

Dokumentácia k riadeniu

Tímový projekt

Vypracovali: Bc. Timotej Betina
Bc. Andrej Lukeš
Bc. Martin Práznovský
Bc. Miroslav Šimulčík
Bc. Marek Tuška
Bc. Tomáš Uherčík
Vedúci projektu: Ing. Peter Lacko, PhD.
Akademický rok: 2010/2011
Kontakt: fiittp6@googlegroups.com

Obsah

Obsah	2
Zoznam tabuliek	3
Zoznam obrázkov	4
1 Úvod	1-1
1.1 Prehľad dokumentu	1-1
2 Ponuka	2-1
3 Úlohy členov	3-1
3.1 Roly	3-1
3.2 Autorstvo častí dokumentácie	3-2
4 Plán projektu	4-1
4.1 Dlhodobý plán	4-1
4.1.1 Verzia 1.0	4-1
4.1.2 Verzia 2.0	4-2
4.1.3 Verzia 3.0	4-3
4.2 Krátkodobé úlohy	4-4
4.3 Ganttov diagram	4-6
5 Podporné prostriedky riadenia	5-1
5.1 Komunikácia v tíme	5-1
5.2 Google Code	5-1
5.3 SVN	5-4
5.4 Webstránka	5-5
6 Konvencie, metodiky	6-1
6.1 Konvencie modelovania	6-1
6.2 Dátové modely	6-4
6.3 Dokumentácia	6-4
6.4 Inštalácia prostredia	6-5

6.4.1	Inštalácia Bitnami Wappstack	6-5
6.4.2	Inštalácia frameworku Yii	6-5
6.4.3	Konfigurácia potrebných rozšírení PHP	6-5
6.4.4	Vytvorenie projektu v Yii	6-6
6.4.5	Inštalácia Eclipse pre PHP	6-6
6.4.6	Inštalácia intellisence do Eclipse	6-6
6.5	SVN	6-6
6.5.1	Inštalácia Subclipse	6-6
6.5.2	Konfigurácia Subclipse	6-7
6.5.3	Používanie Subclipse	6-7
6.5.4	Základné operácie a pravidlá	6-8
6.5.5	Pravidlá pri práci so Subversion	6-9
6.6	Štábná kultúra	6-9
6.6.1	Názvy	6-9
6.6.2	Odsadenia	6-10
6.6.3	Písanie zátvoriek	6-11
6.6.4	Písanie podmienok	6-11
6.6.5	Písanie switch	6-12
6.6.6	Komentovanie	6-12
6.7	Unit testovanie	6-13
6.7.1	Vytvorenie testu	6-14
6.7.2	Spustenie testu	6-15
6.7.3	Implementácia funkcionality	6-16
6.7.4	Spustenie všetkých testov	6-16
6.7.5	Refaktoring vytvoreného kódu	6-16
6.8	Revízia kódu	6-16
7	Zápisnice zo stretnutí	7-1
7.1	Zápisnica z 1. stretnutia	7-2
7.2	Zápisnica z 2. stretnutia	7-4
7.3	Zápisnica z 3. stretnutia	7-6
7.4	Zápisnica zo stretnutia so zákazníkom č. 1	7-9
7.5	Zápisnica zo 4. stretnutia	7-11
7.6	Zápisnica zo stretnutia so zákazníkom č. 2	7-14
7.7	Zápisnica z 5. stretnutia	7-15
7.8	Zápisnica zo stretnutia ku kontrole kvality softvéru	7-17
7.9	Zápisnica zo 6. stretnutia	7-18
7.10	Zápisnica zo 7. stretnutia	7-20
7.11	Zápisnica z 8. stretnutia	7-23
7.12	Zápisnica z 9. stretnutia	7-26

Zoznam tabuliek

3.1	Roly členov tímu	3-2
3.2	Autorstvo častí dokumentácie	3-5
4.1	Verzia plánu 1.0 - zimný semester	4-1
4.2	Verzia plánu 1.0 - letný semester	4-2
4.3	Verzia plánu 2.0 - zimný semester	4-2
4.4	Verzia plánu 2.0 - letný semester	4-3
4.5	Verzia plánu 3.0 - zimný semester	4-3
4.6	Verzia plánu 3.0 - letný semester	4-4
4.7	Krátkodobé úlohy	4-6
6.1	Notácia diagramov a modelov	6-3

Zoznam obrázkov

4.1	Ganttov diagram	4-8
5.1	Google Code - prehľad úloh	5-3
5.2	Google Code - vytvorenie úlohy	5-4
5.3	SVN repozitár na Google Code	5-5
5.4	Úvodná stránka webovej prezentácie tímu Lakeri	5-6
6.1	Diagram aktivít pre postup vývoja metódy testovanej unit testom	6-14
6.2	Diagram procesu revízie	6-17
6.3	Vloženie pripomienky do zdrojového kódu inštruktorom	6-19

Kapitola 1

Úvod

Účelom tohto dokumentu je zdokumentovať riadenie tímu v rámci projektu *Portál pre časopis ACM* na predmete *Tímový projekt*. Projekt je riešený tímom č. 6 - *Lakeri* počas dvoch semestrov v akademickom roku 2010/2011.

1.1 Prehľad dokumentu

Na začiatku sa nachádza ponuka, ktorú sme vypracovali pri výbere témy projektu. Podarilo sa nám získať nami preferovanú tému. V tejto časti sú zároveň krátko predstavení členovia tímu. Nasleduje prerozdelenie rolí vrámci tímu a krátkodobé úlohy, ktoré sme doteraz (14. decembra 2010) riešili. Ďalšou kapitolou je plán projektu na zimný semester. Nasledujúca kapitola sa zaoberá nami používanými podpornými prostriedkami. Predposledná kapitola sa zaoberá štandardami písania kódu, dokumentácie a metodikami. Poslednými sú kópie zápisníc zo stretnutí.

Kapitola 2

Ponuka

Nasleduje ponuka tak, ako sme ju odovzdali (okrem titulnej stránky).

Predstavenie členov tímu

Bc. Timotej Betina – Absolvent bakalárskeho štúdia na FIIT STU v odbore Informatika ocenený diplomom Magna Cum Laude, momentálne pokračuje v odbore Softvérové inžinierstvo. Bakalárske štúdium ukončil prácou zameranou na webové a multimediálne technológie za ktorú bol nominovaný na cenu dekana. V súčasnosti pracuje ako programátor a podieľa sa na vývoji webových systémov. Má skúsenosti s programovacími jazykmi a súvisiacimi technológiami ako java (j2ee, framework seam, ...), C# (ASP.NET), PHP a databázovými systémami MSSQL, MySQL a PostgreSQL.

Bc. Andrej Lukeš – absolvent bakalárskeho programu Informatika na Fakulte informatiky a informačných technológií. Má skúsenosti s programovacími jazykmi JAVA, C#, HTML. V súvislosti s jazykom HTML by spomenul, že vytvoril webovú stránku určenú pre študentov gymnázia, z dôvodu uľahčenia a modernizácie vyučovacieho procesu. V oblasti programovania webových stránok by sa chcel ďalej zdokonaľovať, rozšíriť si vedomosti, pretože je to oblasť, v ktorej ho baví pracovať a vzdelávať sa. Nápomocné v tejto oblasti by mu mohli byť predmety Pokročilé databázové technológie a Bezpečnosť v internete, ktoré má zapísané tento akademický rok.

Bc. Martin Práznovský – podobne ako väčšina členov nášho tímu je čerstvým bakalárom FIIT v odbore informatika. Vo väčšine doterajších projektov pracoval v jazykoch C a C#, je otvorený novým poznatkom, ktoré aj od predmetu Tímový projekt očakáva. V tíme je odhodlaný zastávať akúkoľvek pozíciu, od testera až po manažéra tímu.

Mgr. Richard Strapko – minuloročný absolvent Manažérskej matematiky na FMFI UK. Jeho programovacie skúsenosti sú prevažne z matematických softvérov, no rád si osvojí aj iné jazyky a rozšíri svoje znalosti. Kvôli jeho doterajšiemu štúdiu je skôr vhodný na manažérske pozície, no nebráni sa žiadnej.

Bc. Miroslav Šimulčík – Absolvent bakalárskeho študijného programu Informatika ocenený diplomom Magna Cum Laude. Štúdium ukončil vypracovaním bakalárskej práce Informačný systém na podporu tvorby webových dotazníkov, za ktorú získal pochvalný list dekana. Zaujíma sa o webové technológie, ktorým sa venuje aj mimo štúdia. Má skúsenosti s programovacími jazykmi (PHP, JavaScript + jQuery framework, Java, C) a databázovými technológiami (PostgreSQL, MySQL).

Bc. Marek Tuška – absolvent bakalárskeho programu Informatika na Fakulte Informatiky a Informačných Technológií. Má skúsenosti s tvorbou aplikácií v jazyku Java, C a C#. Vypracoval bakalársku prácu na tému Podpora tvorby a vykonávania semestrálnych rozvrhov. Je schopný vykonávať akúkoľvek funkciu.

Bc. Tomáš Uherčík – absolvoval bakalárske štúdium na FIIT STU v odbore Informatika, ocenený diplomom Magna Cum Laude. Témou jeho bakalárskej práce bol Informačný systém na podporu vyhodnocovania webových dotazníkov, za túto prácu mu bol udelený pochvalný list dekana. Momentálne pokračuje na FIIT STU v odbore Softvérové inžinierstvo. Má skúsenosti s programovacími jazykmi Java, PHP, Javascript, C, s databázovými technológiami PostgreSQL, MSSQL, s technológiami AJAX, XML a pod.

Portál pre časopis (téma 10)

Zaujíma sa o pôsobenie organizácie ACM, preto sa nám páči možnosť podieľať sa na jej fungovaní na Slovensku. S touto organizáciou sme sa stretli pri hľadaní informácií potrebných pre projekty v rámci nášho bakalárskeho štúdia. Umožnenie komunikácie odborníkov a následnú dostupnosť publikovaných materiálov pre širokú odbornú verejnosť považujeme za jeden zo základných prvkov zlepšenia výskumu v oblasti informačných technológií. Mohlo by to pomôcť zvýšiť záujem študentov a vedeckých pracovníkov o publikačnú činnosť a zvýšiť úroveň publikácií, čo by následne zdvihlo celosvetové hodnotenia slovenských univerzít. Zároveň nás zaujala koncepcia tohto zadania, nakoľko sa jedná o projekt situovaný v prostredí webu. Súčasný stav portálu neumožňuje manažment obsahu časopisu, pridávanie článkov a hodnotenie článkov a taktiež chýba digitálna knižnica s prehľadnou organizáciou a možnosťou vyhľadávania. Jedná sa prakticky o statickú internetovú stránku, čo nie je dostačujúce pre daný účel. Vyhovujúcou alternatívou pre túto problematiku by bolo vytvorenie CMS systému použitím dynamických webových technológií.

Náš tím má dostatok predchádzajúcich skúseností s vytváraním CMS systémov, pretože väčšina členov vytvárala podobné systémy v rámci svojich bakalárskych prác (Informačný systém na podporu tvorby webových dotazníkov, Informačný systém na podporu vyhodnocovania webových dotazníkov, Výučbový systém využívajúci multimediálne prezentovanie informácií). Piati zo siedmich členov tímu dostali ocenenie za vynikajúco vypracovanú bakalársku prácu, čo svedčí o zodpovednom prístupe pri riešení projektov. Skúsenosti nadobudnuté v rámci týchto projektov by sme radi premietli do projektu celoslovenského významu.

Koncepcia riešenia:

Východiskom pri riešení tohto projektu bude analýza súčasného portálu pre časopis. Momentálne tento systém umožňuje:

- Zasielanie článkov prostredníctvom webového formulára na server.
- Manuálne publikovanie článkov po ich schválení administrátorom stránky (ide o statickú stránku)
- Prístup k jednotlivým vydaniam časopisu vo formáte PDF dokumentov

Zo súčasných možností systému sme vyvodili nasledovné ciele projektu:

- Doplniť CMS systém, aby sa automatizovala správa obsahu stránky časopisu – vytvorenie nástroja, ktorý by umožňoval vkladanie formátovaného textu a multimediálnych súborov, bez nutnosti použitia html, formátovanie priamo v prehliadači.
- Zabezpečiť dostupnosť časopisu aj vo webovej podobe - možnosť prezerania časopisu vo forme html (bez nutnosti sťahovania PDF dokumentu).
- Naimplementovať podporu posudzovania článkov – možnosť pridania odborného posudku, možnosť užívateľského hodnotenia na určitej stupnici, zoradenie článkov podľa počtu prečítaní a podľa odozvy čitateľov.

- Doplnenie digitálnej knižnice, ktorá by umožňovala správu a vyhľadávanie jednotlivých publikácií – možnosť rôznych prehľadov, podľa tém, ktorých sa články týkajú, možnosť filtrovania článkov a vyhľadávanie s možnosťou voľby rôznych parametrov.
- Pridanie správy užívateľských účtov – možnosť prihlásenia užívateľa, na základe jeho oprávnení v systéme mu budú sprístupnené časti systému, na ktoré má oprávnenie. Napríklad možnosť meniť obsah časopisu, možnosť vkladať odborné posudky, možnosť vkladať články a podobne.

Pri riešení budeme analyzovať medzinárodnú stránku organizácie ACM – www.acm.org. Pokúsime sa o dosiahnutie výsledku na podobnej úrovni. Budeme klásť dôraz na jednoduchosť používania, prehľadnosť, bezpečnosť a interaktivitu. Systém by mal využívať moderné technológie z oblasti webu. Radi by sme použili PHP ako jazyk na strane servera a JavaScript (AJAX) ako klientský jazyk kvôli zvýšeniu interaktivity. Ako databázu navrhujeme použiť PostgreSQL. Tento výber bol podmienený hlavne predchádzajúcimi skúsenosťami s týmito technológiami, ich rozšírenosťou a dostupnosťou a v neposlednom rade aj tým, že sú voľne dostupné.

Crowdsourcing verejných dát (téma 15)

Táto téma nás zaujala nie len z hľadiska jej technickej realizácie, ale hlavne tým, čo je cieľom tohto projektu. Každý jednotlivec v spoločnosti sa prirodzene zaujíma o to, ako sa s jeho financiami nakladá a týmto projektom sa dostaneme o kúsok bližšie k pravde. Jedná sa predsa aj o peniaze z našich daní a už len pomyslenie na toto je dostatočná motivácia pre celý náš tím, dostať zo seba čo najviac počas práce na tomto projekte.

Portál s týmto zameraním zatiaľ na Slovensku neexistuje a bola by to pre nás česť vytvoriť ho. Plusom pre nás by bola možnosť stretnúť sa a komunikovať s reálnym zákazníkom (Aliancia Fair-play) a potenciálna možnosť náš systém nasadiť do reálnej prevádzky. Uvedomujeme si, že k tomu bude treba plne funkčný systém. My v našom tíme sa však neuspokojíme len s nejakými prototypmi, či nedokončenými projektami. Dôkazom je napríklad udelenie pochvál piatim členom nášho tímu za bakalárske práce.

Väčšina členov nášho tímu má skúsenosti s vytváraním webových aplikácií a väčšie či menšie skúsenosti s html, php, css a databázami. Nasadenie systému na linuxový stroj nebude pre nás tiež žiadnym problémom. Nikto z nás sa však doteraz nestretol s Ruby. Neznalosť Ruby sa môže zdať ako nevýhoda, opak je však pravdou. My budeme len radi, že sa môžeme naučiť používať modernú a perspektívnu technológiu a obohatiť svoje vedomosti o niečo nové. Nezoberte nám preto túto jedinečnú príležitosť a vyberte si náš tím!

Koncepcia riešenia:

Použitie technológií/jazykov bolo v podstate zadané spolu s témou

(<http://www2.fiit.stuba.sk/~bielik/courses/tp-slov/org/temy.html#Crowdsourcing>). Využívať sa budú moderné, v súčasnosti populárne technológie a pri vývoji bude použitá metodika SCRUM s dôrazom na testovanie.

Nakoľko aplikácia s podobným zameraním zatiaľ neexistuje (na Slovensku), bude potrebné začať od základov. Ciele projektu a hlavné funkcie systému by sme zatiaľ zhrnuli do niekoľkých nasledovných bodov:

- Vytvorenie web aplikácie určenej na spracovávanie zverejnených dokumentov
- Umožnenie interaktívnej práce s týmito dokumentmi
 - Konvertovanie do spoločného formátu
 - Extrahovanie textu z naskenovaných dokumentov
 - Vyhľadávanie v dokumentoch
 - Označovanie podozrivých častí dokumentov

- Využitie davu (crowdsourcing)
 - Registrácia používateľov, hodnotenie používateľov a príspevkov
 - Prepojenie so sociálnymi sieťami (facebook)
- Dôraz na bezpečnosť celého systému

Správa študentských projektov na fakulte (téma 12)

Systém na správu študentských projektov na fakulte môže výrazne skvalitniť a urýchliť proces spojený s prípravou zadaní, výberom projektov, pridelovaním oponentov a inými činnosťami. Každý člen nášho tímu mal možnosť vyskúšať si systém Yonban, ktorý sa v súčasnosti používa na fakulte už 8. rok. Hneď po prvej návšteve nás zarazil dizajn a absencia moderných technológií. Okrem týchto nedostatkov sa počas dlhodobého používania objavila aj ďalšia funkcionálna, ktorú by bolo vhodné do systému doplniť, prípadne upraviť tak, aby spĺňal súčasné požiadavky. Záujem o webové technológie, chuť zlepšiť správu projektov na fakulte a vízia nasadenia systému v rámci univerzity nás viedli k výberu práve tejto témy.

Náš tím má dostatok predchádzajúcich skúseností s vytváraním CMS systémov, pretože väčšina členov vyvíjala podobné systémy v rámci bakalárskych prác (Informačný systém na podporu tvorby webových dotazníkov, Informačný systém na podporu vyhodnocovania webových dotazníkov, Výučbový systém využívajúci multimediálne prezentovanie informácií). Skúsenosti nadobudnuté v týchto projektoch by sme radi využili na vytvorenie systému, ktorý by nahradil Yonban a pomáhal študentom a vyučujúcim pri ich práci s projektami aspoň ďalších 8 rokov.

Koncepcia riešenia:

Východiskom pri riešení tohto projektu bude analýza súčasného systému (yonban). Keďže v súčasnosti máme prístup len k malej množine funkcionálnych, ktoré yonban obsahuje, je obtiažne ho komplexne zhodnotiť. Z pohľadu študenta možno pozorovať napríklad tieto nedostatky:

- zastaraný dizajn, neprehľadnosť systému.
- komplexnejšie nedostatky ako napríklad minimálna funkcionálna, ktorú študentom yonban poskytuje pri voľbe témy, kde by bolo vhodné použitie viacerých podporných prvkov, ktoré študentovi sprehľadnia a uľahčia výber.

Ciele pri vytváraní systému:

- analýza procesu absolvovaného študentami a pedagógmi a jeho uľahčenie pre jednu aj pre druhú stranu, jednoduchosť používania, prehľadnosť, automatizácia niektorých krokov, bezpečnosť, modulárnosť a iné.

Ďalší aspekt, na ktorý je potrebné sa zamerať, je využitie moderných webových technológií. Väčšina členov nášho tímu má skúsenosti s viacerými technológiami, a preto dokážeme zhodnotiť potreby budúceho systému s ohľadom na rôzne aspekty (rozsah projektu, rozšíriteľnosť do budúcnosti,

atď.). Taktiež sú dôležité určité doplnkové technológie, ktoré zjednodušujú, sprehľadňujú a spríjemňujú prácu používateľa so systémom (napr. AJAX, jQuery ako framework pre Javascript).

Po nasadaní akademického informačného systému nastala situácia, že mnohé vlastnosti sú v yonbane a v AISe duplicitné. Preto je dôležité, aby vlastnosti, ktoré poskytuje AIS, využíval aj budúci systém na správu projektov. Medzi tieto vlastnosti patrí napríklad:

- prihlasovanie pedagógov a študentov, vkladanie študentských prác, priamy náhľad pedagógov na študijné výsledky študentov uchádzajúcich sa o témy, atď.

Vlastnosti budúceho systému:

- Príprava zadaní projektov pedagógmi – v nej sú dôležité veci ako priame vytváranie a úprava zadaní v systéme, pridávanie odkazov na existujúce práce s podobnou tematikou, kategorizovanie prác, upozorňovanie pedagógov o deadlineoch, vytvorenie kalendára s voľnými časmi na konzultáciu prác.
- Výber projektu študentami - prehľadný a organizovaný podľa kategórií; študenti by si mohli označiť niektoré projekty o ktoré majú záujem a prideliť im prioritu; prihlasovanie sa do kalendára daného pedagóga na konzultácie; vyriešenie situácie, keď je študentovi pridelená práca a tým pádom mu už nie je možné prideliť inú, ale daný študent mal prioritný záujem o inú prácu.

Organizácia študentských projektov z minulých rokov - študenti a pedagógovia si budú môcť prehľadávať tieto práce a získavať predstavu o ich budúcej práci a podobne. V tejto časti je dôležitá prehľadná organizácia projektov, ale taktiež efektívne a rýchle vyhľadávanie z implementačnej stránky.

Príloha A - Zoradenie všetkých tém podľa priority

preferencia	číslo témy	Názov
1	10	Portál pre časopis (Časopis)
2	15	Crowdsourcing verejných dát (CrowdPublic)
3	12	Správa študentských projektov na fakulte (Projekty)
4	11	RoboCup - tretí rozmer (RoboCup 3D)
5	5	Tvorba rozvrhov (Rozvrhy)
6	9	Prispôsobiteľný Widget (Widget)
7	7	Tréner mentálnych schopností (Tréner)
8	1	Objektové úložisko dát (Úložisko)
9	4	Platforma pre realizovanie transakcií prostredníctvom mobilných zariadení (Mobily)
10	2	Adaptívny proxy server (Proxy-plugins)
11	19	Model používateľa pre jeho identifikáciu (UserModel)
12	13	Vyhľadávanie a sprístupnenie citácií (Portál)
13	6	Interaktívna vizualizácia grafových štruktúr v 3D priestore (Vizualizácia)
14	14	Dizajn s použitím obohatenej reality (Dizajn)
15	18	3D grafická podpora vyhľadávania znalostí v dokumentoch (3D-Znalosti)
16	8	Virtuálna FIIT (VFIIT)
17	17	Simulated Car Racing Competition 2011 (Car Racing)
18	3	Evolučný simulátor umelého života založený na heuristických pravidlách (HERBAL)
19	16	Imagine Cup 2011: Game Design (ICup2011)

Príloha B - Aktuálny rozvrh všetkých členov tímu

		7.00-7.50	8.00-8.50	9.00-9.50	10.00-10.50	11.00-11.50	12.00-12.50	13.00-13.50	14.00-14.50	15.00-15.50	16.00-16.50	17.00-17.50	18.00-18.50	19.00-19.50	20.00-20.50
Pondelok	Andrej						PDT		prednáška			TPI		VIS	
	Marek						PDT		prednáška			TPI		VIS	
	Martin								ZK			TPI		VSS	
	Timotej								PDT			TPI		VSS	
	Tomáš								PDT			TPI		VSS	
	Miro								PDT			TPI		VSS	
	Richard								ML II			TPI		VIS	
Utorok	Andrej									MSI			MSI		MSI
	Marek									MSI			MSI		MSI
	Martin									MSI			MSI		MSI
	Timotej		ZK							MSI			MSI		MSI
	Tomáš		ZK							MSI			MSI		MSI
	Miro		ZK							MSI			MSI		MSI
	Richard									MSI			MSI		MSI
Streda	Andrej														
	Marek														
	Martin														
	Timotej											ZK			
	Tomáš											ZK			
	Miro											ZK			
	Richard		Marketing						ML II						
Štvrtok	Andrej							ZK					AIS		
	Marek							ZK					AIS		
	Martin				Návrh prekladačov	Návrh prekladačov		ZK							
	Timotej										ASS				
	Tomáš										ASS				
	Miro										ASS				
	Richard								PDT				AIS		
Piatok	Andrej														
	Marek														
	Martin														
	Timotej				PDT										
	Tomáš				PDT										
	Miro				PDT										
	Richard				PDT										

Legenda:

- TPI - Timový projekt I
- PDT - Pokročilé databázové technológie
- VIS - Výskum informačných systémov
- VSS - Výskum softvérových systémov
- MSI - Manažment projektov softvérových a informačných systémov
- ZK - Základy kryptografie
- AIS - Architektúra informačných systémov
- ASS - Architektúra počítačových systémov
- ML II - Matematická logika II

prednáška	cvičenie	vyhovujúci čas	nevyhovujúci čas	vyhovujúci čas všetkým
-----------	----------	----------------	------------------	------------------------

Pozn.

Podľa dohody s vyučujúcim navrhujeme (v poradí podľa preferencií): 1. utorok, 2. streda, 3. iný termín

Kapitola 3

Úlohy členov

3.1 Roly

Obsahom tejto kapitoly je rozdelenie úloh v tíme, ktoré prebehlo po vzájomnej diskusii a po zohľadnení vzájomných priorit a vedomostí jednotlivých členov tímu. ¹.

¹Mgr. Richard Strapko ukončil inžinierske štúdium na FIIT, jeho úlohy boli dodatočne rozdelené vrámci tímu

Zodpovedná osoba	Rola	Popis
Bc. Timotej Betina	Manažér kvality	Testovanie, definovanie používaných metód a štandardov
Bc. Andrej Lukeš	Manažér plánovania, správca webového sídla	Vytvorenie a aktualizovanie webovej stránky tímu, vytváranie plánu projektu
Bc. Martin Práznovský	Manažér komunikácie, manažér PR, vedúci dokumentarista	Komunikácia v rámci tímu, tvorba dokumentácie
Bc. Miroslav Šimulčík	Vedúci tímu, manažér rizík	Pridelovanie úloh v tíme, riadenie a komunikácia v tíme/so zákazníkmi
Bc. Marek Tuška	Zástupca vedúceho, dokumentarista	Pridelovanie úloh, dokumentácia
Bc. Tomáš Uherčík	Manažér podporných prostriedkov	Inštalácia a konfigurácia nástrojov na podporu tvorby softvéru, komunikácie, riadenia

Tabuľka 3.1: Roly členov tímu

3.2 Autorstvo častí dokumentácie

V tabuľke 3.2 sú zosumarizované jednotlivé kapitoly nachádzajúce sa v dokumentácii spolu s ich tvorcami.

Kapitola	Názov kapitoly	Autor
Dokumentácia		
1	Úvod	Práznovský
1.1	Účel a prehľad dokumentu	Práznovský
2	Analýza	Práznovský
2.1	Analýza existujúcich riešení	Práznovský
2.1.1	Taylor & Francis Group Journal	Betina
2.1.2	SpringerLink	Šimulčík
2.1.3	IOS Press	Lukeš
2.1.4	Elsevier	Tuška
2.2	Analýza CMS systémov	Práznovský
2.2.1	WordPress	Lukeš
2.2.2	Joomla!	Práznovský

2.2.3	Drupal	Tuška
2.2.4	Zhodnotenie	Práznovský
2.3	Analýza serverových technológií	Betina
2.3.1	PHP	Betina
2.3.2	ASP.NET	Betina
2.3.3	J2EE	Betina
2.3.4	Ruby on Rails	Betina
2.3.5	Zhodnotenie	Betina
2.4	Analýza PHP frameworkov	Uherčík
2.4.1	Zend	Uherčík
2.4.2	Yii	Uherčík
2.4.3	CakePHP	Uherčík
2.4.4	CodeIgnier	Uherčík
2.4.5	Zhodnotenie	Uherčík
2.5	Databázové systémy	Šimulčík
2.6	Nástroj pre fulltextové vyhľadávanie	Šimulčík
3	Špecifikácia	—
3.1	Kontext systému	Šimulčík
3.2	Identifikácia používateľov	Šimulčík
3.3	Špecifikácia údajov	Šimulčík
3.4	Funkcionálne požiadavky	Uherčík
3.5	Nefunkcionálne požiadavky	Betina
4	Návrh	—
4.1	Prípady použitia	Práznovský
4.1.1	Autor	Lukeš, Tuška, Práznovský
4.1.2	Používateľ	Uherčík, Betina
4.1.3	Školiteľ	Práznovský
4.1.4	Admin	Lukeš, Tuška, Práznovský
4.2	Stavové diagramy	—
4.2.1	Stavové diagramy entity článok	Uherčík
4.2.2	Stavové diagramy entity dizertácia	Uherčík
4.3	Logický dátový model	Šimulčík
4.4	Návrh obrazoviek	Betina
5	Prototyp	—
5.1	Ciele prototypovania	Práznovský
5.2	Implementácia prototypu	—
5.2.1	Architektúra systému	Šimulčík
5.2.2	Fyzický dátový model	Šimulčík
5.3	Ukážky prototypu	Betina
5.4	Dosiahnuté výsledky	Betina

5.5	Inštaláčna príručka	Betina
Dokumentácia k riadeniu		
1	Úvod	Práznovský
1.1	Prehľad dokumentu	Práznovský
2	Ponuka	Všetci
3	Úlohy členov	Tuška
3.1	Roly	Tuška
3.2	Autorstvo častí dokumentácie	Lukeš
4	Plán projektu	Lukeš
4.1	Dlhodobý plán	—
4.1.1	Verzia 1.0	Lukeš, Uherčík
4.1.2	Verzia 2.0	Lukeš, Uherčík
4.1.3	Verzia 3.0	Lukeš, Uherčík
4.2	Krátkodobé úlohy	Tuška, Lukeš
4.3	Ganttov diagram	Lukeš
5	Podporné prostriedky riadenia	Lukeš
5.1	Komunikácia v tíme	Lukeš
5.2	Google code	Lukeš, Uherčík
5.3	SVN	Práznovský
5.4	Webstránka	Lukeš
6	Konvencie, metodiky	—
6.1	Konvencie modelovania	Lukeš, Šimulčík
6.2	Dátové modely	Šimulčík
6.3	Dokumentácia	Práznovský, Tuška
6.4	Inštalácia prostredia	Uherčík
6.4.1	Inštalácia Bitnami Wappstack	Uherčík
6.4.2	Inštalácia frameworku Yii	Uherčík
6.4.3	Konfigurácia potrebných rozšírení PHP	Betina
6.4.4	Vytvorenie projektu v Yii	Betina
6.4.5	Inštalácia Eclipse pre PHP	Uherčík
6.4.6	Inštalácia intellisence pre Eclipse	Uherčík
6.5	SVN	—
6.5.1	Inštalácia Subclipse	Lukeš
6.5.2	Konfigurácia Subclipse	Lukeš
6.5.3	Používanie Subclipse	Lukeš
6.5.4	Základné operácie a pravidlá	Lukeš
6.5.5	Pravidlá pri práci so Subversion	Lukeš
6.6	Štábna kultúra	Betina
6.6.1	Názvy	Betina
6.6.2	Odsadenia	Betina

6.6.3	Písanie zátvoriek	Betina
6.6.4	Písanie podmienok	Betina
6.6.5	Písanie switch	Betina
6.6.6	Komentovanie	Betina
6.7	Unit testovanie	Uherčík
6.7.1	Vytvorenie testu	Uherčík
6.7.2	Spustenie testu	Uherčík
6.7.3	Implementácia funkcionality	Uherčík
6.7.4	Spustenie všetkých testov	Uherčík
6.7.5	Refaktoring vytvoreného kódu	Uherčík
6.8	Revízia kódu	Šimulčík
7	Zápisnice zo stretnutí	Práznovský
7.1	Zápis z 1. stretnutia	Strapko
7.2	Zápis z 2. stretnutia	Uherčík
7.3	Zápis z 3. stretnutia	Tuška
7.4	Zápis zo stretnutia so zákazníkom	Betina
7.5	Zápis z 4. stretnutia	Práznovský
7.6	Zápis zo stretnutia so zákazníkom	Tuška
7.7	Zápis z 5. stretnutia	Lukeš
7.8	Zápis zo stretnutia ku kontrole kvality	Lukeš
7.9	Zápis z 6. stretnutia	Tuška
7.10	Zápis z 7. stretnutia	Šimulčík
7.11	Zápis z 8. stretnutia	Lukeš
7.12	Zápis z 9. stretnutia	Práznovský
	Preberací protokol	Tuška

Tabuľka 3.2: *Autorstvo častí dokumentácie k projektu a dokumentácie k riadeniu*

Kapitola 4

Plán projektu

4.1 Dlhodobý plán

4.1.1 Verzia 1.0

Dátum vypracovania: 30.9.2010

Zimný semester			
Číslo	Názov úlohy	Predpokladaný dátum ukončenia	Dátum ukončenia
1.	Vytvorenie webovej stránky	5.10.2010	
2.	Predbežná analýza a identifikácia požiadaviek	19.10.2010	
3.	Revízia požiadaviek po stretnutí so zákazníkom, úprava analýzy	26.10.2010	
4.	Návrh	30.10.2010	
5.	Vytvorenie prototypu	14.12.2010	

Tabuľka 4.1: *Verzia plánu 1.0 - zimný semester*

Letný semester			
Číslo	Názov úlohy	Predpokladaný dátum ukončenia	Dátum ukončenia
1.	Revízia prototypu podľa požiadaviek zákazníka	Nešpecifikované	
2.	Implementácia kompletného systému	Nešpecifikované	
3.	Testovanie systému	Nešpecifikované	
4.	Nasadenie systému	Nešpecifikované	

Tabuľka 4.2: Verzia plánu 1.0 - letný semester

4.1.2 Verzia 2.0

Dátum vypracovania: 25.10.2010

Zimný semester			
Číslo	Názov úlohy	Predpokladaný dátum ukončenia	Dátum ukončenia
1.	Vytvorenie webovej stránky	5.10.2010	3.10.2010
2.	Predbežná analýza a identifikácia požiadaviek	19.10.2010	18.10.2010
3.	Revízia požiadaviek po stretnutí so zákazníkom, úprava analýzy	26.10.2010	25.10.2010
4.	Návrh fulltextového vyhľadávania	30.10.2010	
5.	Podrobný návrh prípadov použitia	9.11.2010	
6.	Vytvorenie prototypu	14.12.2010	

Tabuľka 4.3: Verzia plánu 2.0 - zimný semester

Úloha 4. pre zimný semester – *Návrh* z verzie 1.0 pokračuje vo verzii 2.0 ako úloha 4. *Návrh fulltextového vyhľadávania* a úloha 5. *Podrobný návrh prípadov použitia*. Úloha 5 verzie 1.0 *Vytvorenie prototypu* má vo verzii 2 číslo 6.

Letný semester			
Číslo	Názov úlohy	Predpokladaný dátum ukončenia	Dátum ukončenia
1.	Revízia prototypu podľa požiadaviek zákazníka	Nešpecifikované	
2.	Implementácia kompletného systému	Nešpecifikované	
3.	Testovanie systému	Nešpecifikované	
4.	Nasadenie systému	Nešpecifikované	

Tabuľka 4.4: Verzia plánu 2.0 - letný semester

4.1.3 Verzia 3.0

Dátum vypracovania: 20.11.2010

Zimný semester			
Číslo	Názov úlohy	Predpokladaný dátum ukončenia	Dátum ukončenia
1.	Vytvorenie webovej stránky	5.10.2010	3.10.2010
2.	Predbežná analýza a identifikácia požiadaviek	19.10.2010	18.10.2010
3.	Revízia požiadaviek po stretnutí so zákazníkom, úprava analýzy	26.10.2010	25.10.2010
4.	Návrh fulltextového vyhľadávania	30.10.2010	30.10.2010
5.	Podrobný návrh prípadov použitia	9.11.2010	8.11.2010
6.	Prototyp vyhľadávania	14.12.2010	
7.	Prototyp vkladania článkov	14.12.2010	
8.	Prototyp zobrazenia článkov	14.12.2010	

Tabuľka 4.5: Verzia plánu 3.0 - zimný semester

Úloha 6. pre zimný semester – *Vytvorenie prototypu* z verzie 2.0 pokračuje vo verzii 3.0 ako úloha 6. *Prototyp vyhľadávania*, úloha 7. *Prototyp vkladania článkov* a úloha 8. *Prototyp zobrazenia článkov*.

Letný semester			
Číslo	Názov úlohy	Predpokladaný dátum ukončenia	Dátum ukončenia
1.	Revízia prototypu podľa požiadaviek zákazníka	Nešpecifikované	
2.	Implementácia kompletného systému	Nešpecifikované	
3.	Testovanie systému	Nešpecifikované	
4.	Nasadenie systému	Nešpecifikované	

Tabuľka 4.6: Verzia plánu 3.0 - letný semester

4.2 Krátkodobé úlohy

Táto kapitola obsahuje úlohy, ktoré riešili členovia tímu v dobe medzi jednotlivými stretnutiami.

Úloha	Začiatok	Ukončenie	Osoba
Vytvorenie webstránky tímu	28.9.2010	5.10.2010	Andrej
Vytvorenie plagátu tímu	28.9.2010	5.10.2010	Andrej, Marek
Výber a konfigurácia podporného prostriedku na komunikáciu – Google Groups	28.9.2010	5.10.2010	Miro, Tomáš
Analýza súčasného portálu, procesy, biznis logika	28.9.2010	5.10.2010	Všetci
Výber a konfigurácia prostriedku na podporu vývoja – Google Code	28.9.2010	5.10.2010	Miro
Vytvorenie prípadov použitia a aktivity diagramov – admin	5.10.2010	12.10.2010	Andrej, Martin
Vytvorenie prípadov použitia a aktivity diagramov – používateľ	5.10.2010	12.10.2010	Marek
Vytvorenie prípadov použitia a aktivity diagramov – recenzant	5.10.2010	12.10.2010	Timo
Vytvorenie stavového diagramu	5.10.2010	12.10.2010	Tomáš
Vytvorenie logického modelu + MARC 21	5.10.2010	12.10.2010	Miro
Analýza Springerlink	5.10.2010	12.10.2010	Miro
Analýza IOSPress	5.10.2010	12.10.2010	Andrej
Analýza Tandf	5.10.2010	12.10.2010	Timo
Analýza Elsevier	5.10.2010	12.10.2010	Marek
Analýza Joomla	12.10.2010	19.10.2010	Martin

Analýza Drupal	12.10.2010	19.10.2010	Marek
Analýza WordPress	12.10.2010	19.10.2010	Andrej
Analýza a výber technológií	19.10.2010	26.10.2010	Miro, Timo, Tomáš
Revízia diagramov (prípady použitia, aktivity a stavové diagramy, logický model)	19.10.2010	26.10.2010	Všetci
Stretnutie so zákazníkmi: prof. Bieliková, prof. Návrat	20.10.2010	20.10.2010	Miro, Timo, Tomáš
Finálna revízia diagramov	26.10.2010	30.10.2010	Andrej, Marek, Martin, Miro, Tomáš
Stretnutie so zákazníkmi: Ing. Andrejčíková	28.10.2010	28.10.2010	Všetci
Návrh obrazoviek	26.10.2010	30.10.2010	Timo
Vytvorenie dokumentácie – časť riadenie	26.10.2010	30.10.2010	Andrej, Marek
Vytvorenie finálnej dokumentácie na odovzdanie	26.10.2010	2.11.2010	Martin
Vytvorenie a zverejnenie metodiky k SVN a Subclipse	09.11.2010	12.11.2010	Andrej
Vytvorenie a zverejnenie štábnej kultúry	09.11.2010	14.11.2010	Timotej
Štúdium a inštalácia frameworku Yii, Bitnami, Subclipse	09.11.2010	14.11.2010	Všetci
Metodika na inštaláciu vývojového prostredia	09.11.2010	14.11.2010	Tomáš
Vytvorenie databázy podľa dátového modelu	16.11.2010	23.11.2010	Miro
Implementácia fulltext vyhľadávania	16.11.2010	30.11.2010	Miro
Vytvorenie základnej štruktúry prototypu - inicializácia	16.11.2010	23.11.2010	Timo, Tomáš
Úpravenie dizajnu a vytvorenie views podľa návrhu obrazoviek	16.11.2010	23.11.2010	Marek, Andrej, Martin
Implementácia vyhľadávania v článkoch	23.11.2010	30.11.2010	Miro, Timo, Tomáš
Implementácia zobrazenia zoznamu, detailov a vkladania	23.11.2010	30.11.2010	Marek, Andrej, Martin

V submission form zmeniť polia titul before a titul after na text-field	30.11.2010	7.12.2010	Marek
V submission form vytvoriť text-field pre titul supervisora	30.11.2010	7.12.2010	Marek
Zmeniť supervisora na usera	30.11.2010	7.12.2010	Marek
Pridať do databázy adresu k inštitúcií a vyriešiť ukladanie používateľov	30.11.2010	7.12.2010	Miro
Opraviť chybu v používateľskom rozhraní	30.11.2010	7.12.2010	Andrej
Písanie dokumentácie k riadeniu a dokumentácie	30.11.2010	7.12.2010	Andrej, Martin, Tomáš
Pridať do databázy adresu k inštitúcií	30.11.2010	7.12.2010	Miro
Dokončenie vyhľadávania(rozšírené aj jednoduché)	30.11.2010	7.12.2010	Miro
Ukladanie fulltextu do databázy zo zdrojových kódov latexu	30.11.2010	7.12.2010	Miro
Kategorizácia článkov	30.11.2010	7.12.2010	Tomáš
Výsledky vyhľadávania, zoznam článkov	30.11.2010	7.12.2010	Timo
Dohodnutie času prezentácie	7.12.2010	14.12.2010	Martin
Uloženie metodiky na wiki	7.12.2010	14.12.2010	Miro
Úpravy databázy	7.12.2010	14.12.2010	Miro

Tabuľka 4.7: Krátkodobé úlohy

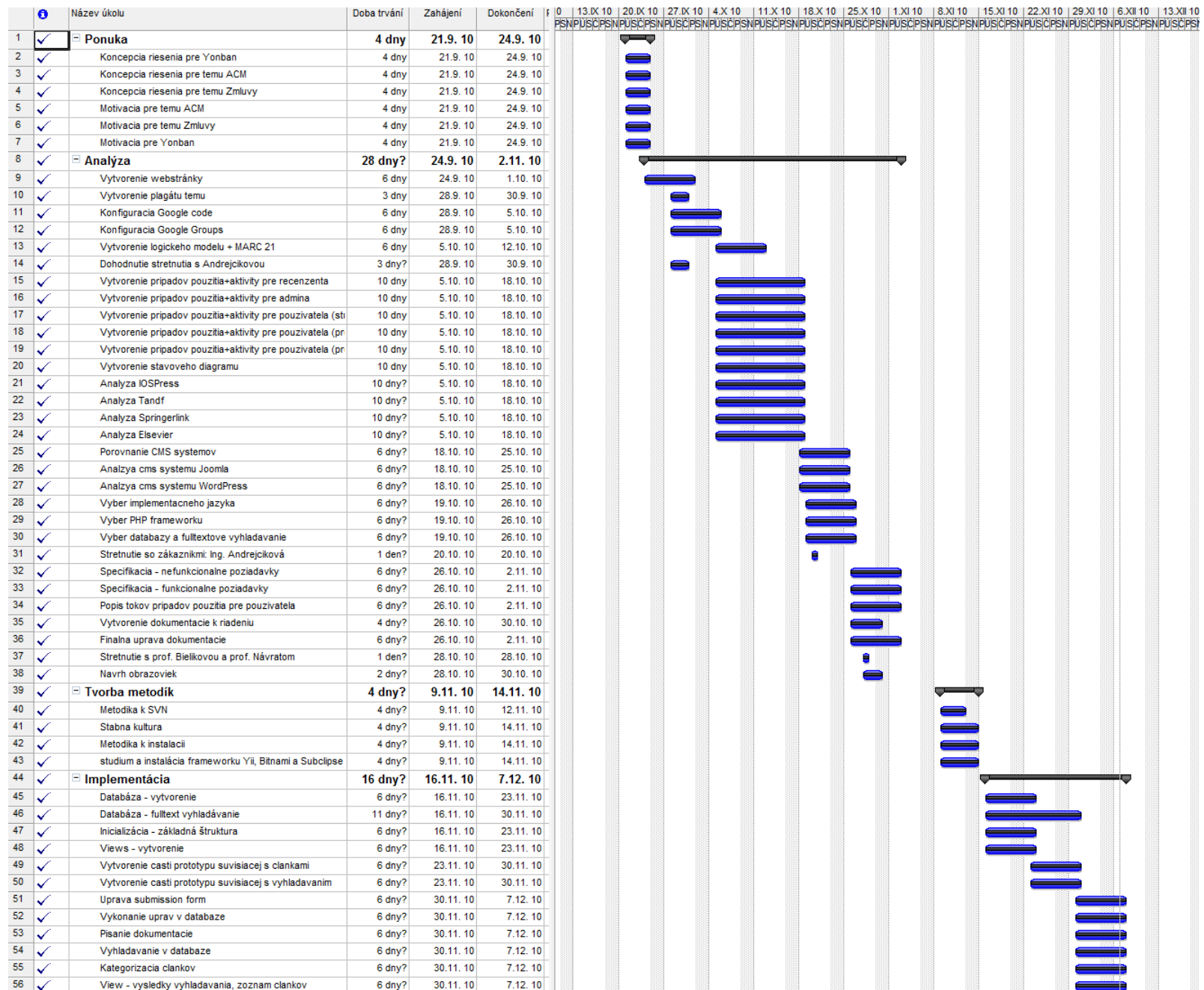
4.3 Ganttov diagram

Na evidenciu úloh sme použili Google Code. Avšak tu sa dajú evidovať len jednotlivé úlohy, no nedá sa vytvoriť Ganttov diagram. Jednotlivé úlohy je však možné exportovať do CSV súboru. Na vytvorenie Ganttovho diagramu sme použili nástroj Microsoft Project.

Microsoft Project je nástroj, ktorý slúži na manažment projektov. Slúži na podporu riadenia projektov, spravovanie úloh, zdrojov a zisťovanie aktuálneho stavu projektu. Poskytuje rôzne možnosti zobrazenia ako napr. Ganttov diagram, kalendáre, používanie úloh, sieťový diagram, zobrazovanie a spravovanie zdrojov.

Pomocou Ganttovho diagramu sa dajú zobraziť jednotlivé úlohy, ktoré sme riešili v rámci projektu. Ganttov diagram pozostáva z časovej osi, na ktorej je zobrazené trvanie

úloh. Úloha je teda reprezentovaná vodorovným úsekom, pričom dĺžka tohto úseku reprezentuje dĺžku trvania úlohy.



Obr. 4.1: Ganttov diagram - časť úloh v zimnom semestri. Vytvorený pomocou nástroja MS Project.

Kapitola 5

Podporné prostriedky riadenia

V tejto kapitole sú popísané podporné prostriedky, ktoré sú využité v tímovom projekte.

5.1 Komunikácia v tíme

Komunikácia v rámci tímu zohráva veľmi dôležitú úlohu. Potreba práce na projekte je veľmi rozsiahla, nestačia len stretnutia k tímovému projektu. Keďže niektorí členovia bývajú mimo Bratislavu, nie je možné stretnúť sa vždy osobne. Preto je použitie nástroja na komunikáciu nevyhnutné. V rámci komunikácie používame chat dostupný na Facebooku a taktiež ICQ. Ďalej sme vytvorili skupinu na Facebooku, kde môžu členovia taktiež diskutovať formou hromadného chatu. Hlavné formu komunikácie však tvorí skupina, ktorú sme založili na portáli groups.google.com. Dajú sa tu zakladať diskusie, pričom členovia tejto grupy môžu do diskusií prispievať a o jednotlivých príspevkoch sú informovaní mailom. Diskusná skupina je pre pozvaných členov dostupná na adrese <http://groups.google.com/group/fiittp6/>.

5.2 Google Code

Požadované vlastnosti nástroja pre manažment úloh:

- Jednoduchosť – budú ho používať všetci členovia tímu
- Možnosť definície predlôh – úlohy majú mať štandardizovaný tvar
- Podpora svn a wiki
- Možnosť definície stavu úloh
- Pridelenie riešiteľov jednotlivých úloh

Rozhodli sme sa použiť nástroj Google Code, viedla nás k tomu najmä možnosť prispôbenia nástroja požiadavkám nášho projektu. Použitie nástroja:

- Nástroj používajú všetci členovia tímu
- Úlohy vkladá
 - Zapisovateľ po stretnutí
 - Člen tímu, ktorý zistil, že je vhodné definovať úlohu, ktorá nebola diskutovaná na stretnutí – nutnosť konzultácie s ostatnými členmi tímu, najmä s vedúcim tímu
- Riešiteľ úlohy mení stav úlohy podľa toho v akej fáze riešenia je. Typické stavy sú napríklad:
 - Pridelená
 - Riešená
 - Splnená
 - Zrušená
- Využitie SVN – z pohľadu sledovania úloh je výhodou najmä umožnenie podrobného sledovania úloh, nakoľko je možné na základe jednotlivých vložení sledovať :
 - Zmeny, ktoré riešiteľ spravil
 - Čas, kedy vkladal čiastočné riešenia
- Prehľady úloh:
 - Filtrovanie úloh na základe príznakov
 - Základný prehľad úloh v zozname
 - Tabulkový prehľad, v ktorom je možné pridelit príznak riadku a stĺpcu na základe čoho je možné vytvorit prispôbené prehľady podľa záujmových atribútov

Select: All None	Actions...	1 - 43 of 43	List Grid				
StartWeek	EndWeek	Type	Status	Priority	Owner	Cc	Summary + Labels
1	2	Task	Done	High	timotej.betina	timotej.betina	Koncepcia riešenia pre Yonban
1	2	Task	Done	High	tomass.u	tomass.u	Motivacia a koncepcia riesenia pre temu ACM
1	2	Task	Done	High	marektuska	martin.praznovsky, andrej.lukes1, marektuska	Vypracovanie ponuky pre temu Zmluvy
1	2	Task	Done	Medium	andrej.lukes1	andrej.lukes1	* Vytvorenie webstránky
1	2	Task	Done	Medium	andrej.lukes1	marektuska, andrej.lukes1	* Vytvorenie plagátu tímu
1	2	Task	Done	High	simulcik.miro	simulcik.miro	Motivacia pre Yonban
1	2	Task	Done	High	simulcik.miro	simulcik.miro	Koncepcia riesenia pre temu ACM
2	3	Task	Done	Medium	simulcik.miro	timotej.betina, simulcik.miro	Konfiguracia Google code
2	3	Task	Done	Medium	simulcik.miro	simulcik.miro	Konfiguracia Google Groups
3	4	Task	Done	High	simulcik.miro	simulcik.miro	Vytvorenie logického modelu + MARC 21
3	4	Task	Done	Medium	simulcik.miro	simulcik.miro	Analýza Springerlink
3	4	Task	Done	Medium	andrej.lukes1	andrej.lukes1	* Analýza IOSPress
3	4	Task	Done	Medium	timotej.betina	timotej.betina	Vytvorenie pripadov použitia+aktivity pre recenzenta

Obr. 5.1: Google Code - prehľad úloh

Prispôsobenie nástroja:

- Vytvorenie predlôh na vytváranie úloh
 - Vytvorenie vlastných popisov úloh:
 - Kvôli faktu, že stretnutia sa konajú každý týždeň je vhodné pridelit úlohám:
 - * Týždeň pridelenia úlohy
 - * Predpokladaný koniec
- Realizácia vlastnými príznakmi úloh
- * Týždeň začatia
 - * Týždeň ukončenia

Vytvorenie úlohy (obrázok 5.2)

1. Člen tímu, zadávajúci úlohu, vyberie vytvorenie novej úlohy
2. Vyberie predlohu podľa toho o aký typ úlohy sa jedná
3. Vyberie príznaky úlohy:
 - Typ úlohy – pri použití predlohy to nie je nutné, lebo je preddefinovaný

- Týždeň pridelenia úlohy
 - Predpokladaný týždeň ukončenia úlohy
4. Vyplní údaje na základe predlohy
 5. Vloží úlohu

Obr. 5.2: Google Code - vytvorenie úlohy

Komentovanie

- Počas riešenia úlohy je vhodné písať komentáre k úlohe, najmä ak sa jedná o nové zistenie pri riešení úlohy a úlohu riešia viacerí členovia tímu.

5.3 SVN

Pre manažment verzií zdrojových kódov a dokumentácie sme sa rozhodli použiť systém SVN (z anglického *subversion*). SVN server je poskytovaný v rámci Google Code. Je na výber z množstva programov na klientskej strane, napr. TortoiseSVN, alebo plugin Subclipse priamo do vývojového prostredia Eclipse.

Verzie (revízie) sú verejne dostupné¹. Zápis majú povolený iba členovia tímu, ich prístupové informácie sa nachádzajú po prihlásení na stránke Google Code.

¹<http://code.google.com/p/acm-portal/source/browse/>

acm-portal
Portal pre acm v rámci tímového projektu 2010/2011

Project Home Downloads Wiki Issues Source

Checkout | Browse | Changes | Search Trunk | Request code review

Source path: svn/ < r2 r3 r4 >

Directories	Filename	Size	Rev	Date	Author
▼svn	dokumentacia_fiitp6_aux	1.2 KB	r2	Oct 24, 2010	martin.praznovsky
branches	dokumentacia_fiitp6_lof	90 bytes	r2	Oct 24, 2010	martin.praznovsky
tags	dokumentacia_fiitp6_log	23.8 KB	r2	Oct 24, 2010	martin.praznovsky
▼trunk	dokumentacia_fiitp6_lot	90 bytes	r2	Oct 24, 2010	martin.praznovsky
docs	dokumentacia_fiitp6.pdf	124 KB	r2	Oct 24, 2010	martin.praznovsky
images	dokumentacia_fiitp6.tex	6.5 KB	r2	Oct 24, 2010	martin.praznovsky
wiki	dokumentacia_fiitp6.toc	652 bytes	r2	Oct 24, 2010	martin.praznovsky

Obr. 5.3: SVN repozitár na Google Code

Systém má danú adresárovú štruktúru: adresáre *trunk*, *tags* a *branches*. V adresári *trunk* sme vytvorili podadresár *docs*, ktorý obsahuje súbory s dokumentáciou a *src* so zdrojovými kódmi projektu. Ďalšie adresáre a súbory môžu pribúdať počas implementácie.

Aby došlo k čo najmenej konfliktom, je potrebné dodržiavať nasledovné pravidlá:

- Robiť malé zmeny a často, nie veľa zmien po dlhšom čase
- Každú zmenu výstižne komentovať
- Zdrojový kód musí byť spustiteľný a otestovaný

5.4 Webstránka

Webová prezentácia tímu je dostupná na adrese stránke tímu ². Táto stránka je pravidelne týždenne aktualizovaná a slúži ako prezentácia tímu a jeho aktivít na tímovom projekte. Je na nej možné nájsť informácie o členoch tímu, zápisnice zo stretnutí, hrubý plán na zimný semester a jednotlivé úlohy členov tímu, pridelené v rámci týždňov.

² <http://labss2.fiit.stuba.sk/TeamProject/2010/team06is-si/index.html>






Obr. 5.4: Úvodná stránka webovej prezentácie tímu Lakeri

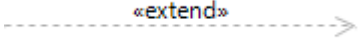
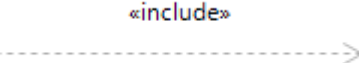



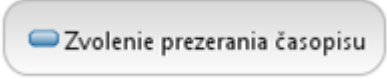
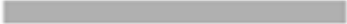

Kapitola 6



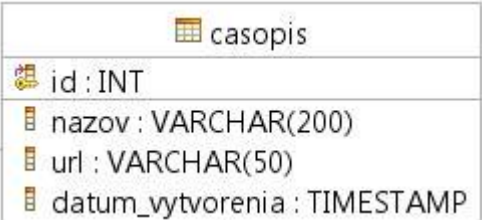

Konvencie, metodiky

6.1 Konvencie modelovania

Diagramy prípadov použitia, diagramy aktivít a stavový diagram boli vypracované pomocou nástroja IBM Rational Software Architect, pričom bola použitá notácia UML 2.0.

Značka	Popis
 Admin	Hráč – reprezentuje používateľa systému
 Vloženie výsledku obhajoby	Prípad použitia - je to interakcia medzi systémom a používateľom systému
Sú údaje správne vyplnené? 	Rozhodovací blok - rozdelenie sledu nasledujúcich aktivít na základe splnenia podmienky

	<p>Rozširuje – jeden prípad použitia rozširuje funkcionality druhého prípadu použitia, prípady použitia sú navzájom spojené pomocou rozšírenia (extend)</p>
	<p>Zahrňa – jeden prípad použitia je nevyhnutnou súčasťou druhého prípadu použitia, prípady použitia sú navzájom spojené pomocou zahrnutia (include)</p>
	<p>Používa - šípka predstavuje priradenie prípadu použitia k hráčovi</p>
	<p>Začiatkový stav - predstavuje začiatok vykonávania</p>
	<p>Koncový stav - predstavuje koniec vykonávania</p>
	<p>Aktivita - činnosť ktorá reprezentuje vykonanie určitej úlohy</p>
	<p>Rozvetvenie (fork)</p>
	<p>Spojenie (join)</p>

↓	Smer vykonávania aktivít
	Entita – v prvej časti obsahuje názov, v druhej primárne kľúče a v tretej atribúty
	Väzba – vyjadruje vzťah medzi entitami. Pozostáva z popisu a násobností. V tomto prípade ide o vzťah 1..n na ľavej strane a 0..1 na pravej strane väzby
	Tabuľka – v prvej časti obsahuje názov, v druhej primárne kľúče a v tretej názvy stĺpcov spolu s typmi
	Väzba – vyjadruje prepojenia medzi tabuľkami. Obsahuje násobnosti. V tomto prípade ide o násobnosť 0..n na ľavej strane a 1 na pravej strane väzby.

Tabuľka 6.1: Notácia diagramov a modelov

6.2 Dátové modely

Názvy tabuliek v logickom modeli aj fyzickom modeli nazývame po slovensky pre lepšie pochopenie významu pre slovenského čitateľa. Pri pomenovávaní dodržiavame tieto pravidlá:

- V logickom modeli budeme používať diakritiku, vo fyzickom nie
- Názvy tabuliek budeme zapisovať v jednotnom čísle malými písmenami a prípadné medzery v názve nahradíme podtržníkom. pomenovaním v množnom čísle jednotiek, ktoré uchovávajú
- Primárny kľúč budeme označovať v tabulkách pomenovaním id
- Cudzie kľúče budeme v tabulkách označovať podľa tabuľky, kde sa pôvodne nachádzajú. Cudzí kľúč sa tvorí podľa konvencie `id_vysklonovany_nazov_tabulky` (napr.: `id_kategorie`).
- Tabuľky, ktoré sú vo väzbe N:M prepojíme väzobnou tabuľkou, ktorú nazveme `nazov_prvej_tabulky_x_nazov_druhej_tabulky`. Cudzie kľúče do spojovacej tabuľky vytvoríme podľa konvencií, ktoré sme si definovali vyššie.

6.3 Dokumentácia

Metodika písania dokumentácie:

- Dokumentácia je písaná po slovensky
- V prípade nutnosti je možné vložiť cudzojazyčné slovo do zátvorky
 - Prípad použitia (use case)
- Je povolené používať skratky
- Všetky obrázky a tabuľky majú popis
- Vkladané dokumenty (zápisnice, ponuka atď.) sú ponechávané v pôvodnej forme
- Pri odrážkach, číslovaniach, popisoch obrázkov, tabuliek a v bunkách tabuliek sa začína veľkým písmenom a bez bodky na konci. Bodka sa použije ak ide o postupnosť viacerých viet, ako v tejto odrážke. Pri opise tokov prípadov použitia sa používa notácia s číslovaním a bodkami na konci.
- Kapitoly majú krátky a výstižný názov

6.4 Inštalácia prostredia

V tejto časti popisujeme inštaláciu odporúčaného vývojového prostredia.

6.4.1 Inštalácia Bitnami Wappstack

1. Stiahnite si Bitnami Wappstack
2. Spustite inštalačný súbor
3. Koreňový adresár webu je implicitne v `\BitNami WAPPStack\apache2\htdocs`, ak ho chcete zmeniť, tak musíte v súbore `BitNami WAPPStack\apache2\conf\httpd.conf` zmeniť `DocumentRoot`.
4. V environment variables / system variables treba do `path` pridať cestu k súboru `php.exe`

6.4.2 Inštalácia frameworku Yii

1. Stiahnite si Yii z <http://www.yiiframework.com/download/>
2. Dekomprimujte do adresára, ktorý je pod koreňovým adresárom webu (pri Bitnami Wappstack je to implicitne adresár `\BitNami WAPPStack\apache2\htdocs`)

6.4.3 Konfigurácia potrebných rozšírení PHP

1. Skontrolujte si vo webovom prehliadači cez `127.0.0.1/cesta/requirements` či máte nainštalované `PDO extension` a `PDO PostgreSQL extension` (majú stav `passed`). Ak nie tak pokračujte v nasledovných krokoch.
2. V `C:\Program Files\BitNami WAPPStack\php\php.ini` odkomentujte riadky `extension=php_pdo.dll` a `extension=php_pdo_pgsql.dll`. Musíte mať nastavené práva, aby ste mohli ukladať zmeny v súboroch v zložke `Program Files`.
3. Reštartnite server pomocou `Start → Programs → BitNami WAPPstack → BitNami WAPPstack service → Stop BitNami WAPPstack service` a následne `Start BitNami WAPPstack service`.
4. Skontrolujte či sa inštalácia podarila na `127.0.0.1/cesta/requirements`.

6.4.4 Vytvorenie projektu v Yii

1. Cez command line spustíte príkaz `yii webapp cesta\k\adresaru`. Musíte byť v adresári `C:\Program Files\BitNami WAPPStack\php` alebo mať tento adresár nastavený v premennej `PATH` v enviroment variables.
2. V `cesta\k\projektu\protected\config\main.php` odkomentujte riadky modulu Gii
3. V tom istom súbore odkomentujte aj časť pripojenia k databáze. Namiesto `mysql` napíšte `pgsql` a vyplňte potrebné údaje na pripojenie k databáze.

6.4.5 Inštalácia Eclipse pre PHP

1. Stiahnite si <http://www.eclipse.org/pdt/downloads/>
2. Skopírujte ho tam, kde ho chcete mať

6.4.6 Inštalácia intellisence do Eclipse

1. Stiahnite si súbor `yii-templates.zip` z <http://www.yiiframework.com/ext/files/?id=255>
2. Dekomprimujte `yii-templates.zip` niekam na disk
3. Spustite eclipse
4. Choďte do *Preferences* → *PHP* → *Editor* → *Templates* → *Import* a vyberte súbor `yii-templates.xml`

6.5 SVN

6.5.1 Inštalácia Subclipse

Podrobný popis inštalácie aj s obrazovým návodom je dostupný na stránke projektu ¹. Inštalácia prebieha priamo v prostredí Eclipse. Nasledovný postup vyžaduje verziu Eclipse vyššiu ako 3.2.

1. V prostredí Eclipse zvolíme *Help* → *Software updates* → *Find and Install*
2. V ďalšom okne zvolíme *Search for new features for install*

¹<http://subclipse.tigris.org/servlets/ProjectProcess?pageID=p4wYuA>

3. Zvolíme možnosť *New remote site* a do zobrazeného formulára zadáme meno a adresu URL v nasledujúcom tvare : Do položky *Name*: Subclipse 1.4.x (Eclipse 3.2+) Do položky *URL*: http://subclipse.tigris.org/update_1.4.x
4. Zaškrtneme nájdenú stránku v tom prípade, ak nie je zaškrtnutá a pokračujeme v inštalácií
5. Na ďalšej obrazovke zaškrtneme všetky položky, ktoré chceme inštalovať a stlačíme *Next*
6. Akceptujeme licenčné podmienky a stlačíme *Next*
7. Dokončíme inštaláciu potvrdením tlačidla *Instal All*
8. Reštartujeme Eclipse

6.5.2 Konfigurácia Subclipse

Po inštalácií zásuvného modulu je potrebná jeho konfigurácia, aby sa mohol pripojiť na repozitár vo vytvorenom projekte.

1. V prostredí Eclipse zvolíme *File* → *New* → *Project*
2. Vyberieme *SVN* → *Checkout project from SVN* a stlačíme *Next*
3. V tomto kroku nastavíme umiestnenie repozitáru, zvolíme možnosť *Create a new repository location* a stlačíme *Next*
4. Do políčka URL zadáme <https://acm-portal.googlecode.com/svn/trunk/>. Adresu nájdeme v záložke *Source*, v projekte, do ktorého chceme na Google code pristupovať. Klikneme na *Finish*. Vykonaním tohto kroku, sa vytvorí lokálna kópia adresára trunk.
5. Objaví sa sprievodca vytvorenia nového projektu. Tu je možné vytvoriť požadovaný projekt, na ktorom chceme pracovať. Projekt predstavuje lokálnu kópiu repozitára.

6.5.3 Používanie Subclipse

Po konfigurácií Subclipse bola na lokálnom disku vytvorená kópia repozitára. Projekt, ktorý sme v Eclipse vytvorili má totožnú adresárovú štruktúru ako repozitár umiestnený na stránke vytvoreného projektu.



Funkcie Subclipse sú prístupné po kliknutí pravým tlačidlom myši na niektorý súbor v súborovom systéme, v časti *Team*.

6.5.4 Základné operácie a pravidlá

Synchronizácia lokálnej kópie a repozitára

- Pred začatím práce je potrebné vykonať synchronizáciu lokálnej kópie s repozitárom. Klikneme pravým tlačidlom na projekt v adresárovej štruktúre a zvolíme položku *Team* → *Synchronize with Repository*.

Vytvorenie súboru

- Po vytvorení nového súboru alebo adresára vo svojej lokálnej kópii je súbor označený ikonou . Pri najbližšom potvrdení zmien (commit) sa tento súbor umiestni do repozitára: klikneme pravým tlačidlom na tento súbor resp. adresár, zvolíme *Team* → *Add to Version Control*. Ikona sa zmení na .

Potvrdenie zmien v pracovnej kópii (Commit súboru)

- Akonáhle sme spokojní s úpravou súboru (s doplnením funkcionality), je potrebné tieto zmeny zapísať do repozitára. Ak máme týchto súborov viac, klikneme na celý projekt pravým tlačidlom, zvolíme *Team* → *Commit*. Otvorí sa okno, v ktorom je možné porovnať commitovaný súbor so súborom v repozitári.

Aktualizácia lokálnej kópie (Update) a odstránenie konfliktov

- Klikneme pravým tlačidlom na projekt a zvolíme *Team* → *Update to HEAD*. Lokálna kópia sa aktualizuje na najnovšiu verziu z repozitára, to môže znamenať okrem doplnenia zdrojového kódu aj pridanie alebo odobranie súborov.

Odstránenie konfliktov

- V prípade, ak nastane konflikt môžeme ho manuálne odstrániť, zvolením : *Team* → *Edit Conflicts*. Otvorí sa *Compare editor*, v ktorom sú zobrazené oba konfliktné súbory.

Zlúčenie hlavnej vetvy vývoja s oddelenou vetvou

- Ak programátor pracuje na oddelenej vetve vývoja a chce držať krok s hlavnou vetvou vývoja, na tento účel sa používa *Team* → *Merge*.

Prechod medzi jednotlivými adresármi

- Pomocou *Team* → *Switch to another Branch/Tag/Revision*, sa môžeme prepínať medzi jednotlivými adresármi v rámci repozitára. V ďalšom okne je potrebné vybrať adresu URL, ktorá špecifikuje umiestnenie adresára a potvrdiť voľbu *Ok*.

6.5.5 Pravidlá pri práci so Subversion

Update

- používame často, aby sme vždy pracovali s najnovšou verziou.

Commit

- necommitovať súbory .classpath, .project
- commitovať len bezchybný kód
- úpravu cudzieho kódu treba najprv prekonzultovať s autorom
- každú zmenu výstižne okomentovať

6.6 Štábna kultúra

Nasledujúca časť obsahuje záväzné pokyny pre písanie kódu v PHP jazyku, ktoré budeme dodržiavať v tímovom projekte.

6.6.1 Názvy

Použitie správnych názvov je kľúčové k sprehľadneniu kódu.

Všeobecne platné princípy

- Používame anglické názvy
- Používame správne tvary slov (napr. tam kde treba sloveso bude sloveso)
- Používame rozumné a zrozumiteľné názvy
- Ak je v jednom názve viac slov ako jedno, budú napísane tak, že začiatkové písmená všetkých ďalších slov okrem prvého budú veľké (SPRÁVNE: monkeyAndElephant, NESPRÁVNE monkey_and_elephant alebo monkeyandelephant)

Triedy

Pre triedy platia nasledujúce body:

- Začínajú veľkým písmenom
- Sú to podstatné mená

Funkcie

Pre funkcie platia nasledujúce body:

- Začínajú malým písmenom

Premenné

Pre premenné platia nasledujúce body:

- Začínajú malým písmenom
- Nepoužívame nič nehovoriace názvy ako `x`; `da` podobne
- Nepoužívame magické čísla, ale vždy ich priradíme do nejakej premennej s prízná-
čným názvom

Konštanty

Pre konštanty platia nasledujúce body:

- Celé sú písané veľkými písmenami

6.6.2 Odsadenia

- Každý nový vnorený riadok musí byť odsadený tabulátorom
- Počet odsadení nie je limitovaný

Správne:

```
function func()
{
  if (something bad)
  {
    if (another thing bad)
    {
      while (more input)
      {
      }
    }
  }
}
```

6.6.3 Písanie zátvoriek

Písanie {}

- Píšeme ich vždy na novom riadku.

Správne:

```
if (catIsHungry)
{
    cat eats mouse;
}
```

Nesprávne:

```
if (catIsHungry) {
    cat eats mouse;
}
```

Písanie ()

- Použitie zátvorkovania v if podmienkach nie je prikázane explicitne pre každý logický výraz

Tieto príklady sú oba správne

```
if (a == 2 || b == 3)
if ((a == 2) || (b == 3))
```

6.6.4 Písanie podmienok

Správne:

```
if (condition)           // Comment
{
}
else if (condition)     // Comment
{
}
else                    // Comment
{
}
```

6.6.5 Písanie switch

Správne:

```
switch (...)
{
  case 1:
    ...
    // FALL THROUGH

  case 2:
    {
      $v = getWeekNumber();
      ...
    }
    break;

  default:
}
```

6.6.6 Komentovanie

- Pred každou triedou, metódou, význačnou if podmienkou a podobne, treba napísať komentár v tvare:

```
// Niečo o metode
// Niečo viac o metode
```

- Slovo TODO: bude značiť niečo, čo je potrebné v budúcnosti spraviť


```
// TODO: pridať tu a tu podmienku
```

- Slovo BUG bude označovať známe bugy v danej metóde, časti kódu a podobne

```
// BUG: [bugId] Nachádza sa tu taký a takýto bug
```

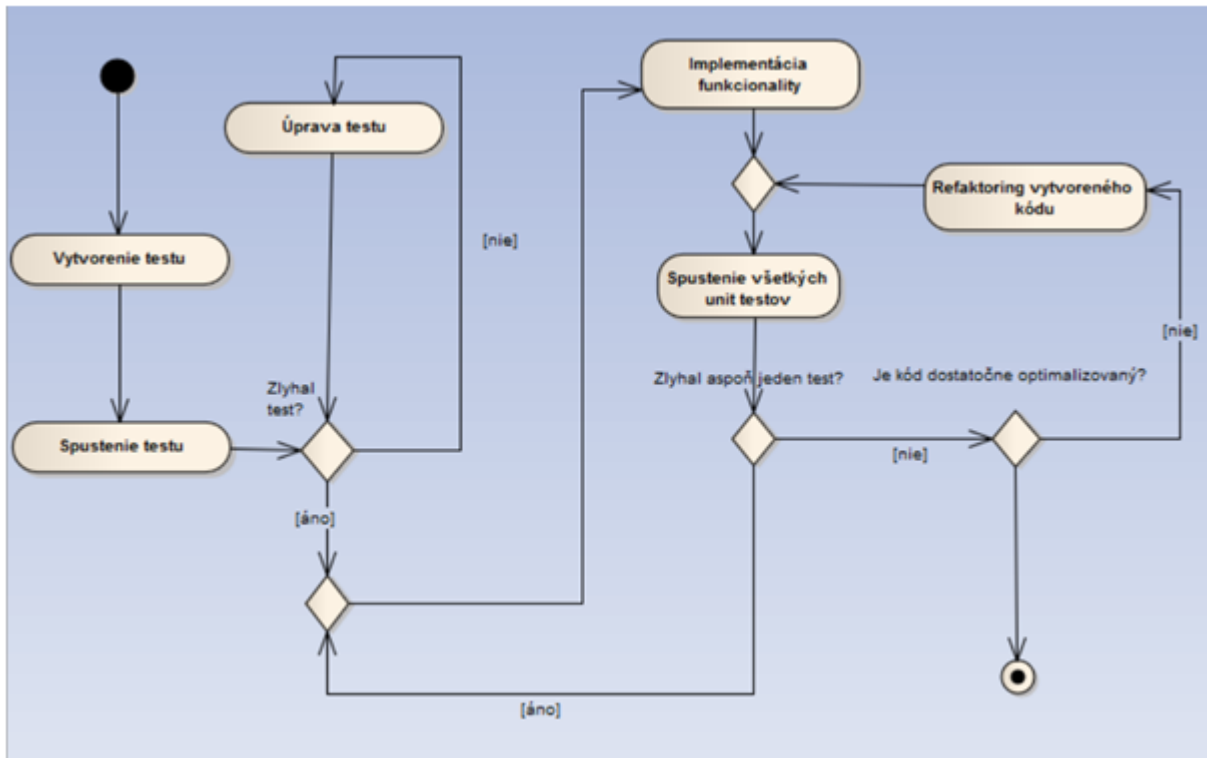
- Pred každou časťou kódu, kde chceme oboznámiť ostatných o nejakom varovaní, budeme písať slovo WARNING

```
// WARNING: Davajte si pozor na nasledujúce body: Chuck Norris nie je Bruce Lee
```

6.7 Unit testovanie

Postup práce pri implementácii metódy testovanej unit testom:

1. Programátor vytvorí test.
2. Ak prázdna metóda prejde testom, tak test určite nie je v poriadku, pokračuje sa krokom 1. Ak prázdna metóda testom neprejde, pokračuje sa krokom 3.
3. Programátor implementuje funkcionálnosť produkčného kódu.
4. Spustia sa všetky unit testy, ktoré akokoľvek závisia od nového kódu.
5. Ak zlyhal aspoň jeden test ide sa na krok 3., ak všetky testy prebehli úspešne pokračuje sa krokom 6.
6. Ak je kód dostatočne optimalizovaný tak je implementovaný kód hotový. Ak nie tak sa pokračuje krokom 7.
7. Refaktoring kódu, pokračuje sa krokom 4.



Obr. 6.1: Diagram aktivít pre postup vývoja metódy testovanej unit testom

6.7.1 Vytvorenie testu

1. Analytik určí vstupy a k nim prislúchajúce výstupy. Do vstupov je potrebné vložiť aj okrajové prípady. Unit testy musia pokryť všetky cesty v kóde testovanej metódy. Tieto prípady sú pre každú oblasť špecifické. Príkladom sú tieto body, ku ktorým je potrebné pristupovať so zvýšením dôrazom:
 - Zahrnutie hraničných hodnôt v testovacej množine.
 - Ak metóda pracuje s referenciou objektu, zahrnúť v testovacích vstupoch aj objekt s hodnotou null.
 - Ak metóda používa databázu, treba vložiť neplatné pripojenie k databáze.
2. Triedy unit testov sa odvodzujú od triedy *PHPUnit_Framework_TestCase*.
3. Testovacie metódy sú metódy s modifikátorom prístupu public a ich názov bude začínať prefixom test.
4. V testovacích metódach sa na overenie toho, či sa skutočná hodnota rovná očakávanej sa používajú metódy triedy *PHP_Framework_Assert*. Všetky tieto metódy sa začínajú prefixom assert a sú zdokumentované na <http://www.phpunit.de/manual>, pre potreby nášho projektu sú dôležité:

- `assertTrue(bool $condition[, string $message = "])` – ak neplatí podmienka definovaná výrazom `$condition`, tak sa nastane chyba definovaná výrazom `$message`.
 - `assertEquals(mixed $expected, mixed $actual[, string $message = "])` – ak sa parametre `$expected` a `$actual` nerovnajú, nastane chyba definovaná v `$message`.
5. V prípade potreby akýchkoľvek chybových výpisov v tele testovacej metódy programátor používa derivácie metódy `assert`.
 6. Závislosti medzi testovacími metódami:
 - Každá testovacia metóda, ktorá je závislá od inej metódy musí deklarovať od ktorej metódy je závislá.
 - Ak zlyhá test, od ktorého sú závislé ostatné testy, tak sa tieto testy nevykonajú, lebo ich výsledky by boli irelevantné.
 - Pri dodržaní pravidla v bode a. je bod b podporovaný priamo testovacím nástrojom a teda je možné jednoducho zistiť od ktorej testovacej metódy, sú závislé iné testovacie metódy.
 7. Príklad jednoduchého testu, vytvoreného na základe predchádzajúcich pravidiel. Metóda `testDependentAddition` je závislá od metódy `testSimpleAddition`.

```
<?php
class DependentTest extends PHPUnit_Framework_TestCase
{
    public function testSimpleAddition()
    {
        $a = 2;
        $b = 3;
        $c = $a+$b;
        $this->assertEquals(5, $c);

        return $c;
    }
    /**
     * @depends testSimpleAddition
     */
    public function testDependentAddition($c)
    {
        $this->assertEquals(5, $c);
        return $c;
    }
}
?>
```

6.7.2 Spustenie testu

Test sa dá spustiť:

1. Použitím konzolového príkazu *phpunit test.php*, kde *test.php* je názov testu, ktorý chceme spustiť. Návod na použitie pokročilých prepínačov sa zobrazí po zadaní *phpunit -help* do príkazového riadku.
2. Priamo v rozhraní vývojového prostredia Eclipse.
 - Používateľ v prehliadači projektu označí test, ktorý chce spustiť
 - Vo vrchnom menu vyberie *run->external tools->PHPUnit*.

6.7.3 Implementácia funkcionality

Programátor vytvorí alebo upraví implementáciu produkčného kódu metódy, ktorú sa ide testovať unit testom.

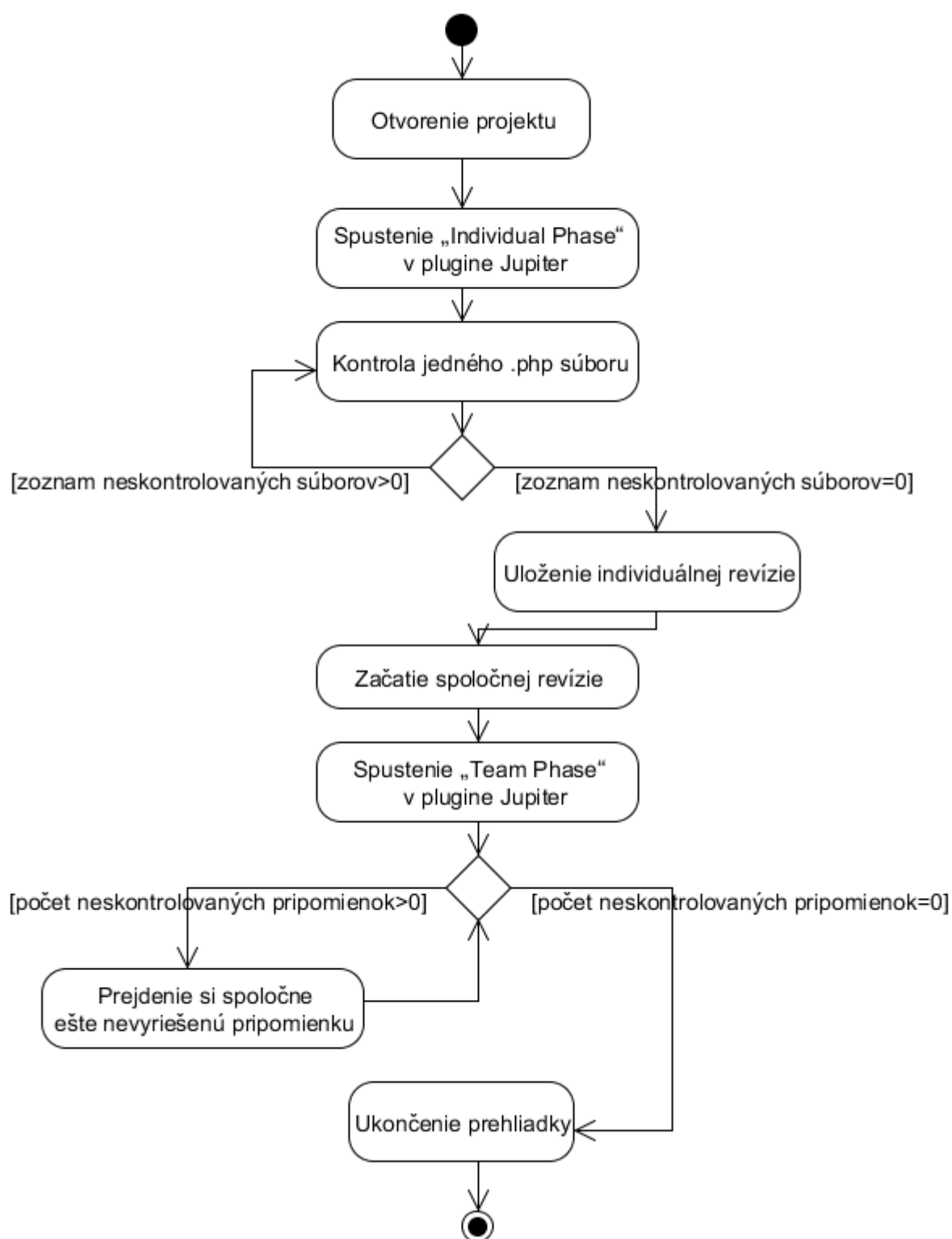
6.7.4 Spustenie všetkých testov

1. Tento krok sa vykonáva, lebo implementácia produkčnej metódy môže ovplyvniť iné časti kódu, preto treba klásť dôraz na nezanedbanie tohto kroku.
2. Všetky testy v testovacom adresári sa spustia príkazom *phpunit test*, kde *test* je adresár v ktorom sú testy.
3. Návod na použitie pokročilých prepínačov sa zobrazí po zadaní *phpunit -help* do príkazového riadku.

6.7.5 Refaktoring vytvoreného kódu

1. Vytvorený produkčný kód prešiel unit testami a teda k určeným vstupom vracia požadované výstupy.
2. Programátor môže optimalizovať kód, lebo pomocou unit testov sa dá jednoducho overiť, či metóda na určené vstupy vracia požadované výstupy.

6.8 Revízia kódu



Obr. 6.2: Diagram procesu revízie

Otvorenie projektu

- Inšpektor spustí vývojový nástroj Eclipse
- Podľa *Metodiky o správe verzií* si inšpektor aktualizuje projekt

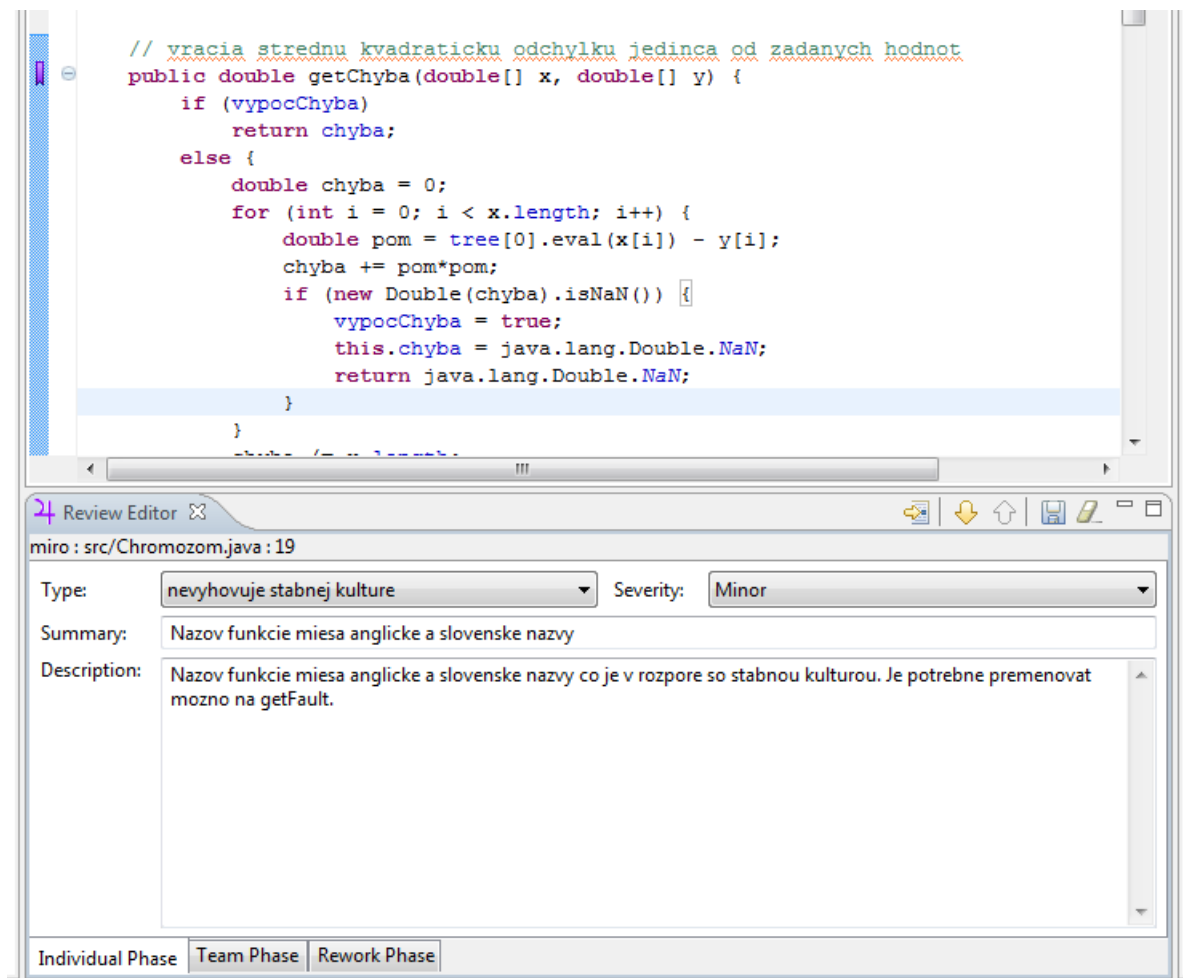
Spustenie *Individual Phase* v plugine Jupiter

- Inšpektor vyberie v Eclipse *Jupiter Perspective*
- Následne zvolí *Individual Phase* z panela nástrojov
- Na vyskočenom dialógovom okne inšpektor zvolí:
 - Projekt *ACMPortal*
 - ID prehliadky
 - Svoje ID/meno aby bolo zrejmé, kto robí danú individuálnu revíziu

Kontrola jedného .php súboru

- Inšpektor vyberie súbor so zdrojovým kódom, ktorý ešte neskontroloval
- Následne skontroluje daný súbor s ohľadom na nasledujúce aspekty kódu:
 - Vhodnosť architektonického návrhu
 - Vhodnosť názvov prvkov v kóde
 - Vhodnosť štruktúry kódu
 - Vhodnosť použitých algoritmov
 - Dodržiavanie definovanej štábnej kultúry
- Postup kontroly je nasledovný:
 - Inšpektor sa oboznámi so zoznamom chýb, ktoré zistí program na automatickú kontrolu zdrojového kódu a prezentáciou zdrojového kódu
 - Inšpektor postupne prechádza všetky riadky kódu a kontroluje ich s ohľadom na aspekty kódu, ktoré kontroluje
- Po nájdení chyby, inšpektor klikne pravým tlačidlom na zvolený riadok a vyberie *Add Review Issue...*
- Inšpektor vyplní všetky informácie o danej chybe:
 - Typ chyby vyberie zo zoznamu vytvoreného podľa (pole *Type*)
 - Chybe priradí kritickosť (pole *Severity*):

- * *Critical* – chyba je veľmi závažná, keďže obmedzuje funkčnosť kódu a závisia od nej iné časti kódu, a preto si vyžaduje neodkladnú opravu
 - * *Normal* – chyba obmedzuje funkčnosť kódu ale nie sú od nej závislé iné časti kódu, a preto je potrebné ju postupom času odstrániť
 - * *Minor* – chyba je mierna, neobmedzuje funkčnosť kódu a jej odstránenie nie je nevyhnutné
- Napíše krátky a jasný názov chyby (pole *Summary*)
 - Napíše výstižný opis chyby (pole *Description*)



Obr. 6.3: Vloženie pripomienky do zdrojového kódu inštruktorom

Uloženie individuálnej revízie

- Inšpektor uloží zoznam chýb klávesovou kombináciou *Ctrl+S*

- Inšpektor podľa *Metodiky o správe verzií* do spoločného repozitára nahrá súbor *.review* (XML súbor), ktorý obsahuje všetky zaznamenané chyby a informácie, ktoré inšpektor spísal

Začatie spoločnej revízie

- Projektový manažér, inšpektori, autor zdrojového kódu, pozorovateľ a zapisovateľ sa stretnú podľa harmonogramu procesu prehliadky
- Projektový manažér spustí vývojový nástroj Eclipse a podľa *Metodiky o správe verzií* si aktualizuje projekt
- Zapisovateľ spíše úvodné informácie do zápisnice

Spustenie *Team Phase* v plugine Jupiter

- Projektový manažér vyberie v Eclipse *Jupiter Perspective*
- Následne zvolí *Team Phase* z panela nástrojov
- Na vyskočenom dialógovom okne projektový manažér zvolí:
 - Projekt ACMPortal
 - ID prehliadky
 - Svoje ID/meno

Prejdenie si spoločne ešte nevyriešenú pripomienku

- Projektový manažér vyberie zaznamenanú pripomienku od ľubovoľného inšpektora, ktorá ešte nebola skontrolovaná
- Inšpektor vysvetlí svoje dôvody, prečo danú pripomienku zaznamenal
- Projektový manažér začne diskusiu o danej pripomienke. Diskusiu je potrebné viesť s ohľadom na:
 - Opodstatnenosť pripomienky
 - Závažnosť chyby
 - Ako ju vyriešiť
- Po vzájomnej diskusii a dohode sa pre danú pripomienku v Jupiter spíšu informácie:
 - Komu je pridelená oprava danej chyby (vždy to bude autor) (pole *Assigned To*)
 - Uznesenie o danej chybe (pole *Resolution*):

- * *Valid needs fixing* – chybu je potrebné čo najskôr odstrániť
 - * *Valid fix later* – chybu je potrebné odstrániť ale nie okamžite
 - * *Valid Duplicate* – daná chyba už bola zaznamená (napr. v prípade, že dvaja inšpektori nájdu rovnakú chybu)
 - * *Valid won't fix* – chyba sa nebude opravovať, pretože nie je závažná a neznížuje funkcionality
 - * *Invalid won't fix* – nie je to chyba
- Krátky opis čo treba urobiť na odstránenie danej chyby (pole *Annotation*)
- Zapisovateľ spíše základné body diskusie a výsledok uznesenia o danej chybe

Ukončenie spoločnej revízie

- Projektový manažér uloží zoznam pripomienok klávesovou kombináciou *Ctrl+s*
- Projektový manažér podľa Metodiky o správe verzií do spoločného repozitára nahrá súbor *.review*, ktorý obsahuje všetky pripomienky, ktoré boli spísané na stretnutí
- Zapisovateľ dopíše zápisnicu
- Projektový manažér ukončí spoločnú revíziu

Kapitola 7

Zápisnice zo stretnutí

Nasledujú zápisnice zo stretnutí, v chronologickom poradí, vo formáte, ako boli zverejnené na webstránke.

Zápis zo stretnutia č. 1	
Dátum:	28.9.2010
Čas:	09:00 – 11:00
Miestnosť:	softvérové štúdio (d07b)
Vedúci:	Ing. Peter Lacko, PhD.
Zúčastnení členovia tímu:	Bc. Andrej Lukeš, Bc. Timotej Betina, Bc. Martin Práznovský, Mgr. Richard Strapko, Bc. Miroslav Šimulčík, Bc. Marek Tuška, Bc. Tomáš Uherčík
Iné zúčastnené osoby:	
Vypracoval:	Mgr. Richard Strapko

Téma stretnutia: Rozdelenie úloh, plán projektu (podľa harmonogramu)

Opis stretnutia:

1. Spoločne sme zvažovali rozdelenie konkrétnych úloh v rámci tímu. Výsledné rozhodnutie sme neprijali, nakoľko sme dostali informáciu, že role budú predmetom nasledujúceho spoločného stretnutia všetkých tímov.
2. Peter podotkol potrebu vytvorenia našej webstránky a plagátu. Túto úlohu si zobral Andrej.
3. Na návrh Petra sme spoločne spravili hrubú analýzu doterajšieho riešenia, ktoré vytvoril minuloročný tím. Peter navrhol a ostatní súhlasili, že dizajn web stránky ponecháme pôvodný.
4. Peter podotkol, že je nutné zistiť spôsob ukladania metadát o publikáciách. Po spoločnej diskusii Miro dostal úlohu zorganizovať stretnutie s p. Andrejčíkovou za týmto účelom.
5. Ďalej Peter stručne predstavil doterajší stav portálu:
 - a. Systém slúži na publikovanie referátov a publikácii, pričom formulár odovzdania súboru len uložil súbor.
 - b. Všetky nasledujúce kroky sú robené ručne, napr. kontrola, zverejnenie.

- c. Za kontrola, resp. review príspevku je považované obhájenie dizertačnej práce.
 - d. Po obhájení sa príspevok dostane do stavu „forthcoming issue“, kde zotrvá do vydania nového čísla.
6. Peter prezentoval niektoré požiadavky na nový portál:
- a. Pre formulár použiť AJAX technológiu.
 - b. Po odovzdaní súboru systém odošle mail editorovi, súbor zostáva čakať na schválenie.
 - c. Zvažovali sme možnosť využitia niekoľkých editorov, ktorým by sa práce prideliť cez vytvorený interface a čakali na ich posúdenie.
 - d. V prípade, že článok určitú dobu (neprimerane dlho) čaká na obhájenie, odoslať mail s požiadavkou o určenie stavu, či už publikácia má byť zverejnená.
 - e. Preformátovanie článkov sa automatizovať nebude kvôli problémom s formátovaním.
 - f. Webové publikovanie ponecháme vo formáte pdf, ale doplníme pri publikovaných prácach rozbaľovacie pole s doplnkovými informáciami o diele, napr. abstrakt.
 - g. Zvažovali sme možnosť zverejňovania kompletných dizertačných prác.
 - h. Administrátor musí mať heslo na prístup. Pre autora je vhodnejší prístup bez hesla pomocou generovaného odkazu.

Zoznam úloh pre ďalšie obdobie:

Číslo úlohy	Popis úlohy	Zodpovedá	Dátum zadania	Termín splnenia
1.1	Zorganizovať stretnutie s p. Andrejčíkovou	Miroslav Šimulčík	28.9.2010	5.10.2010
1.2	Vytvoriť základ pre webstránku nášho projektu	Andrej Lukeš	28.9.2010	19.10.2010
1.3	Vytvoriť plagát tímu	Andrej Lukeš	28.9.2010	5.10.2010
1.4	Spísať zápisnicu z tohto stretnutia	Richard Strapko	28.9.2010	5.10.2010
1.5.	Pridať Petra do mailing listu	Miroslav Šimulčík	28.9.2010	5.10.2010

Zápis zo stretnutia č. 2	
Dátum:	05.10.2010
Čas:	09:00 – 11:30
Miestnosť:	softvérové štúdio (d07b)
Vedúci:	Ing. Peter Lacko, PhD.
Zúčastnení členovia tímu:	Bc. Andrej Lukeš, Bc. Timotej Betina, Bc. Martin Práznovský, Mgr. Richard Strapko, Bc. Miroslav Šimulčík, Bc. Marek Tuška, Bc. Tomáš Uherčík
Iné zúčastnené osoby:	
Vypracoval:	Bc. Tomáš Uherčík

Téma stretnutia: Rozdelenie úloh, rozdelenie rolí, analýza dokumentácie minuloročného projektu.

Opis stretnutia:

1. Plánovanie stretnutia s Ing. Andrejčikovou, ktoré by mohlo byť napríklad budúci týždeň počas nášho stretnutia, lebo ona má stretnutie s jej tímom so začiatkom o 10:00, čiže ak by bola ochotná prísť o niečo skôr, mohli by sme s ňou prebrať formát MARC 21, prípadne sa opýtať ako postupovať vo veciach, súvisiacich s by bolo vhodné použiť.
2. Plánovanie stretnutia s prodekanou prof. Ing. Máriou Bielikovou, PhD. a prodekanom prof. Ing. Pavlom Návratom, PhD., pokúsime sa o stretnutie po prednáške z TP. Pravdepodobne by malo byť postačujúce stretnúť sa iba s prof. Ing. Máriou Bielikovou, PhD.
3. Diskusia novej črty v procese – po neúspešnej obhajobe môže školiteľ odstrániť neúspešne obhájený abstrakt.
4. Diskusia o rozdeľovaní rolí. Miro – vedúci, Timo – manažér kvality, Martin – manažér PR, komunikácia v tíme, Marek – zástupca vedúceho, Andrej –

prezentácia tímu, manažér plánovania, Tomáš – manažér podporných prostriedkov.

5. Identifikácia, analýza a plánovanie rizík
6. Identifikácia hlavných procesov, ktoré majú prebiehať v systéme
 - a. Submitnutie
 - b. Formálna kontrola
 - c. Prideľovanie ľudí na hodnotenie článkov – defaultne vedúci ale do budúcnosti sa počíta aj s inými
 - d. Zaradovanie do zásobníka (poolu)
 - e. Zverejňovanie článkov
 - f. Posudzovanie
 - g. Potvrdenie obhajoby

Zoznam úloh pre ďalšie obdobie:

Číslo úlohy	Popis úlohy	Zodpovedá	Dátum zadania	Termín splnenia
2.1	Preštudovanie dokumentácie minuloročného projektu, naštudovanie procesov, biznis logika	Všetci	5.10.2010	12.10.2010
1.2	Databáza a MARC 21	Miro	5.10.2010	12.10.2010
1.3	Diagram prípadov použitia - administrátor	Andrej, Martin	5.10.2010	12.10.2010
1.4	Diagram prípadov použitia - používateľ	Marek	5.10.2010	12.10.2010
1.5.	Diagram prípadov použitia - recenzent	Timo	5.10.2010	12.10.2010

Zápis zo stretnutia č. 3	
Dátum:	12.10.2010
Čas:	09:00 – 12:00
Miestnosť:	softvérové štúdio (d07b)
Vedúci:	Ing. Peter Lacko, PhD.
Zúčastnení členovia tímu:	Bc. Andrej Lukeš, Bc. Timotej Betina, Bc. Martin Práznovský, Bc. Miroslav Šimulčík, Bc. Marek Tuška, Bc. Tomáš Uherčík
Iné zúčastnené osoby:	
Vypracoval:	Bc. Marek Tuška

Téma stretnutia: analýza, návrh

Opis stretnutia:

1. Zhodnotenie diagramov (use case, activity, state) vytvorených jednotlivými členmi tímu. Návrhy na úpravy + zapracovanie nových skutočností, ktoré sme zistili pri konzultácii s Ing. Petrom Lackom PhD.
2. Spresnenie stretnutia s prof. Ing. Máriou Bielikovou, PhD. a prodekanom prof. Ing. Pavlom Návratom, PhD. ktoré bolo naplánované na 13.10.2010 o 08:00. Dohodli sme sa že toto stretnutie absolvujú Miro, Timo a Tomáš
3. Rozprava o CMS systémoch, aké sú výhody, či sa nám oplatí využiť tieto výhody alebo by bolo ich použitie pre nás nevýhodou. Uzavretie rozpravy s tvrdením že vlastnoručné programovanie nám bude trvať menej ako sa naučiť pracovať s CMS systémom + doprogramovávať určité funkcionality
4. Spresnenie si spôsobu akým funguje tento portál dnes
 - a. Články sú zoradené podľa poradia v akom nám bolo oznámené ich obhájenie

- b. Časopis má okolo 70 strán, takže sa vyberá toľko článkov aby sa táto norma približne dodržala
 - c. Za rok sa vydávajú dva časopisy bez ohľadu či je dostatok článkov, ak je málo vydá sa časopis aj s menej stranami
 - d. V skutočnosti treba rozlišovať dva druhy článkov, klasický článok a rozšírený abstrakt ale momentálne sa pracuje iba s rozšíreným abstraktom
 - e. V prípade rozšíreného abstraktu dáva školiteľ len dátum a obhájené/neobhájené + pri neobhájenom rozhoduje či sa abstrakt stiahne alebo ponechá (ale nikdy nie vymazať)
 - f. Pre klasický časopis treba umožniť recenzentovi aj slovné posúdenie
 - g. Po potvrdení obhájenia sa články presúvajú do forthcomingu
5. Zaoberanie sa témou inovácie a zlepšenia, preberanie návrhov ktoré by sme mohli po schválení na konzultáciách použiť.
- a. Zadelenie recenzentov do kategórií, skúsiť to spraviť podobne ako sú kategórie podkategórie na profesii alebo aspoň oblasti v ktorých vedia podať recenzenti kvalifikovaný názor
 - b. Doriešiť priebeh potvrdenia obhajoby, vygenerovaný link na potvrdenie poslať hneď po pridelení a automaticky alebo ručne po nastavenej dobe ak neodpovedá
 - c. Po kliknutí na vygenerovaný link by mal byť dostupný článok na prečítanie ako aj iné informácie o danom článku (bavíme sa o rozšírenom abstrakte) prečítať dostupný článok ako aj potvrdiť obhájenie
 - d. Pri prezeraní článkov umožniť rozbaľiť ponuku s podrobnejšími informáciami o článku ale umožniť aj pomocou ikonky prekliknutie na tie isté informácie aby bolo možné poslať link na konkrétny článok
 - e. Treba rozdeliť časopis na dve rôzne entity na „rozšírený abstrakt“ a „časopis“ a v procese realizácie rozlíšiť tieto dva druhy článkov ikonkou
 - f. Treba umožniť študentovi zadať pri vkladaní článku email svojho školiteľa, ktorému následne príde notifikácia o vkladaní takéhoto článku študentom s možnosťou potvrdenia obhájenia jeho práce (platí len pre rozšírený abstrakt)
 - g. Ak bude vydaný nový časopis treba prideliť články ku konkrétnemu číslu časopisu
 - h. Treba aby bol každý odoslaný článok poslaný adminovi na základné posúdenie (formálne, obsahové)
 - i. Umožniť existenciu viacerých recenzentov pre klasický článok ale len jedného pre rozšírený abstrakt

Zoznam úloh pre ďalšie obdobie:

Číslo úlohy	Popis úlohy	Zodpovedá	Dátum zadania	Termín splnenia
3.1	Revízia prípadov použitia	Všetci	12.10.2010	19.10.2010
3.2	Stretnutia s prof. Ing. Máriou Bielikovou, PhD. A prof. Ing. Pavlom Návratom	Miro, Timo, Tomáš	12.10.2010	19.10.2010
3.3	Návrh technológií	Miro, Timo, Tomáš	12.10.2010	19.10.2010
3.4	Analýza CMS systémov	Marek	12.10.2010	19.10.2010

Lakeri – Tím 6

Zápis zo stretnutia so zákazníkom

Dátum: 20.10.2010

Miesto: Softvérové štúdio (D07b, FIIT - STU)

Prítomní: zákazníci – prof. Ing. Mária Bieliková, PhD., prof. Ing. Pavol Návrat, PhD.; pedagóg - Ing. Peter Lacko, PhD.; členovia tímu - Bc. Miroslav Šimulčík, Bc. Timotej Betina, Bc. Tomáš Uherčík

Zapisovateľ: Timotej Betina

Téma stretnutia: Zber požiadaviek na systém.

Opis stretnutia:

Prediskutovali sme nasledovné body:

- Na čo sa má systém zamerať:
 - portál dizertácií
 - portál časopisu (rozšírene abstrakty, časopis)
 - vytváranie časopisu
 - vkladanie dizertácii a rozšírených abstraktov
 - vyhľadávanie v abstraktoch/dizertáciách/časopisoch

- Na čo sa systém nemá zamerať:
 - vkladanie a recenzovanie samostatných článkov

- Dôležité body (pre nás ako vývojárov), na ktoré sa zamerať pri tvorbe systému:
 - orientácia na používateľa, ktorého cieľom je čítanie, vyhľadávanie v zdrojoch. Čiže na funkcie digitálnej knižnice, aby mala prehľadnú, jednoduchú, intuitívnu navigáciu
 - rôzne formy vyhľadávania: fulltextové, podľa kľúčových slov, podľa kategorizácie prac, ...
 - Jasná orientácia pre používateľa resp. doktoranda, že kde sa v systéme nachádza (napr. pri prechode doktoranda od portálu ku vkladaniu abstraktu/dizertácie, aby mu bolo jasné, že čo vkladá, čo ešte môže vložiť a podobne)

- Ďalšie poznámky zo stretnutia:

- po vložení dizertácie/abstraktu poslať školiteľovi mail o tom, že má potvrdiť výsledok obhajoby. V prípade, že to dlhšie nepotvrdil, poslať automaticky alebo manuálne ďalší notifikačný mail
- umožniť editorovi, aby ručne mohol potvrdiť výsledky obhajoby pre práce v prípade, že sa školiteľ neozýva
- neobhájené abstrakty/dizertačné práce vylúčiť zo systému
- umožniť bežným používateľom poslať odporúčanie na nejakú prácu

ΘIST.dl

Theses in Information Sciences and Technologies – Digital Library

Grécky: θέση, anglicky thesis

Zápis zo stretnutia č. 4

Dátum:	26.10.2010
Čas:	09:00 – 12:00
Miestnosť:	softvérové štúdio (d07b)
Vedúci:	Ing. Peter Lacko, PhD.
Zúčastnení členovia tímu Lakeri:	Bc. Andrej Lukeš, Bc. Timotej Betina, Bc. Miroslav Šimulčík, Bc. Marek Tuška, Bc. Tomáš Uherčík, Bc. Martin Práznovský
Iné zúčastnené osoby:	
Vypracoval:	Bc. Martin Práznovský

Téma stretnutia: Ďalšia revízia prípadov použitia, rozdelenie úloh pred prvým odovzdávaním dokumentácie

Opis stretnutia:

1. Úvodná všeobecná diskusia o:
 - Možnosti použitia frameworkov pre php
 - O tom, čo bolo na prednáške z TP
 - O stretnutí s p. Andrejčíkovou, čo s ňou potrebujeme diskutovať (štandardy, frameworky, vyhľadávanie, metadáta)
 - O používaní Google Code
2. Zhrnutie dôvodov, prečo nepoužijeme niektoré CMS, ale vytvoríme vlastný systém:
 - Naša oblasť je dosť špecifická a navyše mnoho súčastí CMS systémov by tam bolo zbytočne navyše
 - Prácu by nám to neurýchlilo, nakoľko nikto z nás so žiadnym z týchto systémov nemá skúsenosti; čas potrebný na preštudovanie CMS systému by sa rovnal času vývoju nášho nového systému
 - CMS systémy vnímame skôr ako vhodné pre netechnicky orientovaných ľudí
3. Z dôvodu fulltextového vyhľadávania navrhol Peter preštudovať si textové databázy. Miro sa podujal na túto úlohu a porovná, ktorá (MySQL, PostgreSQL) je na to vhodnejšia, rýchlejšia a či by na toto vyhľadávanie nebolo vhodné použiť nejakú ďalšiu aplikáciu

4. Peter podotkol, že by bolo vhodné navrhnuť obrazovky, najmä pri vyhľadávaní. Je totiž pravdepodobné, že naši zákazníci (Bieliková, Návrat) sa práve o toto budú čoskoro zaujímať. Úlohu si zobral Timo.
5. Wiki na Google Code doteraz nebola použitá, bolo by dobré tu publikovať napr. metodiky
6. Bolo dohodnuté stretnutie s p. Andrejčíkovou na štvrtok 28.10. na 11:00 v softvérovom štúdiu
7. Po stretnutí s p. Bielikovou a Návratom nám boli upresnené požiadavky, čiže bolo potrebné urobiť revíziu prípadov použitia. Spoločne sme spísali role (používateľov) a ich úlohy v systéme (Timo spisoval na papier). Dohodli sme sa, že extended abstract sa bude nazývať ako článok a dizertácia ako dizertácia. Revízie diagramov sme si rozdelili nasledovne:
 - Miro – dátový model
 - Tomáš – stavový model
 - Martin, Andrej – admin
 - Marek – autor, školiteľ
 - Timo – používateľ
8. O týždeň sa odovzdáva prvá verzia dokumentácie, takže treba urobiť nasledovné:
 - Časť dokumentácie o manažovaní a riadení tímu
 - predstavenie tímu
 - role členov v tíme
 - zápisnice
 - plán (aspoň hrubý)
 - ponuka, rozvrh
 - metodiky
 - nástroje (Google Code)
 - Niektoré z týchto vecí sa nachádzajú na webstránke tímu
 - Dopísanie analýzy CMS systémov – Andrej, Martin, Marek
 - Dopísanie analýzy technológií a programovacích jazykov – Miro, Tomáš, Timo
 - Špecifikácia požiadaviek (funkcionálne, nefunkcionálne), asi na jednu stranu
 - Zlepenie jednotlivých častí do jedného dokumentu, dopísanie úvodu – Martin

Zoznam úloh pre ďalšie obdobie:

Číslo úlohy	Popis úlohy	Zodpovedá	Dátum zadania	Termín splnenia
4.1	Štúdium textových databáz	Miro	26.10.2010	2.11.2010
4.2	Návrh obrazoviek	Timo	26.10.2010	2.11.2010
4.3	Vykonať revízie diagramov	Všetci	26.10.2010	2.11.2010
4.4	Práca na dokumentácii k riadeniu	Všetci	26.10.2010	2.11.2010

4.5	Analýza CMS systémov	Martin, Andrej, Marek	26.10.2010	2.11.2010
4.6	Analýza technológií a programovacích jazykov	Timo, Miro, Tomáš	26.10.2010	2.11.2010
4.7	Vytváranie dokumentácie	Martin	26.10.2010	2.11.2010

Lakeri – Tím 6

Zápis zo stretnutia so zákazníkom II

Dátum: 28.10.2010

Miesto: Softvérové štúdio (D07b, FIIT - STU)

Prítomní:

zákazník – Ing. Nadežda Andrejčíková, PhD.;

pedagóg - Ing. Peter Lacko, PhD.;

členovia tímu - Bc. Miroslav Šimulčík, Bc. Tomáš Uherčík, Bc. Andrej Lukeš, Bc. Martin

Práznovský, Bc. Marek Tuška

Zapisovateľ: Marek Tuška

Téma stretnutia: Návrh systému, používané štandardy

Opis stretnutia:

Prediskutovali sme nasledovné body:

- Čo má poskytovať náš systém, požiadavky od iných zákazníkov
- Aké metadáta sa majú zbierať o článkoch, časopisoch, aké by sme mali zbierať my
- Aké systémy môžu byť prepojené s naším, aké systémy môžeme využiť
- Unimarc, Marc21 – prehľad formátovania dokumentov

Dôležité body zo stretnutia:

- spraviť fulltext + metadáta vyhľadávanie nad článkami, časopismi
- fulltexty uložiť mimo databázy, OpenURL
- metadáta o časopise (presne špecifikované aké údaje máme zbierať nám bude upresnené)
 - Názov (názov, podnázov, súbežný názov)
 - Autor (Meno, Priezvisko, rodné priezvisko, pracovisko,...)
 - Ročník čísla
 - Pracovisko
 - Kľúčové slová (Tezaurus – hierarchická štruktúra)
- Umožniť vybrať školiteľa (pracovisko/inštitúciu) zo zoznamu autorít, ak sa tam nenachádza vytvoriť novú autoritu, proces riešenia duplicit autorít a ich prepojení nechať na ďalších ľuďoch (knihovníčka)

Zápis zo stretnutia č. 5	
Dátum:	26.10.2010
Čas:	08:00 – 9:00
Miestnosť:	softvérové štúdio (d07b)
Vedúci:	Ing. Peter Lacko, PhD.
Zúčastnení členovia tímu:	Bc. Andrej Lukeš, Bc. Timotej Betina, Bc. Martin Práznovský, Bc. Miroslav Šimulčík, Bc. Marek Tuška, Bc. Tomáš Uherčík
Iné zúčastnené osoby:	
Vypracoval:	Bc. Martin Práznovský

Téma stretnutia: Ďalšia revízia prípadov použitia, rozdelenie úloh pred prvým odovzdávaním dokumentácie

Opis stretnutia:

1. Úvodná všeobecná diskusia o:
 - a. Možnosti použitia frameworkov pre php
 - b. O tom, čo bolo na prednáške z TP
 - c. O stretnutí s p. Andrejčikovou, čo s ňou potrebujeme diskutovať (štandardy, frameworky, vyhľadávanie, metadáta)
 - d. O používaní Google Code
2. Zhrnutie dôvodov, prečo nepoužijeme niektoré CMS, ale vytvoríme vlastný systém:
 - a. Naša oblasť je dosť špecifická a navyše mnoho súčastí CMS systémov by tam bolo zbytočne navyše
 - b. Prácu by nám to neurýchlilo, nakoľko nikto z nás so žiadnym z týchto systémov nemá skúsenosti; čas potrebný na preštudovanie CMS systému by sa rovnala času vývoju nášho nového systému
 - c. CMS systémy vnímame skôr ako vhodné pre netechnicky orientovaných ľudí
3. Z dôvodu fulltextového vyhľadávania navrhol Peter preštudovať si textové databázy. Miro sa podujal na túto úlohu a porovná, ktorá (MySQL, PostgreSQL)

- je na to vhodnejšia, rýchlejšia a či by na toto vyhľadávanie nebolo vhodné použiť nejakú ďalšiu aplikáciu
4. Peter podotkol, že by bolo vhodné navrhnúť obrazovky, najmä pri vyhľadávaní. Je totiž pravdepodobné, že naši zákazníci (Bieliková, Návrat) sa práve o toto budú čoskoro zaujímať. Úlohu si zobral Timo.
 5. Wiki na Google Code doteraz nebola použitá, bolo by dobré tu publikovať napr. metodiky
 6. Bolo dohodnuté stretnutie s p. Andrejčíkovou na štvrtok 28.10. na 11:00 v softvérovom štúdiu

Zoznam úloh pre ďalšie obdobie:

Číslo úlohy	Popis úlohy	Zodpovedá	Dátum zadania	Termín splnenia
5.1	Finálna Revízia prípadov použitia	Andrej, Marek, Martin, Miro, Tomáš	26.10.2010	19.10.2010
5.2	Stretnutie so zákazníkmi: Ing. Andrejčíková	Všetci	26.10.2010	28.10.2010
5.3	Návrh obrazoviek	Timo	26.10.2010	30.10.2010
5.4	Vytvorenie dokumentácie – časť riadenie	Andrej, Marek	26.10.2010	30.10.2010
5.5	Vytvorenie finálnej dokumentácie	Martin	26.10.2010	2.11.2010

Lakeri – Tím 6

Zápis zo stretnutia ku kontrole kvality softvéru

Dátum: 09.11.2010

Miesto: Softvérové štúdio (D07b, FIIT - STU)

Prítomní: pedagóg: Ing. Ján Suchal, členovia tímu - Bc. Miroslav Šimulčík, Bc. Timotej Betina, Bc. Tomáš Uherčík, Bc. Andrej Lukeš, Bc. Martin Práznovský, Bc. Marek Tuška

Zapisovateľ: Andrej Lukeš

Téma stretnutia: Kontrola kvality vytváraného softvéru v prvých 7 týždňoch zimného semestra

Opis stretnutia:

1. Pripomienky k dokumentácií vytvorenej počas 7 týždňov semestra
 - a. Diskusia k vytváraniu vlastného CMS systému.
 - b. Odporúčenie pozrieť sa na framework NETTE.
 - c. Diskusia o programovacích jazykoch a o našich analýzach k týmto jazykom v dokumentácií. Konkrétne sme diskutovali o jazykoch PHP, Java, Ruby on Rails.
 - d. Diskusia na tému fulltextového vyhľadávania – dostali sme odporúčenie zamerať sa len na PostgreSQL databázu a nepoužívať Sfinx.
 - e. Odporúčenie spraviť model databázy v angličtine, ak by niekto v budúcnosti chcel na ňom ďalej pracovať.
 - f. Diskusia k vytváraniu časopisu, k tvorbe hlavičky a úvodnej strany.
 - g. Diskusia k návrhu rozhrania – vo formulároch zoskupiť textové polia, ktoré navzájom súvisia, dať si pozor na kontrolu vyplnenia údajov vo formulári a upload súboru, spraviť to oddelene.

Zápis zo stretnutia č. 6	
Dátum:	28.10.2010
Čas:	11:00 – 12:00
Miestnosť:	softvérové štúdio (d07b)
Vedúci:	Ing. Peter Lacko, PhD.
Zúčastnení členovia tímu:	Bc. Andrej Lukeš, Bc. Martin Práznovský, Bc. Miroslav Šimulčík, Bc. Marek Tuška, Bc. Tomáš Uherčík
Iné zúčastnené osoby:	Ing. Nadežda Andrejčíková, PhD.
Vypracoval:	Bc. Martin Práznovský

Téma stretnutia: Návrh systému, používané štandardy

Opis stretnutia:

1. Prediskutovali sme nasledovné body:
 - a. Čo má poskytovať náš systém, požiadavky od iných zákazníkov
 - b. Aké metadáta sa majú zbierať o článkoch, časopisoch, aké by sme mali zbierať my
 - c. Aké systémy môžu byť prepojené s naším, aké systémy môžeme využiť
 - d. Unimarc, Marc21 – prehľad formátovania dokumentov

2. Dôležité body zo stretnutia:
 - a. spraviť fulltext + metadáta vyhľadávanie nad článkami, časopismi
 - b. fulltexty uložiť mimo databázy, OpenURL
 - c. metadáta o časopise (presne špecifikované aké údaje máme zbierať nám bude upresnené)
 - Názov (názov, podnázov, súbežný názov)

- Autor (Meno, Priezvisko, rodné priezvisko, pracovisko,...)
 - Ročník čísla
 - Pracovisko
 - Kľúčové slová (Tezaurus – hierarchická štruktúra)
- d. Umožniť vybrať školiteľa (pracovisko/inštitúciu) zo zoznamu autorít, ak sa tam nenachádza vytvoriť novú autoritu, proces riešenia duplicit autorít a ich prepojení nechať na ďalších ľuďoch (knihovníčka)

Zápis zo stretnutia č. 7	
Dátum:	23.11.2010
Čas:	09:00 – 11:00
Miestnosť:	softvérové štúdio (d07b)
Vedúci:	Ing. Peter Lacko, PhD.
Zúčastnení členovia tímu Lakeri:	Bc. Andrej Lukeš, Bc. Timotej Betina, Bc. Martin Práznovský, Bc. Miroslav Šimulčík, Bc. Marek Tuška, Bc. Tomáš Uherčík
Iné zúčastnené osoby:	
Vypracoval:	Bc. Miroslav Šimulčík

Téma stretnutia: Zhodnotenie stavu prototypu a pridelenie ďalších úloh k implementácii.

Opis stretnutia:

1. Diskusia o problémoch a postrehoch jednotlivých členov tímu, s ktorými sa stretli počas implementácie za posledný týždeň.
2. Definovanie potrebných zmien vo funkcionalite prototypu, ktoré treba vykonať.
3. Aktualizácia plánu na najbližšie obdobie, určenie konkrétnych činností ktoré je potrebné vykonať a ich pridelenie jednotlivým členom tímu.

Zhodnotenie predchádzajúcich úloh:

Číslo úlohy	Popis úlohy	Zodpovedá	Dátum zadania	Termín splnenia	Stav
6.1	Vytvorenie databázy podľa dátového modelu	Miro	16.11.2010	23.11.2010	Splnená
6.2	Implementácia fulltext vyhľadávania	Miro	16.11.2010	30.11.2010	Splnená
6.2	Vytvorenie základnej štruktúry prototypu - inicializácia	Timo, Tomáš	16.11.2010	23.11.2010	Splnená
6.3	Upravenie dizajnu a vytvorenie views podľa návrhu obrazoviek.	Andrej, Marek, Martin	16.11.2010	23.11.2010	Prebieha
6.4	Sprístupnenie databázy a dizajnu aktuálneho ACM portálu	Peter	16.11.2010	23.11.2010	Splnená

Zoznam úloh pre ďalšie obdobie:

Číslo úlohy	Popis úlohy	Zodpovedá	Dátum zadania	Termín splnenia
7.1	Implementácia vyhľadávania v článkoch	Miro, Timo, Tomáš	23.11.2010	30.11.2010
7.2	Implementácia zobrazenia zoznamu, detailov a vkladania	Marek, Andrej, Martin	23.11.2010	30.11.2010

	článkov			
--	---------	--	--	--

Zápis zo stretnutia č. 8	
Dátum:	30.11.2010
Čas:	09:00 – 11:00
Miestnosť:	softvérové štúdio (d07b)
Vedúci:	Ing. Peter Lacko, PhD.
Zúčastnení členovia tímu Lakeri:	Bc. Andrej Lukeš, Bc. Timotej Betina, Bc. Miroslav Šimulčík, Bc. Marek Tuška, Bc. Tomáš Uherčík
Iné zúčastnené osoby:	
Vypracoval:	Bc. Andrej Lukeš

Téma stretnutia: Zhodnotenie stavu prototypu, pridelenie ďalších úloh k implementácii, písaní dokumentácie.

Opis stretnutia:

1. Diskusia k prototypu, riešenie problémov, ktoré sa vyskytli pri implementácií
2. Riešenie kategorizácie pri submittnutí článku
3. Diskusia k dátumom – dátum vloženia, dátum obhájenia a dátum, kedy vyšlo číslo, v ktorom bol článok publikovaný
4. Diskusia k search result, articles - autor a názov článku, univerzita, a stav zverejnenia (či bol článok zverejnený alebo nie)

5. Articles – treba doplniť informácie, ktoré sú v pôvodnej verzii (v tabuľke – dátumy, konkrétne dátum submitnutia, dátum obhájenia a dátum publikovania) + navyše treba doplniť inštitúciu
6. Search – autor, title, inštitúcia - doplniť

Zhodnotenie predchádzajúcich úloh:

Číslo úlohy	Popis úlohy	Zodpovedá	Dátum zadania	Termín splnenia	Stav
7.1	Implementácia vyhľadávania v článkoch	Miro, Timo, Tomáš	23.11.2010	30.11.2010	Uzavreté
7.2	Implementácia zobrazenia zoznamu, detailov a vkladania článkov	Marek, Andrej, Martin	23.11.2010	30.11.2010	Prebieha

Zoznam úloh pre ďalšie obdobie:

Číslo úlohy	Popis úlohy	Zodpovedá	Dátum zadania	Termín splnenia
8.1	V submission form zmeniť polia titul before a titul after na	Marek	30.11.2010	7.12.2010
8.2	textfield V submission form vytvoriť textfield pre titul supervisoru	Marek	30.11.2010	7.12.2010
8.3	Zmeniť supervisoru na usera	Marek	30.11.2010	7.12.2010
8.4	Pridať do databázy adresu	Miro	30.11.2010	7.12.2010

	k inštitúcií a vyriešiť ukladanie používateľov			
8.5	Opraviť chybu v používateľskom rozhraní	Andrej	30.11.2010	7.12.2010
8.6	Písanie dokumentácie k riadeniu a dokumentácie	Andrej, Martin, Tomáš	30.11.2010	7.12.2010
8.7	Pridať do databázy adresu k inštitúcií	Miro	30.11.2010	7.12.2010
8.8	Dokončenie vyhľadávania(rozšírené aj jednoduché)	Miro	30.11.2010	7.12.2010
8.9	Ukladanie fulltextu do databázy zo zdrojových kódov latexu	Miro	30.11.2010	7.12.2010
8.10	Kategorizácia článkov	Tomáš	30.11.2010	7.12.2010
8.11	Výsledky vyhľadávania, zoznam článkov	Timo	30.11.2010	7.12.2010

Zápis zo stretnutia č. 9	
Dátum:	7.12.2010
Čas:	09:00 – 11:30
Miestnosť:	softvérové štúdio (d07b)
Vedúci:	Ing. Peter Lacko, PhD.
Zúčastnení členovia tímu Lakeri:	Bc. Andrej Lukeš, Bc. Timotej Betina, Bc. Miroslav Šimulčík, Bc. Marek Tuška, Bc. Tomáš Uherčík, Bc. Martin Práznovský
Iné zúčastnené osoby:	
Vypracoval:	Bc. Martin Práznovský

Téma stretnutia: Zhodnotenie stavu prototypu, prípravy pred odovzdaním

Opis stretnutia:

1. Posledné stretnutie – diskusia k prezentácii a odovzdávaniu
2. Diskusia k prototypu, riešenie problémov, ktoré sa vyskytli pri implementácii
3. Doplnené autorstvo častí dokumentácie
4. Treba dopísať a dokončiť dokumentáciu, najmä dokumentácia prototypu a do riadenia metodiku, ktorú Miro nahrá na wiki
5. Diskusia k niekoľkým potrebným zmenám v databáze
6. Diskusia k detexu – zrejme vytvoríme neskôr vlastný skript s podobnou funkcionalitou

7. Treba dohodnúť dátum a čas prezentovania s ďalším tímom (nám vyhovuje štvrtok o 16.00)

Zhodnotenie predchádzajúcich úloh:

Číslo úlohy	Popis úlohy	Zodpovedá	Dátum zadania	Termín splnenia	Stav
8.1	V submission form zmeniť polia titul before a titul after na textfield	Marek	30.11.2010	7.12.2010	Uzavreté
8.2	V submission form vytvoriť textfield pre titul supervisora	Marek	30.11.2010	7.12.2010	Uzavreté
8.3	Zmeniť supervisora na usera	Marek	30.11.2010	7.12.2010	Zrušené
8.4	Pridať do databázy adresu k inštitúcií a vyriešiť ukladanie používateľov	Miro	30.11.2010	7.12.2010	Uzavreté
8.5	Opraviť chybu v používateľskom rozhraní	Andrej	30.11.2010	7.12.2010	Uzavreté
8.6	Písanie dokumentácie k riadeniu a dokumentácie	Andrej, Martin, Tomáš	30.11.2010	7.12.2010	Prebieha
8.7	Pridať do databázy adresu k inštitúcií	Miro	30.11.2010	7.12.2010	Uzavreté
8.8	Dokončenie vyhľadávania(rozšírené aj jednoduché)	Miro	30.11.2010	7.12.2010	Prebieha

8.9	Ukladanie fulltextu do databázy zo zdrojových kódov latexu	Miro	30.11.2010	7.12.2010	Prebieha
8.10	Kategorizácia článkov	Tomáš	30.11.2010	7.12.2010	Prebieha
8.11	Výsledky vyhľadávania, zoznam článkov	Timo	30.11.2010	7.12.2010	Uzavreté

Zoznam úloh pre ďalšie obdobie:

Číslo úlohy	Popis úlohy	Zodpovedá	Dátum zadania	Termín splnenia
9.1	Dohodnúť čas prezentácie	Martin	7.12.2010	14.12.2010
9.2	Nahratie metodiky na wiki	Miro	7.12.2010	14.12.2010
9.3	Úpravy databázy	Miro	7.12.2010	14.12.2010

Preberací protokol

Typ projektu: Tímový projekt
Názov projektu: Portál pre časopis ACM

Číslo tímu: 6

Členovia tímu: Bc. Andrej Lukeš
Bc. Timotej Betina
Bc. Martin Práznovský
Bc. Miroslav Šimulčík
Bc. Marek Tuška
Bc. Tomáš Uherčík

....., týmto potvrdzuje prevzatie dokumentácie k tímovému projektu v rozsahu strán od tímu číslo 6, zloženej z 2 častí, projektovej dokumentácie a dokumentu k riadeniu projektu.

V Bratislave dňa:

Podpis vedúceho tímu: