

Projektová dokumentace pro tvorbu internetových aplikací

Project documentation for internet projects

Bakalářská práce

Tomáš Kuthan

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

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta

Katedra informatiky

2009

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval 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 21.4.2009

Tomáš Kuthan

Anotace

Bakalářská práce se snaží stanovit vzor pro vytváření projektové dokumentace internetových projektů včetně ukázek na konkrétním příkladě a nasazení v praxi. Teoretická část se zabývá základními pravidly, která by měly být dodržovány při tvorbě dokumentace. Zároveň poskytuje také vzorové dokumenty a obecný model postupů. Naproti tomu praktická část se věnuje aplikování teorie do praxe. Jako vzorový příklad projektové dokumentace v praxi byla použita aplikace SuperOPL projektu „Informační a aplikační portál“, na kterém pracuji jako hlavní programátor.

Abstract

The bachelor work gives the pattern for the creation of a project documentation of internet projects including samples on a concrete example and use in practice. The theoretic part describes basic rules which should be followed during the creation of documentation. At the same time it offers sample documents and the basic pattern of technique. On the other hand the practical part attends to application of the theory to the practice. As an example of the project documentation in practice was used the application SuperOPL of the Informative and application portal at which I work as a head programmer.

Obsah

1	ÚVOD.....	6
2	PROBLEMATIKA	8
2.1	KDY VZNIKÁ PROJEKTOVÁ DOKUMENTACE?.....	9
2.2	ROZDĚLENÍ PROJEKTŮ	10
2.2.1	Informační projekty	11
2.2.2	Aplikační projekty	12
2.3	ROLE V PROJEKTOVÉM TÝMU	14
2.3.1	Projektový manažer.....	15
2.3.2	Další role v projektovém týmu.....	16
2.4	NÁSTROJE ŘÍZENÍ PROJEKTU	16
2.5	SPRÁVA DOKUMENTACE	17
2.5.1	Subversion.....	18
2.6	FÁZE PROJEKTU – ŽIVOTNÍ CYKLUS	18
2.6.1	Definice programu – účel a používání	21
2.6.2	Analýza problému a specifikace požadavků zákazníka	22
2.6.3	Návrh řešení zadání	24
2.6.4	Implementace a kódování.....	26
2.6.5	Testování	28
2.6.6	Nasazení	29
2.6.7	Provoz a údržba	30
2.7	VAZBY MEZI FÁZEMI PROJEKTU – MODEL POSTUPŮ	30
2.8	STRUKTURA A OBSAH PROJEKTOVÉ DOKUMENTACE	34
2.8.1	Organizační část	35
2.8.2	Technická část.....	38
3	PROJEKTOVÁ DOKUMENTACE V PRAXI	44
3.1	ÚVOD K PROJEKTU – PŘEDSTAVENÍ	45
3.2	POŽADAVKY ZÁKAZNÍKA.....	46
3.3	KATEGORIE PROJEKTU A POTŘEBNÉ DOKUMENTY – TEORIE.....	46
3.3.1	Organizační dokumentace.....	47
3.3.2	Technická dokumentace	47
3.4	PRAXE – POUŽITÉ DOKUMENTY A DODRŽENÁ PRAVIDLA	48
3.4.1	Organizační dokumentace.....	48

3.4.2	Technická dokumentace	49
4	ROZDÍLY MEZI TEORIÍ A PRAXÍ	55
4.1	ORGANIZAČNÍ ČÁST DOKUMENTACE	56
4.2	TECHNICKÁ ČÁST DOKUMENTACE	57
5	SHRNUTÍ VÝSLEDKŮ PRÁCE.....	58
6	ZÁVĚR	62
7	PŘÍLOHY.....	67

1 Úvod

Podnětem ke vzniku této práce byl předpoklad, že v současné době nejsou v oblasti vývoje internetových a intranetových aplikací dodržována základní pravidla pro vznik projektové dokumentace. Ačkoli publikací, které jsou určeny pro tvůrce internetových projektů je na trhu nepřehledné množství, věnují se z největší části pouze technické stránce věci. Obsahují vzorové ukázky kódů, referenční příručky nebo učební texty jednotlivých technologií jakými jsou například HTML, PHP, CSS a další. Na českém trhu není v současné době k dispozici taková publikace, která by se zabývala projektovou dokumentací z pohledu tvůrců a zahrnula všechny požadované oblasti. Pro takto zaměřenou literaturu je nutné sáhnout zejména po anglicky psaných textech. Jako dobrý lehký úvod do problematiky v českém jazyce je možné zvolit například knihu *Neobyčejně užitečná kniha o webu* od autorky June Cohenové.¹

Tato práce má za cíl zahrnout konkrétní část problematiky, která uspokojí členy projektového týmu ve všech úrovních. V každém týmu je nutné přesně definovat pozici jednotlivých členů a pozicím přidělit konkrétní úkoly. Stejně tak je nutné, aby z pohledu každé pozice byla zpracována taková dokumentace, aby bylo možné předat úkoly či se později vrátit ke konkrétním úkonům a vědět, jaké další kroky budou následovat.

Samotná projektová dokumentace je hlavní úkolem, na který je zaměřena tato bakalářská práce. Je obtížné připravit a sestavit tak univerzální projektovou dokumentaci, kterou by bylo možné použít na každý projekt. Proto

¹ COHENOVÁ, June. *Neobyčejně užitečná kniha o webu*. Jan Kuklínek. Praha : SoftPress, 2004. 372 s. ISBN 80-86497-63-1.

je nejdříve nutné rozdělit projekty podle druhu do zvláštních kategorií. Pro takto specifikované celky je již možné vytvořit soubor dílčích dokumentů, které tvoří projektovou dokumentaci.

Práce popisuje základní pravidla a obecný model postupů při práci v projektovém týmu a definuje tak jasnou strukturu práce na internetových projektech. Samotný model postupů ovšem není zcela jednoznačnou záležitostí a tak je nutné i jej upravit v závislosti na již výše zmiňovaném druhu projektu.

Součástí je také přehled kompletní projektové struktury a dokumentace, dle kategorie projektu, její vzorové vypracování či grafická podoba modelu postupů.

2 Problematika

Projektová dokumentace a řízení projektů jako takové je činnost, která má za cíl zabezpečit bezproblémový běh projektu a dosažení požadovaných cílů v reálném čase a maximálně možné kvalitě. Část oblasti řízení projektů, která se zabývá oblastí vývoje softwaru, používá anglickou zkratku SPM (Software Project Management).²

Prvním problémem, který je nutné vyřešit, je definice pojmu projektové dokumentace. Běžně je možné se setkat s tím, že se jedná o určitý popis toho, co daný program či aplikace dělá. Ale tato odpověď je pravdivá jen z poloviny. Pod projektovou dokumentací je nutné si představit i samotné zaznamenávání průběhu projektu, stanovování cílů, rozdělování úkolů a mnoho dalšího. Pro potřeby této práce bychom mohli používat označení „Technická“ a „Organizační“ dokumentace. Jejich konkrétní obsah bude popsán až v následujících kapitolách, ale co si pod těmito názvy lze představit je zřejmé již z názvu. Pojmem technická dokumentace zastřešuje požadavky na daný systém, jeho návrh i testování.³ Pod touto skupinou se tak skrývá popis použitých funkcí, algoritmů, použitých metod a všeho co se týká samotného

² ŠEVČÍK, Matěj. Řízení softwarových projektů. [s.l.], 2006. 67 s. Masarykova Univerzita, Fakulta Informatiky. Vedoucí diplomové práce RNDr. Jaroslav Ráček, Ph.D. Dostupný z WWW: http://is.muni.cz/th/4266/fi_m/Rizeni_softwarovych_projektu.pdf.

³ DELANEY, Declan, BROWN, Stephen. *DOCUMENT TEMPLATES FOR STUDENT PROJECTS IN SOFTWARE ENGINEERING* [online]. 2002 [cit. 2009-04-19]. Dostupný z WWW: <<http://www.utdallas.edu/~chung/ieee-templates.pdf>>.

provedení daného projektu. Zatímco organizační dokumentace obsahuje například zápisy z jednání, definice úkolů či informace o průběhu projektu.⁴

Zároveň je nutné říci, že nelze specifikovat projektovou dokumentaci tak obecně, aby pojala všechny druhy internetových projektů. Proto se její podoba u různých projektů může lišit.

2.1 Kdy vzniká projektová dokumentace?

Každá z výše popsaných dvou částí projektové dokumentace začíná v jiném stadiu projektu. Zatímco projektová dokumentace technická, se začíná zpracovávat nejdříve od okamžiku, kdy se práce ujmou návrháři či programátoři, s organizační částí se začíná mnohem dříve.

I přesto, že v současné praxi tomu tak nebývá, samotné programování aplikace či programu by mělo tvořit pouze malou část vzniku projektu. Kdy se začíná na organizační části pracovat, to úzce souvisí s fázemi, kterými by měl každý projekt projít. Jak již bylo uvedeno výše, tato dokumentace by měla provázet projekt od samého začátku, až do úplného konce. Z toho vyplývá, že se vznikem projektu se také pokládají základy projektové dokumentaci.

V praxi se můžeme velmi často setkat s tím, že projektová dokumentace je opomíjena a v určité podobě je dopracovávána v průběhu, či až po skončení projektu. V takovém případě organizační dokumentace plně ztrácí svůj nápomocný význam či je dokonce na obtíž. Technická dokumentace pak v takovém případě bývá neúplná nebo dokonce mylná.

⁴ SHELFORD, Thomas J., REMILLARD, Gregory A. *Eal Web Project Management: Case Studies and Best Practices*. [s.l.] : Addison Wesley Professional, 2002. 336 s. ISBN 0321112555.

I když zákazník vždy vyžaduje nejlevnější a z pravidla také nejrychlejší řešení, šetření na řízení projektu není ideálním způsobem a v konečném důsledku způsobí zvýšení nákladů při dopracovávání dokumentace a prodloužení termínů při úpravách na již nasazeném systému.

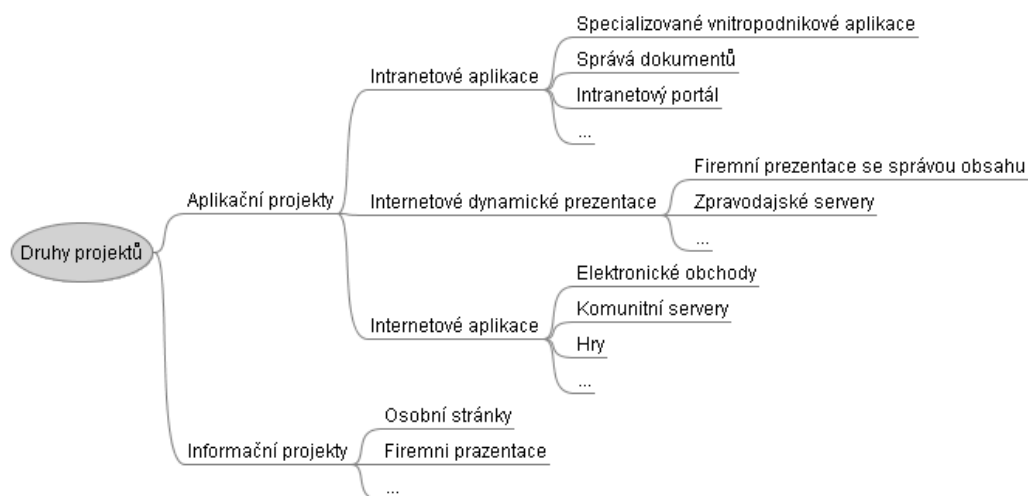
2.2 Rozdělení projektů

Struktura a samotný obsah projektové dokumentace se liší v závislosti na dokumentovaném projektu. Zjednodušeně lze říci, že budou potřebné jiné dokumenty u prezentace, která má za cíl pouze informovat o činnosti firmy, a jiné například u internetového obchodu. Druhy internetových projektů je tedy možné rozdělit na takové, které mají pouze sdělovat informace, a na ty, které vykonávají určitou činnost či mají aplikační část.

Tento způsob rozdělení projektů je velmi vhodný zejména z pohledu obsahu projektové dokumentace. Internetové projekty všeobecně jsou velmi obsáhlé a v případě, že bychom definovali takovou dokumentaci, která by měla být aplikovatelná na libovolný projekt, její obsah by byl až příliš obecný.

Základní dvě kategorie je možné i nadále rozdělit na podskupiny, ovšem v tomto případě nejsou rozdíly v dokumentaci a průběhu projektu tak zásadní, jako je tomu u základního dělení.

Zásadní rozdíly je nutné hledat zejména v oblasti analýzy, vývoje a testování. Konkrétní rozdíly budou rozebrány vždy u daného dokumentu či fáze projektu.



Obrázek 2.1: Druhy projektů

2.2.1 Informační projekty

První skupinu internetových projektů tvoří taková díla, která jsou zaměřena především na sdělení informace. Pod pojmem projekt není možné hledat pouze elektronické obchody, vyhledávače nebo například komunitní servery.

I firemní prezentace, která má za cíl pouze zveřejnění pracovní náplně firmy a kontaktů, je pro tvůrce projektem. Takový projekt, který je tvůrcem naplněn daty a dále se s ním již nepracuje, pak s sebou nepřináší příliš obsáhlou projektovou dokumentaci a některé části odpadají úplně. Mezi takové může patřit například analýza problému.

Název této skupiny je ovšem poněkud zavádějící. Pod informačním projektem je možné si představit například prezentaci města. Na první pohled se může jednat o stejný příklad, jako byl uveden v předcházejícím odstavci, ale není tomu tak. V okamžiku, kdy si zákazník představuje určitou samočinnost, projekt spadá do kategorie druhé. Samočinností aplikace se může rozumět například správa obsahu přímo zadavatelem.

Pro tuto skupinu projektů pak převládá dokumentace, která se zaměřuje převážně na samotné provedení. U projektů této kategorie je kladen důraz zejména na prvky, jako je:

- Design
- Reprezentace
- Množství podané informace

To ovšem neznamená, že by řešení těchto úkolů mělo být časově či obsahově méně náročné. Samozřejmě, že tyto problémy a úkoly s sebou může přinést i projekt aplikační, ale zejména u prezentací, které mají za úkol zaujmout nebo upoutat návštěvníkovu pozornost, může být kladen velký důraz na grafické provedení a podmanivé rozmístění jednotlivých prvků. V takových případech mohou probíhat marketingové průzkumy, studie trhu a další potřebné kroky k dosažení maximálního výsledku.⁵

Mezi základní dokumenty projektové dokumentace této skupiny projektů budou patřit samozřejmě zápisy z jednání, nebude chybět rozdělení úkolů a sledování jejich plnění. Technická dokumentace je v případě těchto projektů výrazně chudší na obsah než u aplikačních projektů. Samotné provedení u statických prezentací nebývá příliš složité a tak se dokumenty vážou spíše k souborové struktuře nebo provázanosti jednotlivých sekcí.

2.2.2 Aplikační projekty

Převážně na projekty této kategorie je zaměřena tato práce. Zatímco v předchozím případě nebylo spektrum dokumentace příliš široké, zde je tomu

⁵ BURDMAN, Jessica. *Collaborative Web Development: Strategies and Best Practices for Web Teams*. [s.l.] : [s.n.], 1999. ISBN 0201433311. s. 9.

přesně naopak. Aplikační projekty vyžadují důkladnou dokumentaci a to jak tu organizační, tak i tu technickou.

Tuto poměrně obsáhlou skupinu lze i dále rozdělit na další podskupiny, které s sebou přináší i rozdíly v obsahu dokumentace. Tyto rozdíly se ovšem nevážou k přítomnosti některých dokumentů, ale spíše k jejich obsahu. Není proto nutné pro jednotlivé podskupiny vytvářet samostatné struktury. Rozdělení je možné například následujícím způsobem:

- Intranetové aplikace
- Internetové dynamické prezentace
- Internetové aplikace

Pod tuto kategorii projektů lze zařadit téměř vše, co nespadá do kategorie první. Patří sem teda například elektronické obchody, zpravodajské servery, seznamovací servery a mnoho dalšího.

Ať už se jedná o internetovou či intranetovou aplikaci, je nezbytné udržovat stále aktuální dokumenty jak organizační, tak i projektové části. U předchozí kategorie je většinou kladen důraz na obsah a vzhled. V tomto případě je pro všechny projekty této kategorie kladen důraz na funkcionalitu. Co se obsahu a designu týká, důležitost se u jednotlivých podskupin liší. Zatímco u jednoúčelové intranetové aplikace, či intranetové aplikace obecně bude kladen důraz zejména na jednoduchost, rychlost a funkčnost, u internetového zábavního portálu půjde krom správného fungování jistojistě také o moderní, přívětivé a intuitivní grafické prostředí.

Krom základních dokumentů, které je nutné použít i u předchozí skupiny, jako jsou zápisy z jednání nebo rozdělení úkolů, je nutné dokumentaci této skupiny doplnit i o další části, které jsou nezbytné. Mezi ně patří například dokumentování fází projektu, které jsme mohli u informačních projektů vynechat či jim nevěnovat přílišnou pozornost. Příkladem může být analýza

požadavků zákazníka, která v tomto případě musí být prováděna mnohem důkladněji. To samé platí také třeba pro testování.

2.3 Role v projektovém týmu

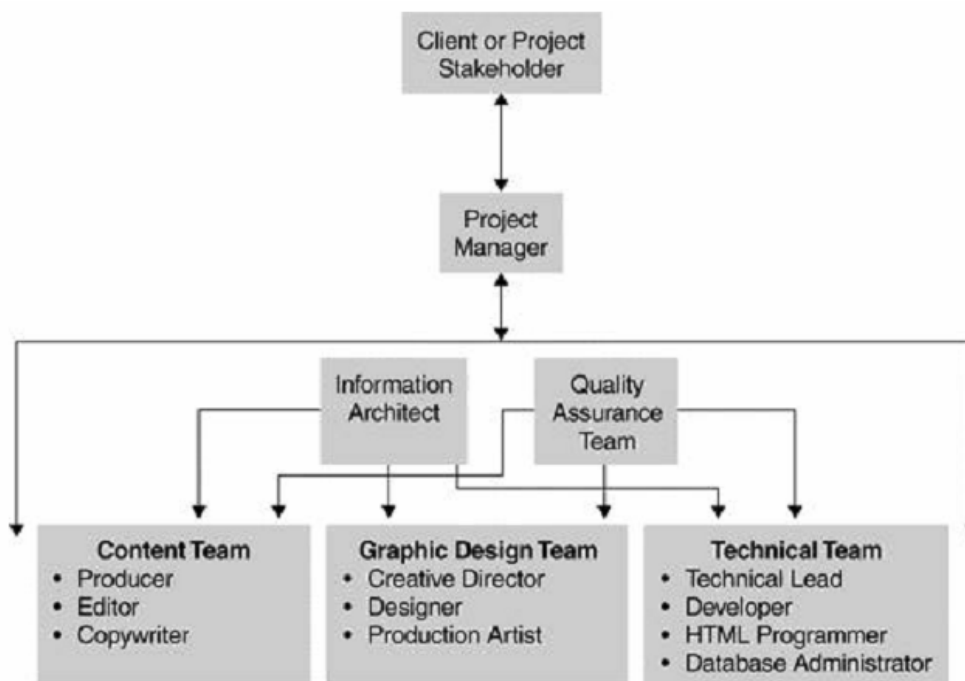
Ať je projektový tým velký nebo malý, musí v něm být zastoupeny různé role. U rozsáhlých projektů, či specializovaných firem může být konkrétní role zastoupena i několika osobami. Naopak u menších projektů, či menších projektových týmů je obvyklé, že jedna osoba zastupuje i více rolí.

V každém týmu, a je jedno, zda je projektový nebo třeba fotbalový, musí existovat vedoucí. Ten se stará zejména o plánování, řízení a kontrolu projektu. Další osoby v týmu jsou jeho členové a i ti mají své konkrétní role. Pokud nebude brát v úvahu role, které záleží na chování jednotlivých osob a budeme se věnovat rolím dle náplně práce, pak by v softwarovém týmu neměl rozhodně chybět například:

- Tvůrce obsahu
- Grafik
- Analytik
- Návrhář systému
- Developer (vývojář, programátor)
- Technický správce (databázový administrátor, správce serveru)
- Tester

Typický projektový tým pro tvorbu webových projektů by měl být rozdělen do tří částí dle náplně jejich činnosti. Tyto základní skupiny jsou:

obsah, grafika a technické provedení. Nad těmito skupinami pak stojí projektový manažer, viz diagram 2.2.⁶



Obrázek 2.2: Rozdělení projektového týmu

2.3.1 Projektový manažer

Projektový manažer je osoba, která se pohybuje napříč všemi oblastmi projektu. Práce projektového vedoucího vyžaduje nejen znalost oboru, ale také například schopnost jednat s lidmi. Hlavní náplní jeho práce je zejména:

- Komunikace mezi týmem a klientem
- Definice technického, grafického a funkčního rozsahu projektu

⁶ SHELFORD, Thomas J., REMILLARD, Gregory A. Real Web Project Management: Case Studies and Best Practices. [s.l.] : Addison Wesley Professional, 2002. 336 s. ISBN 0321112555.

- Vytváření a údržba projektové dokumentace
- Stanovování a udržování časového plánu projektu
- Stanovování milníků
- Řízení jednání
- Rozdělování úkolů a dohled nad jejich plněním
- Motivace a vedení týmu

2.3.2 Další role v projektovém týmu

Náplň práce dalších zmiňovaných rolí projektového týmu je intuitivní a proto není nutné se jí zabývat. Z pohledu projektové dokumentace by z každé pozice měla vznikat taková část dokumentace, která se váže k náplni práce dané role. V projektovém týmu neexistuje osoba, která by pro svou činnost nemusela vést dokumentaci.

2.4 Nástroje řízení projektu

I přesto, že se může zdát, že nástroje řízení projektu nejsou přímou součástí projektové dokumentace, jejich obsah nedílnou součástí dokumentace je. Nástrojů pro řízení projektu je hned několik a liší se zejména svojí komplexností a lze je rozdělit do třech základních kategorií.⁷

Nejzákladnější a nejjednodušší nástroje jsou takové, které primárně nejsou určeny pro řízení projektu, ale jsou k těmto účelům použitelné. Příkladem může být například Microsoft Excel použitý k vedení tabulky úkolů, přehledů a dalších základních informací.

⁷ ŠEVČÍK, Matěj. Řízení softwarových projektů. [s.l.], 2006. 67 s. Masarykova Univerzita, Fakulta Informatiky. Vedoucí diplomové práce RNDr. Jaroslav Ráček, Ph.D. Dostupný z WWW: http://is.muni.cz/th/4266/fi_m/Rizeni_softwarovych_projektu.pdf. s. 11

Pokročilejším nástrojem jsou pak nástroje pro řízení projektu. Mezi takové patří například Microsoft Project. Takovéto nástroje nabízí přímo rozhraní, které je pro řízení větších projektů nezbytné

Poslední kategorii tvoří specializované nástroje, které jsou používány zejména pro rozsáhlé projekty s velmi specifickými požadavky.

Které konkrétní nástroje se budou pro řízení projektu používat, závisí vždy pouze na volbě manažera. Každý má oblíbené nějaké nástroje a proto není možné podle velikosti projektu říci, které nástroje se budou používat.⁸

2.5 Správa dokumentace

Pečlivé vedení projektové dokumentace ještě neznamená zdárný konec. I v případě, že je dokumentace vedena v plném rozsahu, ale její správa nemá jasný řád, může to být stejné, jako kdyby žádná dokumentace neexistovala. Způsobů vedení dokumentace je hned několik a vždy záleží na použitém modelu správy projektu.

Jako dostačující model lze samozřejmě použít úložiště na pevném či síťovém disku doplněné o soubor s přehledem jednotlivých dokumentů. S tímto způsobem je možné se setkat u většiny malých projektů a pro většinu z nich se zároveň jedná o ideální řešení.

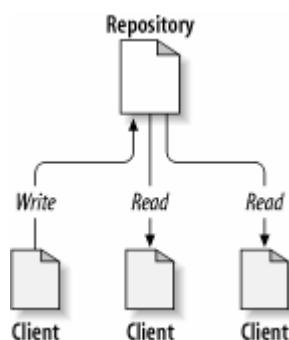
U větších projektů či rozsáhlejších týmu ovšem vzniká s projektovou dokumentací problém stejný, jako u samotného přístupu ke zdrojovým kódům projektu. V takovém případě je na místě použít specializované nástroje. Pro tyto potřeby vznikají programy pod společným označením SCM (Source Code

⁸ BURDMAN, Jessica. *Collaborative Web Development: Strategies and Best Practices for Web Teams*. [s.l.] : [s.n.], 1999. ISBN 0201433311. s. 17-18.

Management) určené pro řízení zdrojových kódů projektu a dokumentů s tím souvisejících.

2.5.1 Subversion

Jedním z nástrojů kategorie SCM je program Subversion (SVN). Jedná se o následníka programu CVS pro správu a verzování zdrojových kódů. Systém pracuje na principu tzv. repository (úložiště) a working copy (pracovní kopie), jak ukazuje obrázek 2.3.⁹ Umožňuje tak pracovat více členům týmu na stejném projektu. Tento systém se hodí zejména u dokumentace technické, kde se předpokládá vznik nových verzí a editace více členy týmu.



Obrázek 2.3: Subversion

2.6 Fáze projektu – životní cyklus

Již bylo nastíněno, že každý projekt prochází několika fázemi, tzv. softwarovým procesem. „Softwarový proces je po částech uspořádaná množina kroků směřujících k vytvoření nebo úpravě softwarového díla. Každá z částí vývoje projektu má své důvody a náležitosti.“¹⁰ Stejně tak, jako člověk nemůže přeskočit fáze svého vývoje od novorozeněte, přes batole po dospělého jedince,

⁹ SVN tutoriál [online]. 2007 [cit. 2009-04-15]. Dostupný z WWW:

<http://merlin.fit.vutbr.cz/wiki/index.php?title=SVN_tutori%C3%A1l>.

¹⁰ VONDRÁK, Ivo. Úvod do softwarového inženýrství. 1.1. aktualiz. vyd. Ostrava, 2002. s. 8.

pak by ani projekt neměl vynechat žádnou ze svých fází a pokud by tak učinil, s největší pravděpodobností by to mělo vliv na jeho další vývoj. Jako základ pro definici fází projektu lze použít model zvaný ADDIE. Tento model postupů definuje 5 základních fází, které se v současné době používají zejména v oblasti tvorby výukových materiálů, ale v upravené podobě je lze aplikovat i pro potřeby vývoje softwaru. Mezi základní fáze definované tímto modelem patří **A**nalysis, **D**esign, **D**evelopment, **I**mplementation a **E**valuation, tedy analýza, návrh, vývoj, implementace a hodnocení. Fáze, kterými prochází softwarový projekt, lze definovat pouze přizpůsobením tohoto modelu a vzniká následující:

1. Definice programu – k čemu bude určen
2. Analýza problému a specifikace požadavků klienta
3. Návrh řešení
4. Implementace, kódování
5. Testování
6. Nasazení
7. Provoz a údržba

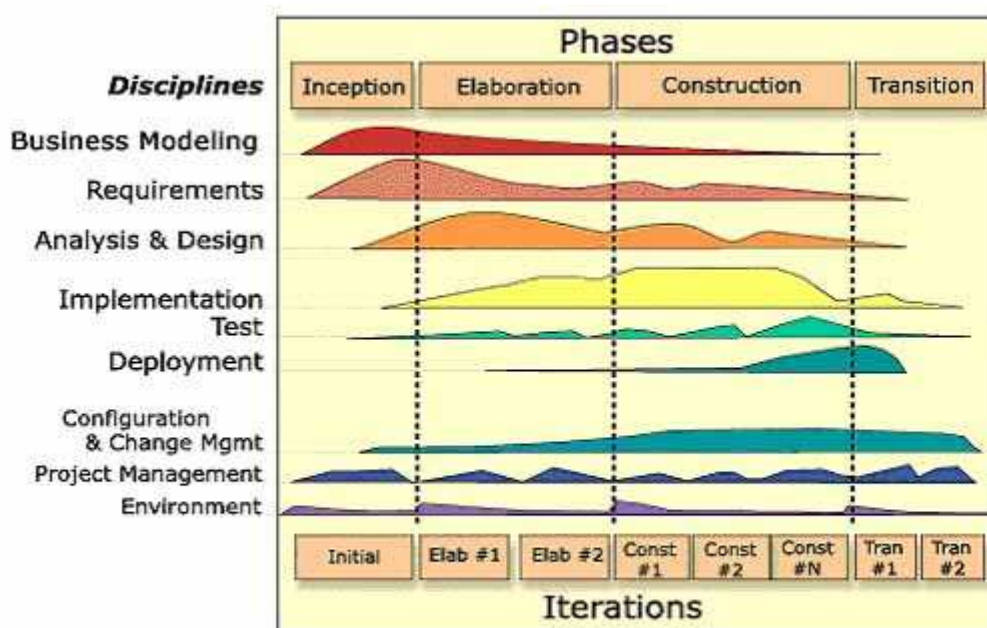
Tyto fáze jsou stejné jak pro klasické aplikace, tak i pro internetové projekty, na které je zaměřena tato práce. Oproti klasickému programování se ovšem mohou lišit z hlediska jejich významu a obsahu. To bude rozebráno konkrétně v následujících kapitolách.

Popisu tohoto celého procesu se také věnuje model zvaný RUP (Rational Unified Process) vyvíjený pod hlavičkou Rational Software.¹¹ Cílem tohoto

¹¹ Rational Software je jedna z pěti softwarových značek IBM, která zahrnuje řešení a nástroje určené k usnadnění vývoje aplikací.

modelu je vytvoření produktu v maximální kvalitě, kterou požaduje zákazník, a zároveň ve stanovené době a s přesně daným rozpočtem.

Délky trvání jednotlivých částí se liší a není ani možné definovat, kolik času by měla jaká část zabrat. Vše záleží také na kategorii daného projektu. Například jinak dlouhá bude doba testování, pokud půjde o bankovní aplikaci, elektronický obchod nebo třeba o osobní stránky. Určitý časový rozvrh jednotlivých fází ovšem nabízí model RUP. Schématické vyjádření ukazuje obrázek 2.4.¹²



Obrázek 2.4: Vyjádření procesu RUP

¹² Rational Software - IBA CZ [online]. 2008 [cit. 2009-04-13]. Dostupný z WWW: <http://www.ibacz.eu/-Rational-Software->.

2.6.1 Definice programu – účel a používání

Dřív, než se začne s jakýmkoliv vývojem, je samozřejmě nutné vědět, co se má vyvíjet. Tento krok, byl měl být iniciován ze strany zadavatele (zákazníka) a zároveň je plně v jeho režii.

Zadavatel by měl v tuto chvíli zpracovat jakýsi úvod do problému samotného, ale zároveň také sepsat „životní cyklus“ řešeného problému a situaci kolem něj. Pro správné vypracování daného projektu by se realizační tým měl vžít do situace a pochopit současné fungování. Zákazník by tudíž měl stručně popsat, jak je daný problém řešen aktuálně, jakých dalších činností se dotýká, případně jaké výhody by mělo nové řešení poskytnout.¹³

Z pohledu lepší spolupráce se zadavatelem je vhodné na straně zákazníka získat osobu, která novým řešením ušetří čas, práci, či je jen zapálena do nového řešení. Takovýto člověk výrazně pomůže při další práci na projektu, ať už se jedná o fázi vývojovou či implementační.

Co se projektové dokumentace týká, tak výstupem první fáze je tzv. „creative brief“, ve kterém zadavatel popisuje výše uvedené.¹⁴ V případě, kdy vykonavatel pracuje na více projektech a nejedná se pouze o jednorázovou akci, je vhodné i zadání klienta dát určitou formu a řád. Projektový tým se na této fázi projektu nijak výrazně nepodílí. Pouze analytik zde slouží jako poradce.

¹³ SHELFORD, Thomas J., REMILLARD, Gregory A. Real Web Project Management: Case Studies and Best Practices. [s.l.] : Addison Wesley Professional, 2002. 336 s. ISBN 0321112555. s. 77 – 104.

¹⁴ SUTHERLAND, Erci. *Requirements for the Creative Brief*. [s.l.] : [s.n.], 2008. 25 s. ISBN 978-1-906440-01-5.

V tuto chvíli je také na čase připravit projektovou strukturu a všechny potřebné dokumenty. Na základě zadání již projektový tým může určit typ projektu a zároveň vybrat a připravit nezbytné dokumenty.

2.6.2 Analýza problému a specifikace požadavků zákazníka

Níže popsané postupy samozřejmě platí pro obě hlavní kategorie internetových projektů (viz kapitola 2.2), ovšem pro informační projekty je situace výrazně jednodušší. Jako ideální příklad nám poslouží statická internetová prezentace firmy. U takového projektu není nutné rozebírat detailní funkcionalitu a zákazníka to ani nezajímá. V takovém případě je více důrazu kladeno spíše na grafickou podobu a ta je ve většině případů až na základní prvky ponechávána grafikovi.

Proto i u jednotlivých fází musíme brát v potaz kategorii daného projektu a k ní přizpůsobit i podobu dokumentace. Pokud cílem projektu je intranetová aplikace, pak je důraz kladen převážně na její funkce, zatímco u prezentace firmy je snaha spíše zaujmout potenciálního zákazníka. Význam této fáze je u informačních projektů zřejmý a tak se dále budeme věnovat projektům aplikačním.

V okamžiku, kdy projektový tým již ví, na čem bude pracovat, dostává se do popředí práce analytika. Ten nejdříve musí prostudovat přání zákazníka a pochopit současné fungování. Tato fáze je v celém životním cyklu projektu velice zásadní a neměla by být podceňována. V případě, že by byla práce v tento okamžik zanedbána, může to negativně ovlivnit, jakým směrem se bude následný vývoj ubírat. Právě zde by měly být rozebrány veškeré požadavky zákazníka na funkčnost a funkcionalitu nového systému, účel používání, možné využití při dalších činnostech a mnoho dalšího.

Pro ilustraci, jak moc je tato fáze důležitá, by mohla posloužit následující ukázka situace, která je v praxi velice častá a v podobné formě ji ve své knize

Software Requirements uvedl Karl E. Wiegers.¹⁵ Předpokládejme, že softwarová firma dostala zakázku na vytvoření systému pro správu přidělování poznávacích značek. V případě, že přípravné fázi nebyla věnována pozornost, mohl by telefonát od zákazníka vypadat následovně:

„Dobrý den, máme problém s tím novým programem.“ „Dobrý den, a co máte za problém?“ „Zákazník si chce změnit poznávací značku, ale ten program to nechce přijmout, jen když se změní i vlastník vozidla.“ „Aha, ale to jsme nevěděli, že si lidé mohou měnit poznávací značku, jak se jim chce. Neříkali jste nám to.“ „Ale my to potřebujeme! Kdy to opravíte, když jste to udělali tak špatně?“ „Ale my nic špatně neudělali, nechtěli jste to po nás...“

Takováto situace může nastat velice snadno a úplně se jí vyvarovat nemusí být zrovna jednoduché. Pečlivá analýza ovšem může riziko výrazně snížit. Pro důkladnou analýzu celého procesu je možné použít metodu BPM (Business Process Modeling), která ze 3 pohledů zkoumá daný proces.¹⁶

Během této fáze vznikají první zásadní dokumenty, které by v projektové dokumentaci neměli chybět. Pro případné reklamace je dobré zdokumentovat veškerá přání a požadavky klienta. Mezi tyto dokumenty patří například:

- Požadavky a přání zadavatele
- Technické možnosti
- Předpokládaný časový harmonogram

¹⁵ WIEGERS, Karl E. Software Requirements. Redmond : Microsoft press, 2003. 516 s. ISBN 0-7356-1879-8.

¹⁶ ERIKSSON, Hans-Erik, PENKER, Magnus. *Business Modeling with UML: Business Patterns at Work*. [s.l.] : [s.n.], 2000. 459 s. ISBN 0471295515.

2.6.3 Návrh řešení zadání

Zatímco předchozí části znamenaly velkou součinnost projektového týmu a zákazníka, během návrhu řešení se rozdělení práce již začíná měnit a zákazník se v tuto chvíli pouze vyjadřuje ke vzniklému návrhu. I v tuto chvíli je nutné fázi rozdělit podle kategorie projektu, protože podle ní se liší samotný obsah a rozdělení práce.

Návrh řešení informačních projektů z pravidla bývá zaměřen pouze na grafický návrh. Jak již bylo uvedeno, u informačních webů se jedná především o formu podání informace a funkcionality až na druhém místě. Někdo může namítnout co vlastně řešit za funkcionality na prezentačních stránkách. Ani u firemních prezentací by ale neměl být opomenut například způsob pohybování po prezentaci, navigace či logické uspořádání jednotlivých částí.

Dříve, než je zákazníkovi předložen grafický návrh, přichází na řadu tzv. wireframe neboli „drátěný model“. Tímto pojmem se označuje zjednodušený model nebo návrh definující funkci a obsah stránek webu.¹⁷ Jak vyplývá z názvu, wireframe není návrhem konkrétní grafické podoby. Definuje ale textový i grafický obsah, rozmístění funkčních prvků, navigaci či znění nadpisů, klíčových slov a tlačítek.

Drátěný model může existovat ve čtyřech základních podobách, ovšem v žádné z nich se nepoužívají ani barvy, ani obrázky. Teprve na základě tohoto modelu připravuje grafik konkrétní grafickou podobu. Ta sice vychází z tohoto hrubého návrhu, není na něm ale úplně závislá. Základními typy jsou:

- Textový wireframe
- Stručný či blokový wireframe

¹⁷ LINOWSKI, Jakub. *Wireframes Magazine* [online]. c2009 [cit. 2009-04-15]. Dostupný z WWW: <<http://wireframes.linowski.ca/>>.

- Podrobný wireframe
- Hypertextový wireframe

Rozdíl mezi těmito typy je zřejmý z obrázku 2.5.¹⁸ Textový wireframe popisuje jednotlivé části pouze slovní formou a jedná se pouze o výčet použitých prvků. Stručný či blokový wireframe již ukazuje velikost a pozici jednotlivých částí a přináší tak více vypovídající informace. Dalším stupněm drátěného modelu je podrobný wireframe. Ten obsahuje přesné definice a velikosti textů, obrázků a funkčních prvků. Pokud je tento podrobný model provázán odkazy s modely dalších stránek, vzniká wireframe hypertextový.



Obr 2.5: Typy wireframe

Wireframe je pouze model a proto není potřeba jej vytvářet v nějakém specializovaném nástroji. Pro základní podobu modelu stejně dobře poslouží MS Word jako třeba MS Powerpoint. U podrobných či hypertextových návrhů se již vyplatí použít software, který je k návrhům určený, jako například MS Visio.

U aplikačních projektů můžeme samozřejmě řešit popsané problémy ohledně grafické podoby a struktury, ale pokračujeme ještě dál a musí být

¹⁸ Wireframe webu [online]. c1999-2009 [cit. 2009-04-15]. Dostupný z WWW: <http://www.symbio.cz/slovník/wireframe-webu.html>.

vytvořen model po stránce funkcionality. Ten ve většině případů vzniká pomocí moderních návrhových metod, tzv. CASE nástrojů.

CASE je zkratkou z anglického Computer Aided Software Engineering, což lze přeložit jako počítačem podporované softwarové inženýrství. Samotné CASE nástroje jsou určeny zejména pro modelování IT systémů pomocí diagramů (UML), generování zdrojového kódu z modelu či vytváření dokumentace z modelu. Díky těmto nástrojům je možné usnadnit práci jednotlivým členům týmu nebo díky grafického návrhu lépe pochopit fungování a vazby.¹⁹

Tato fáze je pro samotný projekt velice důležitá a neměla by být podceňována. Právě v tuto chvíli jsou vyjasňovány veškeré detaily fungování vznikajícího projektu. V případě, že během následujících fází je nalezen nedostatek, tak je nutné se vrátit právě do tohoto kroku a vytvořit nový model.

Návrhový diagram by samozřejmě měl být součástí vznikající projektové dokumentace a zároveň odsouhlasen zadavatel. Vzhledem k přehlednosti takového návrhu je toto ideální způsob pro odhalení případných nedostatků ještě před samotným vývojem a zároveň slouží jako dobrý důkazní materiál v případě reklamací.

2.6.4 Implementace a kódování

Tuto část je možné také nazvat manuální prací. V tuto chvíli přichází na řadu programátoři, kteří do zdrojového kódu přepíší navržený model.

Velmi často se lidé domnívají, že v tuto chvíli vznikají všechny části programu, ale to zdaleka není pravda. Jak již bylo uvedeno, v této chvíli vzniká sice samotná aplikace (její zdrojový kód), ale na základě předem rozhodnutého

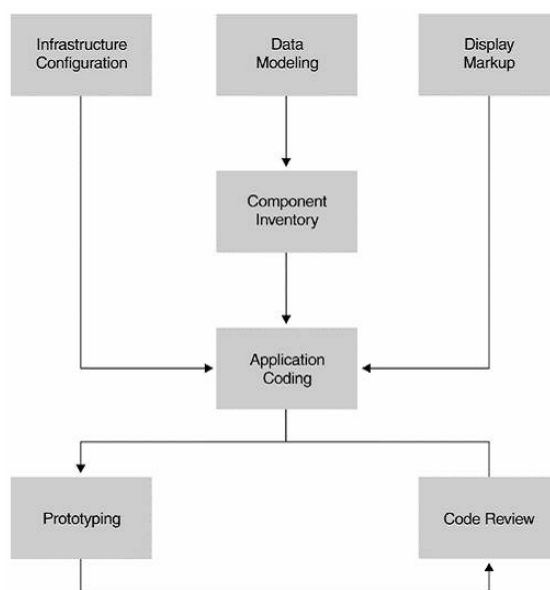
¹⁹ BROWN, Alan W., et al. *Principles of CASE Tool Integration*. [s.l.] : [s.n.], 1994. 288 s. ISBN 0195094786.

modelu. Lze tedy tuto fázi brát jen jako doplnění zdrojového kódu do existující kostry.

Z uvedeného tedy vyplývá, že z pohledu dokumentace se zaměřujeme zejména na dokumentaci technickou. Pro případné úpravy zdrojového kódu, které ve většině případů v bližší či delší době přijdou, je nutné mít pečlivě zdokumentované používané funkce, algoritmy, vazby zdrojového kódu.

Pokud zpracovatel poskytuje zadavateli aplikaci, nikoli zdrojové kódy, pak vypracovaná technická dokumentace zůstává zadavateli skryta a slouží pouze projektovému týmu. To ovšem nemění důležitost takto zpracovávaného dokumentu a i v takovém případě je nutné zachovat řád a pravidla.

Celá vývojová fáze by měla mít jasný řád a pravidla. Nejen, že je nutné dodržovat postupy celého projektu, ale i jednotlivé fáze se skládají z více částí. Jak by měla vypadat vývojová část, ukazuje obrázek 2.6.²⁰



²⁰ SHELFORD, Thomas J., REMILLARD, Gregory A. Real Web Project Management: Case Studies and Best Practices. [s.l.] : Addison Wesley Professional, 2002. 336 s. ISBN 0321112555. s. 192.

Obr 2.6: Průběh vývojové fáze

2.6.5 Testování

Testování znamená ověření, že předmět testu splňuje požadavky na něj kladené.²¹ I přesto, že testování by v rámci životního cyklu podle staré teoretické poučky mělo zabrat minimálně tolik času, co fáze předchozí, v praxi se s takto důkladným testováním lze setkat jen málokdy.

Samozřejmostí zůstává, že čas věnovaný testování je přímo úměrný možným rizikům a důležitosti aplikace. Již několikrát zmiňovaná internetová prezentace firmy jistě projde testováním, zda vše potřebné funguje jak má, ale možných situací, které mohou při fungování nastat je pouze malé omezené množství.

U aplikačního projektu se situace komplikuje a je nutné počítat i s přístupy a zadáními, které nejsou úplně standardní. Proto je vhodné mít testování zdokumentované, část po části, aby se na jednotlivé možnosti nezapomnělo. Z pohledu projektové dokumentace je důležité testování evidovat spíše z důvodu kontroly průběhu a jak již bylo uvedeno také proto, aby testování bylo důkladné a kompletní.

Samotný průběh testování vzhledem ke svému umístění na konci celého projektu bývá touto skutečností silně ovlivněno. Právě v tuto dobu se již důrazněji sleduje dodržování termínů a rozpočet celého projektu. Pro finální podobu vznikající aplikace to může přinést i nemalé problémy.

²¹ Je testování software věda?. *DSM*. 1.1.2005, č. prosinec, s. 5. Dostupný z WWW: <<http://www.pdqm.cz/Publications/TestVeda.pdf>>.

2.6.6 Nasazení

Vyzdvihovat, že v pokročilejších fázích se rozchází přístup k informačnímu a k aplikačnímu projektu již asi není třeba. Fáze nasazení se od předchozích nijak neliší a tak může být nasazení hotového projektu otázkou pár chvil nebo i týdnů.

Dobu nasazení mohou ovlivnit zejména různá propojení na další systémy. U webových stránek, které mají pouze informační charakter, externí propojení neočekáváme, a tudíž celý proces bývá bezproblémový.

U aplikací, které vyžadují komunikaci s dalšími zdroji, se situace komplikuje. Pro bezproblémové nasazení nového systému je potřeba všechna propojení otestovat a připravit na nový systém. Externí zdroje v tomto případě nemusí být pouze zdroje systémové, ale například i lidské a pak je například nutné zajistit i proškolení takových lidí.

Pokud se jedná o důležitou část systému, nejspíše již existuje krizový plán pro případ, že by část systému přestala fungovat. Pokud plán neexistuje, neměla by v projektové dokumentaci chybět zmínka o možnostech dopadu změny na nový systém a možných řešeních. I přesto, že vývoj aplikace byl dlouhý, samotné nasazení je klíčové. Běžně je možné se setkat s kvalitními rozsáhlými systémy jako například SAP, kde samotná implementace do konkrétního prostředí nebyla provedena zrovna šťastným způsobem a proto je nutné implementaci nijak nepodceňovat.²²

²² SHELFORD, Thomas J., REMILLARD, Gregory A. *Eal Web Project Management: Case Studies and Best Practices*. [s.l.] : Addison Wesley Professional, 2002. 336 s. ISBN 0321112555. s. 234 – 245.

2.6.7 Provoz a údržba

Na konci života každého projektu se předpokládá, že bude nasazen do provozu a používán. Připravit projekt v takové podobě, aby se již neměnil, je téměř nemožné a údržba bude nutná v každém případě.

Pokud byl celý průběh vývoje pečlivě zdokumentován, pak je při případných změnách situace výrazně jednodušší a pro tvůrčí tým nebývá zpravidla problém jakoukoli součást opravit či pozměnit.

V každém případě je pro veškeré změny projektu nutné mít stanovená jasná pravidla a postupy. To platí jak pro změny během celého projektu, tak i po jeho skončení. Řízení změn a využití jejich potenciálu se zabývá obor, který se označuje jako „change management“. Jedná se o manažerskou disciplínu, která definuje, popisuje, klasifikuje, hodnotí, schvaluje, realizuje a ověřuje změny.²³

U každé změny je nutné zhodnotit, na jaké další součásti bude mít vliv, jaká bude časová náročnost takové změny, či jestli je možné takovou změnu vůbec zpracovat. Pokud se jedná o změnu ještě před předáním celého projektu, je nutné také domluvit případné prodloužení termínů a přepracování časového plánu projektu.

2.7 Vazby mezi fázemi projektu – model postupů

Základní postupy a vazby mezi jednotlivými částmi byly již nastíněné v předchozí kapitole a jednoduchý pohled by mohl vypadat tak, že jednotlivé fáze na sebe navazují. Tento model se označuje jako vodopádový.

²³ ŠEVČÍK, Matěj. Řízení softwarových projektů. [s.l.], 2006. 67 s. Masarykova Univerzita, Fakulta Informatiky. Vedoucí diplomové práce RNDr. Jaroslav Ráček, Ph.D. Dostupný z WWW: http://is.muni.cz/th/4266/fi_m/Rizeni_softwarovych_projektu.pdf. s. 27 - 30.

Modelů, jakými lze aplikace vyvíjet, je ale mnohem více. Jako základní model lze označit již zmiňovaný model vodopádu. Další modely jsou více či méně jeho obměnou.²⁴ Mezi vývojové modely patří například:

1. Vodopád
2. Inkrementální model
3. Iterativní model
4. Prototypování
5. Spirálový model

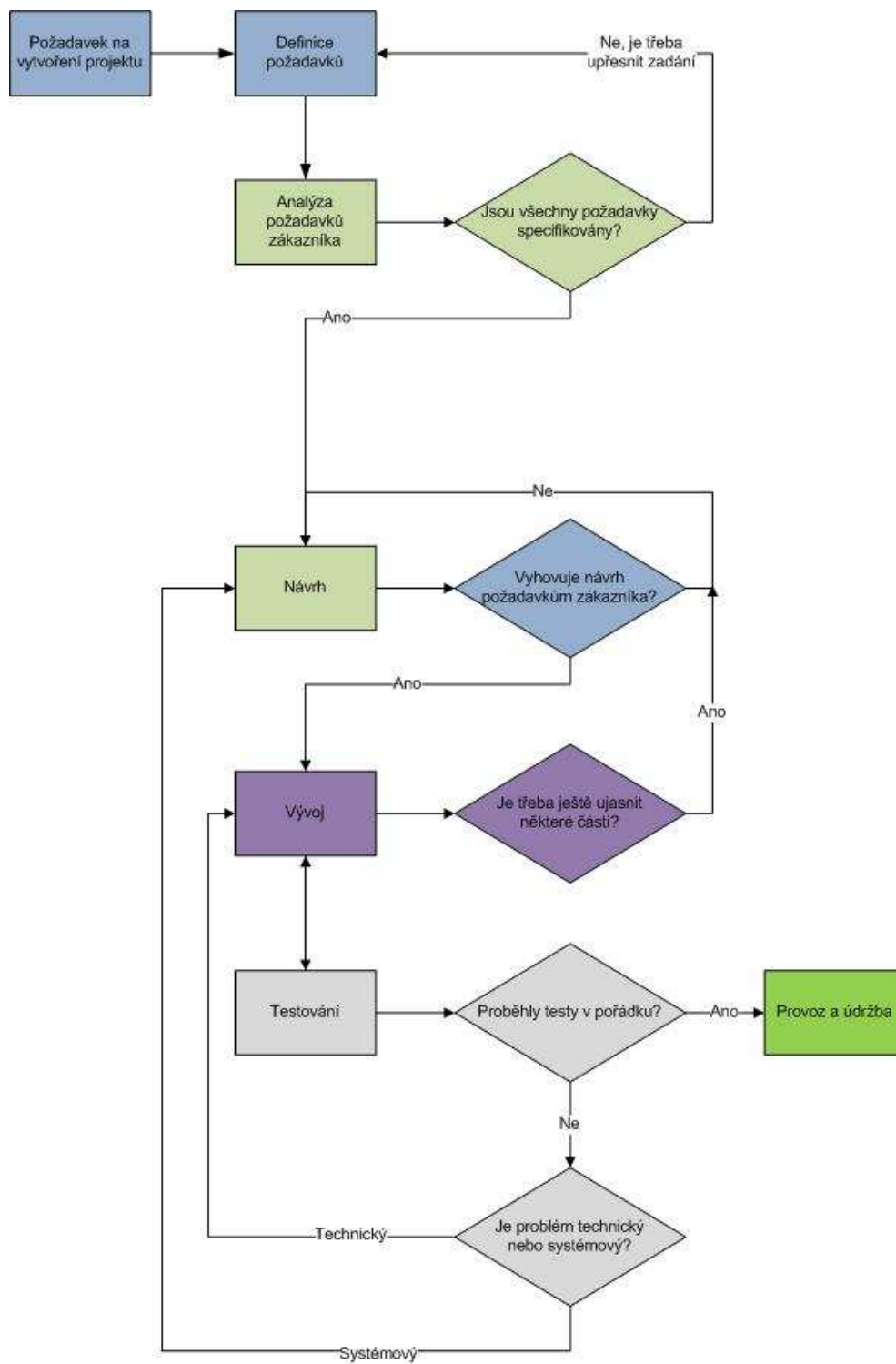
Zkusíme se na průběh podívat detailněji a jednotlivé fáze blíže specifikovat. Pro lepší pochopení by měl posloužit obrázek 2.7. Tento diagram odpovídá vodopádovému modelu vývoje, který byl popisován v kapitole 2.6, a proto není nutné celý proces a jednotlivé části blíže popisovat. Fáze životního cyklu jsou zde provázány a doplněny o potřebné rozhodovací procesy. Barevné odlišení jednotlivých činností pak umožňuje určit, který člen týmu se v danou chvíli na práci podílí.

Pro rychlý a co nejvíce bezproblémový chod projektu je nutné dodržovat pravidelné a včasné kontroly nejen odvedené práce, ale také jednotlivých rozhodnutí. Pokud by takováto kontrola nebyla prováděna, zvyšovalo by se riziko výskytu chyb a například až v okamžiku používání by bylo nutné se vrátit na samý začátek. Riziko chyb přitom není možné úplně eliminovat, ale dodržováním pravidel je možné snížit pravděpodobnost jejich výskytu.

V případě, že během práce byla odhalena chyba či nějaký nedostatek, je nutné dodržet níže vyobrazené rozhodovací procesy. V praxi bývá velice

²⁴ SLOUP, Petr. Vytvoření aplikačního software pro podporu marketingu prodeje automobilů. [s.l.], 2002. 50 s. Diplomová práce.

častým problémem (a nejen u vývoje software) fakt, že po odhalení nedostatku se daná osoba snaží problém vyřešit sama a pokračovat v práci. Takové jednání může být na první pohled rychlejší a efektivnější, ale může zároveň s sebou přinést nemalé problémy do budoucna.



Obrázek 2.7: Vodopádový model postupů

1. technickou část
2. organizační část

Tyto dvě části, ač se váží k jednomu projektu, jsou velice rozdílné a proto je vhodné tyto dvě dílčí části rozdělit. Zatímco technická část, až na několik výjimek, zůstává nejen ve vlastnictví, ale také plně k dispozici pouze projektovému týmu, na organizační části se podílí i samotný zákazník.

Technickou dokumentaci máme samozřejmě na mysli části jako popis algoritmů, datových souborů a podobných prvků. Tím, že tyto dokumenty zůstávají zákazníkovi zapovězeny neříkáme, že spolu s projektem nemůže být předán technický ovšem uživatelský popis aplikace. Výjimku tvoří, pokud je software vyvíjen přímo do vlastnictví zákazníka. V takovém případě je spolu se zdrojovými kódy předávána i technická dokumentace. Takový způsob vývoje není neobvyklý, nýbrž prodej licence k používání je častějším způsobem.²⁶

Tato kapitola se věnuje obsahu jednotlivých dokumentů, ne jejich konkrétní podobě. Jak mohou jednotlivé dokumenty vypadat je možné nalézt v příloze. Adresářová struktura, kterou je možné nalézt na přiloženém CD je připravena zejména pro velké projekty. Pro méně obsáhlá díla se používají jednodušší struktury.

2.8.1 Organizační část

Organizační část dokumentace nemá s konkrétní podobou finální aplikace příliš společného, ovšem pro její vznik je nezbytná. V rámci organizační části se objevují dokumenty, které by měly zahrnovat a obsahovat komunikaci mezi jednotlivými členy týmu včetně zákazníka. Lze tedy říci, že organizační

²⁶ POLANSKÝ, Ondřej. *Software a jeho licence* [online]. 2007 [cit. 2009-04-18]. Dostupný z WWW: <<http://www.dsl.cz/clanky-dsl/clanek-739/software-a-jeho-licence-1-2>>.

dokumentace slouží pouze k zaznamenávání průběhu projektu a jeho řízení.²⁷

Dokumenty spadající do této organizační dokumentace lze rozdělit do dvou základních kategorií:

1. Jednání se zadavatelem
2. Řízení projektu

2.8.1.1 Jednání se zadavatelem

Zápisy z jednání se zadavatelem jsou nezbytnou součástí každého (nejen softwarového) projektu. Případné dokumenty o průběhu, které jsou zákazníkovi předkládány, sice patří do jiné kategorie, ale pro potřeby případných nejasností, nedostatků či problémů by měla být zachycena kompletní komunikace mezi zákazníkem a zadavatelem včetně všech dokumentů předkládaných.

V projektové dokumentaci, ve složce jednání se zákazníkem, by tedy neměly chybět zápisy z jednotlivých jednání a všechny předkládané či diskutované dokumenty. I přesto, že jednání se zadavatelem svým způsobem patří do řízení projektu, svým obsahem přímo do dané složky nezapadá.

Obsah takového zápisu z jednání by v každém případě mělo tvořit několik základních informací.²⁸ V případě jejich absence by takový zápis byl stejně významný, jako kdyby vůbec neexistoval. Aby nemusel být vytvářen speciální

²⁷ SHELFORD, Thomas J., REMILLARD, Gregory A. Real Web Project Management: Case Studies and Best Practices. [s.l.] : Addison Wesley Professional, 2002. 336 s. ISBN 0321112555. s. 77 – 104. s. 83 – 100.

²⁸ MCNAMARA, Carter. *Basic Sample of Board of Directors Meeting Minutes* [online]. c1997-1998 [cit. 2009-04-18]. Dostupný z WWW: <<http://managementhelp.org/boards/minutes.htm>>.

formulář, je možné použít vzorové zápisy z jednání, které nabízí kancelářský software. V každém případě by neměly chybět informace:

- Datum konání schůze
- Zúčastněné osoby
- Řešené problémy, jejich závěry a úkoly z nich plynoucí

2.8.1.2 Řízení projektu

Složka řízení projektu obsahuje zásadní informace pro samotný průběh. Primárně by její obsah měl být tvořen manažerem a vedoucím projektu. Ostatním by měl sloužit jen jako zdroj informací. Nedílnou součástí řízení projektu by mělo být rozdělení úkolů, používané nástroje či projektový plán.²⁹ Dalším dokumentem, který se objevuje v této kategorii, je například sledování nákladů.

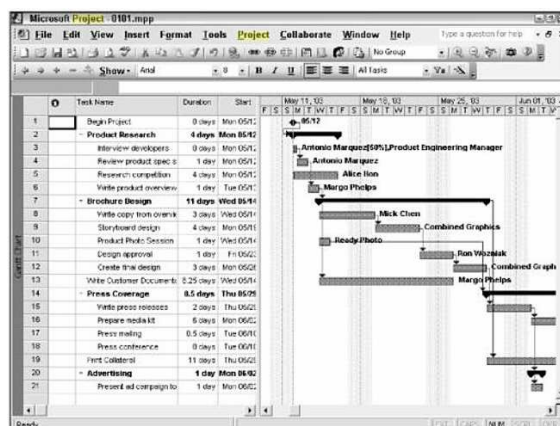
Veškeré tyto dokumenty lze vést za pomoci kancelářského softwaru a i v tomto případě je po obsahové stránce nutné dodržovat určité minimum. Dokument přidělování a sledování úkolů by například měl obsahovat takové informace, jako:

- Popis úkolu
- Zodpovědná osoba
- Datum splnění

Pro účely projektového plánu lze stejně jako pro sledování nákladů použít například tabulkového procesoru. Tím lze nahradit specializovaný software,

²⁹ DELANEY, Declan, BROWN, Stephen. *DOCUMENT TEMPLATES FOR STUDENT PROJECTS IN SOFTWARE ENGINEERING* [online]. 2002 [cit. 2009-04-19]. Dostupný z WWW: <<http://www.utdallas.edu/~chung/ieee-templates.pdf>>.

který již nemusí být pro projektový tým dostupný z hlediska jeho ceny. Specializovaným softwarem pro sledování termínů můžeme mít na mysli například Microsoft Project.³⁰ Jak takový projektový plán v aplikaci Microsoft Project může vypadat, znázorňuje obrázek 2.9.³¹



Obr. 2.9: Ukázka projektového plánu v MS Project

2.8.2 Technická část

Do této části dokumentů řadíme takové části, které jsou jakýmsi duševním vlastnictvím tvůrců. Důvod vzniku takové dokumentace je prostý a pochopitelný. Ať už je projekt malý či velký, obsahuje části, které nemusí být pro druhou osobu, případně po delší přestávce i pro programátora samotného, na první pohled zřejmé.

Tzv. komentování zdrojového kódu je pouze malou částí technické dokumentace. Tuto kategorii dokumentů můžeme rozdělit do několika menších

³⁰ ŠEVČÍK, Matěj. Řízení softwarových projektů. [s.l.], 2006. 67 s. Masarykova Univerzita, Fakulta Informatiky. Vedoucí diplomové práce RNDr. Jaroslav Ráček, Ph.D. Dostupný z WWW: http://is.muni.cz/th/4266/fi_m/Rizeni_softwarovych_projektu.pdf. s. 13.

³¹ MARMEL, Elaine. *Microsoft Office Project 2003 Bible*. [s.l.] : [s.n.], 2003. 960 s. ISBN 978-0-7645-4252-7.

celků. Již bylo zmíněno, že z pohledu každé fáze projektu by měl vzniknout dokument, který popisuje výsledky dosažené v dané fázi. Základní podoba technické dokumentace by tak měla obsahovat popis všech částí, jejichž náplň se váže k technickému provedení projektu.³² Mezi ně patří:

1. Analýza
2. Návrh
3. Vývoj
4. Testování

Vzhledem k tomu, že ve vývojové fázi vzniká nejvíce částí, které mají hodně společného s technickým provedením projektu, můžeme dokumentaci této části rozdělit do více bloků. Nemělo by být opomenuto zejména:

- Popisy metod a funkcí
- Vazby mezi jednotlivými částmi
- Připojení, nastavení a struktura databáze
- Zabezpečení, řízení přístupu
- Souborová struktura
- Grafika

Jednotlivé části mohou být tvořeny samostatnými dokumenty, které vznikají zejména v průběhu vývoje. V tomto stavu je možné dokumenty ponechat i v závěru projektu nebo z nich sestavit jeden souhrnný dokument popisující technické provedení celého projektu.

³² SHELFORD, Thomas J., REMILLARD, Gregory A. Real Web Project Management: Case Studies and Best Practices. [s.l.] : Addison Wesley Professional, 2002. 336 s. ISBN 0321112555. s. 77 – 104. s. 84 – 86.

2.8.2.1 Analýza

Analýza požadavků zákazníka je první velkou činností, která předchází samotný vývoj projektu. Na základě tzv. „creative brief“ a dalších zkoumání by měl být analytik schopný sestavit analýzu problematiky na straně zadavatele.

Důkladná analýza požadavků zákazníka je pro projekt velice důležitou činností. Výzkumem bylo zjištěno, že ve velkém množství projektů bylo po jeho ukončení zákazníkem upozorněno na funkcionalitu, která na začátku nebyla požadována.³³ Důkladná analýza ovšem snižuje riziko takového problému a během vypracovávání návrhu tak může tým brát v úvahu všechny požadavky klienta.

V každém případě by měla analýza obsahovat popis celého procesu, který má být projektem řešen a činností, které jsou na daný problém navázány. Nedílnou součástí by také měl být pohled na danou problematiku ze strany všech zainteresovaných osob.

2.8.2.2 Návrh

Na základě provedené analýzy vzniká návrh. To je již známý fakt a i návrh by měl v projektové dokumentaci své místo. Nejen že na jeho základě probíhá vývoj a je díky němu možné s použitím speciálních nástrojů některé části i zautomatizovat, ale v případě problémů, či stížností ze strany zadavatele, odhalit místo vzniku problému. Návrh by měl být odsouhlasen zákazníkem a až poté předán k dalšímu zpracování.

Návrh může být proveden různými způsoby a jeho podoba také záleží na tom, komu je předkládán. Grafické provedení návrhu, vazeb a postupů je více intuitivní a pokud je takový návrh předkládán například manažerům na straně

³³ Uvedli tak všichni respondenti, kteří zároveň uvedli, že v jejich společnosti nevzniká pro každý projekt projektová dokumentace.

zákazníka, rozhodně je pro ně přehlednější a pochopitelnější než návrh databáze a textový popis celého procesu.

2.8.2.3 Vývoj

Dokumentů, které popisují vývojovou fázi a řešení v ní použité, je hned několik druhů. Konkrétní podoba těchto dokumentů a informací, které by neměly být opomenuty, se ovšem liší v závislosti na mnoha faktorech. Vycházet lze ale ze vzorů a doporučení organizace IEEE (Institute of Electrical and Electronics Engineers).³⁴ Tato organizace vydává standardy a doporučení také ohledně vývoje software.³⁵

2.8.2.3.1 Popisy metod a funkcí

Jedná se o nejklašičtější technickou dokumentaci. Popisy metod, funkcí a algoritmů přímo ve zdrojovém kódu jsou základem každého programování. Pro některá programovací prostředí existují automatické nástroje, které z těchto komentářů vytvoří kompletní dokument. Ovšem tento souhrnný dokument není pro samotnou dokumentaci tak důležitý. Důležitá je existence informací v něm obsažených.

Ve struktuře projektové dokumentace teda může, či nemusí existovat celkový přehled funkcí, metod a jejich popisů. V případě souhrnného dokumentu by pak tento popis funkcí stál samostatně.

2.8.2.3.2 Vazby mezi jednotlivými částmi

³⁴ Mezinárodní nezisková profesionální organizace pro rozvoj technologií souvisejících s elektřinou. <<http://www.ieee.org/>>.

³⁵ DELANEY, Declan, BROWN, Stephen. *DOCUMENT TEMPLATES FOR STUDENT PROJECTS IN SOFTWARE ENGINEERING* [online]. 2002 [cit. 2009-04-19]. Dostupný z WWW: <<http://www.utdallas.edu/~chung/ieee-templates.pdf>>.

Pokud projekt není tvořen jediným souborem, pak s největší pravděpodobností existují propojení a vazby mezi jednotlivými soubory. Většinou se jedná o vkládání a používání souborů, které obsahují definice funkcí společných pro celý projekt, definice hlaviček, jazykových souborů a mnoho dalšího. Nejedná se o dokument, který by byl stěžejní pro projektovou dokumentaci, ovšem jeho přítomnost může zrychlit práci na projektu.

2.8.2.3.3 Připojení, nastavení a struktura databáze

Předpokladem ke vzniku těchto informací je existence databáze v projektu. Ta se v současné době vyskytuje u většiny v dnešní době vznikajících projektů. Pro účely dokumentace by neměla chybět struktura tabulek, jejich vazby a ani definice jednotlivých polí.

Grafický náhled na podobu databáze podá více informací, než popis v textu a tak se struktura tabulek ukazuje právě na grafickém modelu. Takový model je pak možné vygenerovat přímo z některých databázových manažerů.

2.8.2.3.4 Zabezpečení, řízení přístupu

Zabezpečení a řízení přístupu patří mezi další nepovinnou část technické dokumentace. U informačních projektů z pravidla tato část není zastoupena vůbec. U aplikačních projektů je situace již jiná a v případě, že daný projekt obsahuje možnost řízení přístupu, neměla by v dokumentaci chybět informace o způsobu zabezpečení, úrovních přístupu a o způsobu přidělování oprávnění. Vše samozřejmě z pohledu technické stránky věci.

2.8.2.3.5 Souborová struktura

U rozsáhlých projektů s velkým množstvím zdrojových souborů, ale i u menších, se jednotlivé soubory strukturují do několika adresářů. Neděje se tomu tak z vlastní vůle programátorů nýbrž pro lepší přehlednost zdrojových souborů podle jejich typu. I přesto, že názvy jednotlivých adresářů naznačují,

co se v dané složce nachází, je vhodné adresářovou strukturu spolu s jejím popisem do projektové dokumentace zahrnout.

2.8.2.3.6 Grafika

Grafika je samostatnou částí technické dokumentace, která s předchozími částmi nemá příliš mnoho částí společných. Samotné grafické provedení lze u aplikačních projektů brát jako samostatný projekt, kde jedním týmem je vyvíjena programová část a druhým grafické provedení. To samozřejmě platí například pro komunitní weby. U intranetových aplikací, kde není kladen tak vysoký důraz na grafiku, samozřejmě probíhá vývoj v rámci jednoho týmu.

V současné době je grafika věcí zejména kaskádových stylů (CSS) a tak se projektová dokumentace grafického provedení zabývá právě podobou souborů s koncovkou css.

2.8.2.4 Testování

V této fázi již nic nového nevzniká a i přesto se jedná o velice důležitou část. Dokumentace testování neobsahuje pouze report o průběhu samotného testování, nýbrž by měla obsahovat i jeho plán a přehled všech funkcí, které je zapotřebí vyzkoušet.³⁶ Po úspěšně ukončeném testování produktu by tak v dokumentaci neměla chybět informace kdo, kdy a s jakým výsledkem provedl testování dané části. Pro úspěšné zahájení provozu by výsledek měl být samozřejmě pozitivní.

³⁶ DELANEY, Declan, BROWN, Stephen. *DOCUMENT TEMPLATES FOR STUDENT PROJECTS IN SOFTWARE ENGINEERING* [online]. 2002 [cit. 2009-04-19]. Dostupný z WWW: <<http://www.utdallas.edu/~chung/ieee-templates.pdf>>.

3 Projektová dokumentace v praxi

Stejně jako i u jiných lidských činností, tak i u vývoje softwaru se praxe může od teorie poněkud lišit. Pro potřeby této bakalářské práce poslouží následující praktická ukázka pro lepší ilustraci situace kolem projektové dokumentace.

V praxi mohou být některé dokumenty nahrazeny jinou formou předávání informací a proto není žádoucí duplikovat tato data v dalším dokumentu. Z pravidla se jedná například o přehled e-mailové korespondence, která zahrnuje jednání se zákazníkem či předávání informací v rámci týmu. Nejen v závěru projektu, ale i v průběhu, je pak možnost tuto korespondenci exportovat přímo do projektové struktury a dostat tak teoretickému modelu. Toto nahrazování dokumentů je nejvíce časté zejména u menších projektových týmů. Tam, kde se na práci podílí 1-2 lidé, bývá velké množství předávání informací nahrazeno verbální komunikací. Opakem je tomu u velkých komerčních projektů, kde je přítomnost všech dokumentů vyžadována.³⁷

Samotná podoba projektové dokumentace tedy není příliš důležitá a v této práci nabízené dílčí dokumenty lze brát pouze jako návrh její podoby. Důležitý je její obsah, který by měl obsahovat všechny výše popisované části. Je proto možné a zároveň i žádoucí jednotlivé dokumenty nahrazovat například již zmiňovanou archivací elektronické korespondence.

Všeobecně lze říci, že projektová dokumentace je v praxi ovlivňována mnoha faktory a některé její části, které se zdají projektovému týmu méně důležité, jsou na úkor kvality zjednodušovány.

³⁷ JALOTE, Pankaj. Software Project Management in Practice. [s.l.] : Addison-Wesley Professional, 2002. 288 s. ISBN 0201737213.

3.1 Úvod k projektu – představení

Aplikace, kterou jsem vybral pro názornou ukázkou projektové dokumentace, se jmenuje SuperOPL a je součástí "Informačního a aplikačního portálu" jedné českobudějovické firmy. V tomto případě byl požadavek zákazníka jasný, tudíž byl projekt v úvodních fázích bez větších potíží. Požadovaná aplikace není řešením nového problému, ale má za cíl odbourat současné neefektivní a komplikované řešení.

Vzhledem k nutnosti přístupu k této aplikaci odkudkoliv a bez dalších instalací na straně klienta, byla volba umístění na firemním intranetu jediným a zároveň i nejlepším řešením. Současná podoba "Informačního a aplikačního portálu" již obsahuje několik aplikací podobného stylu, včetně řady společných funkcí a proto celý projekt nebyl časově příliš náročný.

Po technické stránce je projekt, stejně jako zbytek systému, postaven na technologii PHP s podporou databáze Microsoft SQL, která je pro potřeby firemního nasazení vhodnější, než například MySQL.

Projektový tým byl i v případě tohoto zadání shodný, jako je tomu u vývoje dalších intranetových aplikací firmy, a skládal se samozřejmě z vedoucího (manažera), analytika, návrháře a programátora v jedné osobě a samozřejmě testera. Vzhledem k umístění aplikace do již existujícího prostředí již nebylo zapotřebí řešit grafické prostředí a tato část je podřízena centrálnímu stylu.

Celý projekt od zadání zadavatelem až po spuštění hotové a otestované aplikace trval 6 týdnů. Okamžitě po spuštění ostrého provozu ovšem začaly přicházet požadavky z dalších stran na aplikaci stejného stylu, jen v menších modifikacích a tak byl projekt opět nastartován a aplikace je do doby publikace této bakalářské práce stále modifikována tak, aby obsáhla požadavky ostatních zákazníků.

3.2 Požadavky zákazníka

Problémem, který by měla aplikace vyvíjena tímto projektem řešit, je rozdělování úkolů a sledování jejich plnění. Z toho plyne také název aplikace. Slovo „super“ by mělo značit rozsáhlost a zkratka OPL je z anglického „Open Point List“. V současné době existuje řešení v podobě souborů MS Excel, kam se jednotlivé úkoly zapisují.

Prvním nedostatkem současného řešení je špatná informovanost o přidělení nového úkolu pracovníkovi. Většina úkolů je přidělována v rámci pravidelných porad týmu. V případě, že osoba odpovědná za splnění nějakého úkolu na poradě chybí, musí následně nahlédnout do seznamu úkolů a hledat, zda nemá přidělený nový úkol. Takový způsob je neefektivní a například po týdenní nepřítomnosti mohlo takových porad proběhnout hned několik v rámci několika různých týmů. Z toho plyne nutnost automatického informování o přidělení nového úkolu.

Druhým, stejně důležitým požadavkem, zadavatele bylo řešení problému, který již byl nastíněn. Každá osoba může pracovat hned v několika týmech a tím pádem mít přidělované úkoly v několika souborech. Řešením by měl být kompletní přehled úkolů pro jednotlivé pracovníky.

V průběhu diskuze o zadání nově vznikající aplikace byla specifikována ještě další funkcionalita, která usnadní práci všem odpovědným osobám a zároveň nebude znamenat výrazné navýšení nákladů na vývoj. Touto novou vlastností je upozorňování na vypršení úkolu.

3.3 Kategorie projektu a potřebné dokumenty – teorie

Jak je již z představení projektu a požadavků zákazníka zřejmé, jedná se o aplikační typ projektu, který bude umístěn na firemním intranetu. Není proto nutné řešit části, jako je grafika. Středem zájmu, alespoň co se technické dokumentace týká, bude samotné programové provedení.

Prozatím jsem vždy dokumentaci rozdělával na dokumentaci organizační a technickou. Ani v tomto případě tomu nebude jinak. Dle teorie, které se tato bakalářská práce věnovala v předchozích kapitolách, by měla dokumentace vypadat následovně.

3.3.1 Organizační dokumentace

Organizační dokumentace by se počtem použitých dokumentů neměla lišit ať se jedná o projekt informační nebo o projekt aplikační. Použity by tedy měly být zejména dokumenty, obsahující jednání se zadavatelem práce a dokumenty, které se vážou k řízení projektu.³⁸

- Zápisy z jednání
- Rozdělení a sledování úkolů
- Rozpočet
- Termínový plán

3.3.2 Technická dokumentace

Svým zadáním projekt jednoznačně spadá do kategorie intranetové aplikace a není proto problém definovat dokumenty, které by měly technickou stránku aplikace popisovat. Domnívám se, že není nezbytné na tomto místě znova popisovat a jmenovat veškeré dokumenty, které jsou nezbytné. Vzhledem k tomu, že se jedná o aplikační projekt, z pohledu technické dokumentace je nutné, aby obsahoval všechny dokumenty, popisované v kapitole 2.8.2.

³⁸ DELANEY, Declan, BROWN, Stephen. *DOCUMENT TEMPLATES FOR STUDENT PROJECTS IN SOFTWARE ENGINEERING* [online]. 2002 [cit. 2009-04-19]. Dostupný z WWW: <<http://www.utdallas.edu/~chung/ieee-templates.pdf>>.

I přesto zde však jedna modifikace bude. Aplikační projekt by sice měl obsahovat všechny výše uvedené části, ale pouze v případě, že by se jednalo o projekt internetový. Vzhledem k nasazení na firemním intranetu do již existujícího prostředí, nebude se v průběhu projektu řešit problematika grafiky. Ta je v případě intranetových projektů buď úplně vynechána a vývoj je zaměřen pouze na funkčnost nebo, jako je tomu v tomto případě, je grafika řešena centrálním stylem celého systému.

3.4 Praxe – použité dokumenty a dodržená pravidla

Do této chvíle bylo vše věnováno především teorii, ale nyní je nutné se podívat také na praktickou stránku věci a ukázat, jak projektová dokumentace může také vypadat. Níže uvedené vzorové dokumenty slouží pouze pro představu projektové dokumentace v praxi. V žádném případě není nutné, aby dokumentace u jiných projektů měla stejnou formu.

3.4.1 Organizační dokumentace

Již bylo zmíněno, že dle teorie by organizační část dokumentace měla obsahovat zejména dokumenty o řízení projektu a zápisy z jednání. Ani u vybraného projektu SuperOPL tomu nebylo jinak.

Pro zápisy z jednání, ať už se zadavatelem nebo v rámci týmu, byly používány poupravené šablony aplikace MS Word určené pro zápisy z jednání. Tyto dokumenty obsahují všechny potřebné informace a proto není nutné pro tento účel vytvářet speciální dokumenty. Příklad takového zápisu z jednání je k dispozici na přiloženém CD. Dokumenty, které řeší řízení projektu, byly oproti teorii poněkud upraveny. Součástí každého projektu by měl být dokument, který sleduje náklady. V případě našeho projektu tomu ovšem bylo jinak a protože aplikace SuperOPL byla vyvíjena pouze interně s využitím interních nákladů, nebylo toto sledování důležité. Ostatní dokumenty již ovšem zastoupeny byly.

Pro sledování termínového plánu bylo použito nástroje MS Project, který je pro dané účely z mého pohledu nejideálnější. Přidělování úkolů a sledování jejich plnění bylo řešeno v rámci standardního interního souboru MS Excel, který by měl být nahrazen právě touto vyvíjenou aplikací.

Z pohledu organizační dokumentace nebylo nutné v tomto projektu používat žádné další dokumenty a to zejména i z důvodu, že členové týmu osobně znají řešenou problematiku.

3.4.2 Technická dokumentace

Výhodou projektu, který jsem použil pro potřeby této práce, je fakt, že daný projekt je součástí již existujícího celku a proto je velké množství součástí řešeno v rámci celého systému. To ovšem neznamená, že tyto části budou v bakalářské práci přehlíženy. V kapitole 2.8.2 byly rozebrány jednotlivé části technické dokumentace, které by dle teorie měly být v projektu zahrnuty. Jaký je obsah dokumentů v praxi, tomu se věnují následující odstavce, které v příloze doplňují vzorová data. Celá technická stránka této aplikace je řešena v jednom souboru.

3.4.2.1 Analýza

Již bylo zmíněno, že analýza požadavků byla velice snadná. Nejen, že v tomto případě byla představa zadavatele jasná, ale i tým podílející se na projektu znal danou problematiku. Výstupem analýzy požadavků zadavatele byl v tomto případě zároveň i zápis z prvního jednání doplněný o náčrtek požadovaného procesu.

3.4.2.2 Návrh

I přesto, že požadavky byly jasné a i náčrt z prvního jednání by mohl sloužit jako návrh aplikace, bylo nutné vše shrnout a zpracovat veškeré technické možnosti. Vzhledem k faktu, že následný návrh měl být předložen

k posouzení manažerům, byl vypracován v grafické podobě za použití nástroje MS Visio, na jehož výstup jsou manažeři již zvyklí a tudíž je pro ně takový model snadno pochopitelný. Zároveň konzultace nad takovým návrhem je srozumitelná a tak je riziko vzniku nedorozumění velmi málo pravděpodobné.

Předložený návrh byl schválen hned napoprvé a proto se mohlo okamžitě začít s vývojem programové části.

3.4.2.3 Vývoj

Vývojová fáze tohoto projektu trvala 3 týdny a na samotném kódování pracoval jeden člověk.

3.4.2.3.1 Metody, funkce a procedury

Díky zasazení do již funkčního celku a přítomnosti centrálních funkcí je velká část činností řešena právě v rámci algoritmů, které jsou dostupné pro celý systém. I přes to, že teorie říká, že přehled funkcí a jejich popisů je výhodnější mít v externím souboru, v případě této aplikace tomu tak není a to z jednoho důvodu. Centrální funkce mají své přesně stanovené místo a soubor se zdrojovým kódem obsahuje pouze definice těchto funkcí. Zdrojový kód je tak sám o sobě přehledem všech funkcí, jejichž popisy jsou prováděny v podobě komentářů. Podobně je tomu tak i u funkcí, které jsou specifické pro daný projekt.

Tento model popisu metod, funkcí a procedur se nám osvědčil a je proto používán u všech vypracovaných projektů. Dle mého názoru není nutné u takto méně rozsáhlých projektů vytvářet speciální soubory s extrahovaným popisem funkcí.

3.4.2.3.2 Vazby

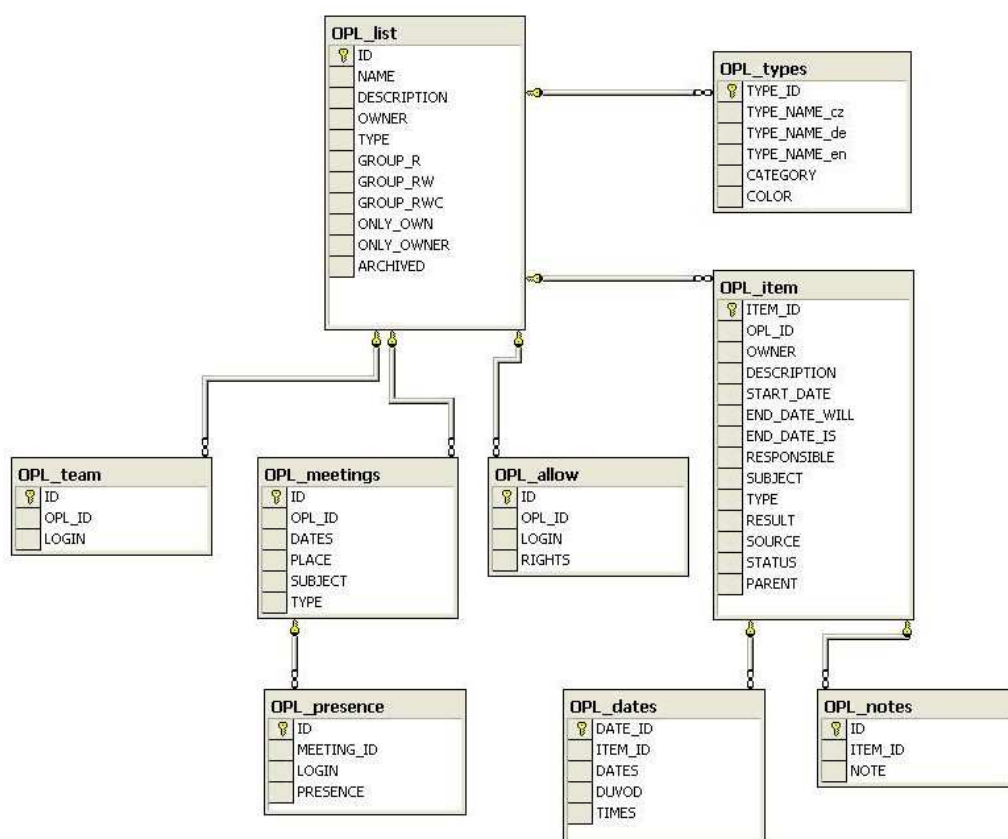
Celý vyvíjený systém má mnoho společného a proto se některé zdrojové kódy neopakují pro jednotlivé aplikace, ale používá se vždy jeden zdrojový

soubor pro všechny aplikace. Vzhledem k jejich podobnosti je to nejideálnější způsob pro případné rychlé změny v celém systému. V tomto konkrétním případě se používají především soubory obsahující definice funkcí, grafiky, nastavení a v neposlední řadě také jazykové soubory.

V případě, že počet vkládaných souborů je takto velký, osvědčil se nám při práci přehled „includovaných“ souborů s krátkým popisem, co daný soubor obsahuje a k čemu se používá.

3.4.2.3.3 Databáze

Správné navržení databáze je pro celý systém velice důležité. V případě, že by byla navržena nevhodným způsobem, pak by celý projekt ztrácel na efektivitě. Zároveň by měla být při návrhu dodržena 3. normální forma. V praxi se osvědčil grafický návrh, který je snadno pochopitelný a lze díky němu snadno odhalit některé nedostatky pouhým pohledem. Dalším způsobem již není nutné návrh databáze dále dokumentovat. Jak takový návrh může vypadat je znázorněno na obrázku 3.1.



Obrázek 3.1: Návrh databáze

3.4.2.3.4 Zabezpečení

Zabezpečení je poměrně choulostivá část a je proto nutné mít vše správně nejen pro programové stránce. Zejména u větších aplikací, které mají více přístupových úrovní, by měl existovat koncept oprávnění. Ten by měl obsahovat informace o způsobu autentizace, přidělování práv či o možnostech jednotlivých úrovní.

Aplikace SuperOPL umožňuje takové řízení přístupu a to hned v několika rovinách. Autentizace v každém případě probíhá na základě ověření NT uživatele proti active directory a následné ověření úrovně je nutné provádět proti více zdrojům. Jak konkrétně je řešeno oprávnění u této aplikace a jak

vypadá dokument, který se právě těmto informacím věnuje, naleznete v příloze.

3.4.2.3.5 Struktura

Pokud se nejedná o jednoduchou aplikaci, která je tvořena jedním či dvěma soubory, měla by být popsána struktura zdrojových souborů. V našem případě se domnívám, že umístění jednotlivých souborů v adresářích je intuitivní. Základní popis je ovšem umístěn v dokumentu, který popisuje celou aplikaci.

3.4.2.3.6 Grafika

Vzhledem k tomu, že celý projekt je umístěn v rámci firemního intranetu a již existujícím prostředí, není proto nutné řešit věc grafiky. Ta je podřízena centrálnímu kaskádovému stylu a tudíž není nutné ji pro tento projekt dokumentovat zvlášť.

3.4.2.4 Testování

Vyvíjená aplikace sice není přímo svázána s činností, která by mohla způsobit finanční či hmotnou újmu na straně zákazníka, ale možná ztráta dat by byla poměrně zásadní pro běh projektů, z nichž plynoucí úkoly se do tohoto systému zadávají. Je proto nutné provést takové testování, aby nemohlo dojít k neúmyslnému nebo i úmyslnému odstranění dat nepovolanou osobou.

To ovšem není jedinou akcí, kterou je potřeba prověřit. Vyzkoušen a otestován musí být samozřejmě celý proces od vytváření nových záznamů, přidělování úkolů až po zaznamenávání jejich plnění a automatické upozorňování. Na celý tento proces byl v projektovém plánu vyhrazen 1 týden, což znamená přibližně $\frac{1}{4}$ času, který připadal na vývoj. Testování bylo přiděleno osobě, která do té doby neměla s vývojem nic společného.

Výsledkem testování tak bylo nejen vyzkoušení funkčnosti, ale také feedback od prvního uživatele.

Záznam o průběhu testování naleznete také jako přílohu této práce. Plán pro testování byl sestaven v rámci porady projektového týmu, kde byly diskutovány možné problémy.

4 Rozdíly mezi teorií a praxí

Ani u projektové dokumentace, stejně jako i u jiných oborů lidské činnosti, není zpravidla možné přesně dodržet teoretické postupy a tak se praxe mnohdy liší. U řízení projektu je tato skutečnost ovlivňována faktory, které jsou v dnešní době pro většinu zákazníků důležité. Čas a peníze. I přesto, že ovlivnění kvality dokumentace těmito faktory je velmi krátkozraké, jejich vliv je poměrně velký.³⁹

Důvodů, proč je ze strany zákazníka vyžadována nižší cena a kratší dodací lhůty je několik, ale pouze jediný důvod bývá ve většině případů zásadní. Vývoji softwaru firmy nevěnovaly po dlouhou dobu velkou pozornost ani tomu nepřikládaly váhu a tak se o vedení těchto projektů ve firmě staraly osoby, které to braly jako „trest“. V současné době se situace zlepšuje, neboť zákazníci objevují potenciál v dobře a přesně nasazeném softwarovém řešení. Proto platí fakt, který již byl uveden v kapitole 2.6.1, že je dobré na straně zákazníka získat takovou osobu, která je pro věc nadšená.

Formou krátkého dotazníku, který je k dispozici jako příloha, bylo osloveno 60 firem po celé ČR, které působí v oblasti tvorby internetových prezentací a aplikací. Výběr byl uskutečněn tak, aby byly zahrnuty jak velké profesionální firmy s několika zaměstnanci, tak i malé společnosti, kde tým tvoří dva lidé. Návratnost tohoto dotazníku byla ovšem nižší, než jsem původně očekával a dosahovala pouze necelých 50%. Vzhledem k tomu, že vrácené dokumenty i tak pokryly spektrum trhu, lze výsledky získané tímto průzkumem považovat za relevantní.

Záměrem tohoto průzkumu nebylo statistické zkoumání aktuální situace na trhu tvorby internetových aplikací, ale spíše základní informace o tom, zda se

³⁹ Ve výzkumu to uvedlo přibližně 15% respondentů.

v současné době dodržují určitá pravidla pro tvorbu internetových projektů či nikoli. Na základě tohoto výzkumu bylo zjištěno, že průběh samotného projektu se u jednotlivých dotazovaných společností výrazně liší a to na úkor kvalitní práce. Zatímco u velkých firem, které mají více zkušeností i zdrojů pro jednotlivé části, jsou ve velké míře dodržována výše popsaná pravidla. Společnosti s méně zkušenostmi dle informací získaných průzkumem některá pravidla a zásady přímo negují.

Zatímco u velkých týmů s velkými zkušenostmi by bylo možné pochopit, že u některých fází mohou čerpat ze svých zkušeností a tak jim nevěnovat takový důraz, jsou to možná právě ty zkušenosti, které společnosti nutí k přesnému řádu. Vliv na tuto skutečnost v určité podobě ovšem mají také samotní zákazníci. Zatímco na projektové týmy s velkým zázemím se obrací zákazníci, kteří jsou zvyklí na řád a pravidla práce v týmu, menší programátorské týmy si vybírají zákazníci s poměrně neurčitým zadáním a ponecháním všeho přímo na tvůrcích.

Čím je tato skutečnost způsobena ovšem není předmětem této práce a důvody takového jednání by bylo třeba hledat nejspíše u psychologických prací.

4.1 Organizační část dokumentace

Organizační dokumentace je výše popisovaným faktem zasažena více, než dokumentace technická. Důvodem je zejména fakt, že pokud na projektu pracuje jeden či dva lidé, není pro ně zaznamenávání informací a jejich předávání mezi sebou takovým problémem. Většina takových informací se předává pouze slovně a uchování informace je na každém zvlášť. Pro potřeby předávání informací tedy není tak velká potřeba dodržovat daná pravidla. Druhý účel dodržování organizační dokumentace, tedy minimalizace chyb a případné hledání jejich vzniku, v takovém případě ovšem není možné splnit.

Na základě informací z průzkumu tedy není organizační dokumentace dodržována v plném rozsahu. Zejména u menších projektových týmů chybí organizační dokumentace téměř úplně. Dokumenty ohledně jednání se zadavatel v takovém případě bývají pouze ve formě uložené e-mailové korespondence a podobně tomu tak je i u dokumentování řízení projektu. Rozsáhlejší týmy a tím i zároveň týmy, podílející se na rozsáhlejších projektech, jsou si naopak vědomy možných rizik a v určité podobě dokumentují celý průběh projektu. U profesionálních týmů také bývá samozřejmostí přesně definovaný projektový plán včetně řízení lidských zdrojů.

4.2 Technická část dokumentace

Technická dokumentace je na tom o něco lépe než část organizační. Určitá podoba technické dokumentace je vedena dle průzkumu ve většině vypracovávaných projektů. Výjimku tvoří jednoduché informační projekty, kde by ani při pečlivém dodržování teoretických zásad nebyla dokumentace příliš obsáhlá.

Ale i zde je možné najít rozdíly v přístupu jednotlivých vývojových týmů a to na základě stejných vlivů, jako u dokumentace organizační. Technická dokumentace je dle průzkumu vypracovávána opravdu u většiny projektů, ale opět zejména u menších projektových týmů se jednotlivé kategorie spojují do komentování zdrojového kódu, kde je pak možné nalézt ve větším či menším množství požadované informace. Z hlediska rychlosti přístupu k požadovaným informacím se pak jedná o výrazné zpomalení. Údaje jsou sice umístěny přímo u algoritmu, který daný problém řeší, ale například u problému řízení přístupu je nutné nejdříve najít tu část, která problematiku řízení přístupu řeší.

5 Shrnutí výsledků práce

Definované cíle výzkumného záměru práce jsou stanoveny v rovině teoretické, empirické a praktické. Projektová dokumentace internetových projektů je v této práci chápána jako nástroj pro zvýšení kvality práce na projektu a zvýšení efektivity celého procesu. V teoretické části se jednalo zejména o představení celého vývojového cyklu s důrazem na projektovou dokumentaci. Rovina praktická se věnovala realizaci intranetového projektu na základě výsledků teoretické části.

Cílem bylo několik dílčích úkolů, které se ovšem všechny zabývaly projektovou dokumentací internetových projektů. Pro splnění těchto cílů bylo nutné ověřit několik dílčích hypotéz, které byly ověřovány formou analýzy dostupných pramenů.

Předpokladem pro jakoukoliv další činnost bylo nejdříve nutné specifikovat pojem projektová dokumentace. Ta je zastupována ve dvou základních rovinách. V první rovině se jedná o dokumentaci organizační, věnující se řízení projektu, a v rovině druhé o dokumentaci technickou. Ta zastupuje požadavky na systém, návrh a testování. Toto tvrzení se opírá o standardy IEEE.⁴⁰ Vzhledem k profesionalitě této organizace lze toto rozdělení projektové dokumentace chápat jako správné.

Nezbytnou součástí práce v projektovém týmu je stanovení postupů. Dostupné prameny nabízejí několik možných přístupů pro práci na internetových projektech, které ovšem všechny vycházejí ze dvou základních.

⁴⁰ DELANEY, Declan, BROWN, Stephen. *DOCUMENT TEMPLATES FOR STUDENT PROJECTS IN SOFTWARE ENGINEERING* [online]. 2002 [cit. 2009-04-19]. Dostupný z WWW: <<http://www.utdallas.edu/~chung/ieee-templates.pdf>>.

Mezi tyto hlavní přístupy patří vodopádový model a model iterativní.⁴¹ Nejtypičtějším způsobem přístupu pro internetové projekty je v současné době model vodopádový.⁴² Ten ovšem s sebou přináší několik nevýhod mezi něž patří například dlouhá doba mezi zadáním a představením hotového produktu. I přesto je ovšem prvky tohoto modelu možno nalézt i v jiných přístupech.

Iterativní přístup se v současné době zdá jako nejrychlejší možné řešení pro dosažení cíle v co nejkratším čase a nejvyšší možné kvalitě. Mezi tyto modely se také řadí model RUP (Rational Unified Process). Jedná se o model s iterativním přístupem, který využívá již existující komponenty. RUP je vyvíjen primárně pro softwarové projekty, ale internetové aplikace lze také chápat jako software.

Postupů pro dosažení požadovaného cíle je několik. Nejjednodušším a zároveň nejpoužívanějším způsobem je vodopádový model. Ten sice není nejefektivnější, ale pro svou jednoduchost je možné jej použít pro velkou část internetových projektů. Pro rozsáhlé internetové či intranetové aplikace, které v dnešní době dosahují vysoké úrovně, je ovšem nutné zvolit efektivnější řešení, kterým může být právě RUP.⁴³ Vzhledem k tomu, že na vývoji tohoto modelu se podílejí velké společnosti působící v oblasti vývoje software, lze tento model brát jako nejefektivnější řešení.

Výstupem každé fáze životního cyklu projektu je nějaká součást projektu. Jak tato součást pracuje, či jak vznikala, by mělo být pro další potřeby popsáno

⁴¹ SLOUP, Petr. Vytvoření aplikačního software pro podporu marketingu prodeje automobilů. [s.l.], 2002. 50 s. Diplomová práce.

⁴² ŠEVČÍK, Matěj. Řízení softwarových projektů. [s.l.], 2006. 67 s. Masarykova Univerzita, Fakulta Informatiky. Vedoucí diplomové práce RNDr. Jaroslav Ráček, Ph.D. Dostupný z WWW: http://is.muni.cz/th/4266/fi_m/Rizeni_softwarovych_projektu.pdf.

⁴³ VONDRÁK, Ivo. Úvod do softwarového inženýrství. 1.1. aktualiz. vyd. Ostrava, 2002. 74 s.

a zaznamenáno. Specifika těchto dílčích dokumentů je opět možné rozdělit do dvou základních rovin dle typu dokumentu. Podoba dokumentů, které se váží k organizaci, vychází ze základních pravidel manažerské práce a jejich podoba se téměř neliší, ať se jedná o projekt softwarový nebo například o vývoj automobilu.⁴⁴ Jak by měla vypadat technická projektová dokumentace, tomu se již věnují odborníci přímo z oboru. Z jakých informací by se projektová dokumentace měla skládat pak uvádí například již zmiňovaná organizace IEEE ve svých pravidlech a doporučeních.⁴⁵

I v tomto případě lze považovat doporučení organizací IEEE pro vývoj softwaru jako správné a zároveň i relevantní pro internetové projekty. Tento fakt potvrzují také prameny, které jsou na tuto problematiku zaměřeny.⁴⁶

Významným ověřením výsledků teoretické roviny této práce byla také rovina praktická. Dotazníkovým šetřením bylo ověřeno, že u projektů, u kterých je dodržována projektová dokumentace, je výrazně menší pravděpodobnost výskytu problému po předání hotového projektu. Problémem mějme v tomto případě na mysli například upozornění na novou funkcionalitu, kterou zákazník vyžaduje, ale která nebyla na začátku práce ani zmíněna.

⁴⁴ HAUGHEY, Duncan. *Project Planning A Step by Step Guide* [online]. c2000-2009 [cit. 2009-04-19]. Dostupný z WWW: <<http://www.projectsmart.co.uk/project-planning-step-by-step.html>>.

⁴⁵ DELANEY, Declan, BROWN, Stephen. *DOCUMENT TEMPLATES FOR STUDENT PROJECTS IN SOFTWARE ENGINEERING* [online]. 2002 [cit. 2009-04-19]. Dostupný z WWW: <<http://www.utdallas.edu/~chung/ieee-templates.pdf>>.

⁴⁶ SHELFORD, Thomas J., REMILLARD, Gregory A. *Eal Web Project Management: Case Studies and Best Practices*. [s.l.] : Addison Wesley Professional, 2002. 336 s. ISBN 0321112555.

Dalším ověřením správnosti výsledků bylo vypracování modelového projektu s využitím faktů získaných v předchozích částech. Intranetový projekt dodržoval stanovené postupy a v rámci jeho vypracovávání nedocházelo k žádným problémům. Tento fakt lze proto také chápat jako ověření správnosti zjištěných faktů.

V kontextu předložených výsledků a jejich výše uvedenému shrnutí, lze konstatovat, že byl nalezen správný postup pro práci na internetových projektech a zároveň byl předložen vzorový soubor dílčích dokumentů projektové dokumentace.

6 Závěr

Tato práce se v kontextu s cíli snaží představit projektovou dokumentaci jako nástroj pro zvýšení kvality vypracovávaných internetových projektů a zároveň jako možnost ke zvýšení efektivity celého vývojového procesu. Dále předkládá postupy pro práci v projektovém týmu, dílčí dokumenty projektové dokumentace a ukázkou projektové dokumentace s využitím faktů získaných v teoretické analýze celého problému. Uvedené cíle vedly k objasnění některých otázek zkoumané problematiky. Základním problémem v probírané tématice byla podoba projektové dokumentace internetových projektů.

Projektová dokumentace neslouží pouze jako místo pro uchovávání informací o projektu. Pokud není její význam podceňován a je jí využíváno v plné míře, může výrazně usnadnit práci všech členů týmu.

Velkým problémem, se kterým se projektová dokumentace setkává v praxi, je povědomí, že vedení projektové dokumentace je zbytečné, zdlouhavé a zbytečně prodražuje celý vývoj. Dle průzkumů není projektová dokumentace dodržována zejména menšími projektovými týmy. Velké profesionální týmy si jsou vědomy možností projektové dokumentace a zároveň je její přítomnost vyžadována zákazníky, kteří vyžadují profesionální přístup.

Teoretická část se snažila nalézt pravidla pro tvorbu projektové dokumentace formou teoretické analýzy problému. Poukazuje také na vazby mezi projektovou dokumentací a fázemi životního cyklu projektu. Vzhledem ke zvyšování nároků na kvalitu a schopnosti internetových aplikací ze strany zákazníků se také mění postupy práce v týmu. Dílčí dokumenty projektové dokumentace se tak také vyvíjí.

Na základě uvedené teoretické analýzy došlo k definování základních postupů pro práci v projektovém týmu s důrazem na projektovou dokumentaci

a její podobu v jednotlivých fázích. Ta se mění nejen v závislosti na fázi, ve které se projekt nachází, ale také na typu daného projektu. Pokud bychom chtěli zachovat stejnou podobu dokumentů u všech druhů projektů, byla by pak její podoba jen velmi obecná.

Výše překládaná teoretická analýza se snaží poskytnout základ pro dokumentaci internetových projektů podložený analýzou dostupných pramenů a podpořený dotazníkovým šetřením. To bylo realizováno prostřednictvím dotazníku ve formátu MS Word, který byl e-mailem rozeslán na společnosti poskytující služby vývoje internetových aplikací.

Vedení projektové dokumentace neznamena pro vývojový tým žádné dodatečné náklady a jediným faktorem, který tak může ovlivnit její vznik, je čas. Čas jsou sice peníze, ale dodatečné náklady, které s sebou mohou přinést problémy během vývoje či po jeho skončení, bývají výrazně vyšší. Používání a vedení projektové dokumentace nic nebrání a jako podklad může projektovým týmům posloužit vzorová projektová struktura s dílčími dokumenty, která je připravena na přiloženém CD.

Práce si nekladla za cíl obsáhnout celou problematiku vývoje internetových projektů, ale snaží se přispět ke zvýšení kvality internetových aplikací a celého vývojového procesu.

Použitá literatura

- [1] BROWN, Alan W., et al. *Principles of CASE Tool Integration*. [s.l.] : [s.n.], 1994. 288 s. ISBN 0195094786.
- [2] BURDMAN, Jessica. *Collaborative Web Development: Strategies and Best Practices for Web Teams*. [s.l.] : [s.n.], 1999. ISBN 0201433311.
- [3] COHENOVÁ, June. *Neobyčejně užitečná kniha o webu*. Jan Kuklínek. Praha : SoftPress, 2004. 372 s. ISBN 80-86497-63-1.
- [4] COLBY, John. *Practical Intranet Development*. [s.l.] : [s.n.], 2003. 358 s. ISBN 190415123X.
- [5] DELANEY, Declan, BROWN, Stephen. *DOCUMENT TEMPLATES FOR STUDENT PROJECTS IN SOFTWARE ENGINEERING* [online]. 2002 [cit. 2009-04-19]. Dostupný z WWW: <<http://www.utdallas.edu/~chung/ieee-templates.pdf>>.
- [6] ERIKSSON, Hans-Erik, PENKER, Magnus. *Business Modeling with UML: Business Patterns at Work*. [s.l.] : [s.n.], 2000. 459 s. ISBN 0471295515.
- [7] GUCKENHEIMER, Sam, PEREZ, Juan J. *Efektivní softwarové projekty*. [s.l.] : Zoner Press, 2007. 288 s. ISBN 9788086815626.
- [8] HAUGHEY, Duncan. *Project Planning A Step by Step Guide* [online]. c2000-2009 [cit. 2009-04-19]. Dostupný z WWW: <<http://www.projectsmart.co.uk/project-planning-step-by-step.html>>.
- [9] JALOTE, Pankaj. *Software Project Management in Practice*. [s.l.] : Addison-Wesley Professional, 2002. 288 s. ISBN 0201737213.

- [10] Je testování software věda?. *DSM*. 1.1.2005, č. prosinec, s. 5. Dostupný z WWW: <<http://www.pdqm.cz/Publications/TestVeda.pdf>>.
- [11] KRÁL, Jaroslav. *Přednášky předmětu PA102* : Fakulta informatiky Masarykovi univerzity, 2005.
- [12] LINOWSKI, Jakub. *Wireframes Magazine* [online]. c2009 [cit. 2009-04-15]. Dostupný z WWW: <<http://wireframes.linowski.ca/>>.
- [13] MARMEL, Elaine. *Microsoft Office Project 2003 Bible*. [s.l.] : [s.n.], 2003. 960 s. ISBN 978-0-7645-4252-7.
- [14] MCNAMARA, Carter. *Basic Sample of Board of Directors Meeting Minutes* [online]. c1997-1998 [cit. 2009-04-18]. Dostupný z WWW: <<http://managementhelp.org/boards/minutes.htm>>.
- [15] NOVÁK, Milan. *Využití webcastingových systémů ve vzdělávání*. [s.l.], 2007. 211 s. Vedoucí dizertační práce PhDr. Vladimír Rambousek, CSc.
- [16] POLANSKÝ, Ondřej. *Software a jeho licence* [online]. 2007 [cit. 2009-04-18]. Dostupný z WWW: <<http://www.dsl.cz/clanky-dsl/clanek-739/software-a-jeho-licence-1-2>>.
- [17] Rational Software - IBA CZ [online]. 2008 [cit. 2009-04-13]. Dostupný z WWW: <<http://www.ibacz.eu/-Rational-Software->>.
- [18] SHELFORD, Thomas J., REMILLARD, Gregory A. *Real Web Project Management: Case Studies and Best Practices*. [s.l.] : Addison Wesley Professional, 2002. 336 s. ISBN 0321112555.
- [19] SLOUP, Petr. *Vytvoření aplikačního software pro podporu marketingu prodeje automobilů*. [s.l.], 2002. 50 s. Diplomová práce.

- [20] SUTHERLAND, Erci. *Requirements for the Creative Brief*. [s.l.] : [s.n.], 2008. 25 s. ISBN 978-1-906440-01-5.
- [21] *SVN tutoriál* [online]. 2007 [cit. 2009-04-15]. Dostupný z WWW: <http://merlin.fit.vutbr.cz/wiki/index.php?title=SVN_tutori%C3%A1l>.
- [22] ŠEVČÍK, Matěj. *Řízení softwarových projektů*. [s.l.], 2006. 67 s. Masarykova Univerzita, Fakulta Informatiky. Vedoucí diplomové práce RNDr. Jaroslav Ráček, Ph.D. Dostupný z WWW: http://is.muni.cz/th/4266/fi_m/Rizeni_softwarovych_projektu.pdf.
- [23] VONDRÁK, Ivo. *Úvod do softwarového inženýrství*. 1.1. aktualiz. vyd. Ostrava, 2002. 74 s.
- [24] WIEGERS, Karl E. *Software Requirements*. Redmond : Microsoft press, 2003. 516 s. ISBN 0-7356-1879-8.
- [25] *Wireframe webu* [online]. c1999-2009 [cit. 2009-04-15]. Dostupný z WWW: <<http://www.symbio.cz/slovník/wireframe-webu.html>>.

7 Přílohy

Přílohou této bakalářské práce je CD-ROM, který obsahuje:

- Elektronickou podobu této práce
- Dílčí dokumenty projektové dokumentace
- Předpřipravené projektové struktury dle typu projektu
- Ukázky vypracované projektové dokumentace
 - o Vzorový zápis z jednání
 - o Příklad přidělování a sledování úkolů
 - o Záznam o testování
 - o Přehled vkládaných souborů
 - o Zabezpečení