovny QtCore4.dll, QtGui4.dll, QtSql4.dll, QtSql4.dll, mingwm10.dll a libgcc\_s\_dw2-1.dll.

Implementace sleduje architektonický návrh zejména jako logickou strukturu, a to především díky několika průběžně měnícím se požadavkům je samotné umístění funkcí především z datové a logické vrstvy i ve vrstvě prezentační. Architektura jako taková je rozčleněna na tři vrstvy. Databázová část se stará o práci s databází. Část uprostřed je zodpovědná za logické a datové operace. Poslední a pro uživatele nejbližší je část zobrazovací, která

---

<sup>1</sup><http://qt.nokia.com/>

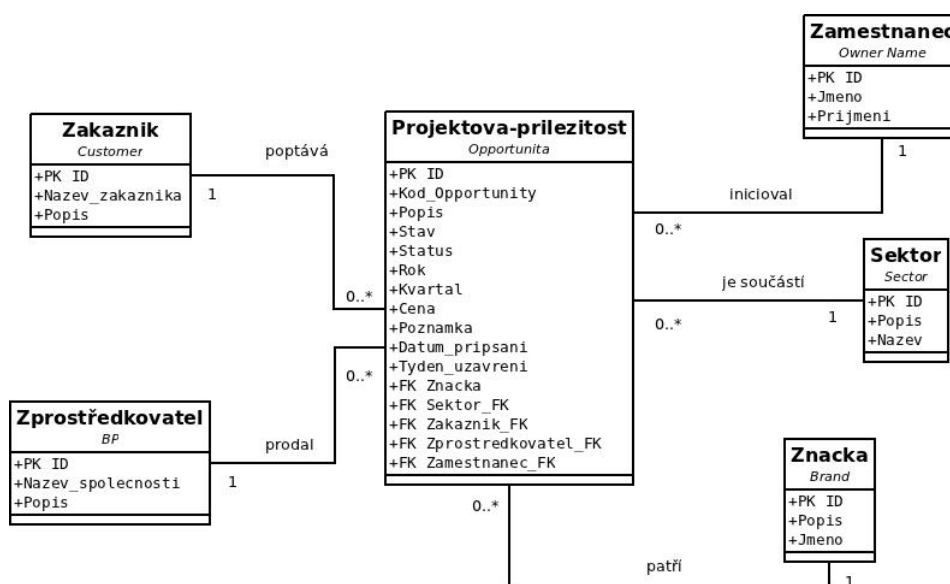
prezentuje data uživateli.

Následuje bližší popis těchto částí, rozdělený do jednotlivých podkapitol v pořadí od nejnižší po nejvyšší architektonickou vrstvu.

## 6.1 Databázová vrstva

První, a tedy nejnižší z pohledu uživatele je vrstva databázová, je nutná vyřešit lokálně, bez pomoci nainstalovaného databázového systému. Podmínka byla splněna pomocí SQLite.[5] Jedná se o volně dostupnou softwarovou knihovnu implementující SQL transakční databázi. Není zapotřebí žádné konfigurace nebo serveru. Samotná databáze je uložena v jediném souboru. Zcela samostatně tak poskytuje plnohodnotný databázový systém pro vyvíjenou aplikaci.

Samotná databáze se jednoduše propojila s Qt. Při startu programu se vytvoří soubor s názvem database, který je pak onou databází. Okamžitě poté je vytvořena databázová struktura pro perzistentní uložení dat. Struktura je popsána pomocí E-R diagramu (Entity-Relationship Diagram), který je zobrazen na obrázku 6.1.



Obrázek 6.1: E-R diagram

Pro kontrolu ukládání dat do databáze a obecný pohled dovnitř byl při vývoji používán prohlížeč SQLite Database Browser, který je dostupný volně ke stažení.<sup>2</sup>

SQLite má jednu zvláštnost, která se však dala bezproblémově využít. Pro své vnitřní potřeby dává tato databáze veškerým záznamům unikátní číslo řádku, které je přístupné pod názvem ROWID. Tato hodnota je autoinkrementální. Toho bylo využito při tvoření databázové struktury. Všechny primární klíče jednotlivých tabulek jsou tak navázány na tuto hodnotu. Díky tomu nebylo nutné vkládat další atribut, ale byla využita stávající

<sup>2</sup><http://www.sqlitebrowser.sourceforge.net>

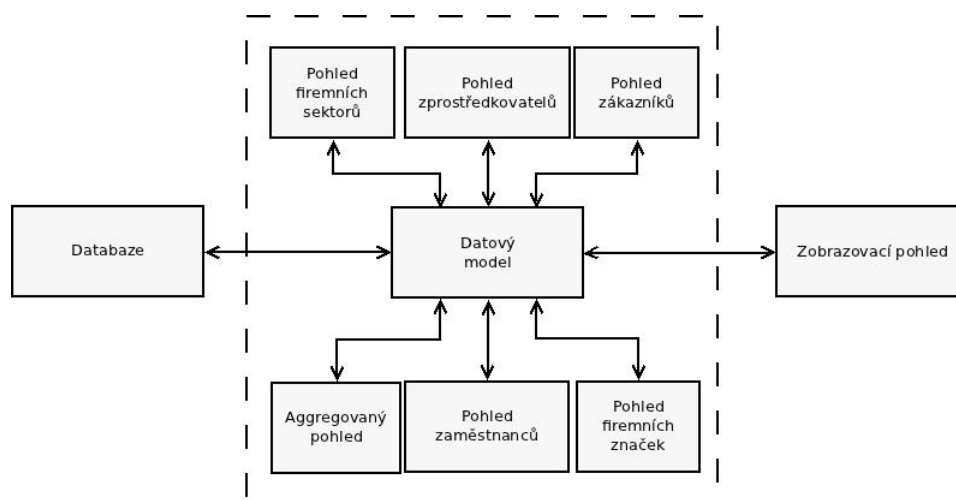


implementace.

## 6.2 Logická a datová vrstva

Operace s daty, zpracovávání, kontrola a vypočítávání ukazatelů je součástí prostřední vrstvy architektury aplikace pro správu projektových příležitostí.

Bylo nezbytné vytvořit nový kompletní datový model. Žádné již připravené řešení neplnilo dostatečně požadovanou funkčnost. Operace s daty, jako je přidávání, mazání nebo editace bylo třeba nově vytvořit, a to do všech databázových tabulek. Pro tento účel je pak využito několika tříd, které využívají dědičnosti a dědí dvě základní funkce `safeToDb` a `loadToDb`. Ty jsou pak přizpůsobeny pro jednotlivé tabulky. Toto řešení je znázorněno na obrázku 6.2. Takto se může jednotně přistupovat k operacím s daty v databázi.



Obrázek 6.2: Princip přístupu k ukládání a načítání dat

Zpracovávání vrstva provádí při nahrávání a ukládání do databáze. Z důvodu, že se v agregovaném pohledu vyskytují sloupce jako zaměstnanec nebo rok, které obsahují data uložená pod více atributy v databázové entitě, je nutné je složit dohromady a zobrazit společně. Z toho vychází otázka kontroly.

Většina údajů se při zadávání kontroluje pomocí regulárních výrazů. Funkce pak pomocí návratové hodnoty předává informaci o správnosti, a podle toho se buď ukládá přípustná hodnota do databáze, nebo se pouze červeně podbarveným textem upozorní uživatel, že je hodnota nepřipustná. V té chvíli jsou data nechána na opravení, a až poté uložena.

Důležitou součástí je výpočet ukazatelů, který je řešen dvěma způsoby. Ukazatele, které mají za úkol zobrazovat hodnoty za aktuální rok bez ohledu na případnou selekci uživatele na obrazovce, jsou čerpány pomocí SQL agregační funkce. Ta od aplikace dostává aktuální rok jako parametr. Následně jsou předány k zobrazení. Hodnoty, které jsou vypočítávány podle uživatelského pohledu, jsou řešeny pomocí seznamu zobrazených řádků a předem definovaných konstant ve sloupci, podle kterého jsou roztříděny do jednotlivých stavů prodeje.

## 6.3 Prezentační vrstva

Grafické uživatelské rozhraní je poslední a pro uživatele nejvíce viditelnou částí, která je v návrhu ve vrstvě prezentační.

Grafické prvky jsou tvořeny pomocí implementované funkce `aggregateView`. Ta generuje všechny prvky, které uživatel může vidět v okně aplikace. Tak, aby byl obraz přizpůsobivý, a tím i lepší pro používání na různých rozlišeních, jsou ovládací prvky vloženy do několika layoutů. Layout je schéma, které se stará o grafické rozvržení jednotlivých prvků a o jejich automatické změny velikosti v závislosti na změně velikosti hlavního okna aplikace. Bylo použito třídy `QGridLayout`, které můžeme dávat souřadnice v pomyslné matici a tím určovat umístění jednotlivých prvků. Ty jsou použity celkem tři, první je pro horní vyhledávací a řadící prvky. Druhý je pak pro zobrazení ročních finančních ukazatelů v jednotlivých firemních značkách. A poslední je využit pro ukazatele, které jsou vypočítávány z těch projektových příležitostí, které uživatel může právě vidět.

Kvůli možnosti mít co největší prostor pro informace o jednotlivých projektových příležitostech, byly finanční ukazatele implementovány na spodní části tak, aby nezabíraly tolik potřebné místo pro záznam projektové příležitosti.

Software, který by byl propracovaný do detailu, avšak náročný na naučení nebo ovládnání, nebude možné nasadit do reálného prostředí. Tento názor byl pro vývoj mimořádně důležitý, proto je celá aplikace tvořena tak, aby intuitivně budila dojem jednoduchosti z pohledu použití, i naučení pracovat s tímto nástrojem.

Pro lepší vzhled bylo použito stylování pomocí kaskádových stylů (CSS - Cascading Style Sheets).

Pro názorný přehled je níže na obrázku 6.3 uveden snímek aplikace s již naplněnými testovacími daty. V horní části je možné vidět souhrn tlačítek, které nesou název jednotlivých sloupců. Jedinou výjimkou je poslední sloupec, který nemá přiřazen název a slouží pouze jako možnost rychlé poznámky pro konkrétní projektovou příležitost. Přímo pod názvy jsou prázdná textová pole, které umožňují vložení vzoru, podle kterého chce uživatel vyhledat pouze ty projektové příležitosti, které začínají na zadanou posloupnost vzorového řetězce. Zde je výjimkou první sloupec, kde bylo z důvodu uživatelského komfortu přistoupeno k možnosti vyhledání jednotlivých kvartálů pomocí prokliku. Při stisku více kvartálů dochází ke sloučování výsledků vyhledávání.

Hlavní částí je pak tabulka, která má za úkol zobrazovat jednotlivé záznamy. Při stisknutí svislého číslování tabulky se příležitost na daném řádku barevně zvýrazní, a toto zvýraznění je nadále uchováno až do doby, než je znovu odebráno pomocí prokliku na stejném místě.

Při zadávání samotné projektové příležitosti je snaha o co nejmenší zatěžování uživatele, proto je při špatném zadání text zbarven do červené barvy. Jedná se o zadávání roku, kde je nutné správně uvést rok a kvartál. Sloupec s názvem FOR má čtyři povolené hodnoty (At risk, Solid, Stretch, Not in Roadmap). U zadávání jména je nutné uvést jméno i příjmení zaměstnance. Poté se kontrolují i správná zadání data, ceny a stavu.

V levém dolním rohu jsou ukazatele, které počítají příjmy za jednotlivé firemní zna-

**Management of project opportunities**

	Q	Customer	Op Number	Op Name	Sector	Brand	FOR	Owner	BP	Stage	Close Override	Forecast Override	Revenue
X	1	2	3	4									
1	2012 Q4	COMPANY D	15PKRD1...	Second software	FTT	WebSphere		Owner D	direct	4	NIR	30.12.2012	1500
2	2012 Q4	COMPANY A	12PKD18FW	First software	GGQ	Data	solid	Owner A	direct	4	Roadmap	22.12.2012	6000
3	2012 Q3	COMPANY B	13PKFD1...	Second software	FTT	Data	Not in Roadmap	Owner B	direct	5	Roadmap	31.12.2012	4500
4	2012 Q3	COMPANY E	16PKD1-8...	First software	FWW	IndSol	Not in Roadmap	Owner E	direct	3	Roadmap	16.12.2012	1500
5	2012 Q2	COMPANY C	14PKD18FW	First software	FWW	cross		Owner A	direct	6	NIR	31.12.2012	3300
6	2012 Q2	COMPANY CH	20PKD18FM	First software	GGQ	BusAna	Not in Roadmap	Owner A	direct	6		31.10.2011	500
7	2012 Q1	COMPANY F	17PKTD1...	Second software	FTT	Rational	Not in Roadmap	Owner F	direct	6	Roadmap	30.12.2012	1000
8	2012 Q1	COMPANY G	18PKD18FW	First software	FTT	Rational	Not in Roadmap	Owner G	direct	8	Roadmap	31.12.2012	1000
9	2010 Q4	COMPANY H	19PK-D18FW	Second software	FTT	Tivoli	Not in Roadmap	Owner A	direct	9	NIR	31.12.2012	500
10	2010 Q4	COMPANY I	21PKD18FW	Second software	FWW	WebSphere	Not in Roadmap	Owner A	direct	7	NIR	16.12.2011	10

<b>Revenue Brands:</b>				<b>Revenue from selection:</b>			
Tivoli:	Q1	Q2	Q3	Q4	Risk:	0	
WebSphere:	0	0	0	0	Stretch:	0	Total: 19810
Rational:	2000	0	0	0			
Lotus:	0	0	0	0			
Data:	0	0	4500	6000			
IndSol:	0	0	1500	0			
BusAna:	0	500	0	0			
Cross Brand:	0	3300	0	0			
Other Brand:	0	0	0	0			
SUM:	2000	3800	6000	7500			

Obrázek 6.3: Aplikace pro správu projektových příležitostí

čky pouze v aktuálním roce. V pravém rohu je pak možné nalézt příjmy, které jsou vypočítávány z projektových příležitostí, které uživatel právě vidí.

Menu má několik nabídek, ve chvíli, když vybereme pohled na firemní značky, zobrazí se nám pohled, který můžeme vidět na dalším obrázku 6.4.

## 6.4 Zhodnocení výsledků a návrh rozšíření

Výsledkem je funkční aplikace, která splňuje všechny nejdůležitější požadavky, které na ni byly kladeny. Koncového uživatele se snaží co nejméně rušit při práci, ikdyž se může zdát, že jde tato vlastnost na úkor neinformovanosti uživatele. Aplikace například neříká, kde je chyba ve vkládaných datech, pouze jej barevně upozorní, že k chybě došlo. Nástroj je optimalizován tak, aby mohl být použit na různých rozlišeních. Všechny tyto funkční vlastnosti jsou ukázkou funkčního řešení.

Aplikace byla testována na ostrých datech a i pro ně se chovala korektním způsobem. Požadavek s nižší prioritou, na schopnost zobrazit veškeré informace o projektové příležitosti bez nutnosti použití horizontálního posuvníku, nemohl být zcela splněn. Přednější v tomto případě byla schopnost zachovat velikost textu a velikost jednotlivých buněk v tabulce.

Je několik rozšíření vzhledem k aplikovatelnosti do praxe. Možností, jak rozšířit uživatelské schopnosti, je doimplementovat správu prostředí tak, aby uživatel mohl rozhodovat, jakým způsobem se budou jednotlivé sloupce třídit. Mohl by si tak vybrat zdali chce, aby se sloupec třídil od začátku nebo od konce abecedy. V nastavení by se mohl vybrat cílové místo uložení databáze, případně nastavit automatickou zálohu dat.

**Management of project opportunities**

Q	Customer	Op Number	Op Name	Sector	Brand	FOR	Owner	BP	Stage	Close Override	Forecast Override	Revenue	
X	1	2	3	4									
Customer						Description							
1	not set up						not set up						
2	COMPANY A												
3	COMPANY B												
4	COMPANY C												
5	COMPANY D												
6	COMPANY E												
7	COMPANY F												
8	COMPANY G												
9	COMPANY H												
10	COMPANY CH												
11	COMPANY I												
12													

Revenue Brands:					Revenue from selection:	
	Q1	Q2	Q3	Q4		
Tivoli:	0	0	0	0	Risk:	5500
WebSphere:	0	0	0	1500		
Rational:	2000	0	0	0		
Lotus:	0	0	0	0		
Data:	0	0	4500	6000	Stretch:	9000
IndSol:	0	0	1500	0		
BusAna:	0	500	0	0		
Cross Brand:	0	3300	0	0		
Other Brand:	0	0	0	0	Solid:	0
SUM:	2000	3800	6000	7500		

Obrázek 6.4: Aplikace pro správu projektových příležitostí - pohled na firemní značky

Další z možných rozšíření je, aby si uživatel mohl skrývat a znovu zobrazovat jednotlivé sloupce podle svého uvážení. Tím by se získal prostor pro zvětšení ostatních sloupců, a tedy i možnost zobrazit více informací v daných buňkách.

Rozšíření, které by pomohlo uživatelskému komfortu je umožnění používání klávesových zkratk, například pro přepnutí mezi pohledy do jednotlivých tabulek. Tento výčet zcela jistě není konečný a je možné, že se funkcionality bude rozšiřovat podle potřeb, které bude detekovány při delším firemním testování.

## Shrnutí

Aplikace vychází z požadavků, které byly ve firmě identifikovány. Byla vytvořena v programovacím jazyce C++ za pomoci frameworku Qt a SQLite. Nejnižší databázová vrsta se stará o vytvoření a komunikaci s SQLite relační databází. Prostřední datová a logická vrsta různě manipuluje s vkládanými daty a tvoří finanční ukazatele. Poslední a tedy nejvyšší je vrsta prezentační. Ta se stará o zobrazení všech informací uživateli. Jedná se o funkční aplikaci, která splňuje hlavní požadavky, které na ni jsou kladeny.

# Kapitola 7

## Závěr

Při důkladném studiu problematiky projektového managementu jsem se utvrdil v přesvědčení, že se jedná o rozsáhlou tematiku, kterou nezle podrobně obsáhnout v rozsahu jedné bakalářské práce. Proto jsem se podrobněji zabýval předprojektovou fází, neboť je z mého podledu jednou z nejzásadnějších fází celého projektového cyklu. Cílem práce bylo vytvořit nástroj pro správu projektových příležitostí pro firmu IBM Česká republika. Tato práce čtenáře uvedla do kontextu projektového managementu jako celku, a pak jej podrobněji seznámila právě s předprojektovou fází.

V protředí firmy probíhala na počátku důkladná analýza stávajícího řešení. Několik etap analýzy vedlo postupně ke slovnímu zadání, které pak bylo základem sestavení matic funkčních a nefunkčních požadavků na aplikaci.

Aplikace byla vytvořena ve volně dostupném frameworku Qt od společnosti Nokia, který vyhovoval vývojovým požadavkům. Nástroj splňuje všechny hlavní požadavky, které na něho byly kladeny. Jako u každého softwarového produktu jsou i zde pravděpodobně mezery, které se ukáží v ostrém provozu, nicméně aplikace je otestována na ostrých datech a bez problémů zvládá úkoly, pro které byla navržena.

Velice si vážím spolupráce s reálným firemním zákazníkem, přestože se objevili situace, kdy se měnilo zadání ať už z důvodu nepochopení nebo změny samotné. Produkt byl tvořen nejen jako povinnost, ale s nadšením jako reálná možnost jak poskytnout efektivnější způsob správy projektových příležitostí

# Literatura

- [1] DOLEŽAL Jan, MÁCHAL Pavel, LACKO Branislav, a kol.: *Projektový management podle IPMA*. 1. vydání, Praha: Grada Publishing, 2009, 507 s., ISBN 978-80-247-2848-3.
- [2] DVOŘÁK Drahoslav, RÉPAL Martin, MAREČEK Martin: *Řízení portfolia projektů*. 1. vydání, Brno: Computer Press, 2011, 198 s., ISBN 978-80-251-3075-9.
- [3] FOTR Jiří, SOUČEK Ivan: *Investiční rozhodování a řízení projektů*. Praha: Grada Publishing, 2011, 408 s., ISBN 978-80-247-3293-0.
- [4] HAJKR Josef, HAVLÍK Jan, PITÁŠ Jaromír, a kol.: Výkladový slovník pojmů [online]. 2009 [cit. 2012-03-16].  
Dostupné z: <http://www.ipma.cz/web/files/IPMA-CzNCB-slovník-pojmu-v3.1.pdf>
- [5] HIPPIE Richard: SQLite [online]. [cit. 2012-03-16].  
Dostupné z: <http://www.sqlite.org/about.html>
- [6] IBM: Rational Focal Point [online]. [cit. 2012-03-16].  
Dostupné z: <http://www-142.ibm.com/software/products/cz/cs/ratifocapoin/>
- [7] KOČÍ Radek, KŘENA Bohuslav: Úvod do softwarového inženýrství [online]. 2010 [cit. 2012-03-16].  
Dostupné z: <https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/IUS-IT/texts?cid=%6837>
- [8] MICROSOFT: Microsoft Project 2010 [online]. [cit. 2012-03-16].  
Dostupné z: <http://www.microsoft.com/cze/project2010/>
- [9] ROSENAU, M. D.: *Řízení projektů*. 2. vydání, Brno: Computer Press, 2003, 344 s., ISBN 80-7226-218-1.
- [10] SCHWALBE, K.: *Řízení projektů v IT*. 1. vydání, Brno: Computer Press, 2007, 720 s., ISBN 978-80-251-1526-8.

# Příloha A

## Obsah CD

- Zdrojové kódy textové zprávy a elektronická verze textové zprávy
- Zdrojové kódy projektu a spustitelná verze
- README - textový soubor s popisem obsahu CD a návodem na sestavení projektu