

Tím č. 16 (GrafIT)

***INTERAKTÍVNA VIZUALIZÁCIA GRAFOVÝCH ŠTRUKTÚR V
3D PRIESTORE***

Dokumentácia k riadeniu tímového projektu
Tímový Projekt

Vedúci tímového projektu: Ing. Peter Kapec

Členovia tímu: Bc. Tomáš Hurban, Bc. Milan Laslop, Bc. Jaroslav Prokop, Bc. Matúš Juhas

Akademický rok: 2010/2011

Obsah

Úvod	iii
1 Ponuky	1-1
2 Plán projektu	2-1
3 Úlohy členov tímu.....	3-1
4 Metodiky potrebné pri vývoji	4-1
4.1 Komunikácia v tíme.....	4-1
4.2 Úlohy	4-1
4.3 Komentáre pri vkladaní súborov do repozitára	4-2
4.4 Metodika manažmentu zmien.....	4-3
4.5 Manažment úloh v distribuovanom projekte	4-17
4.6 Metodika písania dokumentácie zdrojových kódov a komentárov	4-32
4.7 Dokumentovanie zdrojových kódov	4-37
5 Zápisy zo stretnutí.....	5-1
5.1 Zápis zo stretnutia č. 1	5-1
5.2 Zápis zo stretnutia č. 2	5-3
5.3 Zápis zo stretnutia č. 3	5-6
5.4 Zápis zo stretnutia č. 4	5-8
5.5 Zápis zo stretnutia č. 5	5-12
5.6 Zápis zo stretnutia č. 6.....	5-15
5.7 Zápis zo stretnutia č. 7	5-18
5.8 Zápis zo stretnutia č. 8.....	5-21
5.9 Zápis zo stretnutia č. 9	5-24
5.10 Zápis zo stretnutia č. 10	5-26
5.11 Zápis zo stretnutia č. 11	5-29
5.12 Zápis zo stretnutia č. 12	5-31
5.13 Zápis zo stretnutia č. 13	5-33
5.14 Zápis zo stretnutia č. 14	5-35
5.15 Zápis zo stretnutia č. 15	5-37
5.16 Zápis zo stretnutia č. 16	5-38
5.17 Zápis zo stretnutia č. 17	5-40
5.18 Zápis zo stretnutia č. 18	5-41

5.19	Zápis zo stretnutia č. 19	5-43
6	Preberacie protokoly	6-1

Úvod

V tejto práci je zdokumentovaný manažment tímu č. 16 s názvom GraFIT v akademickom roku 2010/2011. Práca obsahuje dokumenty súvisiace s riadením a plánovaním projektu. Tento dokument obsahuje nasledovné časti:

- **Ponuka** – obsahuje odovzdané ponuky. Neobsahuje ponuku na tému *INTERAKTÍVNA VIZUALIZÁCIA GRAFOVÝCH ŠTRUKTÚR V 3D PRIESTORE*.
- **Plán projektu** – obsahuje plán projektu na zimný semester po jednotlivých týždňoch
- **Úlohy členov tímu** – obsahuje rozdelenie hlavných rolí na projekte medzi jednotlivých členov
- **Metodiky potrebné pri vývoji** – obsahuje popis zavedených metodík pre komunikáciu, plánovanie a dokumentovanie
- **Záznamy zo stretnutí** – obsahuje zápisnice zo stretnutí k tímovému projektu
- **Preberacie protokoly** – obsahuje preberacie protokoly k odovzdanej dokumentácii

1 Ponuky

Projektový tím

Náš tím obsahuje študentov prvého ročníka inžinierskeho štúdia na FIIT STU, absolventov bakalárskeho štúdia FIIT v odbore Informatika. Každý z členov má osobité skúsenosti z rôznych oblastí informačných technológií, prípadne manažmentu. To nám poskytuje veľkú výhodu pri riešení problémov či už v oblasti webových technológií, alebo aj v oblasti špecifických problémov v rámci umelej inteligencie. Kontakt na tím je jaro.prokop@gmail.com.

Bc. Laslop Milan

Dva roky pracuje v oblasti návrhu a implementácie softvérových systémov a tvorby webových stránok. Ovláda jazyky Java a C++ na pokročilej úrovni, základy jazykov C# a VisualBasic, má skúsenosti s používaním rôznych API, knižníc a technológií (J2SE, J2ME, Hibernate, Junit, Swing, SWT, STL, wxWidgets, .NET framework, Linq, Eclipse API). Tiež používal jazyky PHP, CSS, XHTML a JavaScript, rôzne webové technológie (GWT, Ajax, SOAP) a pracoval s databázami (najmä MySQL). Venoval sa aj tvorbe aplikácií pre mobilné telefóny (Java – J2ME), skúšal vytvoriť aplikáciu pre operačný systém Android (Java) a vytvoril niekoľko pluginov do programu Eclipse. Konfiguroval serverové aplikácie a služby (Apache, Tomcat, MySQL, SVN+SSH).

Bc. Susedík Jakub

Rok pracuje v oblasti webových technológií, kde pracoval s jazykmi HTML, CSS, Asp .net, C#, s redakčným systémom Dotnetnuke a využíval technológiu WCF na komunikáciu medzi serverom a klientom. Ďalej má skúsenosti s programovaním v programovacích jazykoch C/C++, Visual basic a s databázovými systémami MySql, PostgreSQL. Medzi jeho záujmy patrí oblasť umelej inteligencie, ktorej základy získal počas bakalárskeho štúdia na predmete Umelá inteligencia a nadviazal naň počas inžinierskeho štúdia predmetom Neurónové siete.

Bc. Prokop Jaroslav

Pol roka pôsobil ako procesný analytik v oblasti štátnej správy. Pracoval rok ako programátor na rozsiahlom projekte zaoberajúcim klient-server riešením pre colnú správu. Na projekte bol využitý .NET framework v prostredí VSTS 2008 (Visual Studio Team System),

jazyk C#. Boli využité technológie WCF, NHibernate, Oracle a vzor MVC pre GUI. Rok pracoval v manažmente na pozícii spolutvorcu a administrátora harmonogramu projektu v stavebno-technologickej oblasti (výstavba oceliarne). Má skúsenosti s jazykom Java, Visual Basic, C++, s technológiami a knižnicami J2SE, J2ME, s databázami MySQL, MSSQL, a rozsiahle skúsenosti s MS Project. Na inžinierskom štúdiu študuje predmety Pokročilé databázové systémy a Objektovo orientovaná analýza a návrh systému.

Bc. Hurban Tomáš

Počas štúdia nadobudol znalosti z programovania v jazykoch C/C++, Java. Štyri roky pracuje v oblasti webových technológií ako webmaster a webdizajnér. Počas tohto pôsobenia sa zdokonalil v jazykoch HTML, XHTML, CSS, PHP a získal základné znalosti jazyka JavaScript. Taktiež získal skúsenosti s prácou s rôznymi nástrojmi na tvorbu a správu webových stránok a základné znalosti databázového systému MySQL.

Bc. Fábik Pavol

Zaujíma sa o tvorbu aplikácií pre mobilné telefóny (pre OS Symbian). Programovacie znalosti získaval od strednej školy. V súčasnosti sa venuje programovaniu v C/C++. Takisto ovláda jazyky java, HTML, PHP, MYSQL, CSS, javascript, actionscript. Počas práce na bakalárskom projekte sa začal venovať skriptovaciemu jazyku flash. Práca s grafickými editormi Adobe Photoshop, Illustrator mu nerobí problém. Skúsenosti s vytváraním rozsiahlejších aplikácií a tímovou prácou zatiaľ nemá. Absolvoval stáž v zahraničí, kde sa venoval IT supportu. V budúcnosti by sa chcel viac venovať webovým aplikáciám.

Marek Švec

Počas štúdia nadobudol znalosti z programovania v jazykoch C/C++, Java. Tri roky pracoval ako analytik dát, teda nadobudol skúsenosti s prácou s rozsiahlymi dátami a získal skúsenosti v prístupe k viacerým druhom databáz. V súčasnosti pracuje ako analytik a vývojár aplikácií pre pobočky banky. Okrem návrhu tieto aplikácie programuje a to v jazyku Visual Basic a C++. Okrem týchto pracovných skúseností získal ďalšie skúsenosti s návrhom a implementáciou väčšieho IS pre študentskú organizáciu, pričom riadil vývojársky tím. Pri tomto projekte získal skúsenosti v jazyku PHP a MySQL. Na inžinierskom štúdiu študuje predmety Pokročilé databázové systémy a Architektúra informačných systémov.

Tréner mentálnych schopností

Motivácia

V dnešnom uponáhľanom svete plnom techniky je veľmi pozitívnym prínosom zaradenie do mnohých počítačových programov množstvo rôznych hier, či už logických hlavolamov, hádaniek a rôznych podobných úloh. Pomáhajú mnohým ľuďom, ale najmä deťom a mládeži odreagovať sa od bežných náročných povinností, a pritom si potrénovať myslenie a snažiť sa niekedy prekonať aj samých seba a dokázať stále viac. V súčasnej dobe sa mladí ľudia učia pracovať s počítačom v čoraz mladšom veku. Stále rozšírenejšie je používanie počítača už v predškolských zariadeniach, kde sa deti hrovou formou snažia rozvíjať svoje schopnosti. Už v tomto veku je dôležité snažiť sa ich zaujať, aby sami hľadali riešenie rôznych hádaniek a hlavolamov. Mnohí, prevažne starší ľudia často krát odsudzujú dnešnú mládež, ktorá nedokáže odísť od počítačov. Je pravda, že pohyb vonku to nenahradí, ale čo sa týka rozvíjania schopností, vedomostí a myslenia je to pozitívna vec. Veľmi oceňujeme aj to, že mnoho takýchto vecí sa už praktizuje aj vo viacerých jazykoch. Prevažne angličtina pomáha najmä mladým zdokonaľovať sa aj v cudzích jazykoch a rozvíjať svoje vedomosti.

Pri takýchto programoch rozvíjajúcich naše myslenie je veľmi účinná aj tímová práca. Navzájom sa ľudia zdokonaľujú, vymieňajú si vedomosti, každý používa a využíva to najlepšie, čo v sebe má, dokonca sa snaží byť vždy lepší ako sú ostatní a spoločný výsledok je potom oveľa lepší. Často krát napriek veľkej vzdialenosti alebo rozdielnemu veku sa vytvorí úžasné veci a tomu všetkému predchádzajú i obyčajné detské hry.

Koncepcia riešenia

Na internete existuje množstvo stránok, kde môžeme nájsť veľa rôznych testov a hier, ktoré rozvíjajú naše schopnosti. Ako je už spomenuté v motivácii, jednou z možností ako predísť stereotypu, čo odrádza účastníkov tréningu, sú napríklad hry a hlavolamy pre viac hráčov, či už hráči hľadajú riešenie hlavolamu spoločne, alebo spolu súperia.

Dobrym osviežením pre tímové práce by možno bolo zapojiť do ich riešenia čo najviac ľudí, pretože každý dokáže vniesť do problému nové poznatky, a tým obohatiť o svoje vedomosti aj ostatných. Viac kratších a menšie množstvo príliš dlhých tímových aktivít, pretože neúmerne dlhé hry či tímové práce často vedú k únave a nesústredenosti.

Tento program by rozvíjal všeobecné a špeciálne schopností detí, mládeže a dospelých. U detí je možnosť využitia tohto programu na základných školách v rámci niektorých predmetov, počítačových krúžkov, či v centrách voľného času na rozvoj a zistenie inteligencie numerických, pamäťových, umeleckých a verbálnych schopností. Mohol by pomáhať či už deťom alebo rodičom, ktorým záleží na tom, aby ich deti trávili čas strávení pri počítači zmysluplne.

Jednotlivé úlohy a hlavolamy by boli rozdelené do rôznych kategórií podľa:

- veku
- typu hlavolamu (logické, vedomostné, jazykové, ...)

Program by takisto mal obsahovať možnosť si tieto schopnosti otestovať v podobe rôznych testov (IQ testy, testy na rýchlosť, presnosť, ...) a získané výsledky ukladať, aby si používateľ mohol porovnať po uplynutí určitej doby.

Dôležitým prvkom je taktiež, aby takýto systém hier a úloh bol multiplatformový, pretože niektorí by ho častejšie využívali napríklad v práci, kde sa potrebujú počas prestávky odreagovať a uvoľniť, iní by ho zasa používali počas cesty vlakom.

Prispôsobiteľný Widget

Motivácia

V dnešnej dobe, keď je k dispozícii nesmierne množstvo informácií rozmiestnených na mnohých miestach je pre používateľa nesmierne ťažké mať dostatočný prehľad o aktuálnych informáciách, v tomto prípade o podujatiach, ktoré by ho mohli zaujímať. Agregácia informácií podobného zamerania tak, aby uľahčila používateľovi rýchle vyhľadanie je zaujímavý problém ako z hľadiska možností pridávania lokalít z ktorých možno čerpať (napríklad možnosť automatického vyhľadávania web stránok s požadovaným obsahom respektíve využitie už existujúcich služieb), tak aj z hľadiska triedenia a následného výberu relevantných dát pre konkrétneho používateľa a nakoniec vhodnej prezentácie dát (prostredníctvom widgetu v rámci webových stránok, alebo ako aplikácie na strane klienta). Dátovým aspektom sa venujú viacerí členovia nášho tímu na predmete Pokročilé databázové systémy. Aspektom prezentácie dát sa venovali niektorí členovia na predmete webové publikovanie v rámci bakalárskeho štúdia. Našou snahou bude vytvoriť taký systém, ktorý nie len napomôže k lepšiemu prístupu používateľov k informáciám o podujatiach, ale bude rozšíriteľný aj o iné záujmové oblasti. Uvedomujeme si aj aspekt prívetivého a prehľadného rozhrania, aby bolo používanie tohto systému čo najjednoduchšie a najefektívnejšie. Tému sme si vybrali kvôli jej aktuálnosti, ale hlavne kvôli pestrému záberu problémových oblastí počnúc automatizáciou zbierania informácií prostredníctvom internetu, cez komunikáciu klienta a servera, končiac ich prezentáciou. Našou hlavnou snahou bude vytvoriť životaschopný produkt, ktorý sa uchytí aspoň medzi používateľmi z akademickej obce. Náš tím je po všetkých stránkach schopný splniť túto úlohu a vytvoriť kvalitný produkt.

Koncepcia riešenia

Architektúra systému

Vzhľadom na požiadavku vkladania informácií (gadget), ale aj kvôli čo najrýchlejšiemu vyhľadávaniu informácií a hlavne kvôli požiadavke centralizovaného úložiska informácií navrhujeme klient-server riešenie.

Úlohou aplikácie na strane servera bude vyhľadávanie, zhromažďovanie a triedenie informácií. Aplikácia bude mať za úlohu vyhľadanie podujatia zo zadaných zdrojov. Nájdené podujatie zaradí do úložiska a pridá mu meta tagy, označenia, na základe nájdeného textu. Prostredníctvom týchto tagov potom bude môcť používateľ filtrovať podujatia. Aplikácia na strane servera bude klientovi priebežne poskytovať dáta o tagoch (v prípade doplnenia, alebo vymazania), ktoré sa nachádzajú v úložisku. Ak klient požiadava, server odošle dáta obsahujúce hľadané podujatia. Všetky používateľské nastavenia klienta týkajúce sa nastavenia filtrov a podobne, budú uchovávané na serveri.

Úlohou aplikácie na strane klienta bude prezentácia dát z centrálného úložiska. Toto oddelenie umožní vytvorenie klienta formou klasickej aplikácie, alebo formou webovej aplikácie ktorú bude možné vložiť na web stránku. V takejto forme potom bude možné implementovať aj používateľské nastavenia a autentifikáciu. Autentifikácia je potrebná, ak chceme aby si mohol každý používateľ nastaviť filtre pre požadované podujatia, a aby tieto filtre boli použité pri jeho prihlásení sa na klienta (umožní prihlásiť sa z rôznych klientov – čo sa týka ich umiestnenia).

Kľúčovou časťou je plnenie centrálného úložiska dátami. V prvom kroku by sa prehľadávali prednastavené stránky (na základe kľúčových slov). Napríklad by sa na stránke FIIT vyhľadávali medzi oznamami a v náležitých preddefinovaných sekciách príspevky obsahujúce kľúčové slová ktoré by mohli naznačovať že ide o informáciu týkajúcu sa podujatia. Bude potrebné optimalizovať tento algoritmus, aby vyhľadával čo najpresnejšie práve požadovaný obsah.

Po dokončení prvého kroku bude možné pristúpiť k automatizovanému prehľadávaniu preddefinovaných domén.

Návrh riešenia

Na riešenie problému je možné použiť .NET framework. Komunikáciu medzi klientom a serverom zabezpečuje služba WCF (Windows Communication Foundation services). Vrstva GUI bude implementovaná podľa vzoru MVC (Model View Controller pattern):

1. Model – predstavuje prepojenie medzi View, Controller a dátovým modelom
2. View – predstavuje časť, ktorá sa zobrazí používateľovi
3. Controller – slúži na prácu s dátami

Volanie na server sa uskutočňuje vždy vo vrstve Controller. Použitie tejto technológie umožňuje obojstrannú komunikáciu medzi klientom a. Toto riešenie zabezpečí budúcu rozšíriteľnosť systému.

Simulated Car Racing Competition 2011

Motivácia

Téma je zaujímavá najmä tým, že poskytuje možnosť venovať sa viacerým rôznym oblastiam počas riešenia projektu, rôzne tieto oblasti kombinovať a rozhodovať sa, ktoré využiť na riešenie konkrétnej časti projektu. Vieme si predstaviť napríklad využitie týchto oblastí:

- **umelá inteligencia** (spracovanie vnemov, učenie sa, neurónové siete, genetické algoritmy, prehľadávanie stavov, plánovanie, algoritmy riešenia problémov - napr. rozkladom) – je tu možnosť využiť poznatky z predmetu Umelá inteligencia, zdokonaľiť sa v nich alebo hľadať ďalšie využiteľné oblasti
- **fyzika** - využitie známych fyzikálnych vzťahov, hľadanie ich vhodných kombinácií, odvodzovanie vhodných vzorcov na riešenie problému
- **matematika** (pravdepodobnosť, štatistika) – využiteľné napríklad pri hľadaní najvhodnejšej možnosti (počas rozhodovania sa pri rôznych možných akciách), výpočte kombinácie viacerých možností (ktorá ich najlepšie charakterizuje), rozhodovaní sa medzi akciami na základe pravdepodobnosti určitého javu alebo pri výpočte vhodných parametrov do používaných vzťahov
- **návrh softvéru** (softvérové architektúry, návrhové vzory) – pri navrhovaní štruktúry systému, delení na čo najmenej závislé časti a delení zodpovednosti

Práca v rôznych oblastiach (napríklad skúmanie možností použitia niektorej konkrétnej oblasti umelej inteligencie) a možnosť vhodne deliť systém pri jeho návrhu je výhodou pri projekte pre tím. Je tu možnosť samostatnej práce na nejakej jednoznačne identifikovanej úlohe a tiež možnosť pracovať v tíme pri návrhu úloh (a tým toho, ktorým smerom sa uberať), kombinovaní výstupov do uceleného systému a testovaní. Výhodou teda je, že riešenie sa dá rozdeliť na mnoho elementárnych problémov (napr. čo urobiť po vyjdení z trate alebo ako sa správať po zrážke) a riešenie každého takéhoto problému prispeje určitou časťou k lepšiemu celkovému riešeniu. Zaujímavé je hľadať aj spôsob spájania jednotlivých častí logiky (ktoré môžu byť vytvorené na základe poznatkov z úplne odlišných oblastí), teda napríklad určenie prioritnej akcie (ak máme na výber viac vhodných akcií) alebo spôsobu zjednocovania akcií. Téma je vhodná na dlhodobé riešenie a vhodne motivuje členov tímu venovať úsilie projektu, keďže na začiatku nevieme, aký bude výsledok a čo všetko sa podarí dosiahnuť. Je to rozdiel oproti témam, kde už po fáze návrhu je jasné, ako bude výsledný systém vyzeráť. Pri tejto

téme by sme nepoznali cieľ a snažili by sme sa (možno najmä zo zvedavosti) dotiahnuť projekt do čo najlepšieho konca a počas práce by bolo pozorovateľné zlepšovanie.

Koncepcia riešenia

Zo zadania vyplýva, že ide o problém spadajúci najmä do oblasti umelej inteligencie. Konkrétne ide o vytvorenie rozumného agenta, ktorý na základe vstupov vie v danom čase vytvoriť výstup, ktorý by mal v danej situácii prispieť k jeho čo najväčšej úspešnosti. Táto transformácia vstupov na výstupy sa bude diať v zložitom systéme, ktorý treba počas riešenia projektu vytvoriť.

Počas riešenia projektu sa budeme musieť zaoberať riešením nasledujúcich problémov.

1. Ako sa dopracovať k vhodnému výstupu?

Cieľom je teda nejakým spôsobom vytvoriť výstup, ktorý náš „systém“ vráti v konkrétnom kroku. Musíme si uvedomiť, od čoho tento výstup závisí:

- aktuálny vstup
- preddefinované správanie sa - výsledok návrhu, hľadania vhodných algoritmov, parametrov, vzorcov
- skúsenosti – naučené parametre počas aktuálneho testu – môžu byť začleňované do správania sa ako nové získané informácie o problémovej oblasti, ktoré sú všeobecne použiteľné aj pre ďalšie „spustenia“ systému

2. Architektúra systému

Musíme vymyslieť vhodnú architektúru (alebo viac použitých architektúr). Je možné napríklad vytvoriť systém s viacerými vrstvami s rôznymi úrovňami abstrakcie. Vhodný výber architektúry nám uľahčí ďalšiu prácu, keď bude treba vytvárať komponenty riešiace niektoré konkrétne situácie a budeme musieť ich výstupy vhodne spojiť.

3. Tvorba výsledného výstupu

Systém má len jeden výstup (v danom kroku), ktorý musí byť „tou najlepšou“ kombináciou toho, čo systém dokáže vypočítať. Preto treba vhodne kombinovať výsledky poskytnuté jednotlivými komponentmi. Môžeme identifikovať napríklad tieto prípady:

- dva komponenty poskytnú rôzne návrhy vykonanej akcie, ktoré majú podobný charakter a rovnakú prioritu (napríklad akcia, ktorá musí byť vykonaná, aby sme sa vyhli zrážke so súperom a akcia, ktorou sa môžeme vyhnúť nárazu do okraja trate) – tieto dva výsledky môžeme vhodne skombinovať do najviac prijateľného výsledku (teda takého, ktorý tieto dva výsledky čo najlepšie charakterizuje – môže vzniknúť na základe váh jednotlivých výsledkov, podľa ktorých výsledky prispievajú svojou časťou do konečného riešenia)
- komponent poskytne výsledok, ktorý má vyššiu prioritu (napríklad to, že sa musíme vyhnúť zrážke, má vyššiu prioritu, ako to, že treba zrýchľovať na rovnej trati) – jeden výsledok môžeme ignorovať a použiť ten s vyššou prioritou

Kroky súvisia aj s použitím rôznych návrhových vzorov (nielen architektonických) – napríklad pomocou vzoru reťaz zodpovednosti sa dá dosiahnuť to, že ak nejaký komponent nevie poskytnúť vhodný výsledok v danom čase, pokračujeme ďalším komponentom (ktorý má nižšiu prioritu).

4. Aké komponenty by mohli poskytovať jednotlivé medzivýsledky?

- komponent poskytujúci akcie z naučenej postupnosti – napríklad návrat na trať, nájdenie správneho smeru po zrážke alebo obídenie súpera
- komponent, ktorý hľadá stav na základe aktuálnej situácie – napríklad zrýchlenie na rovnej trati, brzdenie pred zákrutou
- komponent vytvárajúci plány (niekoľko krokov dopredu) a poskytujúci akcie na základe plánu

5. Pomocné komponenty systému

Na zjednodušenie práce by bolo možné vytvoriť pomocné komponenty, napríklad:

- spracovanie vstupu (uloženie do jednoducho spracovateľnej štruktúry v pamäti)
- úložisko znalostí (vhodná dátová štruktúra a vhodné rozhranie na prístup k nej) – uchovávanie všeobecne použiteľných parametrov, vzorcov, postupností akcií (získané testovaním, výpočtami, algoritmami umelej inteligencie)
- úložisko skúseností použiteľných na aktuálnu trať
- správca priorít (uchovanie zrežazenia komponentov podľa priority)
- kombinácia výsledkov (časť systému poskytujúca vhodné výpočty na kombináciu výstupov do celkového výstupu)
- zjednodušenie ovládania auta (napríklad simulácia automatickej prevodovky, teda rozhranie pozostávajúce len s akcie zrýchľovania a spomaľovania)

6. Získanie vhodných algoritmov, parametrov

Počas tvorby správania sa môžeme získať údaje:

- testovaním (hľadanie parametrov na základe štatistických výpočtov)
- algoritmami umelej inteligencie (genetické algoritmy, neurónové siete, prehľadávanie stavového priestoru)
- z fyzikálnych vzťahov

Príloha A - Zoradenie všetkých tém podľa priority

1. Tréner mentálnych schopností
2. Simulated Car Racing Competition 2011
3. Prispôsobiteľný Widget
4. Evolučný simulátor umelého života založený na heuristických pravidlách
5. Portál pre časopis
6. Správa študentských projektov na fakulte
7. Objektové úložisko dát
8. Virtuálna FIIT
9. Tvorba rozvrhov
10. Vyhľadávanie a sprístupnenie citácií
11. Platforma pre realizovanie transakcií prostredníctvom mobilných zariadení
12. Crowdsourcing verejných dát
13. Model používateľa pre jeho identifikáciu
14. Dizajn s použitím obohatenej reality
15. RoboCup tretí rozmer
16. 3D grafická podpora vyhľadávania znalostí v dokumentoch
17. Interaktívna vizualizácia grafových štruktúr v 3D priestore
18. Adaptívny proxy server

2 Plán projektu

V tejto kapitole je plán projektu rozvrhnutý po jednotlivých týždňoch (viď obrázok 2-1). Ďalej je priložený výpis z nástroja pre plánovanie s názvom Trac. V ňom sú uvedené všetky detailné úlohy podľa jednotlivých členov k jednotlivým míľnikom počínajúc 4. týždňom, končiac 6. týždňom.

Tabuľka 2-1: Hlavný plán projektu

Týždeň	Plánované činnosti	Stav	Zodpovedný	Ukončenie
1.	Zostavenie projektového tímu	Splnené		
	Určenie priorít jednotlivých tém	Splnené	Celý tím	28.9.
	Vypracovanie a odovzdanie ponúk na témy	Splnené	J. P., T. H., J. S.	28.9.
2.	Uchádzanie sa o témy a pridelenie témy	Splnené	Celý tím	5.10.
	Rozdelenie úloh v rámci tímu	Splnené	J. P.	5.10.
3. -5.	Naštudovanie a oživenie použitých knižníc v najnovších verziách	Splnené	M. L., T. H.	12.10.
	Oživenie aplikácie, ktorá sa má rozšíriť	Splnené	T. H., M. L.	12.10.
	Zavedenie nástroja pre plánovanie, zadávanie a sledovanie úloh	Splnené Splnené	J. P.	19.10.
	Vytvorenie tímovej stránky		J. P.	19.10.
4. -7.	Analýza existujúceho systému	Splnené	T. H.	26.10
	Analýza možností rozšírenia existujúceho systému	Splnené Splnené	M. L.	26.10.
	Špecifikácia požiadaviek projektu	Splnené	J. P.	26.10.
	Vytvorenie dokumentácie k časti analýza	Splnené Splnené	M. L., T. H., J. S, J. P.	2.11.
	Vytvorenie predbežného návrhu	Splnené	M. L.	2.11.
	Vytvorenie dokumentácie k časti riadenia	Splnené	J. P.	2.11.
7. - 9.	Vytvorenie finálnej verzie návrhu	Splnené	M. L., J. P.	16.11.
	Oddelenie modulu pre parsovanie	Splnené	M. L.	23.11.
	Implementácia funkcionality databázy	Splnené	T. H.	30.11.
	Implementácia parsera RSF	Nesplnené	J. S.	30.11.

	Implementácia zobrazenia multi-hrán	Splnené	J. P	30.11.
9. -11.	Testovanie funkcionality databázy	Splnené	M. L.	7.12.
	Testovanie parsera RSF	Zrušené	T. H.	7.12.
	Testovanie multi-hrán	Nesplnené	J. S.	7.12.
12.	Odovzdanie tímového projektu v zimnom semestri		Celý tím	14.12.
	Vytvorenie prezentácie prototypu		J. P., M. L.	14.12.
	Prezentácia tímového projektu		J. P., M. L., T. H.	15.12.

Plánovanie, pridelovanie úloh a ich sledovanie je realizované v nástroji Trac. Na nasledujúcich obrázkoch 2-1, 2-2 a 2-3 je zjednodušený zoznam všetkých priradených úloh (v nástroji sú označené ako „Ticket“) k jednotlivým míľnikom a ich stav. V hlavičke je napísaný názov míľnika, napríklad „Tímové stretnutie 5“. Pod hlavičkou sú uvedené jednotlivé úlohy, ich číslo, názov, aktuálny stav, riešenie a ich vlastník.

Tímové stretnutie 6. - odovzdanie dokumentácie (14 matches)				
Ticket	Summary	Status	Resolution	Owner
#27	Analýza XPath	new		sveco86@gmail.com
#28	Analýza knižníc na parsovanie	new		sveco86@gmail.com
#33	Analýza rozmiestnenia uzlov v priestore pri načítaní grafu	new		sveco86@gmail.com
#34	Analýza nižšej a vyššej úrovne grafov a uzlov	new		sveco86@gmail.com
#30	Analýza podpory rôznych typov grafov v projekte, vnorené grafy	closed	resolved	milan33@gmail.com
#36	Vypracovanie návrhu – Parser dátového formátu GXL	closed	resolved	milan33@gmail.com
#26	Overenie typu a formátu dát ukladaných do databázy	closed	resolved	susedik.jakub@gmail.com
#35	Analýza a opis problémovej oblasti	closed	resolved	jaro.prokop@gmail.com
#37	Vypracovanie návrhu – Vyhľadávanie a filtrovanie v databáze	closed	resolved	susedik.jakub@gmail.com
#39	Vypracovanie dokumentácie k riadeniu	closed	resolved	jaro.prokop@gmail.com
#38	Vypracovanie návrhu – Zobrazovanie multihrán	closed	resolved	hurbant@gmail.com
#32	Analýza iných možností layoutovania	closed	resolved	hurbant@gmail.com
#29	Rozchodiť projekt v Debug mode	closed	resolved	hurbant@gmail.com
#31	Analýza snímania tváre používateľa pre pohyb kamery v programe	closed	resolved	susedik.jakub@gmail.com

Obrázok 2-1

Timové stretnute 5 (11 matches)

Ticket	Summary	Status	Resolution	Owner
#19	Analýza parsovania veľkých xml súborov	new		sveco86@gmail.com
#20	Analýza ako osg podporuje vizualizáciu rozsiahlych dát	new		sveco86@gmail.com
#22	Vypracovať analýzu knižnice Bullet	closed	resolved	susedik.jakub@gmail.com
#21	Analýza ako dosiahnuť rozklikávanie uzlov	closed	resolved	hurbant@gmail.com
#15	Analýza existujúceho systému	closed	resolved	hurbant@gmail.com
#24	Analýza databázy - čo sa do nej teraz ukladá	closed	resolved	susedik.jakub@gmail.com
#23	Vypracovať analýzu dátového formátu GXL	closed	resolved	milan33@gmail.com
#16	Analýza možností rozšírenia systému - čo možno doplniť	closed	resolved	milan33@gmail.com
#18	Spísať osnovu dokumentácie	closed	resolved	jaro.prokop@gmail.com
#17	Analýza zobrazenia multihrán	closed	resolved	jaro.prokop@gmail.com
#25	Zostavenie plánu na prvý semester	closed	resolved	jaro.prokop@gmail.com

Obrázok 2-2

Timové stretnute 4 (13 matches)

Ticket	Summary	Status	Resolution	Owner
#10	Preskúmať aké options sa dajú nastaviť	new		sveco86@gmail.com
#11	Spísať použité a použiteľné technológie.	new		sveco86@gmail.com
#12	Naštudovať Bullet	closed	resolved	susedik.jakub@gmail.com
#14	Zostaviť popis architektúry systému	closed	resolved	susedik.jakub@gmail.com
#13	Zistiť aké options sa dajú v aplikácii nastaviť	closed	resolved	susedik.jakub@gmail.com
#8	Preskúmať grafické zobrazovanie aplikácie	closed	resolved	hurbant@gmail.com
#7	Overiť funkcionality vrcholov a uzlov	closed	resolved	hurbant@gmail.com
#6	Preskúmať grafickú časť aplikácie	closed	resolved	milan33@gmail.com
#4	Naštudovanie GXL	closed	resolved	milan33@gmail.com
#1	Sprevádzkovanie Tracku	closed	resolved	jaro.prokop@gmail.com
#9	Vybrať nástroj na plánovanie	closed	resolved	jaro.prokop@gmail.com
#5	Zistiť dopad nových knižníc na aplikáciu	closed	resolved	milan33@gmail.com
#3	Prvotné zostavenie plánu projektu	closed	resolved	jaro.prokop@gmail.com

Obrázok 2-3

Z týchto výstupov vyplýva že sa jeden člen nezapojil do práce na projekte, alebo nevyhodnotil stav svojich úloh napriek tomu, že bol o to požiadany. Jeho úlohy majú nezmenený status.

3 Úlohy členov tímu

V nasledujúcej tabuľke (viď Tabuľka 3-1) je rozpis členov tímu a ich pozícia (rola) v rámci tímu:

Tabuľka 3-1: rozdelenie pozícií medzi členmi tímu

Člen tímu	Pozícia
Bc. Tomáš Hurban	Manažér podporných prostriedkov, Manažér správy databáz
Bc. Milan Laslop	Manažér vývoja, Manažér kvality
Bc. Jaroslav Prokop	Manažér tímu, Plánovač, Dokumentarista

Každý člen sa zapája do všetkých činností jednotlivých manažérov v rámci projektu a dáva návrhy na zlepšenie. Jednotlivé časti dokumentácie sú rozdelené medzi jednotlivých členov podľa ich zamerania.

4 Metodiky potrebné pri vývoji

4.1 Komunikácia v tíme

Na komunikáciu v tíme používame službu Google Groups. Aby sa informácie ľahšie vyhľadávali, vypracovali sme si metodiku písania názvov jednotlivých vlákien. V nasledujúcom odseku je popis metodiky:

Metodika písania nadpisov jednotlivých tém: Pred textom nadpisu môže byť umiestnených niekoľko značiek, každá v hranatých zátvorkách (Tabuľka 4-1):

Tabuľka 4-1: označenie názvu témy v Google Groups

značka	opis
organizačné	stretnutia, termíny...
problém	otázka ohľadom problému
riešenie	riešenie problému, ktoré by mohlo zaujímať ostatných
nápad	návrh člena tímu na prehodnotenie
podpora	podporné systémy a nástroje

4.2 Úlohy

Na plánovanie, zadávanie a sledovanie úloh používame nástroj Trac. Vychádzame z troch možných situácií, ktoré potrebujeme zatiaľ riešiť:

- Vytvorenie úlohy a jej priradenie konkrétnemu členovi tímu
- Aktualizácia úlohy riešiacim členom
- Ukončenie úlohy

K týmto trom činnostiam týkajúcich sa plánovania sú vypracované metodiky uvedené v nasledujúcich podkapitolách.

4.2.1 Vytvorenie a pridelenie úlohy

Vytvorená úloha musí mať taký názov, aby bolo jednoznačne identifikované, o ktorú úlohu zo zápisu sa jedná. V popise úlohy je udané jej číslo zo zápisu. Ďalej sa k úlohe priradí dátum do kedy má byť vypracovaná, dátum zadania, míľnik ku ktorému sa vzťahuje (do kedy má byť vypracovaná), jej priorita a úloha musí byť priradená konkrétnemu jednému členovi tímu. Tieto časti musia byť uvedené pri každej úlohe ešte pre jej vytvorením. V popise môže byť uvedený upresňujúci popis úlohy.

4.2.2 Aktualizácia úlohy

Ak sa jedná o novú úlohu, člen ktorý ju dostal na zodpovednosť najprv zmení jej stav na „Accepted“. Potom pri každej zmene môže do komentára zadať problémy pri riešení úlohy a ukončené časti. Do percenta plnenia uvedie jeho odhad na koľko má danú úlohu rozpracovanú.

4.2.3 Ukončenie úlohy

Po ukončení úlohy zmení člen tímu jej stav na:

- „Resolved“ - ak danú úlohu ukončil, do komentára napíše potrebné vysvetlenie k riešeniu úlohy.
- „Invalid“ – ak je úloha nezmyselná (už je splnená), alebo ju nie je potrebné riešiť. Do komentára uvedie prečo danú úlohu nie je potrebné riešiť.
- „WontFix“ – ak danú úlohu nevie vyriešiť. Do komentára uvedie na aké problémy narazil, ktoré bránia vyriešeniu úlohy.

Ďalej priradí ukončenej úlohe (so stavom Resolved) percento plnenia na 100%.

Až po splnení týchto požiadaviek odošle člen tímu potvrdenie o zmene stavu úlohy.

4.3 Komentáre pri vkladaní súborov do repozitára

Pre vkládanie do repozitára sme vypracovali nasledujúcu metodiku:

Komentáre budú pozostávať z riadkov, pričom každý bude mať nasledujúcu štruktúru:

[modul] [značky] text

Každá významná zmena by mala byť opísaná textom v samostatnom riadku, pričom *modul* a *značky* slúžia iba na upresnenie a kategorizáciu zmeny v danom riadku.

Vysvetlenia:

modul – vyskytuje sa, iba ak sa zmena týka konkrétneho modulu (modul je v tomto prípade názov časti na najvyššej úrovni v projekte – Core, Data, Layout...; ak sa zmena netýka žiadneho modulu, časť sa vynechá, aj so zátvorkami)

značky – znaky označujúce kategórie, do ktorých zmena patrí – znaky nie sú oddelené medzerami ani inými oddeľovačmi (ak nie sú žiadne kategórie, časť sa vynechá, aj so zátvorkami)

text – text zmeny – začína malým písmenom, nekončí bodkou

Používané značky sú znázornené v nasledujúcej tabuľke (Tabuľka 4-2):

Tabuľka 4-2: značky používané v komentároch pri vkladaní do repozitára

značka	opis
+	pridanie funkcionality
-	zmena funkcionality (správania sa)
i	zmena rozhrania modulu, ktoré môžu používať iné moduly
a	prispôsobenie sa zmenám rozhrania (knížnic alebo iných modulov)
r	refactoring – preusporiadanie kódu bez zmeny funkcionality
!	oprava chyby
t	zmena testov
f	zmena podporných súborov – resources, CMake súbory, knižnice (v dependencies)

Príklad jedného komentára:

```
[Data][ri] príprava na možnosť importu viacerých formátov
[Data][+tf] pridanie importu formátu GraphML, testy importu, súbory na test
importu
[!f] oprava chyby v CMake súboroch
```

4.4 Metodika manažmentu zmien

4.4.1 Pre koho je metodika určená

Metodika je určená najmä vývojovým tímom, ktoré sa zaoberajú vytváraním a testovaním softvérových systémov.

4.4.2 1.2 Zoznam nadväzujúcich metodík a dokumentov

- Ibbs, C.W., Wong, C.K., Kwak, Y.H.: *Project Change Management System*. Journal of Management in Engineering, 2001.
- Paton, R., McCalman J.: *Change Management: A Guide to Effective Implementation*, Third Edition. SAGE Publications Ltd., 2008.

4.4.3 2. Použité pojmy

- **Visual Studio Team Foundation Server 2010 (TFS)** - je platforma pre spoluprácu od spoločnosti Microsoft. Automatizuje proces dodávania softvéru a poskytuje nástroje

pre efektívne riadenie projektov vývoja softvéru počas celého životného cyklu softvéru [1].

- **Team Foundation Version Control (TFVC)** – centrálny repozitár, ktorý slúži na správu revízií zdrojových kódov, dokumentácie, požiadaviek na zmenu či iných informácií, s ktorými sa pracuje počas celého trvania vývoja projektu [2]. Všetky súbory ako aj všetky zmeny v súboroch sa ukladajú do SQL databázy.
- **Visual Studio 2010 (VS 2010)** – Microsoft Visual Studio je integrované prostredie, ktoré zjednodušuje celý proces vývoja od návrhu až po nasadenie [3].

4.4.4 3. Role a ich zodpovednosti

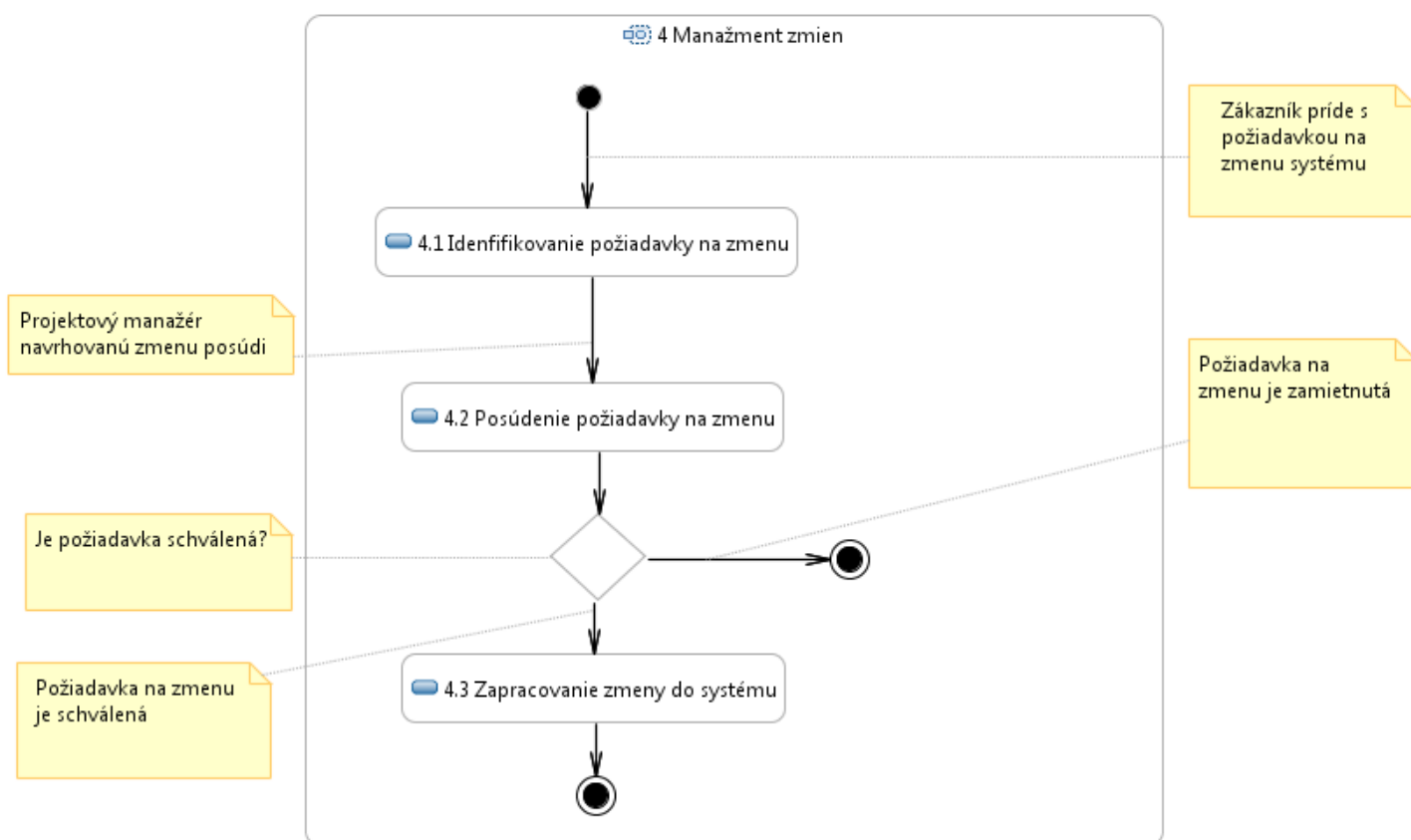
Tabuľka 4-3: pridelenie zodpovednosti jednotlivým rolám

Rola	Zodpovednosť
Projektový manažér	<ul style="list-style-type: none"> • Pridelovanie úloh jednotlivým členom tímu • Posudzovanie a hodnotenie požiadaviek na zmeny v systéme • Riadenie organizácie členov v tíme
Zákazník	<ul style="list-style-type: none"> • Vytvorenie požiadavky na zmenu v systéme • Kontrola vypracovaných zmien
Projektový analytik	<ul style="list-style-type: none"> • Zbieranie informácií a vypracovanie analýz k navrhovaným zmenám • Analyzovanie dopadu vykonaných zmien v systéme
Vývojár	<ul style="list-style-type: none"> • Implementácia schválených zmien • Integrácia implementovaných zmien do systému
Tester (kontrolór)	<ul style="list-style-type: none"> • Testovanie implementovaných zmien • Testovanie zmien integrovaných do systému
Dokumentarista	<ul style="list-style-type: none"> • Dokumentovanie požiadaviek na zmenu v systéme • Dokumentovanie zmien vykonaných v systéme

4.4.5 Manažment zmien

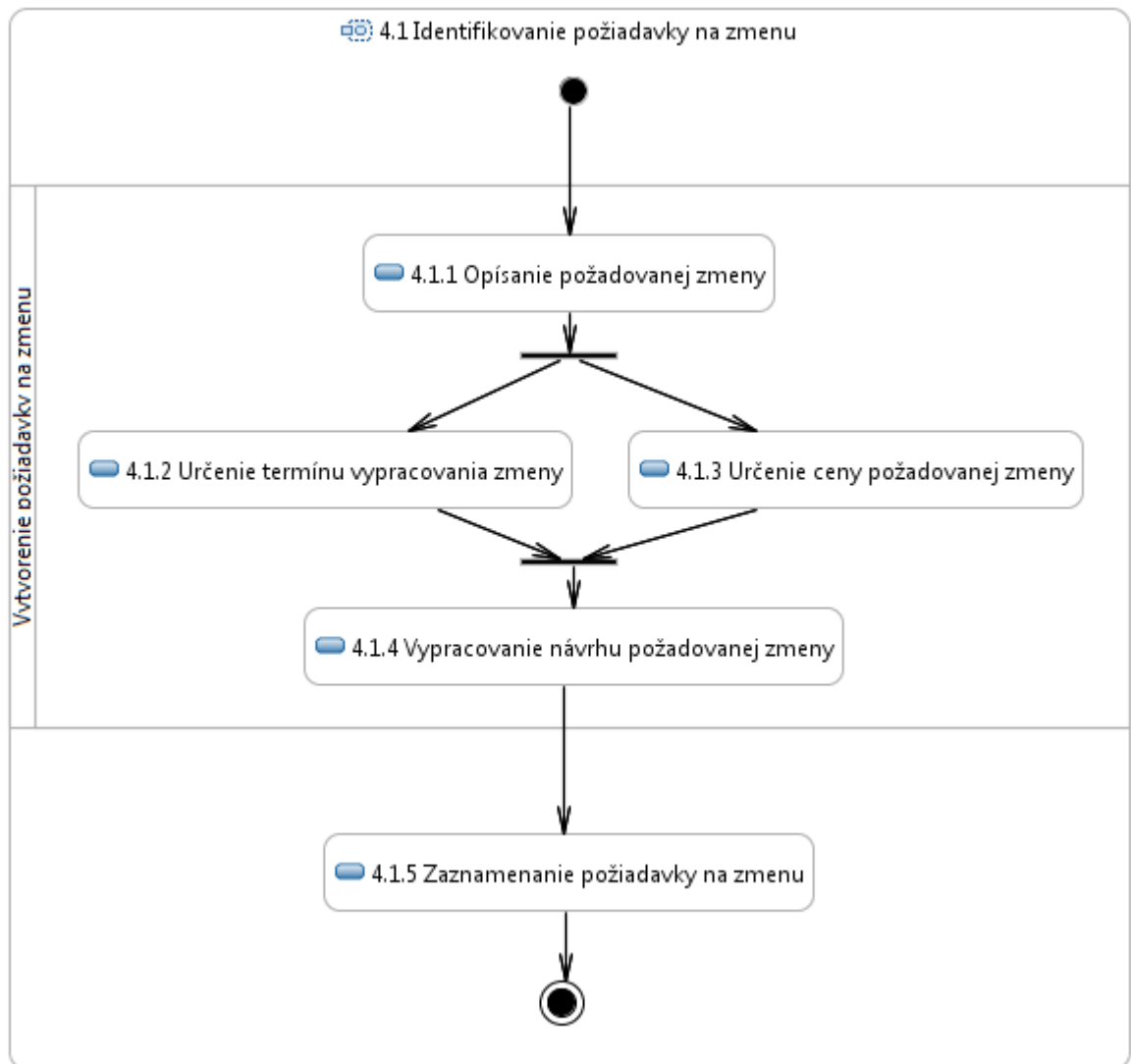
Táto kapitola opisuje procesy manažmentu zmien z pohľadu vývojového tímu. Jednotlivé procesy sú rozdelené do troch skupín:

1. Identifikovanie požiadavky na zmenu
2. Posúdenie požiadavky na zmenu
3. Zapracovanie zmeny do systému



Obrázok 4-1: Diagram aktivít v manažmente zmien

4.4.5.1 Identifikovanie požiadavky na zmenu



Obrázok 4-2: Proces identifikovania požiadavky na zmenu

4.4.5.1.1 Opisanie požadovanej zmeny

Vstup: požiadavka na vytvorenie zmeny v systéme

Výstup: podrobne opísaná požiadavka na zmenu

Zodpovedný: zákazník

Zákazník príde s myšlienkou vytvorenia zmeny v existujúcom systéme. Svoju požiadavku na zmenu podrobne opíše.

4.4.5.1.2 Určenie termínu vypracovania zmeny

Vstup: podrobne opísaná požiadavka na zmenu

Výstup: určenie termínu dokedy je potrebné vypracovať zmenu

Zodpovedný: zákazník, projektový manažér

Zákazník na základe dohody s projektovým manažérom stanoví termín, dokedy sa zmena v systéme vypracuje.

4.4.5.1.3 Určenie ceny požadovanej zmeny

Vstup: podrobne opísaná požiadavka na zmenu

Výstup: určenie ceny za vypracovanie zmeny v systéme

Zodpovedný: zákazník, projektový manažér

Zákazník na základe dohody s projektovým manažérom stanoví cenu za vykonanie navrhovanej zmeny v systéme.

4.4.5.1.4 Vypracovanie návrhu požadovanej zmeny

Vstup: podrobne opísaná požiadavka na zmenu s určeným termínom vypracovania

Výstup: vypracovanie podrobného návrhu požadovanej zmeny

Zodpovedný: zákazník

Zákazník vypracuje návrh požadovanej zmeny v systéme.

4.4.5.1.5 Zaznamenanie požiadavky na zmenu

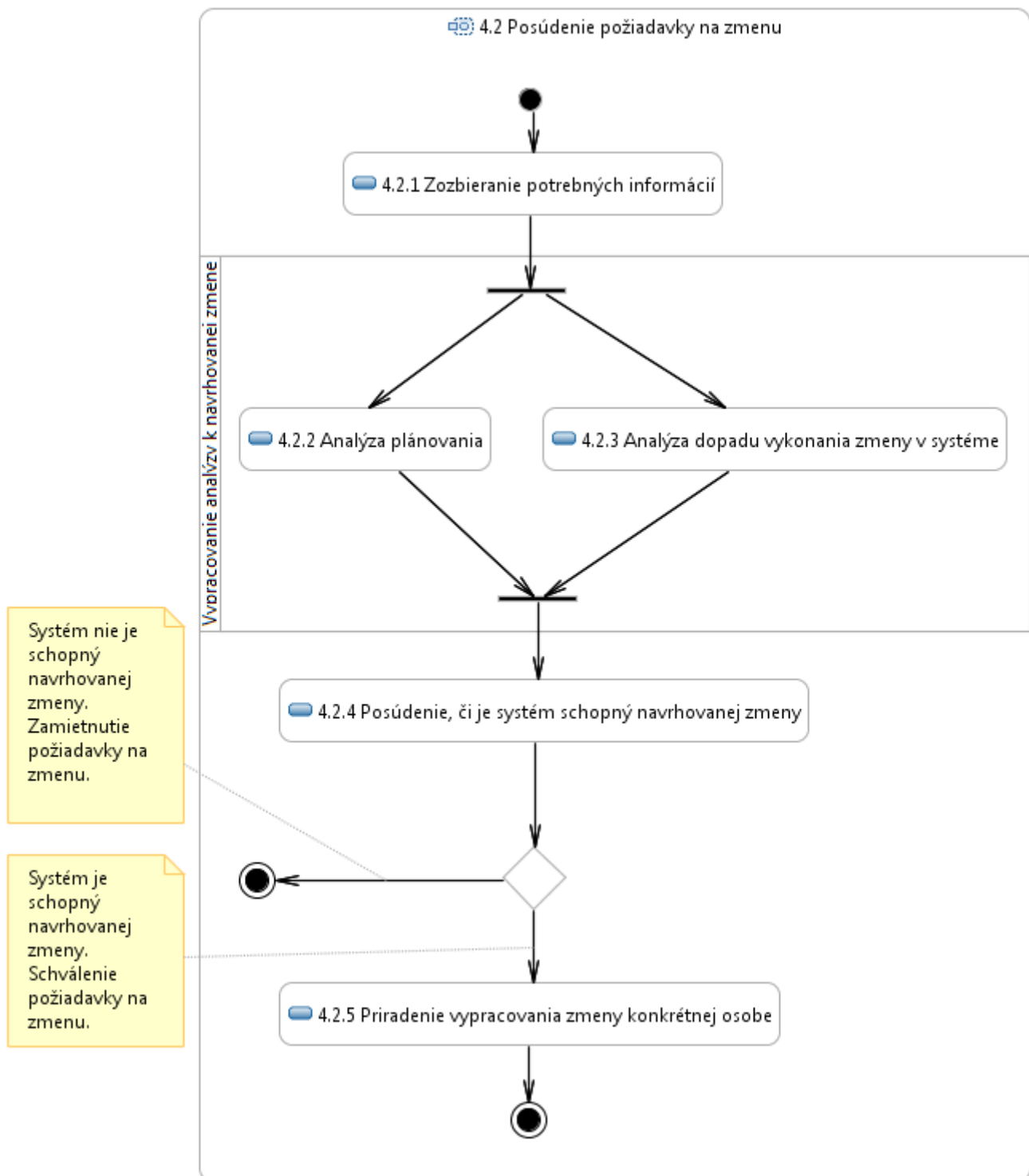
Vstup: vypracovaný návrh požadovanej zmeny

Výstup: zaznamenanie požiadavky na zmenu do systému

Zodpovedný: dokumentarista

Dokumentarista vloží do systému záznam o vytvorenej požiadavke na zmenu.

4.4.5.2 Posúdenie požiadavky na zmenu



Obrázok 4-3: Proces posúdenia požiadavky na zmenu

4.4.5.2.1 Zozbieranie potrebných informácií

Vstup: požiadavka na zmenu evidovaná v systéme

Výstup: spracovanie všetkých potrebných informácií pre posúdenie návrhu zmeny

Zodpovedný: projektový analytik

Projektový analytik vyhledá všetky potrebné informácie a materiály, ktoré je treba, aby sa mohli vypracovať potrebné analýzy plánovania a dopadu navrhovanej požiadavky na zmenu, a aby sa požadovaná zmena mohla implementovať do systému.

4.4.5.2.2 Analýza plánovania

Vstup: získané všetky potrebné informácie pre vytvorenie analýzy

Výstup: vypracovaná analýza plánovania

Zodpovedný: projektový analytik

Projektový analytik vypracuje analýzu plánovania, t. j. ako sa musia zmeniť, resp. doplniť naplánované činnosti, aby bolo možné požadovanú zmenu vykonať v systéme.

4.4.5.2.3 Analýza dopadu vykonania zmeny v systéme

Vstup: získané všetky potrebné informácie pre vytvorenie analýzy

Výstup: vypracovaná analýza dopadu vykonania zmeny

Zodpovedný: projektový analytik

Projektový analytik na základe získaných informácií vypracuje analýzu, kde určí, aký dopad môže mať navrhovaná zmena na systém a či je systém schopný takejto zmeny.

4.4.5.2.4 Posúdenie, či je systém schopný navrhovanej zmeny

Vstup: vypracované analýzy plánovania a dopadu

Výstup: rozhodnutie, či je systém schopný navrhovanej zmeny

Zodpovedný: projektový manažér

Projektový manažér na základe vypracovaných analýz plánovania a dopadu vykonania zmeny a na základe znalostí existujúceho systému zhodnotí a rozhodne, či je systém schopný implementácie a integrácie navrhovanej zmeny. V prípade, že projektový manažér na základe analýz usúdi, že systém je schopný takejto zmeny, pokračuje sa kapitolou 4.2.5, inak sa požiadavka na zmenu zamietne.

4.4.5.2.5 Priradenie vypracovania zmeny konkrétnej osobe

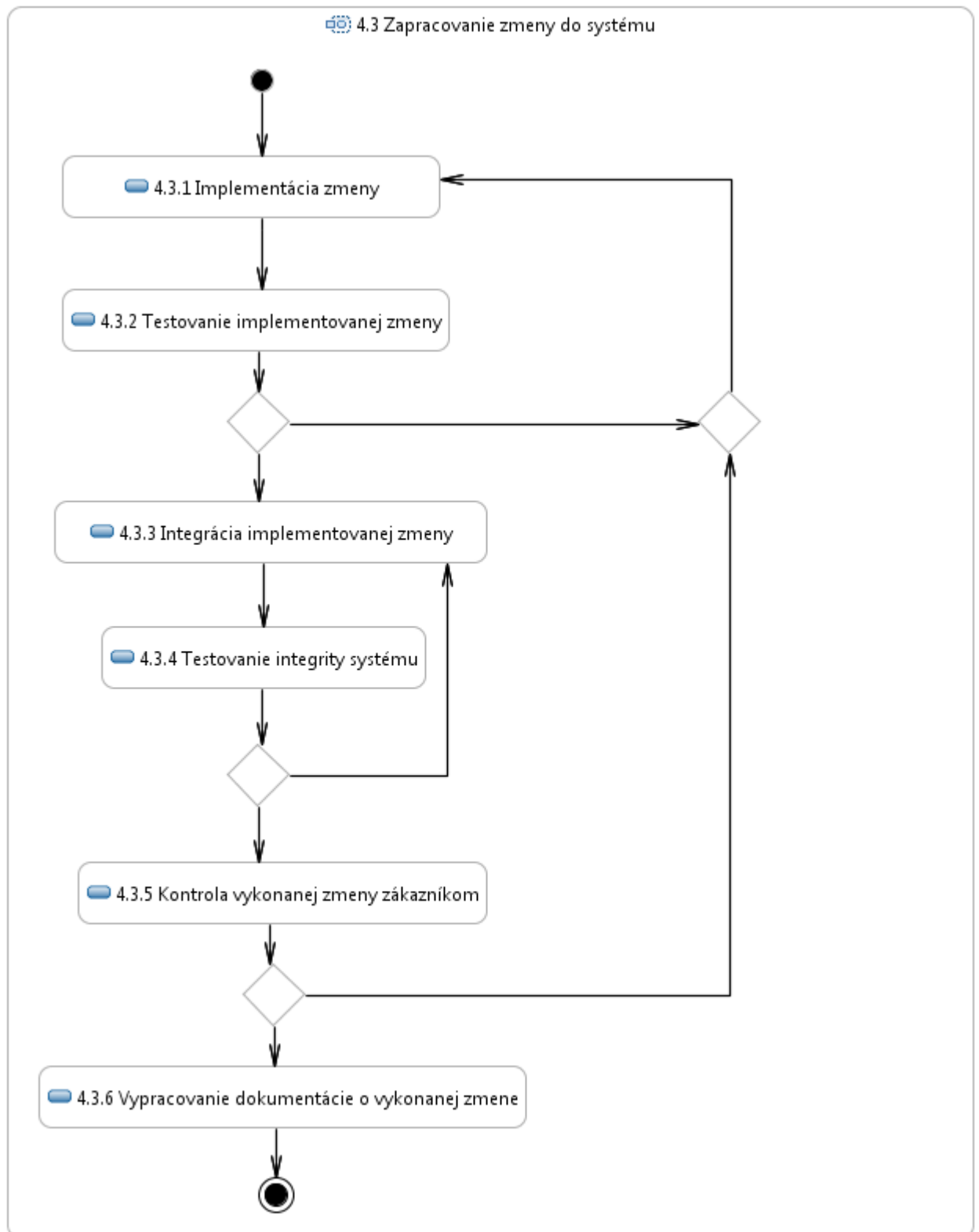
Vstup: rozhodnutie na vypracovanie zmeny v systéme

Výstup: pridelenie úlohy na vypracovanie zmeny konkrétnej osobe

Zodpovedný: projektový manažér

Projektový manažér priradí úlohu na vypracovanie požadovanej zmeny konkrétnemu členovi tímu.

4.4.5.3 Zapracovanie zmeny do systému



Obrázok 4-4:Proces zapracovania zmeny do systému

4.4.5.3.1 Implementácia zmeny

Vstup: pridelenie úlohy na vypracovanie zmeny v systéme

Výstup: implementovaná zmena

Zodpovedný: vývojár

Vývojár po pridelení úlohy na vypracovanie požadovanej zmeny od projektového manažéra a na základe vypracovaných analýz túto zmenu implementuje.

4.4.5.3.2 Testovanie implementovanej zmeny

Vstup: implementovaná zmena

Výstup: vypracovanie záznamu o vykonanom testovaní implementovanej zmeny

Zodpovedný: tester

Tester implementovanú zmenu otestuje a zo získaných poznatkov vypracuje záznam o vykonanej kontrole implementovanej zmeny. Na základe tohto záznamu posúdi, či bola zmena implementovaná úspešne (pokračuje sa kapitolou 4.3.3) alebo nie (pokračuje sa kapitolou 4.3.1).

4.4.5.3.3 Integrácia implementovanej zmeny

Vstup: úspešné otestovanie implementovanej zmeny

Výstup: integrovaná zmena v systéme

Zodpovedný: vývojár

Vývojár integruje implementovanú zmenu do existujúceho systému.

4.4.5.3.4 Testovanie integrity systému

Vstup: integrovaná zmena v systéme

Výstup: vypracovanie záznamu o vykonanej kontrole integrity systému

Zodpovedný: tester

Tester skontroluje integritu systému po zapracovaní implementovanej zmeny. Na základe vykonaných testov vypracuje záznam o vykonanej kontrole integrity systému a posúdi či zmena narušila integritu systému (pokračuje sa kapitolou 4.3.3) alebo nie (pokračuje sa kapitolou 4.3.5).

4.4.5.3.5 Kontrola vykonanej zmeny zákazníkom

Vstup: vypracované záznamy o testovaní zmeny a vykonanej kontrole integrity systému

Výstup: rozhodnutie, či vykonaná zmena súhlasí s navrhovanou zmenou

Zodpovedný: zákazník

Zákazník na základe vypracovaných záznamov o kontrole potvrdí, že vykonaná zmena spĺňa očakávania (v tom prípade sa pokračuje kapitolou 4.3.6) alebo nie (pokračuje sa kapitolou 4.3.1).

4.4.5.3.6 Vypracovanie dokumentácie o vykonanej zmene

Vstup: potvrdenie zákazníka, že vykonaná zmena súhlasí s navrhovanou zmenou

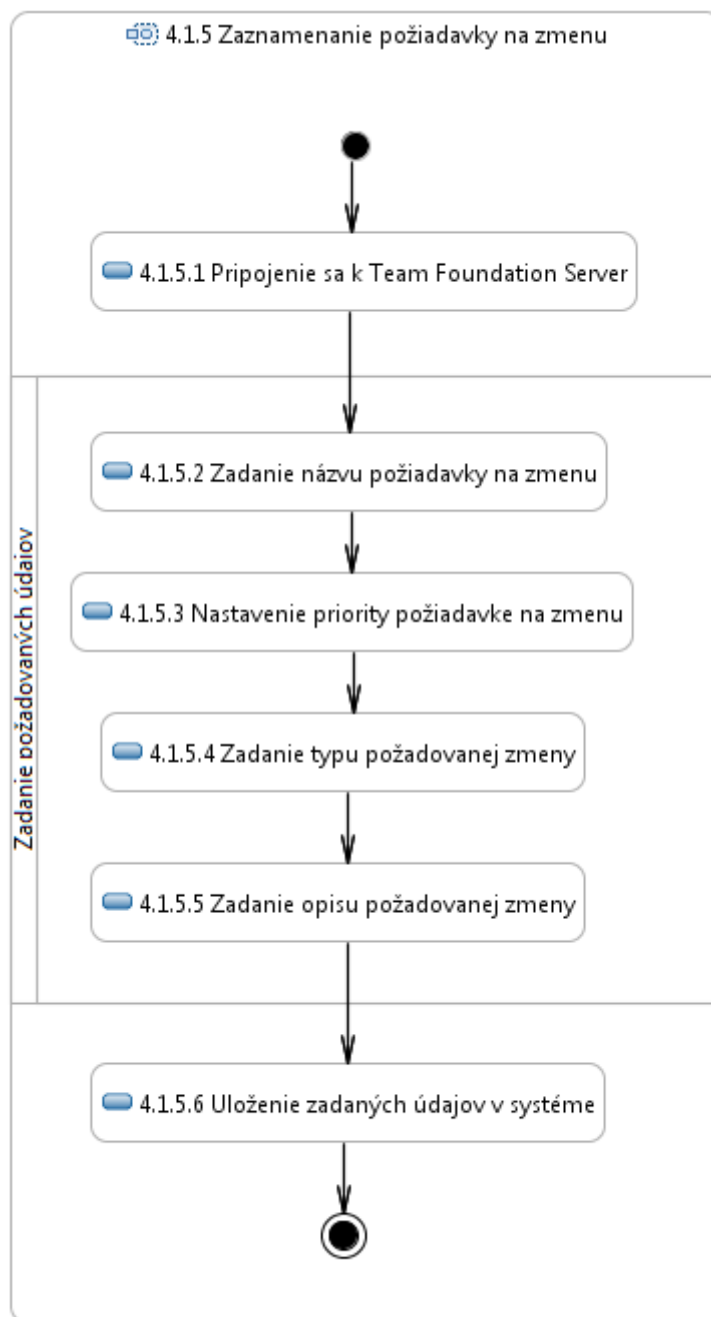
Výstup: vypracovanie dokumentu o vykonanej zmene

Zodpovedný: dokumentarista

Na základe vypracovaných záznamov o vykonanej kontrole implementácie a integrácie, dokumentarista vypracuje dokument o vykonanej zmene v existujúcom systéme.

4.4.6 Podrobný opis procesu Zaznamenanie požiadavky na zmenu

Popis procesu zaznamenania požiadavky na zmenu v prostredí Visual Studio Team Foundation Server 2010.



Obrázok 4-5:Proces zaznamenania požiadavky na zmenu do systému

Vstup: vypracovaný návrh požadovanej zmeny

Výstup: zaznamenanie požiadavky na zmenu do systému

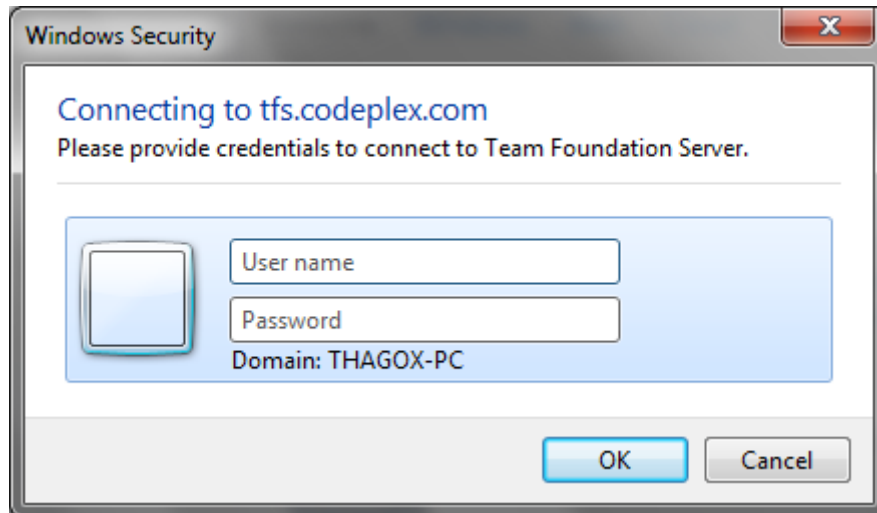
Zodpovedný: dokumentarista

Dokumentarista vloží do systému záznam o vytvorenej požiadavke na zmenu.

4.4.6.1 Pripojenie sa k Team Foundation Server

1. Spustenie Microsoft Visual Studio 2010.

2. V dialógovom okne, ktoré sa zobrazí hneď po spustení programu (obr. 4-6), zadáme do políčka **Používateľské meno** (User name) svoje prihlasovacie meno k Team Foundation Server, do políčka **Heslo** (Password) zadáme svoje heslo a potvrdíme stlačením tlačidla **OK**.



Obrázok 4-6: Dialógové okno pre zadanie používateľského mena a hesla

3. Na úvodnej obrazovke, ktoré sa zobrazí, klikneme na tlačidlo **Pripojiť sa k Team Foundation Server** (Connect To Team Foundation Server).

Informácie o inštalovaní a konfigurovaní Team Foundation Server:

<http://www.microsoft.com/downloads/en/details.aspx?displaylang=en&FamilyID=2d531219-2c39-4c69-88ef-f5ae6ac18c9f>

4. V zobrazenom dialógovom okne si v zozname **vyberieme TFS server**, na ktorý sa chceme pripojiť, vyberieme si **Kolekcie tímových projektov** (Team Project Collections), v ktorej je projekt, na ktorom pracujeme a následne si zo zoznamu **Tímové projekty** (Team projects) vyberieme konkrétny projekt. K vybranému projektu sa pripojíme stlačením tlačidla **Pripojiť** (Connect).

Ak sme postupovali správne, v okne **Tímový prehliadač** (Team explorer) vidíme otvorený náš projekt.

V okne **Tímový prehliadač** (Team explorer) si vyberieme náš projekt a pravým tlačidlom myši klikneme na položku **Pracovné položky** (Work Items). V zobrazenom zozname si vyberieme **Nová pracovná položka** (New Work Item) a klikneme na **Pracovná položka** (Work Item). Zobrazí sa nové okno s názvom **Nová pracovná položka [ID]*** (New Work Item [number]*) (Obr. 4-7.).

4.4.6.2 Zadanie názvu požiadavky na zmenu

Do políčka **Názov** (Summary) (na obr. 4-7. je to políčko s číslom 1) zadáme názov požadovanej zmeny. Tento názov musí byť jasný a výstižný a musí vystihovať hlavnú myšlienku požadovanej zmeny.

4.4.6.3 Nastavenie priority požiadavke na zmenu

V políčku **Priorita** (Priority) (na obr. 4-7. je to políčko s číslom 2) vyberieme jednu z nasledujúcich možností:

- *Vysoká* (High) – túto možnosť vyberieme, ak požadovaná zmena má najvyššiu prioritu, t. j. je potrebné ju vypracovať čo najskôr. Do tejto kategórie patria napríklad chyby, ktoré súvisia so stabilitou systému alebo zmeny kľúčových funkcií systému.
- *Stredná* (Medium) - túto možnosť vyberieme, ak požadovaná zmena má strednú prioritu, t. j. nie je nevyhnutná - nevplyva na stabilitu systému alebo jeho správnu funkčnosť, ale chceme ju vypracovať čo najskôr. Do tejto kategórie patrí napríklad pridanie novej funkcionality do systému.
- *Nízka* (Low) - túto možnosť vyberieme, ak požadovaná zmena má nízku prioritu, t. j. nie je nevyhnutná a nie je potrebné sa s vykonaním tejto zmeny ponáhľať. Do tejto kategórie patria napríklad zmena vzhľadu používateľského rozhrania alebo oprava drobných chýb v systéme.

4.4.6.4 Zadanie typu požadovanej zmeny

V políčku **Typ pracovnej položky** (Work Item Type) (na obr. 4-7. je to políčko s číslom 3) nastavíme jednu z nasledujúcich možností:

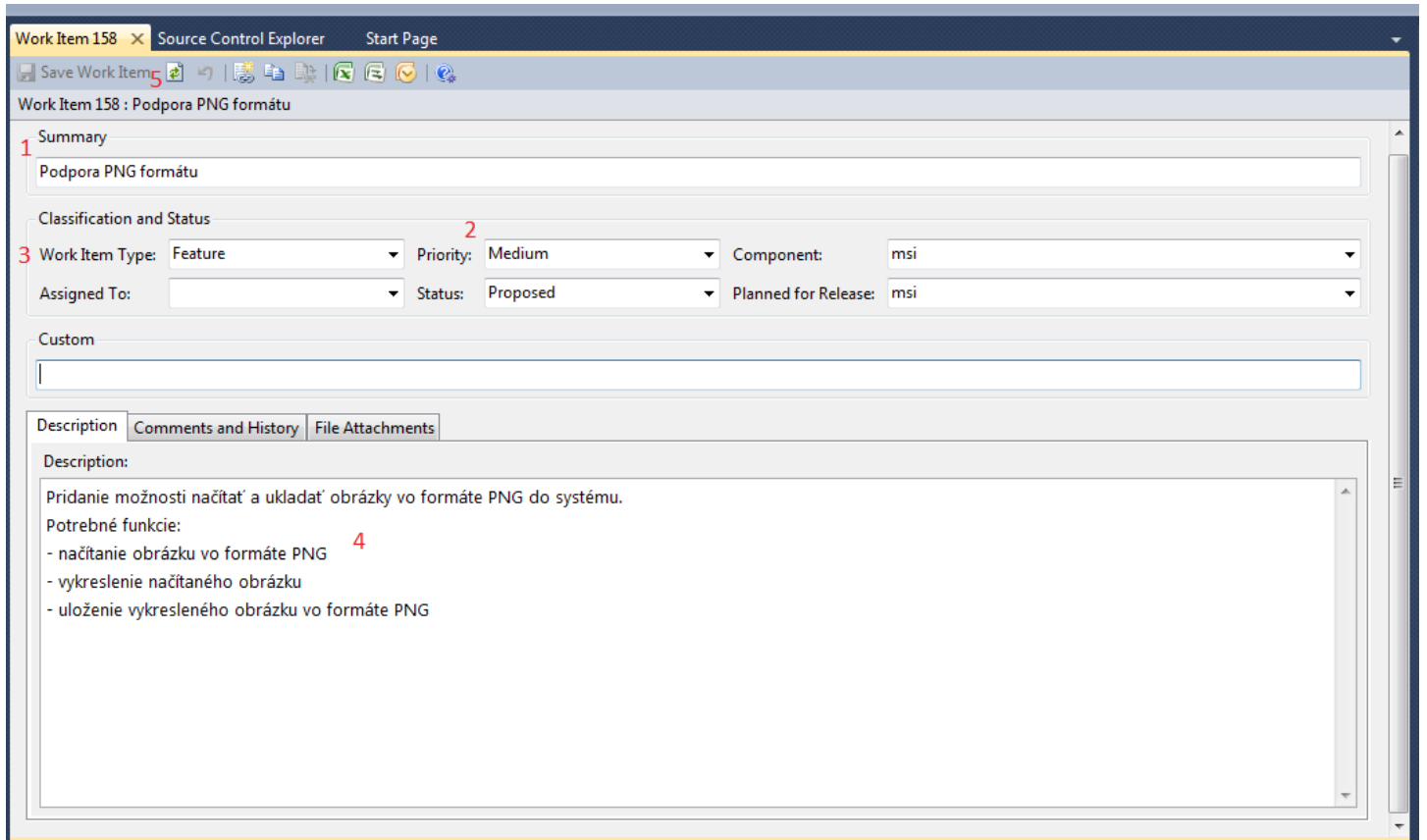
- *Vylepšenie* (Feature) – túto možnosť vyberieme, ak chceme do systému pridať nové vylepšenie alebo chceme vylepšiť už existujúcu časť systému. Napríklad pridanie novej funkcie.
- *Chyba* (Issue) – túto možnosť vyberieme, ak chceme opraviť chybu, ktorá už je v systéme.
- *Úloha* (Task) – túto možnosť vyberieme, ak sa zmena netýka priamo zmeny časti systému.

4.4.6.5 Zadanie opisu požadovanej zmeny

Do políčka **Opis** (Description) (na obr. 4-7. je to políčko s číslom 4) zadáme podrobne opísanú požiadavku na zmenu. V opise musí byť presne špecifikované, čo sa má v systéme zmeniť, resp. dopracovať.

4.4.6 Uloženie zadaných údajov v systéme

Ak sú všetky vyššie spomenuté políčka vyplnené, uloženie týchto údajov do systému vykonáme stlačením tlačidla **Uložiť pracovnú položku** (Save Work Item) v hornej časti obrazovky (na obr. 4-7. je to políčko s číslom 5). Tým sa požiadavka na zmenu uloží do TFVC.



Obrázok 4-7: Okno pre pridávanie požiadaviek na zmenu do systému

4.5 Manažment úloh v distribuovanom projekte

4.5.1 Použitie metodiky

Metodika je určená všetkým členom tímu, ktorý sa zaoberá vývojom softvérových produktov.

4.5.2 Pojmy použité v metodike

Trac – systém určený pre plánovanie, priradovanie a sledovanie úloh v softvérovom projekte.

Ticket – označenie úlohy v systéme *Trac*

Summary – označenie názvu úlohy v systéme *Trac*

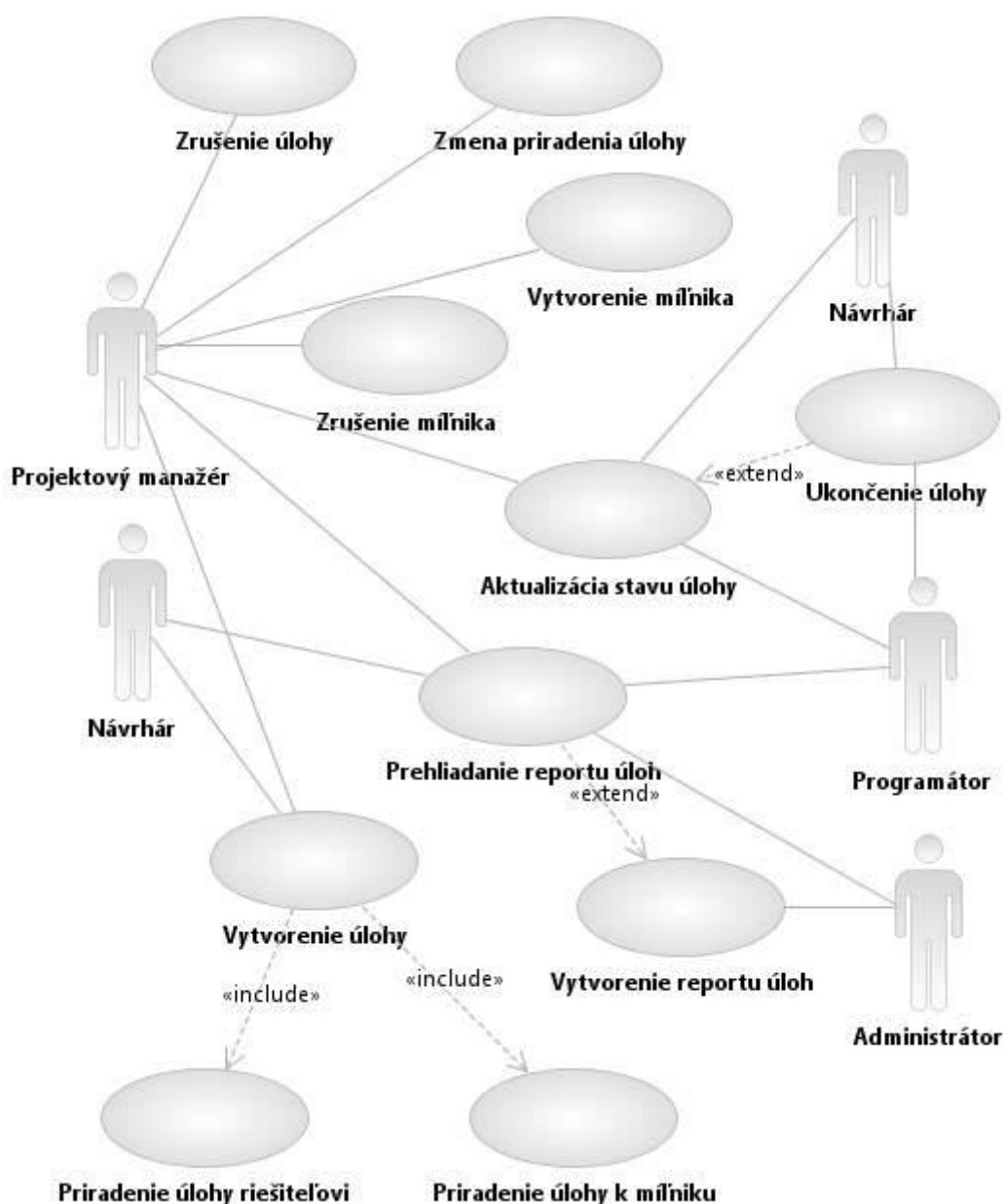
Reporter – označenie osoby zodpovednej za sledovanie úlohy v systéme *Trac*

Tester – osoba zodpovedná za testovanie implementovanej funkcionality

Task – anglické označenie úlohy

4.5.3 Role a zodpovednosti

Táto časť metodiky definuje roly a z nich vyplývajúce zodpovednosť všetkých členov tímu, pre ktorých je metodika určená. V diagrame prípadov použitia uvedenom na obrázku **Chyba! Nenašiel sa žiaden zdroj odkazov.** sú znázornené role a k nim prislúchajúce prípady použitia relevantné pre manažment úloh.



Obrázok 4-8: Diagram prípadov použitia

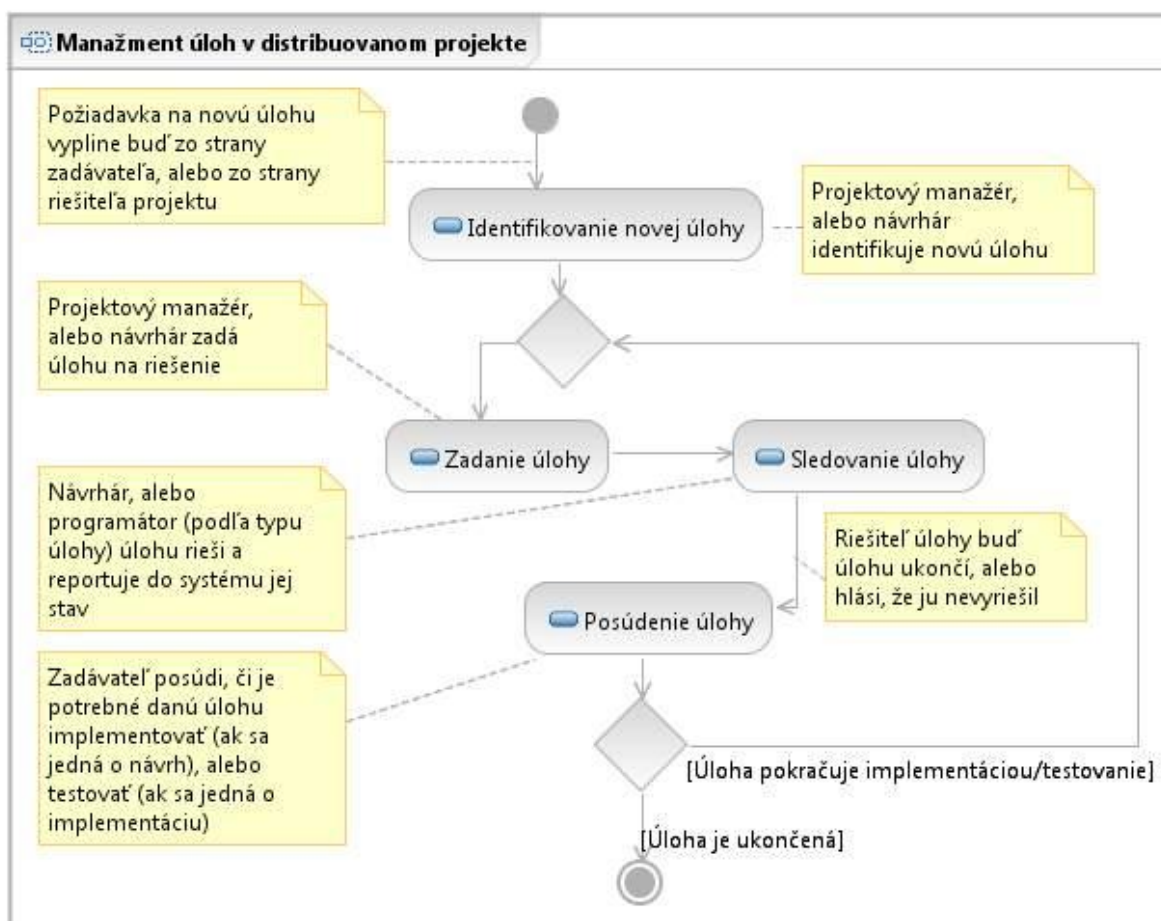
V nasledujúcej tabuľke tabuľke 4- je znázornené priradenie role k zodpovednosti. Rola *Návrhár* je zlúčená s rolou *Analytika* a rola programátora je zlúčená s rolou *Testera*. Z tohto dôvodu sa role *Analytika* a *Testera* neuvádzajú.

Tabuľka 4-4: Roly a zodpovednosti členov tímu

Rola	Zodpovednosť
Projektový manažér	Vytvorenie úlohy, zrušenie úlohy, vytvorenie míľnika, zrušenie míľnika, zmena priradenia úlohy (na iného člena tímu), aktualizácia stavu úlohy, prehliadanie reportu úloh
Systémový administrátor	Vytvorenie reportu úloh, prehliadanie reportu úloh
Návrhár	Vytvorenie úlohy, prehliadanie reportu úloh, aktualizácia stavu úlohy, ukončenie úlohy
Programátor	Aktualizácia stavu úlohy, prehliadanie reportu úloh, ukončenie úlohy

4.5.4 Procesy manažmentu úloh

V tejto časti metodiky sú opísané jednotlivé kľúčové procesy manažmentu úloh. Proces manažmentu úloh je znázornený na obrázku 4-.



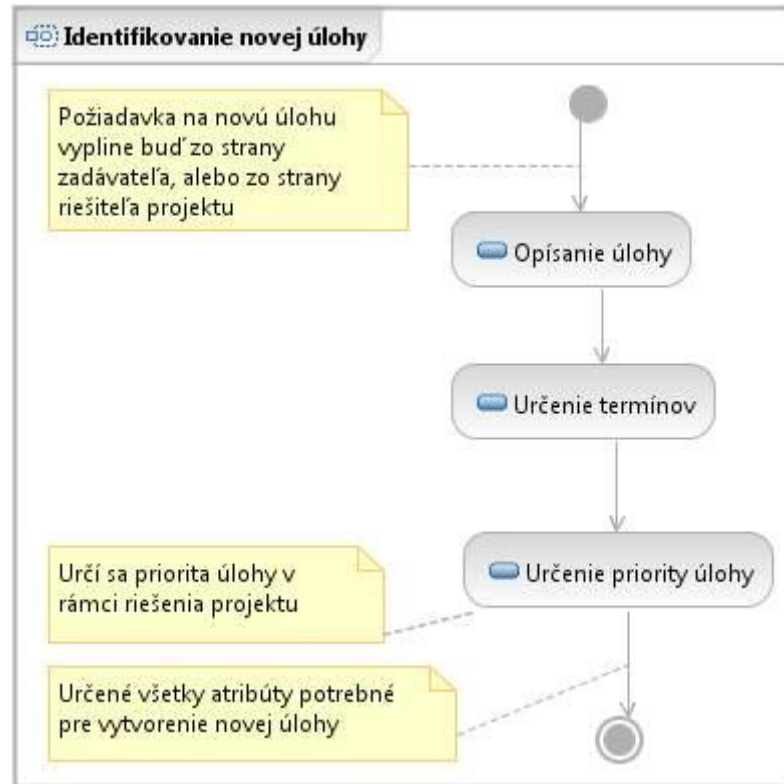
Obrázok 4-8: Aktivity v manažmente úloh

Proces manažmentu úloh má za úlohu spravovať celý životný cyklus úloh počnúc ich identifikáciou a končiac ich realizáciou do finálnej podoby.

V nasledujúcich podkapitolách sú podrobne popísané jednotlivé procesy manažmentu riadenia úloh:

4.5.4.1 Identifikácia novej úlohy

Proces identifikácie novej úlohy slúži na zadefinovanie a vytvorenie rámca pre úlohu identifikovanú ako relevantnú pre riešenie v projekte.



Obrázok 4-9: Aktivity procesu identifikovania novej úlohy

Jednotlivé aktivity uvedeného procesu sú uvedené v nasledujúcich podkapitolách:

4.5.4.1.1 Opísanie úlohy

Vstup: *požiadavka na novú úlohu*

Výstup: *vytvorený opis úlohy týkajúci sa požiadaviek na funkcionality*

Zodpovedný: *projektový manažér, návrhár*

Cieľom aktivity opísania úlohy je počiatočné zadefinovanie funkcionality, ktorú má nová úloha obsahovať.

4.5.4.1.2 Určenie termínov

Vstup: *opis úlohy*

Výstup: *termínový rámec pre riešenie úlohy spolu s jej opisom*

Zodpovedný: *projektový manažér, návrhár*

Cieľom určenia termínov je vytvorenie termínového rámca v rámci ktorého je potrebné sa danou úlohou zaoberať.

4.5.4.1.3 Určenie priority úlohy

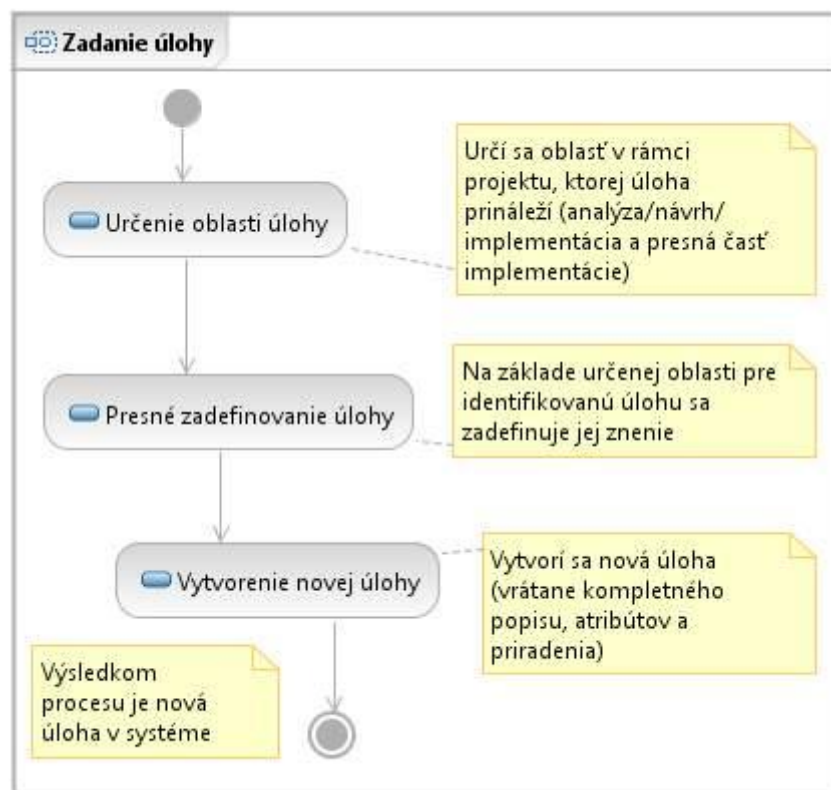
Vstup: *termínový rámeček pre riešenie úlohy spolu s jej opisom*

Výstup: *rámeček úlohy so zadanou prioritou úlohy, termínmi riešenia a opisom úlohy*

Zodpovedný: *projektový manažér, návrhár*

Cieľom aktivity určenia priority úlohy je určiť, či sa má daná úloha riešiť okamžite (napríklad z dôvodu nefunkčnosti vytváraného systému bez tejto úlohy), alebo či sa má odsunúť za vyriešenie prioritnejších úloh v rámci rovnakého časového rámca.

4.5.4.2 Zadanie úlohy



Obrázok 4-10: Aktivity procesu zadania úlohy

Jednotlivé aktivity uvedeného procesu sú uvedené v nasledujúcich podkapitolách:

4.5.4.2.1 Určenie oblasti úlohy

Vstup: *rámeček úlohy so zadanou prioritou úlohy, termínmi riešenia a opisom úlohy*

Výstup: *úloha zaradené do oblasti v rámci vývoja projektu*

Zodpovedný: *návrhár*

Cieľom aktivity je určenie o akú oblasť v rámci vývoja sa bude jednať. Môže ísť o oblasť návrhu, implementácie alebo o oblasť testovania.

4.5.4.2.2 Presné zadefinovanie úlohy

Vstup: *úloha zaradené do oblasti v rámci vývoja projektu*

Výstup: *presne zadefinovaná úloha*

Zodpovedný: *návrhár*

Presné zadefinovanie úlohy znamená, že úloha obsahuje všetky potrebné informácie po obsahovej stránke na to, aby ju bolo možné riešiť.

4.5.4.2.3 Vytvorenie novej úlohy

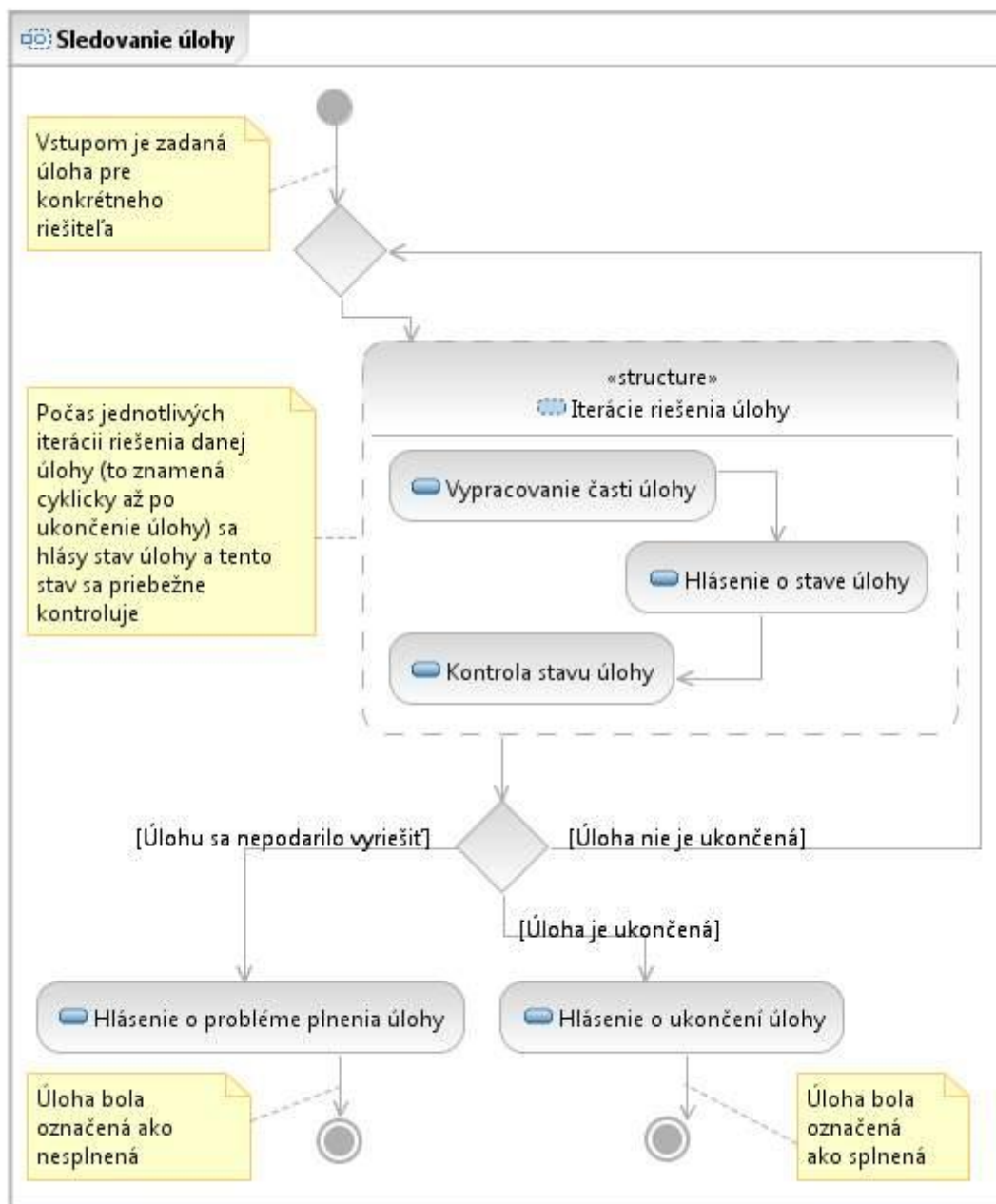
Vstup: *presne zadefinovaná úloha*

Výstup: *nová úloha*

Zodpovedný: *návrhár, projektový manažér*

Vytvorenie novej úlohy má za cieľ priradenie všetkých atribútov novej úlohe vrátane termínov a priradenie úlohy na riešenia konkrétnemu členovi projektového tímu.

4.5.4.3 Sledovanie úlohy



Obrázok 4-11: Aktivity procesu sledovania úlohy

Na obrázku **Chyba! Nenašiel sa žiaden zdroj odkazov.** je znázornený proces sledovania úlohy, ktorý zahŕňa aj jej riešenie. Riešenie úlohy, hlásenie o jej stave a kontrola jej stavu je robené v denných iteráciách. Po ukončení iterácie riešenia úlohy je výstupom úloha v stave *ukončená*, alebo v stave *neukončená*.

Jednotlivé aktivity uvedeného procesu sú uvedené v nasledujúcich podkapitolách:

4.5.4.3.1 Vypracovanie časti úlohy

Vstup: *nová úloha*

Výstup: *vypracovaná časť úlohy*

Zodpovedný: *návrhár, programátor*

Vypracovanie časti úlohy znamená, že je vypracovaná taká časť systému, ktorá do neho môže byť zaradená, môže byť otestovaná, alebo v prípade návrhu môže byť implementovaná.

4.5.4.3.2 Hlásenie o stave úlohy

Vstup: *vypracovaná časť úlohy*

Výstup: *úloha v stave ukončená / úloha v stave neukončená*

Zodpovedný: *návrhár, programátor*

Hlásenie o stave úlohy slúži na to aby bolo možné v rámci projektu sledovať plnenie úloh na báze denného sledovania. Pre to musí každý riešiteľ hlásiť stav úloh každý deň.

4.5.4.3.3 Kontrola stavu úlohy

Vstup: *hlásenie obsahujúce opis riešených častí a percento plnenia úlohy*

Výstup: *doplnenie hlásenia členom tímu zodpovedným za sledovanie danej úlohy*

Zodpovedný: *návrhár*

Kontrolu úloh má na starosti návrhár, ktorý navrhol riešenie pre túto úlohu. Tento bude sledovať jej vývoj až do jej ukončenia. Toto riešenie je použité pre to, aby sa zamedzilo zdržaniam kvôli chybám v návrhu.

4.5.4.3.4 Hlásenie o probléme plnenia úlohy

Vstup: *úloha v stave neukončená*

Výstup: *nesplnená úloha*

Zodpovedný: *návrhár, programátor*

Hlásenie slúži pre prípad, že danú úlohu sa nepodarilo splniť kvôli návrhovým, alebo implementačným problémom. Nesplnená úloha bude obsahovať v popise hlásenie o príčinách zamedzujúcich riešeniu úlohy.

4.5.4.3.5 Hlásenie o ukončení úlohy

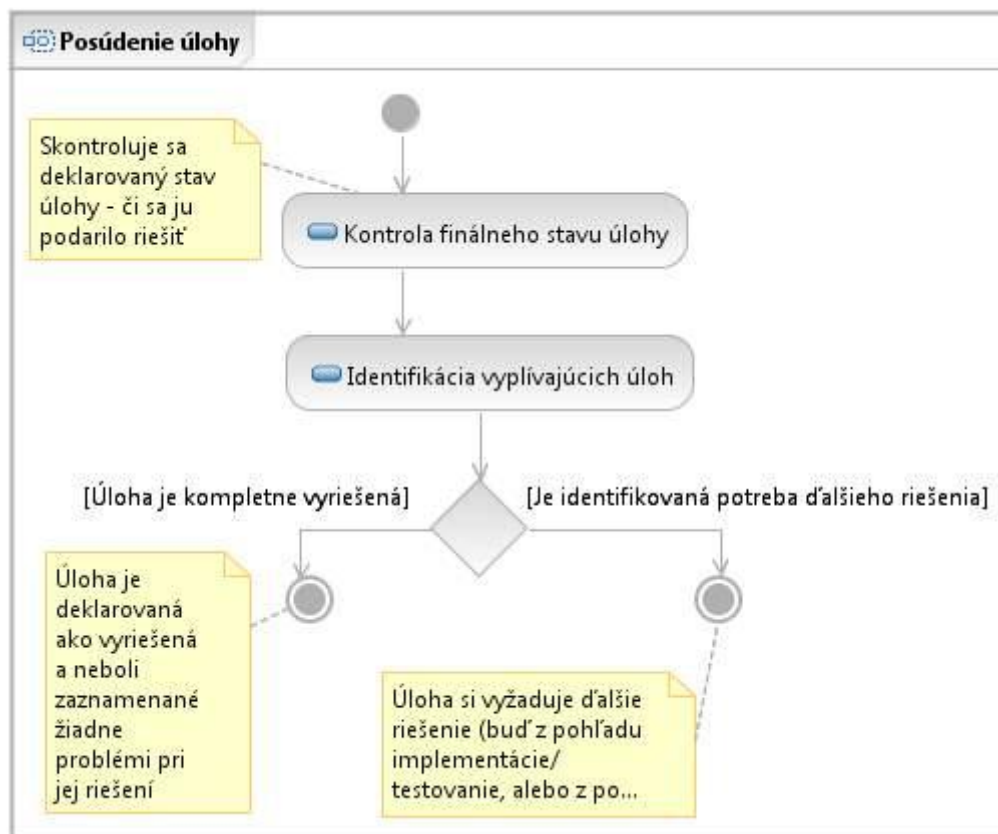
Vstup: *úloha v stave ukončená*

Výstup: *splnená úloha*

Zodpovedný: *návrhár, programátor*

Hlásenie slúži na zachytenie ukončenia úlohy.

4.5.4.4 Posúdenie úlohy



Obrázok 4-12: Aktivity procesu posúdenia úlohy

Jednotlivé aktivity uvedeného procesu sú uvedené v nasledujúcich podkapitolách:

4.5.4.4.1 Kontrola finálneho stavu úlohy

Vstup: *splnená úloha/ nesplnená úloha*

Výstup: *úloha so zmeneným stavom*

Zodpovedný: *návrhár, programátor*

Kontrola slúži na posúdenie správnosti výstupného stavu prideleného riešiteľom úlohy.

4.5.4.4.2 Identifikácia vyplivajúcich úloh

Vstup: *úloha so zmeneným stavom*

Výstup: *požiadavka na novú úlohu/ ukončenie úlohy*

Zodpovedný: *návrhár, programátor*

Výsledkom identifikácie úloh môže byť buď to, že daná úloha je ukončená a nenadväzujú na ňu žiadne ďalšie úlohy, alebo identifikáciu úloh nadväzujúcich na danú úlohu z pohľadu dosiahnutia požadovaného cieľa (požadovaná funkcionálnosť).

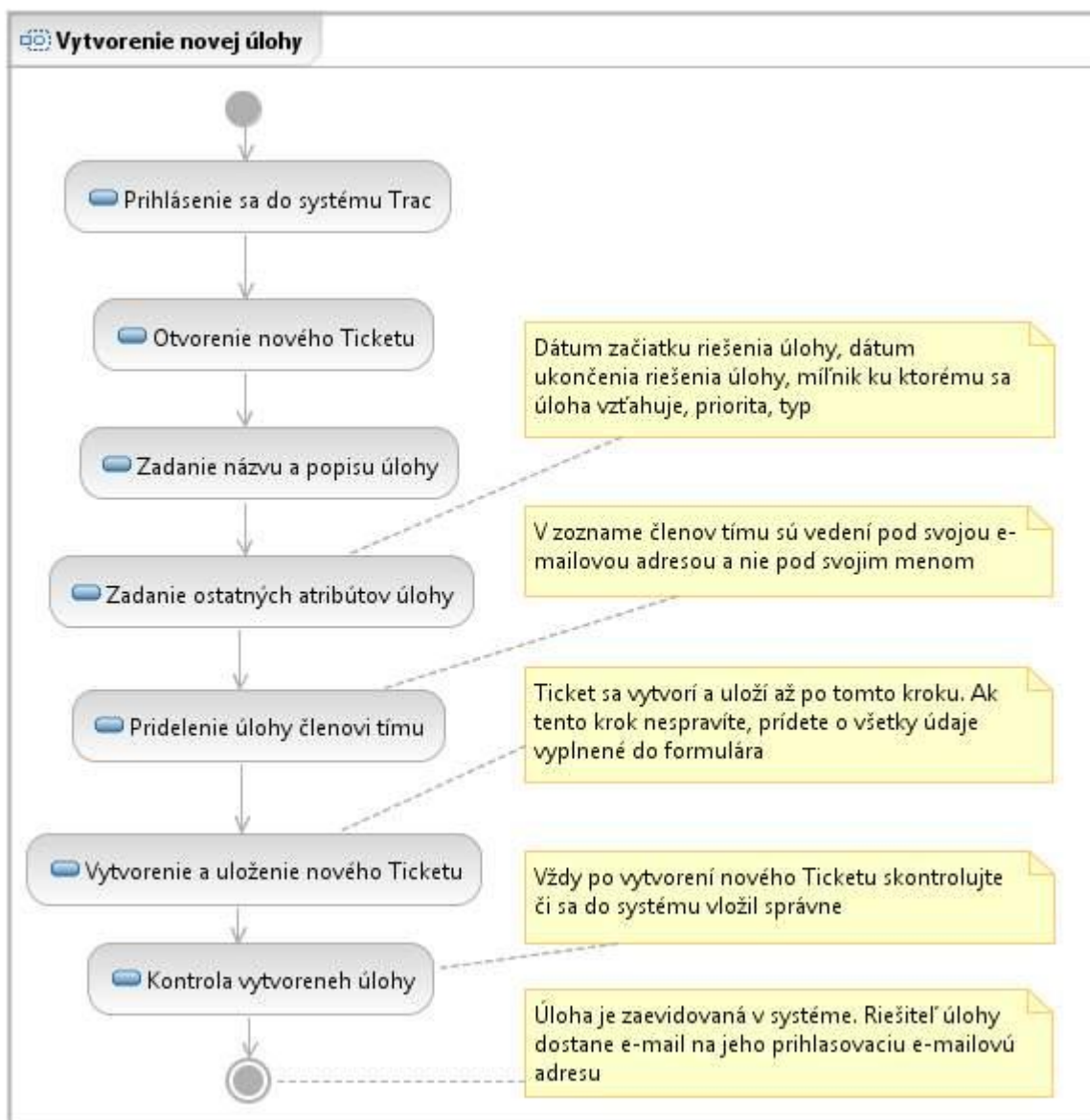
4.5.5 Podrobný opis konkrétneho procesu

Táto časť metodiky popisuje podrobne postup vytvorenia novej úlohy pre konkrétneho člena tímu.

Táto časť metodiky je určená pre tímových manažérov a pre návrhárov.

4.5.5.1 Vytvorenie novej úlohy

V tejto kapitole je podrobne rozpísaný proces vytvorenia novej úlohy. Priebeh procesu je znázornený na obrázku **Chyba! Nenašiel sa žiaden zdroj odkazov..**



Obrázok 4-13: Diagram aktivít vytvorenia úlohy

Úlohy sú v systéme Trac reprezentované ako „Ticket“. Pre to v používateľskom rozhraní nenájde bežne používaný pojem „Task“ pre označenie úlohy.

- 1) Prihláste sa do systému Trac. Prihlasovacie meno je vaša e-mailová adresa. Heslo prideliuje systémový administrátor.
- 2) V ponuke menu zobrazenom na obrázku **Chyba! Nenašiel sa žiaden zdroj odkazov.** kliknite na *New Ticket*



Obrázok 4-14: Ponuka menu v systéme Trac

Zobrazí sa okno s nasledujúcou ponukou tak ako je to znázornené na obrázku **Chyba! Nenašiel sa žiaden zdroj odkazov..**

- 3) Zadajte názov a popis úlohy do polí označených ako *A** na obrázku **Chyba! Nenašiel sa žiaden zdroj odkazov..**
 - a) Summary – názov novej úlohy
 - b) Reporter – člen tímu, ktorý bude sledovať aké zmeny na úlohe prebiehajú
 - c) Description – napíšte popis úlohy a jej číslo uvedené v zápisnici z tímového stretnutia
- 4) Zadajte ostatné atribúty úlohy do polí označených ako *B** na obrázku **Chyba! Nenašiel sa žiaden zdroj odkazov..**
 - a) Type – zvolte typ úlohy z nasledovných možností podľa uvedených kritérií:
 - i) task – úloha vytvára úplne novú časť aplikácie, alebo jej modulu, alebo pridáva funkcionality
 - ii) bug – úloha je zameraná na odstránenie chyby
 - iii) change request – úloha je zameraná na zmenu existujúcej funkcionality
 - b) Priority – zvolte prioritu úlohy z uvedených možností podľa popísaných kritérií:
 - i) critical – riešenie danej úlohy má najvyššiu prioritu, úlohu treba riešiť okamžite aj na úkor iných rozpracovaných úloh
 - ii) major – riešenie úlohy má normálnu prioritu, úloha musí byť vyriešená k uvedenému termínu uvedenom v *End*
 - iii) minor – riešenie úlohy má malú prioritu, úloha nemusí byť vyriešená k termínu uvedenému v položke *End*
 - c) Milestone – míľnik, ku ktorému má byť úloha vyriešená
 - d) Component – komponent aplikácie, ktorým sa úloha zaoberá – nemusí byť uvedená

- e) Version – číslo verzie úlohy, ak sa jedná o doplnenie inej úlohy typu *bug*, alebo *change request*
- f) Keywords – kľúčové slová popisujúce úlohu - neuvádzať
- g) Cc - neuvádzať
- h) Completed – percento plnenia nastavte na 0
- i) End – zadajte koncový dátum, ku ktorému má byť úloha najneskôr vyriešená. Dátum sa uvádza vo formáte *YYYY/MM/DD*, napríklad pre dátum 1.12.2010 to bude *2010/12/01*
- j) Start – zadajte dátum zadania úlohy. Dátum sa uvádza vo formáte *YYYY/MM/DD*, napríklad pre dátum 1.12.2010 to bude *2010/12/01*

5) *Pridel'te úlohu členovi tímu*

- a) Assign to – zo zoznamu označenom ako *C** na obrázku **Chyba! Nenašiel sa žiaden zdroj odkazov.** vyberte člena tímu, ktorému bude daná úloha priradená. Členovia tímu sú v zozname vedený pod svojou prihlasovacou e-mailovou adresou.

Create New Ticket

Properties

Summary:

Reporter:

Description: A*

B *I* **A**

B*

Type:

Milestone:

Version:

Cc:

End (YYYY/MM/DD):

Priority:

Component:

Keywords:

Completed [%]:

Start (YYYY/MM/DD):

Assign to: C*

I have files to attach to this ticket

D*

Obrázok 4-15: Pridanie novej úlohy do systému

6) Stlačte tlačidlo *Create ticket označenom ako D** na obrázku **Chyba! Nenašiel sa žiaden zdroj odkazov..**

7) Skontrolujte vytvorenú úlohu

Po vytvorení novej úlohy vždy skontrolujte či sú všetky jej parametre správne zadané nasledujúcim postupom:

- a) V ponuke menu zobrazenom na obrázku **Chyba! Nenašiel sa žiaden zdroj odkazov.** kliknite na *View Tickets*
- b) Z ponuky možných reportov vyberte *All Tickets By Milestone (Including closed)*, zobrazí sa výpis všetkých úloh zoradených podľa míľníkov
- c) Skontrolujte, či sa nová úloha nachádza v zozname reportu, či je zatriedená pod správnym míľníkom a či má vo výpise priradené požadované atribúty. Pri kontrole môžu nastať nasledujúce tri situácie:
 - i) Úloha sa nenachádza v zozname reportu – systém nevytvoril novú úlohu (napríklad kvôli prerušeniu internetového spojenia). V tomto prípade postupujte všetkými krokmi od 1. kroku, ktoré sú uvedené v kapitole 4.5.5.1
 - ii) Úloha nemá priradené správne atribúty – zadali ste nesprávne niektorý z atribútov. Postupujte nasledovne:
 - (1) Kliknite na danú úlohu a zobrazí sa formulár zobrazený na obrázku **Chyba! Nenašiel sa žiaden zdroj odkazov.**
 - (2) V zobrazenom formulári zmeňte požadovaný atribút zo sekcie *A**, *B**, alebo *C** vyznačenej na obrázku **Chyba! Nenašiel sa žiaden zdroj odkazov.**
 - (3) V zobrazenom formulári kliknite na tlačidlo „*Save Changes*“ (nie je uvedené na obrázku **Chyba! Nenašiel sa žiaden zdroj odkazov.**)

Add/Change #43 (Mapovanie atribútov)

Comment (you may use [WikiFormatting](#) here): D*

B I A [link] [img] [del] [list] [undo] [redo] [table]

Change Properties B*

Summary:

Reporter:

Description: **B I A** [link] [img] [del] [list] [undo] [redo] [table]

Analýza a návrh mapovania atribútov (mapovanie vlastností grafu na vizuálne vlastnosti).

Type: Priority:

Milestone: Component:

Version: Keywords:

Cc:

Completed [%]:

End (YYYY/MM/DD): Start (YYYY/MM/DD):

Action E*

leave as accepted

resolve as The resolution will be set. Next status will be 'closed'

reassign to The owner will change from jaro.prokop@gmail.com. Next status will be 'assigned'

accept Next status will be 'accepted'

Obrázok 4-16: Zmena atribútov existujúcej úlohy

iii) Do poľa *Comment* označenom ako D* na obrázku **Chyba! Nenašiel sa žiaden zdroj odkazov.** vyplňte popis zmeny danej úlohy. Popis má nasledujúci tvar:

Zmena úlohy na základe chybného atribútu: <chybne zvolený atribút 1; chybne zvolený atribút 2; ...; chybne zvolený atribút n >

Zmenu zapracoval: <vaše meno; e-mail>

iv) V podformulári *Change Properties* označenom ako B* na obrázku **Chyba! Nenašiel sa žiaden zdroj odkazov.** zmeňte chybné položky podľa pravidiel uvedených v kapitole 4.5.5.1 v kroku 3)

- d) Úloha sa nachádza v zozname reportu so správnymi atribútmi – úloha sa vytvorila správne, nie sú potrebné ďalšie úpravy

4.5.6 Použitá literatúra a zdroje

[1] The Trac Project

<http://trac.edgewall.org/> (15. November 2010)

4.6 Metodika písania dokumentácie zdrojových kódov a komentárov

Dokument obsahuje súhrn metodík, ktoré sa budú používať pri vytváraní textov, ktoré slúžia na dokumentovanie projektu alebo na jednoduchšiu orientáciu v diskusiách.

Cieľom bolo zjednotiť štýl písania textov, ktoré dokumentujú projekt (dokumentácia v zdrojovom kóde a komentáre pri posielaní súborov do repozitárov) a textov, ktoré tvoria nadpisy diskusií. Bola navrhnutá štruktúra a používané značky tak, aby sa zjednodušila orientácia v dokumentácii a diskusiách a aby navrhnutý spôsob písania týchto textov nebol príliš zložitý a nesťažoval samotné dokumentovanie.

V príkladoch budú povinné časti vyznačené hrubým písmom.

4.6.1 Štýl dokumentovania zdrojových kódov

Spôsob písania komentárov bude vychádzať z už používaného spôsobu predchádzajúceho tímu, ktorý na projekte pracoval.

Používa sa nástroj Doxygen¹ na generovanie dokumentácie, preto budú používané značky vychádzať z požiadaviek tohto nástroja.

4.6.1.1 Komentáre tried, funkcií a premenných – v hlavičkových (.h) súboroch

Časti kódov (triedy, funkcie), ktoré majú svoje rozhranie definované v hlavičkových (.h) a implementáciu v zdrojových (.cpp) súboroch, sa komentujú iba v hlavičkových súboroch.

Komentáre sa vyčleňujú spôsobom známym z *Javadoc* syntaxe:

```
/**
 * <obsah komentára>
 */
```

Komentár sa umiestňuje:

- pred komentovanú časť kódu
- bez vynechania riadku po komentári

Pre všetky komentáre v hlavičkových súboroch platia pravidlá:

- v komentári nie sú vynechávané riadky, ani na oddelenie častí komentára (výnimka: vytvorenie zlomu riadku vo vygenerovanej dokumentácii)
- maximálna dĺžka riadku komentára je 80 znakov

1 <http://www.doxygen.org/>

- musí byť dodržané poradie položiek uvedené v príkladoch

Komentáre na začiatku hlavičkových súborov sa nepoužívajú. Dôvod: informácie, ktoré sa zvyknú uvádzať v týchto komentároch, sa dajú vyčítať z iných miest:

- mená autorov, verzia, dátum vytvorenia, dátum zmeny – z dát repozitára
- účel súboru – je ekvivalentný s komentárom triedy, ktorá sa v súbore nachádza

4.6.1.1.1 Komentovanie tried

Komentár triedy vyzerá nasledovne:

```
/**
 * \brief <krátky komentár>
 * <značky>
 * <dlhý komentár>
 */
```

Vysvetlenia častí komentára sú uvedené v Tab. 1.

Tab. 1: Časti komentára triedy.

časť komentára	popis	povinné
krátky komentár	<ul style="list-style-type: none"> • význam triedy • zodpovednosti objektov triedy • veľké začiatkové písmeno, končí bodkou 	áno
značky	<ul style="list-style-type: none"> • postupnosť dohodnutých značiek (Tab. 2) 	nie
dlhý komentár	<ul style="list-style-type: none"> • poskytovaná funkcionálnosť • spôsob práce triedy • prípady použitia • príklady použitia 	nie

Tab. 2: Značky používané v komentároch tried.

značka	význam
[singleton]	v systéme existuje jedna inštancia triedy (splňa vzor <i>singleton</i>)
[immutable]	dáta objektu sa nedajú po vytvorení objektu meniť
[thread safe]	všetky funkcie je možné volať z viacerých vlákien
[data only]	trieda obsahuje iba dáta a operácie na prácu s nimi (pridávanie, vymazávanie, nastavovanie, získavanie); neobsahuje ďalšiu logiku
[operations]	trieda obsahuje operácie vykonávané nad niektorou [data only] triedou
[interface]	rozhranie – všetky funkcie sú <i>virtual</i> a abstraktné (= 0)
[abstract]	abstraktná trieda – aspoň jedna funkcia je <i>virtual</i> a abstraktná (= 0)
[obsolete]	trieda by sa nemala viac používať, bola nahradená inou funkcionálnosťou (dôvod a opis náhrady sa umiestňuje do časti <i>dlhý komentár</i>)

Príklad okomentovanej triedy:

```
/**
```

```

* \brief Skladanie textových reťazcov.
* [thread safe]
* Objekty reprezentujú textové reťazce, ku ktorým sa
* dajú pridávať ďalšie znaky.
* V konštruktori sa dá určiť počiatočná vyhradená pamäť.
* Funkciou append sa pridávajú ďalšie znaky.
*/
class StringCreator {
...

```

4.6.1.1.2 Komentovanie funkcií

Komentár funkcie:

```

/**
* \brief <krátky komentár>
* \param<smernik> <názov premennej> <komentár parametra>
* \param<smernik> <názov premennej> <komentár parametra>
* \return <komentár k návratovej hodnote>
* <značky>
* <dlhý komentár>
*/

```

Vysvetlenia častí komentára sú uvedené v Tab. 3.

Tab. 3: Časti komentára funkcie.

časť komentára	popis	povinné
krátky komentár	<ul style="list-style-type: none"> význam funkcie veľké začiatkové písmeno, končí bodkou 	áno
smernik	<ul style="list-style-type: none"> označenie parametra ako vstupného alebo výstupného (Tab. 5) 	nie
komentár parametra	<ul style="list-style-type: none"> význam parametra možné hodnoty komentáre musia byť usporiadané podľa poradia parametrov časť <i>komentár</i> začína veľkým začiatkovým písmenom, končí bodkou 	áno, pre každý parameter
komentár k návratovej hodnote	<ul style="list-style-type: none"> možné hodnoty a ich význam <i>komentár k návratovej hodnote</i> začína veľkým začiatkovým písmenom, končí bodkou 	áno, ak vracia hodnotu
značky	<ul style="list-style-type: none"> postupnosť dohodnutých značiek (Tab. 4) 	nie
dlhý komentár	<ul style="list-style-type: none"> poskytovaná funkcionálnosť spôsob práce funkcie prípady použitia príklady použitia 	nie

Tab. 4: Značky používané pri funkciách.

značka	význam
[thread safe]	funkcia môže byť volaná z viacerých vlákien
[getter]	funkcia slúži na nastavenie hodnoty členskej premennej

[setter]	funkcia slúži na získanie hodnoty členskej premennej
----------	------------------------------------------------------

Tab. 5: Označenie vstupných a výstupných parametrov.

označenie	význam
in	vstupný parameter
out	výstupný parameter
in,out	vstupno-výstupný parameter

Príklad okomentovanej funkcie:

```
/**
 * \brief Pridanie textu.
 * \param[in] str Text, ktorý sa má pridať.
 * \param[in] maxLength Maximálna dĺžka výsledného reťazca.
 * \return Aktuálna dĺžka reťazca po pridaní nového reťazca.
 * [thread safe]
 * Pridanie textu na koniec aktuálneho reťazca. Znaký sú
 * pridávané iba do dosiahnutia určenej maximálnej dĺžky,
 * ostatné znaky sú ignorované.
 */
int appendString (const std::string &str, int maxLength) {
...
}
```

Ak je funkcia členskou funkciou triedy a prekrýva virtuálnu funkciu v nadradenej triede, môžu nastať dva prípady:

1. chceme, aby funkcia zdedila celý komentár od pôvodnej funkcie v nadradenej triede; vtedy použijeme označenie, že funkciu netreba komentovať (označenie sa používa iba na to, aby bolo na prvý pohľad zrejmé, že sa komentár dedí):

```
/***/
2. chceme pridať iný komentár funkcii; vtedy funkciu nanovo okomentujeme
```

Ak funkcia slúži iba na získanie alebo nastavenie hodnoty členskej premennej triedy, používajú sa tieto pravidlá²:

1. komentáre týkajúce sa premennej sa píšú iba do komentára premennej
2. v komentári funkcie, ktorá premennú nastavuje alebo vracia, sa použije iba značka [getter] alebo [setter] tvoriaca odkaz na premennú

Premenná musí byť okomentovaná, aby sa správne vygeneroval odkaz.

Štruktúra komentára funkcie na získanie alebo nastavenie hodnoty:

```
/**
 * \brief \link <názov triedy>::<názov premennej> <značka> \endlink
 */
```

Príklad funkcie na nastavenie hodnoty:

```
/**
 * \brief \link User::age [setter] \endlink
 */
int setAge (const int age) {
...
}
```

Ak funkcia má aj iný účinok ako vrátenie alebo nastavenie hodnoty premennej, používa sa

² Tento spôsob sa vo vygenerovanej dokumentácii správne prejaví iba vtedy, ak je zapnuté zahrnutie *private* členov do dokumentácie.

klasická rozšírená forma komentára opísaná vyššie.

4.6.1.1.3 Komentovanie premenných

Premenné sa komentujú nasledovne (vysvetlenia sú v Tab. 6):

```
/**  
 * \brief <krátky komentár>  
 * <dlhý komentár>  
 */
```

Tab. 6: Časti komentára premennej.

časť komentára	popis	povinné
krátky komentár	<ul style="list-style-type: none">• význam premennej• veľké začiatkové písmeno, končí bodkou	áno
dlhý komentár	<ul style="list-style-type: none">• možné hodnoty a ich význam• prípady použitia• príklady použitia	nie

Príklad okomentovanej premennej:

```
/**  
 * \brief Autor príspevku.  
 * NULL, ak príspevok vytvoril anonymný používateľ.  
 */  
User *user;
```

4.6.1.2 Komentáre implementácie – v zdrojových (.cpp) súboroch

Komentáre umiestnené vo funkciách sa nevkladajú do vygenerovanej dokumentácie.

V implementácii funkcií (napríklad na opis menej zrozumiteľnej časti) sa použijú komentáre umiestnené pred opisovanou časťou alebo na konci riadku:

```
// <text komentára>
```

Na oddelenie častí funkcií sa môže používať komentár³:

```
// <názov časti> =====
```

4.6.2 Štýl komentárov pri vkladaní súborov do repozitárov

Komentáre budú pozostávať z riadkov, pričom každý bude mať nasledujúcu štruktúru:

```
[modul][značky] text
```

Každá významná zmena by mala byť opísaná textom v samostatnom riadku, pričom *modul* a *značky* slúžia iba na upresnenie a kategorizáciu zmeny v danom riadku.

Vysvetlenia:

- *modul* – vyskytuje sa, iba ak sa zmena týka konkrétneho modulu (modul je v tomto prípade názov časti na najvyššej úrovni v projekte – Core, Data, Layout...; ak sa zmena netýka žiadneho modulu, časť sa vynechá, aj so zátvorkami)
- *značky* – znaky označujúce kategórie, do ktorých zmena patrí – znaky nie sú oddelené medzerami ani inými oddeľovačmi (ak nie sú žiadne kategórie, časť sa vynechá, aj so

3 Pri používaní komentárov na oddelenie častí funkcií treba zväziť, či sa funkcia nedá rozdeliť na viac samostatných funkcií, ktoré zodpovedajú týmto identifikovaným častiam.

zátvorkami)

- text – text zmeny – začína malým písmenom, nekončí bodkou

Používané značky:

značka	opis
+	pridanie funkcionality
-	zmena funkcionality (správania sa)
i	zmena rozhrania modulu, ktoré môžu používať iné moduly
a	prispôsobenie sa zmenám rozhrania (knížnic alebo iných modulov)
r	refactoring – preusporiadanie kódu bez zmeny funkcionality
!	oprava chyby
t	zmena testov
f	zmena podporných súborov – resources, CMake súbory, knižnice (v dependencies)

Príklad jedného komentára:

```
[Data][ri] príprava na možnosť importu viacerých formátov  
[Data][+tf] pridanie importu formátu GraphML, testy importu, súbory na test  
importu  
[!f] oprava chyby v CMake súboroch
```

4.6.3 Štýl nadpisov v diskusiách

Pred textom nadpisu môže byť umiestnených niekoľko značiek, každá v hranatých zátvorkách:

značka	opis
organizačné	stretnutia, termíny...
problém	otázka ohľadom problému
riešenie	riešenie problému, ktoré by mohlo zaujímať ostatných
nápad	návrh člena tímu na prehodnotenie
podpora	podporné systémy a nástroje

4.7 Dokumentovanie zdrojových kódov

Pre dokumentovanie zdrojových kódov použijeme metodiku zavedenú minuloročným tímom, aby bola zachovaná konzistencia. Na generovanie dokumentácie sa používa nástroj Doxygen⁴, preto budú používané značky vychádzať z požiadaviek tohto nástroja. Podrobný opis metódy je uvedený v dokumentácii z minulého roka.

4 <http://www.doxygen.org/>

5 Zápisy zo stretnutí

5.1 Zápis zo stretnutia č. 1

Tím č. 16

Dátum: 28. 9. 2010

Miesto: D 003; FEI STU

Prítomní: Pedagóg: Ing. Peter Kapec
Členovia tímu: Bc. Tomáš Hurban, Bc. Milan Laslop, Bc. Jaroslav Prokop,
Bc. Jakub Susedík, Bc. Marek Švec

Zapísal: Bc. Milan Laslop

Overil: ...

Priebeh stretnutia

3. Ing. Peter Kapec oboznámil členov tímu s dosiahnutými výsledkami predchádzajúceho tímu, vysvetlil stav projektu, technológie a knižnice, ktorým sa máme venovať.
 - **dôležité nástroje / technológie / postupy:**
 - SVN / GIT (rozhodnúť, čo budeme používať)
 - XML parsing
 - C / C++
 - SQL (PostgreSQL)
 - niekto by mal mať projekt funkčný v Linuxe
 - treba pozrieť predchádzajúci projekt + dokumentáciu
 - **v projekte použité:**
 - „layoutovací“ algoritmus
 - uzly náhodne rozmiestnené v priestore
 - medzi nimi definované sily – príťažlivé / odpudivé
 - simulačný algoritmus: pôsobenie síl vedie k usporiadaniu uzlov
 - dátové štruktúry reprezentujúce grafy
 - načítané grafy uložené v databáze (teraz vypnuté, nepoužíva sa) – SQL
 - knižnica QT – používateľské rozhranie (menu, 1 základné okno, OpenGL context)
 - OpenSceneGraph
 - nadstavba OpenGL

- User guide – niekto by mal študovať
- použiteľné napr. na vizualizáciu mesta, krajiny
- pojem „graf scény” – graf (najčastejšie strom) obsahujúci vykresľované objekty, definované napr. posunutia (napr. koleso auta vykreslené viac krát na rôznych miestach)
- CMake
 - generátor projektov / makefiles
 - skriptovací jazyk – generuje VS projekt alebo linuxový makefile
- **mali by sme si rozdeliť úlohy, napr.:**
 - tvorba dokumentácie (zahŕňa aj web stránku)
 - vedúci tímu
 - podporné nástroje – kompilácia, príprava prostredia
 - programátori
 - hlavný architekt

Úlohy, ktoré vyplynuli zo stretnutia

(úlohy boli určené dodatočne 1. 10. 2010 – nie priamo na stretnutí)

ID	Úloha	Vypracovať do	Zodpovední
1.1	Študovanie dokumentácie vytvorenej tímom v predchádzajúcom roku.	5. 10. 2010	všetci
1.2	Naštudovanie architektúry - aká je vytvorená, aká je koncepcia spolupráce jednotlivých modulov, knižníc a pod.	5. 10. 2010	Marek
1.3	Naštudovanie údajových prostriedkov - databázy a pod.	5. 10. 2010	Jakub
1.4	Študovanie grafických a GUI knižníc (OpenSceneGraph, QT).	5. 10. 2010	Milan
1.5	Študovanie a testovanie podporných nástrojov (SVN, GIT).	5. 10. 2010	Tomáš
1.6	Tvorba tímovej stránky .	12. 10. 2010	Jaroslav

5.2 Zázpis zo stretnutia č. 2

Tím č. 16

Dátum: 5. 10. 2010

Miesto: D 003; FEI STU

Prítomní: Pedagóg: Ing. Peter Kapec

Členovia tímu: Bc. Tomáš Hurban, Bc. Milan Laslop, Bc. Jaroslav Prokop,
Bc. Jakub Susedík, Bc. Marek Švec

Zapísal: Bc. Milan Laslop

Overil: ...

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

ID	Stav	Splnené
1.1	Naštudovali sme si dokumentáciu predchádzajúceho tímu.	OK
1.2	Naštudovaná architektúra predchádzajúceho systému.	OK
1.3	Naštudované údajové prostriedky.	OK
1.4	Skúšanie knižníc QT a OpenSceneGraph na jednoduchých príkladoch (z návodov). Zistené základné princípy práce s knižnicami a zistenie ich použitia v projekte z dokumentácie.	OK
1.5	Zistenie rozdielov medzi SVN a GIT. Zvolili sme GIT. Bude založený repozitár na openhub.com.	OK
1.6	Tímovú stránku zatiaľ nebolo treba robiť.	

Priebeh stretnutia

1. Ing. Peter Kapec vysvetlil, čomu sa majú členovia tímu ďalej venovať, pričom ho členovia tímu informovali o aktuálnom stave.
 - **organizácia práce tímu:**
 - zdrojové kódy od predchádzajúceho tímu budú
 - web stránka tímu – musí sa dať urobiť statická kópia
 - vyskúšať podporné nástroje na prácu s GIT (napr. GIT extensions do VS)
 - zvoliť nástroj na manažment úloh a bugov (napr. Trac)
 - dokumentácia – používaný systém Doxygen – ujednotiť štýl dokumentovania
 - GIT – ujednotiť štýl písania komentárov, napr.:
 - značky + (pridané), ! (opravená chyba)
 - za značkou napr. názov modulu v zátvorkách
 - Google Groups – navrhnúť štýl písania nadpisov (napr. [web] [bug])

- **do projektu bude treba dorobiť:**
 - podpora iných XML formátov (napr. GXL) – teraz je len základné parsovanie GraphML
 - vymyslieť, ako reprezentovať vnorené grafy, hypergrafy
 - navrhnutie dátových štruktúr a vizualizácie, aby sa dal reprezentovať akýkoľvek GraphML súbor
 - mapovanie zložitejších grafov na graf scény – teraz obsahuje 2 hlavné vetvy (jednu pre všetky hrany a jednu pre všetky uzly)
 - využitie databázy – zatiaľ nie je využívaná
 - pridávanie metauzlov
- **analýza:**
 - knižnica Bullet
 - zistiť previazanie na OSG
 - fyzikálne simulácie, detekcia kolízií
 - analyzovať možnosť prepísania „layoutovacieho“ algoritmu pomocou tejto knižnice
 - problém: struny museli mať pevný bod (my ale potrebujeme 2 voľné konce)
 - formát GXL – analýza dátového modelu
- **príprava prostredia:**
 - aktualizovať zdrojové kódy na najvyššie verzie knižníc
 - QT – predkompilované je asi len pre MinGW – použitie aj pluginu na komunikáciu s PostgreSQL
 - OSG – použijeme poslednú development (nie stable) – použitie aj knižníc napr. na prácu s PNG
- **naštudovať nástroje:**
 - CMake
 - skripty generujúce VS projekty aj makefiles
 - 2 fázy – configure (hľadá knižnice), generate (generujú sa makefiles)
- **upozornenia:**
 - VS – OSG pomalé, keď skompilované v debug konfigurácii
 - vyhýbať sa adresárom s medzerami (adresáre s projektom, s knižnicami, s nástrojmi)
 - pozor na platformovo závislé typy – 32 / 64 bit

Úlohy, ktoré vyplynuli zo stretnutia

ID	Úloha	Vypracovať do	Zodpovední
----	-------	---------------	------------

1.6	Pripraviť tímovú stránku . Bude sa musieť dať urobiť statická kópia.	12. 10. 2010	Jaroslav
2.1	Naštudovať zdrojové kódy od predchádzajúceho tímu.	12. 10. 2010	všetci
2.2	Naštudovať podporné nástroje na prácu s GIT (napr. GIT extensions do VS).	12. 10. 2010	Marek
2.3	Zvoliť a pripraviť nástroj na manažment úloh a bugov (napr. Trac).	12. 10. 2010	Tomáš
2.4	Navrhnuť jednotný spôsob písania komentárov a nadpisov : - štýl dokumentovania kódu - štýl písania komentárov pri commitoch do GIT - štýl písania nadpisov v Google Groups Výstupom úlohy by mal byť dokument s navrhnutými štýlmi a značkami.	12. 10. 2010	Jaroslav
2.5	Vypracovanie časti analýzy – knižnica Bullet .	<i>(zatiaľ neurčené)</i>	<i>(zatiaľ neurčené)</i>
2.6	Vypracovanie časti analýzy – dátový model formátu GXL .	<i>(zatiaľ neurčené)</i>	<i>(zatiaľ neurčené)</i>
2.7	Nainštalovať najnovšie verzie knižníc , aktualizovať zdrojové kódy projektu na tieto verzie: - QT + plugin na komunikáciu s PostgreSL - OSG (posledná dev verzia) + plugin na PNG (možno aj iné pluginy)	12. 10. 2010	všetci
2.8	Naštudovať CMake .	12. 10. 2010	všetci
2.9	Vytvoriť účet na github.com (ostatní sa zaregistrujú).	12. 10. 2010	Tomáš
2.10	Tvorba plánu na prvý semester.	<i>(zatiaľ neurčené)</i>	Jaroslav
2.11	Naplánovanie úloh – čo bude treba dorobiť do projektu (poradie, priority).	<i>(zatiaľ neurčené)</i>	<i>(zatiaľ neurčené)</i>

5.3 Zázpis zo stretnutia č. 3

Tím č. 16

Dátum: 13. 10. 2010

Miesto: D 003; FEI STU

Prítomní: Pedagóg: Ing. Peter Kapec

Členovia tímu: Bc. Tomáš Hurban, Bc. Milan Laslop, Bc. Jaroslav Prokop,
Bc. Jakub Susedík

Zapísal: Bc. Jaroslav Prokop

Overil: Bc. Tomáš Hurban

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

ID	Stav	Zodpovední	Vypracovať do	Splnené
1.6	Tímová stránka spojzdená	J.P	12.10.2010	Áno
2.1	Zdrojové kódy zatiaľ našťudoval len Milan, ostatní členovia tímu pracujú na spojzdení prostredia na Windows	M.L.	12.10.2010	Áno
2.2	Marek sa nezúčastnil stretnutia ani nedal žiadnu informáciu	M.S.	12.10.2010	Nie
2.3	Tomáš navrhol použiť priamo nástroj v Git	T.H.	12.10.2010	Áno
2.4	Milan dal metodiku do Google Groups	M.L.	12.10.2010	Áno
2.7	Nainštalovali sme si najnovšie knižnice QT a OSG	všetci	12.10.2010	Áno
2.8	Naštudovali a odskúšali sme CMake	T.H., M.L., J.S.	12.10.2010	Áno
2.9	Tomáš vytvoril účet na GitHubu	T.H.	12.10.2010	Áno

1. Skontrolovali sme stav úloh z predošlých stretnutí
 - Milan rozhodil prostredie a aplikáciu v Linuxe
 - Jaro publikoval web stránku tímu
 - Tomáš a Jakub nainštalovali najnovšie knižnice a našťudovali CMake
2. Vedúci tímu Ing. Peter Kapec upozornil na potrebu spojzdenia debugovania vo Windows
3. Vedúci tímu Ing. Peter Kapec vzniesol nasledovné požiadavky na písanie zázpisnice:
 - Pri zodpovedných používať iniciálky mien
 - K úlohám uvádzať dátum splnenia
 - K úlohám zadávať dátum zadania

- K úlohám uvádzať stav plnenia
- 4. Ukázali sme vedúcemu tímu Ing. Petrovi Kapcovi stav spojzdenia prostredia vo Windows. Narážali sme na problém pri kompilácii v CMake. Vedúci tímu nám ukázal ako pracuje CMake. Nastavili sme všetky potrebné cesty pre jednotlivé knižnice. Počas stretnutia sa nám nepodarilo spojzdníť prostredie.
 - Členom tímu pracujúcim v operačnom prostredí Windows spôsobovala chybu knižnica OSG
- 5. Vedúci tímového projektu prezentoval svoju víziu o našej práci na projekte. Mala by sa týkať grafickej, databázovej časti a časti venovanej parsovaniu zdrojových kódov.
- 6. Rozdelili sme si úlohy ktoré máme splniť do nasledovného stretnutia.

Úlohy, ktoré vyplynuli zo stretnutia

ID	Úloha	Zadané	Vypracovať do	Zodpovední
2.5	Vypracovanie časti analýzy – knižnica Bullet.	28.9.2010	26.10.2010	J.S.
2.6	Vypracovanie časti analýzy – dátový model formátu GXL.	28.9.2010	26.10.2010	M.L.
2.10	Tvorba plánu na prvý semester.	6.10.2010	26.10.2010	J.P.
2.11	Naplánovanie úloh – čo bude treba dorobiť do projektu (poradie, priority).	6.10.2010	19.10.2010	
3.1	Rozchodenie prostredia, odhalenie príčiny chýb	13.10.2010	19.10.2010	M.S., T.H., J.S.
3.2	Spísať zmeny spôsobené inštaláciou novších knižníc a dopad na aplikáciu	13.10.2010	19.10.2010	M.L.
3.3	Overiť funkcionality meta uzlov a vrcholov	13.10.2010	19.10.2010	T.H., M.L.
3.4	Overenie možností nastavení v aplikácii (options)	13.10.2010	19.10.2010	J.S., M.S
3.5	Naštudovať spôsob zobrazovania v aplikácii, možnosti zmeny vykresľovania vrcholov a uzlov	13.10.2010	19.10.2010	M.L., T.H
3.6	Analyzovať použité technológie, vypracovať zoznam technológií ktoré by sa ešte dali použiť. Tento zoznam bude základom pre úlohy na ktorých budeme v rámci systému pracovať	13.10.2010	19.10.2010	M.S.
3.7	Vybrať a nainštalovať nástroj na sledovanie úloh	13.10.2010	19.10.2010	J.P.
3.8	Vložiť do plánu a aj na stránku základné míľniky tímového projektu	13.10.2010	19.10.2010	J.P

5.4 Zázpis zo stretnutia č. 4

Tím č. 16

Dátum: 19. 10. 2010

Miesto: D 003; FEI STU

Prítomní: Pedagóg: Ing. Peter Kapec

Členovia tímu: Bc. Tomáš Hurban, Bc. Milan Laslop, Bc. Jaroslav Prokop,
Bc. Jakub Susedík, Bc. Marek Švec

Zapísal: Bc. Jakub Susedík

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

ID	Úloha	Stav
2.5	Vypracovanie časti analýzy – knižnica Bullet	<i>Rozpracované</i>
2.6	Vypracovanie časti analýzy – dátový model formátu GXL	<i>Rozpracované</i>
2.10	Tvorba plánu prvý semester	<i>Rozpracované</i>
2.11	Naplánovanie úloh – čo bude treba dorobiť do projektu (poradie, priority)	<i>Rozpracované</i>
3.1	Rozchodenie prostredia, odhalenie príčiny chýb	<i>Rozpracované</i>
3.2	Spísať zmeny spôsobené inštaláciou novších knižníc a dopad na aplikáciu	<i>OK</i>
3.3	Overiť funkcionality metauzlov a vrcholov	<i>Rozpracované</i>
3.4	Overenie nastavení v aplikácií (options)	<i>Rozpracované</i>
3.5	Naštudovať možnosti zobrazovania v aplikácií, možnosti zmeny vykresľovania vrcholov a uzlov	<i>Rozpracované</i>
3.6	Analyzovať použité technológie, vypracovať zoznam technológií, ktoré by sa ešte dali použiť, tento zoznam bude základom pre úlohy na ktorých budeme v rámci systému pracovať	<i>Rozpracované</i>
3.7	Vybrať a nainštalovať nástroj na sledovanie úloh	<i>OK</i>
3.8	Vložiť do plánu a aj na stránku základné míľniky tímového projektu	<i>OK</i>

Priebeh stretnutia:

1. Ing. Petra Kapca sme informovali o aktuálnom stave úloh z predchádzajúceho týždňa a o stave spojzdenia projektu vo Windows.
2. Ing. Peter Kapec určil čomu by sme sa mali ďalej venovať
 - Preskúmať v dokumentácií RSF formát
 - Aplikácia by mala podporovať viacero vstupných formátov, nezávisle na vstupe, výstup by mal byť v jednom spoločnom formáte
 - Vymyslieť dátové štruktúry pre GraphML a GXL
 - Zanalyzovať ako sa majú vykresľovať vrcholy, či sa každý vrchol bude vykresľovať sám, alebo bude špeciálne funkcia vykresľovať všetky vrcholy
 - Spísať osnovu dokumentácie
 - Analyzovať možnosti parsovania veľkých XML súborov, ak sú dáta väčšie ako dostupná pamäť
 - Analyzovať ako dosiahnuť rozklikávanie uzlov v aplikácií, čo už je v systéme implementované aby sa to dalo vykonávať
3. Do analýzy za zimný semester by sme mali zahrnúť
 - Analýza existujúceho systému, čo je v systéme a ako spravené, čo v systéme chýba
 - Ako spracovávať multihrany, viac hrán smerujúcich na jeden vrchol
 - Ako OSG podporuje vizualizáciu rozsiahlych dát, ak sú dáta väčšie ako dostupná pamäť

Úlohy, ktoré vyplynuli zo stretnutia:

ID	Úloha	Zadané	Vypracovať do	Zodpovední	Stav
2.5	Vypracovanie časti analýzy – knižnica Bullet	28.9.2010	26.10.2010	J. S.	
2.6	Vypracovanie časti analýzy – dátový model formátu GXL	28.9.2010	26.10.2010	M.L.	
2.10	Tvorba plánu prvý semester	6.10.2010	26.10.2010	J.P.	
3.1	Rozchodenie prostredia, odhalenie príčiny chýb	6.10.2010	26.10.2010	M.S., T.H., J.S.	
3.3	Overiť	6.10.2010	26.10.2010	T.H., M.L.	

	funkcionalitu metauzlov a vrcholov				
3.4	Overenie nastavení v aplikácií (options)	6.10.2010	26.10.2010	J.S., M.S.	
4.1	Zistiť čo všetko sa ukladá do databázy	19.10.2010	26.10.2010	J.S.	
4.2	Vypracovanie časti analýzy - analýza existujúceho programu, čo je ako spravené	19.10.2010	26.10.2010	T.H.	
4.3	Vypracovanie časti analýzy – čo chýba v systéme	19.10.2010	26.10.2010	M.L.	
4.4	Analyzovať ako by sa dali robiť multihrany, viac hrán na jeden vrchol	19.10.2010	26.10.2010	J.P.	
4.5	Spísať osnovu dokumentácie	19.10.2010	26.10.2010	J.P.	
4.6	Analyzovať parsovanie veľkých xml súborov	19.10.2010	26.10.2010	M.S.	
4.7	Analyzovať ako osg podporuje vizualizáciu rozsiahlych dát	19.10.2010	26.10.2010	M.S.	
4.8	Analyzovať ako dosiahnuť rozklikávanie uzlov, čo už je v systéme implementované	19.10.2010	26.10.2010	T.H.	
4.9	Vymyslieť dátové štruktúry pre GraphML a	19.10.2010			

	GXL				
--	-----	--	--	--	--

5.5 Zázpis zo stretnutia č. 5

Tím č. 16

Dátum: 26. 10. 2010

Miesto: D 003 – Softvérové štúdio; FEI STU

Prítomní: Pedagóg: Ing. Peter Kapec

Členovia tímu: Bc. Tomáš Hurban, Bc. Milan Laslop, Bc. Jaroslav Prokop,
Bc. Jakub Susedík, Bc. Marek Švec

Zapísal: Bc. Tomáš Hurban

Overil: Bc. Jaroslav Prokop

ID	Úloha	Stav	Dátum dokončenia	Zodpovední
2.5	Vypracovanie časti analýzy – knižnica Bullet	<i>Rozpracované</i>		<i>J.S.</i>
2.6	Vypracovanie časti analýzy – dátový model formátu GXL	<i>OK</i>	<i>26. 10. 2010</i>	<i>M.L.</i>
2.10	Tvorba plánu na prvý semester	<i>Rozpracované</i>		<i>J.P.</i>
3.1	Rozchodenie prostredia, odhalenie príčiny chýb	<i>OK</i>	<i>26. 10. 2010</i>	<i>M.S., T.H., J.S.</i>
3.3	Overiť funkcionality metauzlov a vrcholov	<i>OK</i>	<i>26. 10. 2010</i>	<i>T.H., M.L.</i>
3.4	Overenie nastavení v aplikácii (options)	<i>OK</i>	<i>26. 10. 2010</i>	<i>J.S., M.S.</i>
4.1	Zistiť čo všetko sa ukladá do databázy	<i>Rozpracované</i>		<i>J.S.</i>
4.2	Vypracovanie časti analýzy - analýza existujúceho programu, čo je ako spravené	<i>Rozpracované</i>		<i>T.H.</i>
4.3	Vypracovanie časti analýzy – čo chýba v systéme	<i>Rozpracované</i>		<i>M.L.</i>
4.4	Analyzovať ako by sa dali robiť multihrany, viac hrán na jeden vrchol	<i>Rozpracované</i>		<i>J.P.</i>
4.5	Spísať osnovu dokumentácie	<i>OK</i>	<i>26. 10. 2010</i>	<i>J.P.</i>
4.6	Analyzovať parsovanie veľkých xml súborov	<i>Rozpracované</i>		<i>M.S.</i>

4.7	Analyzovať ako osg podporuje vizualizáciu rozsiahlych dát	<i>Rozpracované</i>		<i>M.S.</i>
4.8	Analyzovať ako dosiahnuť rozklikávanie uzlov, čo už je v systéme implementované	<i>Rozpracované</i>		<i>T.H.</i>
4.9	Vymyslieť dátové štruktúry pre GraphML a GXL	<i>Rozpracované</i>		<i>M.L.</i>

Tab. 1. Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

Priebeh stretnutia:

4. Ing. Petra Kapca sme informovali o aktuálnom stave úloh z predchádzajúceho týždňa. Jednotlivé úlohy sme preddiskutovali a podľa potrieb doplnili.
5. Ing. Peter Kapec určil čomu by sme sa mali ďalej venovať:
 - jazyk Xpath (XML Path Language)
 - skompilovanie a spustenie projektu v Debug mode
 - funkcionalita metauzlov – zmena hĺbky metauzlov v závislosti od vzdialenosti kamery
 - dopracovanie jednotlivých častí analýzy
 - vnorené grafy, hypergrafy, multihrany, vnoreným grafom v hranách
 - pohyb kamery v 3D priestore v závislosti od snímania tváre používateľa
 - layout do roviny, gule (napr. pre vnorené grafy), ...
 - knižnice na parovanie XML
 - rozmiestnenie uzlov pri načítaní grafu – explózia uzlov vs náhodné rozmiestnenie uzlov v priestore
6. Pozreli sme si existujúce riešenia – Skyrails
7. Prediskutovali sme činnosť Mareka Šveca v tímovom projekte, ktorú vyhodnotil manažér tímu ako nedostatočnú.
8. Venovali sme sa projektu, diskutovali sme o jednotlivých funkciách a vlastnostiach programu, riešení chýb v Debug mode

Úlohy ktoré vyplynuli zo stretnutia:

ID	Úloha	Vypracovať do	Zodpovední
2.5	Vypracovanie analýzy – knižnica Bullet	<i>2.11.2010</i>	<i>J.S.</i>
2.10	Tvorba plánu na prvý semester	<i>14.12.2010</i>	<i>J.P.</i>
4.1	Zistiť, čo všetko sa ukladá do databázy	<i>2.11.2010</i>	<i>J.S.</i>
4.2	Vypracovanie analýzy - analýza existujúceho	<i>2.11.2010</i>	<i>T.H.</i>

	programu, čo je ako spravené		
4.3	Vypracovanie analýzy – čo chýba v systéme	2.11.2010	M.L.
4.4	Analyzovať ako by sa dali robiť multihrany, viac hrán na jeden vrchol	2.11.2010	J.P.
4.6	Analyzovať parsovanie veľkých xml súborov	2.11.2010	M.S.
4.7	Analyzovať ako osg podporuje vizualizáciu rozsiahlych dát	2.11.2010	M.S.
4.8	Analyzovať ako dosiahnuť rozklikávanie uzlov	2.11.2010	T.H.
4.9	Analýza dátových štruktúr pre GraphML a GXL formát	2.11.2010	M.L.
5.1	Analyzovať a naštudovať XPath	2.11.2010	M.S.
5.2	Analýza knižníc na parsovanie	2.11.2010	M.S.
5.3	Rozchodiť projekt v Debug mode	2.11.2010	T.H., J.S.
5.4	Analýza podpory rôznych typov grafov v projekte, vnorené grafy	2.11.2010	M.L., J.P.
5.5	Analýza snímania tváre používateľa pre pohyb kamery v programe	2.11.2010	J.S.
5.6	Analýza iných možností layoutovania – layout do roviny, gule	2.11.2010	T.H.
5.7	Analýza rozmiestnenia uzlov v priestore pri načítaní grafu	2.11.2010	M.S.
5.8	Analýza nižšej a vyššej úrovne grafov a uzlov	2.11.2010	M.S.
5.9	Analýza a opis problémovej oblasti	2.11.2010	J.P.
5.10	Vypracovanie návrhu – Parser dátového formátu GXL	2.11.2010	M.L.
5.11	Vypracovanie návrhu – Vyhľadávanie a filtrovanie v databáze	2.11.2010	J.S.
5.12	Vypracovanie návrhu – Zobrazovanie multihrán	2.11.2010	T.H.
5.13	Vypracovanie dokumentácie k riadeniu	2.11.2010	J.P.

Tab. 2. Úlohy, ktoré vyplynuli zo stretnutia

5.6 Zázpis zo stretnutia č. 6

Tím č. 16

Dátum: 3. 11. 2010

Miesto: D 003 – Softvérové štúdio; FEI STU

Prítomní: Pedagóg: Ing. Peter Kapec

Členovia tímu: Bc. Tomáš Hurban, Bc. Milan Laslop, Bc. Jakub Susedík

Zapísal: Bc. Milan Laslop

Overil:

ID	Úloha	Stav	Dátum dokončenia	Zodpovední
2.5	Vypracovanie analýzy – knižnica Bullet	OK	2.11.2010	J.S.
2.10	Tvorba plánu na prvý semester	rozpracované	14.12.2010	J.P.
4.1	Zistiť, čo všetko sa ukladá do databázy	OK	2.11.2010	J.S.
4.2	Vypracovanie analýzy - analýza existujúceho programu, čo je ako spravené	OK	2.11.2010	T.H.
4.3	Vypracovanie analýzy – čo chýba v systéme	OK	2.11.2010	M.L.
4.4	Analyzovať ako by sa dali robiť multihraný, viac hrán na jeden vrchol	OK	2.11.2010	J.P.
4.6	Analyzovať parsovanie veľkých xml súborov	chýba	2.11.2010	M.S.
4.7	Analyzovať ako osg podporuje vizualizáciu rozsiahlych dát	chýba	2.11.2010	M.S.
4.8	Analyzovať ako dosiahnuť rozklikávanie uzlov	OK, pridané 6.3	2.11.2010	T.H.
4.9	Analýza dátových štruktúr pre GraphML a GXL formát	OK	2.11.2010	M.L.
5.1	Analyzovať a naštudovať XPath	chýba	2.11.2010	M.S.
5.2	Analýza knižníc na parsovanie	chýba	2.11.2010	M.S.
5.3	Rozchodiť projekt v Debug mode	OK	2.11.2010	T.H., J.S.
5.4	Analýza podpory rôznych typov grafov v projekte, vnorené grafy	OK, pridané 6.1	2.11.2010	M.L., J.P.
5.5	Analýza snímania tváre používateľa pre pohyb kamery v programe	OK	2.11.2010	J.S.
5.6	Analýza iných možností layoutovania – layout do roviny, gule	OK, pridané 6.2	2.11.2010	T.H.
5.7	Analýza rozmiestnenia uzlov v priestore pri načítaní grafu	chýba	2.11.2010	M.S.
5.8	Analýza nižšej a vyššej úrovne grafov a uzlov	chýba	2.11.2010	M.S.
5.9	Analýza a opis problémovej oblasti	OK	2.11.2010	J.P.
5.10	Vypracovanie návrhu – Parser dátového formátu GXL	OK	2.11.2010	M.L.
5.11	Vypracovanie návrhu – Vyhľadávanie a filtrovanie v databáze	OK, pridané 6.8	2.11.2010	J.S.
5.12	Vypracovanie návrhu – Zobrazovanie multihrán	OK	2.11.2010	T.H.
5.13	Vypracovanie dokumentácie k riadeniu	OK	2.11.2010	J.P.

Tab. 1. Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

Priebeh stretnutia:

- Ing. Peter Kapec prebral prvú časť dokumentácie (analýza a návrh) a dokumentácie k riadeniu
- prešli sme stav vypracovania aktuálnych úloh
 - knižnica Bullet – ešte budeme uvažovať, či ju je vhodné použiť
 - analýza databázy – vo Windows bol problém so skompilovaním postgresql pluginu – problém sa vyriešil počas stretnutia
- treba sa zamyslieť nad modulárnym layoutovacím algoritmom
 - „layoutovacia časť“ a „obmedzovacia časť“ (obmedzenie do gule, kocky...)
 - vznikne zameniteľný layoutovací algoritmus, nezávislý od časti, ktorá v priestore rozloženie obmedzuje (vznikne layoutovacia časť pracujúca iba v rovine, priestore obmedzenom kruhom...)
 - analyzovať súvis s metauzlami a inou používateľskou interakciou
- návrh vyhľadávania a filtrovania v databáze – rozšírenie úlohy
 - ako riešiť dopyt na rozbalenie uzla – problémy: nové uzly môžu už byť načítané (ak napr. načítavame susedov uzla), nový uzol môže byť hranami pospájaný s už načítanými
- rozklikávanie uzlov – rozšírenie úlohy
 - grafické zobrazenie, že existujú ešte ďalšie susedné uzly, ktoré ale nie sú ešte zobrazené
- nová úloha – analyzovať mapovanie atribútov
 - mapovanie vlastností grafu na vizuálne vlastnosti
 - napr. veľkosť uzla podľa počtu susedov; ale aj atribúty grafu, súvisiace s oblasťou, kde sa graf používa
- Ing. Peter Kapec spomenul nástroj TopoLayout
 - vizualizačný nástroj
 - modulárny
 - obsahuje layoutovací algoritmus
 - študovanie zdrojových kódov by nám v niečom mohlo pomôcť, napríklad pri návrhu architektúry
- preberali sme možnosti testovania
 - unit testy – analyzovať knižnice na ich podporu
 - analyzovať nástroje na detekciu memory leaks

ID	Úloha	Vypracovať do	Zodpovední
2.10	Tvorba plánu na prvý semester	14.12.2010	J.P.
4.6	Analyzovať parsovanie veľkých xml súborov	2.11.2010	M.S.
4.7	Analyzovať ako osg podporuje vizualizáciu rozsiahlych dát	2.11.2010	M.S.
5.1	Analyzovať a naštudovať XPath	2.11.2010	M.S.
5.2	Analýza knižníc na parsovanie	2.11.2010	M.S.
5.7	Analýza rozmiestnenia uzlov v priestore pri načítaní grafu	2.11.2010	M.S.
5.8	Analýza nižšej a vyššej úrovne grafov a uzlov	2.11.2010	M.S.

6.1	Zistiť, čo presne bude treba zmeniť v moduloch aplikácie (Data, databáza, Viewer...), ak chceme podporovať vnorené grafy, multihrany a hypergrafy.	9.11.2010	J.S.
6.2	Analýza modulárneho layoutovacieho algoritmu (napríklad aj oddelená „layoutovacia“ a „obmedzovacia“ časť).	9.11.2010	T.H.
6.3	Navrhnuť grafické zobrazenie, že existujú ďalšie susedné uzly, ktoré ale nie sú zobrazené.	9.11.2010	T.H.
6.4	Analýza a návrh mapovania atribútov (mapovanie vlastností grafu na vizuálne vlastnosti).	16.11.2010	J.P.
6.5	Analyzovať knižnice na podporu unit testov.	9.11.2010	M.L.
6.6.	Analyzovať spôsoby a nástroje na detekciu memory leaks (Windows, Visual Studio).	9.11.2010	J.S.
6.7.	Analyzovať spôsoby a nástroje na detekciu memory leaks (Linux).	9.11.2010	M.L.
6.8	Vyhľadávanie a filtrovanie v databáze – navrhnuť, ako riešiť identifikované problémy.	9.11.2010	J.S.
6.9	Pozrieť projekt TopoLayout, čo by sa dalo použiť (napríklad spôsob návrhu architektúry).	neurčené	neurčené

Tab. 2. Úlohy, ktoré vyplynuli zo stretnutia

5.7 Zázpis zo stretnutia č. 7

Tím č. 16

Dátum: 9. 11. 2010

Miesto: D 003 – Softvérové štúdio; FEI STU

Prítomní: Pedagóg: Ing. Peter Kapec

Členovia tímu: Bc. Tomáš Hurban, Bc. Milan Laslop, Bc. Jakub Susedík

Zapísal: Bc. Tomáš Hurban

Overil: Bc. Milan Laslop

ID	Úloha	Stav	Dátum dokončenia	Zodpovední
2.10	Tvorba plánu na prvý semester	Rozpracované		J.P.
4.6	Analýza parsovania veľkých xml súborov	Rozpracované		M.S.
4.7	Analýza ako osg podporuje vizualizáciu rozsiahlych dát	Rozpracované		M.S.
5.1	Analýza a naštudovanie XPath	Rozpracované		M.S.
5.2	Analýza knižníc na parsovanie	Spojená s úlohou 4.6		M.S.
5.7	Analýza rozmiestnenia uzlov v priestore pri načítaní grafu	Rozpracované		M.S.
5.8	Analýza nižšej a vyššej úrovne grafov a uzlov	Rozpracované		M.S.
6.1	Zistiť, čo presne bude treba zmeniť v moduloch aplikácie (Data, databáza, Viewer...), ak chceme podporovať vnorené grafy, multihypry a hypergrafy.	Rozpracované		J.S.
6.2	Analýza modulárneho layoutovacieho algoritmu (napríklad aj oddelená „layoutovacia“ a „obmedzovacia“ časť).	Rozpracované		T.H.
6.3	Navrhnuť grafické zobrazenie, že existujú ďalšie susedné uzly, ktoré ale nie sú zobrazené.	Rozpracované		T.H.
6.4	Analýza a návrh mapovania atribútov (mapovanie vlastností grafu na vizuálne vlastnosti).	Rozpracované		J.P.
6.5	Analýza knižnice na podporu unit testov.	OK	9.11.2010	M.L.
6.6	Analýza spôsobov a nástrojov na detekciu memory leaks (Windows, Visual Studio).	OK	9.11.2010	J.S.
6.7	Analýza spôsobov a nástrojov na detekciu memory leaks (Linux).	OK	9.11.2010	M.L.
6.8	Vyhľadávanie a filtrovanie v databáze – navrhnuť, ako riešiť identifikované problémy.	Rozpracované		J.S.
6.9	Pozrieť projekt TopoLayout, čo by sa dalo použiť (napríklad spôsob návrhu architektúry).	Neurčené		Neurčené

Tab. 1. Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

Priebeh stretnutia:

- Ing. Peter Kapec nás informoval o stave odovzdanej dokumentácie. Zhodnotil, že je potrebné dokumentáciu poopravovať, najmä časť venovanú riadeniu.

10. Keďže jeden člen tímu – Marek Švec – sa nám už dlhší čas neozval, rozdelili sme jeho úlohy medzi zvyšných členov tímu.
11. Ing. Petra Kapca sme informovali o aktuálnom stave úloh z predchádzajúceho stretnutia.
 - Na základe analýzy spôsobov a nástrojov na detekciu memory leaks (úlohy 6.6 a 6.7) sme sa rozhodli, že tieto nástroje používať nebudeme.
12. Venovali sme sa rôznym možnostiam rozmiestnenia uzlov.
13. Ing. Peter Kapec spomenul plugin OpenOrg, ktorý slúži na rýchle rozmiestnenie uzlov v priestore.
14. Ing. Peter Kapec spomenul, že by sme sa mali začať venovať prototypu, ktorý je potrebné odovzdať na konci semestra. Diskutovali sme o rôznych chýbajúcich častiach systému, ktorým by sme sa tento semester mali venovať a implementovať ich.
 - pohyb kamery v priestore v závislosti od pohybu tváre používateľa (na tomto sme sa rozhodli pracovať priebežne)
 - práca s databázou
 - implementácia parsera RSF formátu

Dohodli sme sa, že tento semester sa budeme venovať najmä implementácií a začleneniu parsera RSF formátu do projektu.
15. Diskutovali sme o modularizovaní kódu.
16. Ing. Peter Kapec spomenul, že v používateľskom rozhraní aplikácie by bolo vhodné doplniť ovládanie pre pripojenie k databáze a výber jednotlivých grafov uložených v databáze.
17. Venovali sme sa rôznym typom grafov (multigrafy, hypergrafy, vnorené grafy) a možnostiam ich implementácie. Taktiež sme sa venovali vhodnej štruktúre databázy na ukladanie týchto grafov.
18. V závere stretnutia sme sa venovali formátu GXL a jeho modelu.

ID	Úloha	Vypracovať do	Zodpovední
2.10	Tvorba plánu na prvý semester	14.12.2010	J.P.
4.6	Analyzovať parsovanie veľkých xml súborov	16.11.2010	J.P.
4.7	Analyzovať ako osg podporuje vizualizáciu rozsiahlych dát	16.11.2010	M.L.
5.1	Analyzovať a naštudovať XPath	16.11.2010	J.S.
5.7	Analýza rozmiestnenia uzlov v priestore pri načítaní grafu	16.11.2010	T.H.
5.8	Analýza nižšej a vyššej úrovne grafov a uzlov	16.11.2010	M.L.
6.1	Zistiť, čo presne bude treba zmeniť v moduloch aplikácie (Data, databáza, Viewer...), ak chceme podporovať vnorené grafy, multihrany a hypergrafy.	16.11.2010	J.S.
6.2	Analýza modulárneho layoutovacieho algoritmu (napríklad aj oddelená „layoutovacia“ a „obmedzovacia“ časť).	16.11.2010	T.H.
6.3	Navrhnuť grafické zobrazenie, že existujú ďalšie susedné uzly, ktoré ale nie sú zobrazené.	16.11.2010	T.H.

6.4	Analýza a návrh mapovania atribútov (mapovanie vlastností grafu na vizuálne vlastnosti).	16.11.2010	J.P.
6.8	Vyhľadávanie a filtrovanie v databáze – navrhnúť, ako riešiť identifikované problémy.	16.11.2010	J.S.
6.9	Pozrieť projekt TopoLayout, čo by sa dalo použiť (napríklad spôsob návrhu architektúry).	Neurčené	Neurčené
7.1	Analyzovať plugin OpenOrd – zistiť, či by sa dal využiť v aplikácii	16.11.2010	T.H.
7.2	Návrh parsera RSF formátu	16.11.2010	J.S.
7.3	Preštudovať modulárnosť parserov (abstraktná trieda, dedenie)	16.11.2010	J.P.
7.4	Navrhnúť štruktúru databázy	16.11.2010	M.L.
7.5	Návrh doplnenia rozhrania aplikácie pre pripojenie a výber grafu z databázy používateľom	16.11.2010	T.H.

Tab. 2. Úlohy, ktoré vyplynuli zo stretnutia

5.8 Zázpis zo stretnutia č. 8

Tím č. 16

Dátum: 19. 10. 2010

Miesto: D 003; FEI STU

Prítomní: Pedagóg: Ing. Peter Kapec

Členovia tímu: Bc. Tomáš Hurban, Bc. Milan Laslop, Bc. Jaroslav Prokop,
Bc. Jakub Susedík

Zapísal: Bc. Jakub Susedík

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

D	Úloha	Stav	Dátum dokončenia	Zodpovední
.10	Tvorba plánu na prvý semester		14.1 2.2010	J.P.
.6	Analýza parsovania veľkých xml súborov	rozpracované	16.1 1.2010	J.P.
.7	Analýza ako osg podporuje vizualizáciu rozsiahlych dát	Rozpracované	16.1 1.2010	M.L.
.1	Analýza a nastudovanie XPath	Rozpracované	16.1 1.2010	J.S.
.7	Analýza rozmiestnenia uzlov v priestore pri načítaní grafu	Ok	16.1 1.2010	T.H.
.8	Analýza nižšej a vyššej úrovne grafov a uzlov	Ok	16.1 1.2010	M.L.
.1	Zistiť, čo presne bude treba zmeniť v moduloch aplikácie (Data, databáza, Viewer...), ak chceme podporovať vnorené grafy, multihrany a hypergrafy.	rozpracované	16.1 1.2010	J.S.
.2	Analýza modulárneho layoutovacieho algoritmu (napríklad aj	Rozpracované	16.1 1.2010	T.H.

	oddelená „layoutovacia” a „obmedzovacia” časť).			
.3	Navrhnuť grafické zobrazenie, že existujú ďalšie susedné uzly, ktoré ale nie sú zobrazené.	Ok	16.1 1.2010	T.H.
.4	Analýza a návrh mapovania atribútov (mapovanie vlastností grafu na vizuálne vlastnosti).	Ok	16.1 1.2010	J.P.
.8	Vyhľadávanie a filtrovanie v databáze – navrhnuť, ako riešiť identifikované problémy.	Rozpr acované	16.1 1.2010	J.S.
.9	Pozrieť projekt TopoLayout, čo by sa dalo použiť (napríklad spôsob návrhu architektúry).		Neu rčené	Neur čené
.1	Analyzovať plugin OpenOrd – zistiť, či by sa dal využiť v aplikácii	Ok	16.1 1.2010	T.H.
.2	Návrh parsera RSF formátu	Rozpr acované	16.1 1.2010	J.S.
.3	Preštudovať modulárnosť parserov (abstraktná trieda, dedenie)	Rozpr acované	16.1 1.2010	J.P.
.4	Navrhnuť štruktúru databázy	Rozpr acované	16.1 1.2010	M.L.
.5	Návrh doplnenia rozhrania aplikácie pre pripojenie a výber grafu z databázy používateľom	rozpra cované	16.1 1.2010	T.H.

Priebeh stretnutia:

- Ing. Petra Kapca sme informovali o aktuálnom stave úloh z predchádzajúceho stretnutia
- Vedúci Ing. Peter Kapec nám odporučil si pozrieť dokumentáciu predminuloročného tímu, na ktorého prácu nadväzujeme
- Ing. Peter Kapec spomenul, že by sme sa mali začať venovať implementácií
- Mali by sme vyčleniť a modularizovať parser od graph manageru
- Implementovať RSF parser

- Identifikovať problémy, prečo sa aplikácia nechce pripojiť k databáze a následne implementovať ukladanie dát o grafe do databázy a vyberanie dát z databázy
- Implementovať multihrany

Úlohy, ktoré vyplynuli zo stretnutia

D	Úloha	Vypracovať do	Zodpovední
.6	Analyzovať parsovanie veľkých xml súborov	23.11 .2010	J.P.
.7	Analyzovať ako osg podporuje vizualizáciu rozsiahlych dát	23.11 .2010	M.L.
.1	Analyzovať a naštudovať XPath	23.11 .2010	J.S.
.2	Analýza modulárneho layoutovacieho algoritmu (napríklad aj oddelená „layoutovacia“ a „obmedzovacia“ časť).	23.11 .2010	T.H.
.2	Návrh a implementácia parsera RSF formátu	23.11 .2010	J.S.
.3	Preštudovať modulárnosť parserov (abstraktná trieda, dedenie)	23.11 .2010	J.P.
.4	Navrhnuť štruktúru databázy	23.11 .2010	M.L.
.5	Návrh doplnenia rozhrania aplikácie pre pripojenie a výber grafu z databázy používateľom	23.11 .2010	T.H.
.1	Oddeliť parser	23.11 .2010	M.L.
.2	Implementácia multihrán	23.11 .2010	J.P.
.3	Implementovať ukladanie a vyberanie dát z databázy	23.11 .2010	T.H.

5.9 Zázpis zo stretnutia č. 9

Tím č. 16

Dátum: 23. 11. 2010

Miesto: D 003; FEI STU

Prítomní: Pedagóg: Ing. Peter Kapec

Členovia tímu: Bc. Tomáš Hurban, Bc. Milan Laslop, Bc. Jaroslav Prokop,
Bc. Jakub Susedík

Zapísal: Bc. Jaroslav Prokop

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

D	Úloha	Stav	Dátum dokončenia	Zodpovední
.6	Analyzovať parsovanie veľkých xml súborov	Ok	23.1 1.2010	J.P.
.7	Analyzovať ako osg podporuje vizualizáciu rozsiahlych dát	Ok	23.1 1.2010	M.L.
.1	Analyzovať a naštudovať XPath	Rozpracované	23.1 1.2010	J.S.
.2	Analýza modulárneho layoutovacieho algoritmu (napríklad aj oddelená „layoutovacia“ a „obmedzovacia“ časť).	Rozpracované	23.1 1.2010	T.H.
.2	Návrh a implementácia parsera RSF formátu	Rozpracované	23.1 1.2010	J.S.
.3	Preštudovať modularnosť parserov (abstraktná trieda, dedenie)	Ok	23.1 1.2010	J.P.
.4	Navrhnuť štruktúru databázy	Ok	23.1 1.2010	M.L.
.5	Návrh doplnenia rozhrania aplikácie pre pripojenie a výber grafu z databázy používateľom	Rozpracované	23.1 1.2010	T.H.

.1	Oddeliť parser	Ok	23.1 1.2010	M.L
.2	Implementácia multihrán	Rozpr acované	23.1 1.2010	J.P.
.3	Implementovať ukladanie a vyberanie dát z databázy	Rozpr acované	23.1 1.2010	T.H.

Priebeh stretnutia:

- Ing. Petra Kapca sme informovali o aktuálnom stave úloh z predchádzajúceho stretnutia
- Prediskutovali sme s Ing. Petrom Kapcom možnosti zobrazenia multihrán a dopad na celkovú dátovú štruktúru.
- Rozobrali sme podrobnosti verzionovania údajov v systéme Git.

Úlohy, ktoré vyplynuli zo stretnutia

D	Úloha	Vypr acovať do	Zodpo vední
.1	Analyzovať a naštudovať XPath	29.11 .2010	J.S.
.2	Analýza modulárneho layoutovacieho algoritmu (napríklad aj oddelená „layoutovacia“ a „obmedzovacia“ časť).	29.11 .2010	M.L.
.2	Návrh a implementácia parsera RSF formátu	29.11 .2010	J.S.
.5	Návrh doplnenia rozhrania aplikácie pre pripojenie a výber grafu z databázy používateľom	29.11 .2010	T.H.
.2	Implementácia multihrán	29.11 .2010	J.P.
.3	Implementovať ukladanie a vyberanie dát z databázy	29.11 .2010	T.H.

5.10 Zázpis zo stretnutia č. 10

Tím č. 16

Dátum: 30. 11. 2010

Miesto: D 003 – Softvérové štúdio; FEI STU

Prítomní: Pedagóg: Ing. Peter Kapec

Členovia tímu: Bc. Tomáš Hurban, Bc. Milan Laslop, Bc. Jaroslav Prokop

Zapísal: Bc. Milan Laslop

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

D	Úloha	Stav	Dátu m dokončenia	Zodp ovední
.1	Analyzovať a naštudovať XPath	Rozpra cované	29.1 1.2010	J.S.
.2	Analýza modulárneho layoutovacieho algoritmu (napríklad aj oddelená „layoutovacia“ a „obmedzovacia“ časť).	OK	29.1 1.2010	M.L.
.2	Návrh a implementácia parsera RSF formátu	Rozpra cované	29.1 1.2010	J.S.
.5	Návrh doplnenia rozhrania aplikácie pre pripojenie a výber grafu z databázy používateľom	OK	29.1 1.2010	T.H.
.2	Implementácia multihrán	OK	29.1 1.2010	J.P.
.3	Implementovať ukladanie a vyberanie dát z databázy	Rozpra cované	29.1 1.2010	T.H.

Priebeh stretnutia:

- preberanie možností ukladania grafov do databázy
 - úloha je rozpracovaná
 - možnosti ukladania ID do databázy: auto increment / určované v programe

- problém: každý objekt musí mať ID, vo vstupných súboroch sú aj dáta, ktoré nemajú
- preberanie odlišnosti databázy od súborov
 - databáza poskytuje rozšírené možnosti dopytovania, preto nie je vhodné zjednotiť súbory a databázu ako úložiská dát s rovnakým rozhraním
- diskusia o dopytovaní databázy – napr. možnosti výberu susedných uzlov v hypergrafe
- diskutovanie príkladu: reprezentácia hypergrafov pomocou pomocných uzlov a hrán
- diskusia o vetvách projektu – napr. rozšírenie možností layoutovača iným študentom v rámci diplomového projektu
- diskusia o mapovaní atribútov na vizuálne vlastnosti

Úlohy, ktoré vyplynuli zo stretnutia

D	Úloha	Vypracovat' do	Zodpovední
.1	Analyzovať a naštudovať XPath	7.12.201 0	J.S.
.2	Návrh a implementácia parsera RSF formátu	7.12.201 0	J.S.
.3	Implementovať ukladanie a vyberanie dát z databázy	7.12.201 0	T.H.
0.1	Analyzovať načítavanie XML súborov pomocou triedy QDomStreamReader v QT.	7.12.201 0	M.L.
0.2	Implementovať parser GXL podporujúci jednoduché grafy (najlepšie s použitím QDomStreamReader).	7.12.201 0	M.L.
0.3	Analyzovať, ako sa dajú oddeliť dáta od vizualizácie (triedy v module Data).	7.12.201 0	J.S.
0.4	Analyzovať, ako reprezentovať zložité typy grafov (zjednotiť zatiaľ nájdené spôsoby a nápady).	7.12.201 0	J.P.
0.5	Dokumentácia – doplniť špecifikáciu (funkcionálne a nefunkcionálne požiadavky, use cases).	7.12.201 0	M.L.
	Dokumentácia – opis architektúry	7.12.201	T.H.

0.6	predchádzajúceho systému. Doplnenie, čo je zle navrhnuté, prípadne čo sa porušuje v kóde.	0	
0.7	Dokumentácia – upraviť návrh modulu Importer (premenovať, doplniť detaily o parseroch).	0	7.12.201 M.L.
0.8	Dokumentácia – návrh – práca s multihranami.	0	7.12.201 J.P.
0.9	Dokumentácia – návrh – práca s databázou.	0	7.12.201 T.H.
0.10	Dokumentácia – úprava formátovania, štylizácií.	0	7.12.201 J.P.
0.11	Dokumentácia – analýza dátového modelu (analýzu databázy prerobiť).	0	7.12.201 T.H.
0.12	Dokumentácia – návrh – reprezentácia zložitých typov grafov (v pamäti, databáze).	0	7.12.201 J.P.

5.11 Zázpis zo stretnutia č. 11

Tím č. 16

Dátum: 07. 11. 2010

Miesto: D 003 – Softvérové štúdio; FEI STU

Prítomní: Pedagóg: Ing. Peter Kapec

Členovia tímu: Bc. Tomáš Hurban, Bc. Milan Laslop, Bc. Jaroslav Prokop,

Zapísal: Bc. Tomáš Hurban

D	Úloha	Stav	Dátu m dokončenia	Zodp ovední
.1	Analyzovať a naštudovať XPath	Zruše né	-----	J.S.
.2	Návrh a implementácia parsera RSF formátu	Zruše né	-----	J.S.
.3	Implementovať ukladanie a vyberanie dát z databázy	OK	7. 12. 2010	T.H.
0.1	Analyzovať načítavanie XML súborov pomocou triedy QXMLStreamReader v QT.	OK	7. 12. 2010	M.L.
0.2	Implementovať parser GXL podporujúci jednoduché grafy (najlepšie s použitím QXMLStreamReader).	OK	7. 12. 2010	M.L.
0.3	Analyzovať, ako sa dajú oddeliť dáta od vizualizácie (triedy v module Data).	Zruše né	-----	J.S.
0.4	Analyzovať, ako reprezentovať zložité typy grafov (zjednotiť zatiaľ nájdené spôsoby a nápady).	OK	7. 12. 2010	J.P.
0.5	Dokumentácia – doplniť špecifikáciu (funkcionálne a nefunkcionálne požiadavky, use cases).	OK	7. 12. 2010	M.L.

0.6	Dokumentácia – opis architektúry predchádzajúceho systému . Doplnenie, čo je zle navrhnuté, prípadne čo sa porušuje v kóde.	OK	7. 12. 2010	T.H.
0.7	Dokumentácia – upraviť návrh modulu Importer (premenovať, doplniť detaily o parseroch).	OK	7. 12. 2010	M.L.
0.8	Dokumentácia – návrh – práca s multihranami .	OK	7. 12. 2010	J.P.
0.9	Dokumentácia – návrh – práca s databázou .	OK	7. 12. 2010	T.H.
0.10	Dokumentácia – úprava formátovania, štylizácií .	OK	7. 12. 2010	J.P.
0.11	Dokumentácia – analýza dátového modelu (analýzu databázy prerobiť).	OK	7. 12. 2010	T.H.
0.12	Dokumentácia – návrh – reprezentácia zložitých typov grafov (v pamäti, databáze).	OK	7. 12. 2010	J.P.

Tab. 1. Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

Priebeh stretnutia:

19. je potrebné sa dohodnúť s tímom č. 9 sa čase prezentácií projektov
 - treba pripraviť prezentáciu
 - v prezentácii treba spomenúť (minimálne): z čoho sme vychádzali, čo sme spravili, úlohy kolegov, ktoré zostali nesplnené
20. taktiež je treba upraviť dokumentáciu do finálnej formy
 - v dokumentácii k projektu: najmä oddeliť časti analýza a návrh
 - v dokumentácii k riadeniu: pridať metodiky, upraviť zápisnice, štyly
21. venovali sme sa štýlom písania kódu, komentárov pri implementácii a o vhodnosti používania výnimiek
22. tiež sme sa venovali tomu, čo už je implementované a čo ešte chýba pre prácu s databázou
 - chýba pridávanie metaузлов a metahrán do databázy
 - ukladanie pozícií uzlov
 - ukladanie pozície kamery do databázy (dlhodobý cieľ)

D	Úloha	Vypr acovať do	Zodp ovední
1.1	Opraviť chyby v odovzdaných častiach dokumentácie	14.1 2.2010	T.H., J.P., M.L.
1.2	Skompletizovať dokumentáciu	14.1 2.2010	J.P.
1.3	Pripraviť prezentáciu	14.1 2.2010	M.L.
1.4	Implementovať chýbajúce časti pre prácu s databázou (ukladanie pozícií, ukladanie metauzlov a metahrán)	14.1 2.2010	T.H.

Tab. 2. Úlohy, ktoré vyplynuli zo stretnutia

5.12 Zápis zo stretnutia č. 12

Tím č. 16

Dátum: 17. 2. 2011

Miesto: D; FIIT STU

Prítomní: Členovia tímu: Bc. Tomáš Hurban, Bc. Milan Laslop, Bc.

Jaroslav Prokop, Bc. Matúš Juhas

Zapísal: Bc. Milan Laslop

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

D	Úloha	Stav	Dátu m dokončenia	Zodpo vední
1.1	Opraviť chyby v odovzdaných častiach dokumentácie	OK	14.1 2.2010	T.H., J.P., M.L.
1.2	Skompletizovať dokumentáciu	OK	14.1 2.2010	J.P.
	Pripraviť prezentáciu	OK	14.1	M.L.

1.3			2.2010	
1.4	Implementovať chýbajúce časti pre prácu s databázou (ukladanie pozícií, ukladanie metauzlov a metahrán)	Rozpracované	14.1 2.2010	T.H.

Priebeh stretnutia:

- zhrnutie aktuálneho stavu a úloh, ktoré treba dokončiť:
 - **podpora hypergrafov a vnorených grafov**
 - môže byť použitý špeciálny typ hrany/uzla v dátovom modeli
 - **podpora databázy**
 - treba skontrolovať, či sa správne ukladajú a načítavajú typy (napr. používateľské uzly a multihrany) a pozície uzlov
 - **parsery**
 - dokončiť podporu hypergrafov a vnorených grafov do existujúcich parserov (GraphML a GXL) – keď bude podpora v dátovom modeli, alebo aspoň navrhnuté rozhranie
 - vytvoriť nové parsery – RSF a GraphXML
- diskusia o rozhraní dátového modelu, pomocou ktorého bude parser posielať informácie o načítanom grafe:
 - **hyperhrany**
 - metóda na poslanie informácie o hyperhrane (vytvorí sa uzol špeciálneho typu)
 - metóda na poslanie „konca“ hyperhrany, ktorý sa pošle ako hrana spájajúca špeciálny uzol s iným existujúcim
 - **vnorené grafy**
 - uzly a hrany vnorených grafov sa budú posielat' existujúcim spôsobom
 - treba zabezpečiť unikátne ID uzlov a hrán – ak nie sú unikátne ID vyžadované už vstupnými formátmi, treba pridávať prefix (napr. názov vnoreného grafu)
 - nové metódy na pripojenie vnoreného grafu k hrane alebo k uzlu (vstupom je hrana alebo uzol, ku ktorej sa vnorený graf pripája, a prvý uzol vnoreného grafu)
- zhrnutie možných nasledujúcich dlhodobých úloh:
 - mapovanie vlastností grafu na vizuálne vlastnosti
 - zobrazenie vnorených grafov
 - označenie uzla alebo hrany, že má vnorený graf – zatiaľ napevno, neskôr možno pomocou mapovania vlastností
 - prechod na zobrazenie vnoreného grafu

Úlohy, ktoré vyplynuli zo stretnutia

D	Úloha	Vypracovať do	Zodpovední
1.4	Implementovať chýbajúce časti pre prácu s databázou (ukladanie pozícií , ukladanie metauzlov a metahrán) –	23.2.2011 1	T.H.

	možno len skontrolovať, či sa správne ukladajú a načítavajú typy (napr. používateľské uzly a multihrany) a pozície uzlov.		
2.1	Pridať podporu hypergrafov a vnorených grafov v dátovom modeli .	2.3.2011	J.P.
2.2	Dokončiť podporu hypergrafov a vnorených grafov do existujúcich parserov (GraphML a GXL).	2.3.2011	M.L.
2.3	Skompilovať knižnice a projekt.	23.2.2011 1	M.J., M.M.
2.4	Implementovať jednoduchý RSF parser (iba jednoduché grafy).	2.3.2011	M.J.
2.5	Implementovať jednoduchý GraphXML parser (iba jednoduché grafy).	2.3.2011	M.M.

5.13 Zápís zo stretnutia č. 13

Tím č. 16

Dátum: 23. 2. 2011

Miesto: D; FIIT STU

Prítomní: Pedagóg: Ing. Peter Kapec

Členovia tímu: Bc. Tomáš Hurban, Bc. Milan Laslop, Bc. Jaroslav Prokop,

Bc. Matúš Juhas, Bc. Michal Masliš

Zapísal: Bc. Tomáš Hurban

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

D	Úloha	Stav	Dátum dokončenia	Zodpovední
1.4	Implementovať chýbajúce časti pre prácu s databázou (ukladanie pozícií , ukladanie metauzlov a metahrán) – možno len skontrolovať, či sa správne ukladajú a načítavajú typy (napr. používateľské uzly a multihrany) a pozície uzlov.	Rozpracované		T.H.
	Pridať podporu hypergrafov a	Rozpra		J.P.

2.1	vnorených grafov v dátovom modeli.	cované		
2.2	Dokončiť podporu hypergrafov a vnorených grafov do existujúcich parserov (GraphML a GXL).	Rozpra cované		M.L.
2.3	Skompilovať knižnice a projekt.	OK	23. 2. 2011	M.J., M.M.
2.4	Implementovať jednoduchý RSF parser (iba jednoduché grafy).	Rozpra cované		M.J.
2.5	Implementovať jednoduchý GraphXML parser (iba jednoduché grafy).	Odlože né		M.M.

Priebeh stretnutia:

- spustenie projektu u nových členov tímu
- preberanie, čo treba ešte dorobiť
 - podpora nových typov grafov – hypergrafy, vnorené grafy (pomocou už existujúcich prvkov – uzol, hrana)
 - dokončiť funkcie pre podporu databázy (v budúcnosti zvážiť použitie ORM)
 - opraviť ukladanie a načítanie pozícií uzlov
 - ukladanie pozície kamery
 - ukladanie viacerých layoutov
 - RSF parser
 - podpora zložitejších grafov (hypergrafy, vnorené grafy) v už implementovaných parseroch
- nová úloha – priestorové obmedzovanie layoutovacieho algoritmu
 - obmedzenia do gule, roviny, štvorca

Úlohy, ktoré vyplynuli zo stretnutia

D	Úloha	Vypracovať do	Zodpovední
1.4	Implementovať chýbajúce časti pre prácu s databázou (ukladanie pozícií , ukladanie metauzlov a metahrán , layoutov , pozície kamery)	2.3.2011	T.H.
	Implementácia podpory vnorených	2.3.2011	J.P.

2.1	grafov		
2.2	Dokončiť podporu hypergrafov a vnorených grafov do existujúcich parserov (GraphML a GXL)	2.3.2011	M.L.
2.4	Implementovať RSF parser	2.3.2011	M.J.
3.1	Implementácia podpory hyperhrán	2.3.2011	M.M.
3.2	Implementácia obmedzovača layoutovacieho algoritmu	9.3.2011	M.L., T.H.

5.14 Zápis zo stretnutia č. 14

Tím č. 16

Dátum: 3. 3. 2011

Miesto: D; FIIT STU

Prítomní: Pedagóg: Ing. Peter Kapec

Členovia tímu: Bc. Tomáš Hurban, Bc. Milan Laslop, Bc. Jaroslav Prokop,
Bc. Matúš Juhas, Bc. Michal Masliš

Zapísal: Bc. Jaroslav Prokop

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

D	Úloha	Stav	Dátum dokončenia	Zodpovední
1.4	Implementovať chýbajúce časti pre prácu s databázou (ukladanie pozícií , ukladanie metauzlov a metahrán , layoutov , pozície kamery)	Rozpracované		T.H.
2.1	Implementácia podpory vnorených grafov	Rozpracované		J.P.
2.2	Dokončiť podporu hypergrafov a vnorených grafov do existujúcich parserov (GraphML a GXL)	Rozpracované		M.L.
2.4	Implementovať RSF parser	Rozpracované		M.J.
3.1	Implementácia podpory hyperhrán	Rozpracované		M.M.

3.2	Implementácia obmedzovača layoutovacieho algoritmu	Rozpracované	M.L., T.H.
-----	----------------------------------------------------	--------------	---------------

Priebeh stretnutia:

- kontrola úloh
- diskusia o podpore všeobecných grafov
 - vytvorenie vyššej vrstvy zastrešujúcej súčasnú reprezentáciu pomocou uzla a čiary
 - vytvorenie tried zastrešujúcich jednotlivé typy uzlov a hrán, ktoré sa v grafe môžu nachádzať
 - ukladanie všeobecných grafov do databázy
- diskusia o spôsobe riešenia vnorených grafov na dátovej úrovni a na vizuálnej úrovni
- diskusia o spôsobe obmedzovania zobrazenia pre vnorené grafy

Úlohy, ktoré vyplynuli zo stretnutia

D	Úloha	Vypracovať do	Zodpovední
1.4	Implementovať chýbajúce časti pre prácu s databázou (ukladanie pozícií , ukladanie metauzlov a metahrán , layoutov , pozície kamery)	9.3.2010	T.H.
2.1	Implementácia podpory vnorených grafov	16.3.2010	J.P.
2.2	Dokončiť podporu hypergrafov a vnorených grafov do existujúcich parserov (GraphML a GXL)	16.3.2010	M.L.
2.4	Implementovať RSF parser	9.3.2010	M.J.
3.1	Implementácia podpory hyperhrán	9.3.2010	M.M.
3.2	Implementácia obmedzovača layoutovacieho algoritmu	16.3.2010	M.L., T.H.
4.1	Preskúmať možnosti pre inštancionalizáciu triedy Graph	9.3.2010	J.P.
4.2.	Premyslieť vyššiu úroveň reprezentácie grafov (všeobecný graf)	16.3.2010	J.P., M.L.
4.3.	Podpora RSF parseru pre hyperhrany	16.3.2010	M.J.

5.15 Zápis zo stretnutia č. 15

Tím č. 16

Dátum: 9. 3. 2011

Miesto: D; FIIT STU

Prítomní: Pedagóg: Ing. Peter Kapec

Členovia tímu: Bc. Tomáš Hurban, Bc. Milan Laslop, Bc. Matúš Juhas

Zapísal: Bc. Matúš Juhas

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

D	Úloha	Stav	Dátum dokončenia	Zodpovední
1.4	Implementovať chýbajúce časti pre prácu s databázou (ukladanie pozícií , ukladanie metauzlov a metahrán , layoutov , pozície kamery)	Dokončené	9.3.2011	T.H.
2.1	Implementácia podpory vnorených grafov	Rozpracované		J.P.
2.2	Dokončiť podporu hypergrafov a vnorených grafov do existujúcich parserov (GraphML a GXL)	Dokončené		M.L.
2.4	Implementovať RSF parser	Dokončené	9.3.2011	M.J.
3.1	Implementácia podpory hyperhrán	Rozpracované		M.M.
3.2	Implementácia obmedzovača layoutovacieho algoritmu	Rozpracované		M.L., T.H.
4.1	Preskúmať možnosti pre inštancionalizáciu triedy Graph	Rozpracované	9.3.2011	J.P.
4.2.	Premyslieť vyššiu úroveň reprezentácie grafov (všeobecný graf)	Rozpracované	16.3.2011	J.P., M.L.
4.3.	Podpora RSF parseru pre hyperhrany	Rozpracované	16.3.2011	M.J.

Priebeh stretnutia:

- kontrola úloh
- diskusia o získavaní hyperhrán z RSF súborov
 - pri pridávaní novej hrany overiť, či už je vytvorená hrana s daným zdrojovým vrcholom

- diskusia o spôsobe zobrazenia grafov do priestoru gule
- diskusia o spôsobe obmedzovania zobrazenia pre vnorené grafy

Úlohy, ktoré vyplynuli zo stretnutia

D	Úloha	Vypracovať do	Zodpovední
2.1	Implementácia podpory vnorených grafov	16.3.201 1	J.P.
3.1	Implementácia podpory hyperhrán	16.3.201 1	M.M.
3.2	Implementácia obmedzovača layoutovacieho algoritmu	16.3.201 1	M.L., T.H.
4.1	Preskúmať možnosti pre inštancionalizáciu triedy Graph	16.3.201 1	J.P.
4.2.	Premyslieť vyššiu úroveň reprezentácie grafov (všeobecný graf)	16.3.201 1	J.P., M.L.
4.3.	Podpora RSF parseru pre hyperhrany	16.3.201 1	M.J.
5.1.	Podpora uloženia viacerých grafov z jedného súboru v databáze	16.3.201 1	

5.16 Zápis zo stretnutia č. 16

Tím č. 16

Dátum: 16. 3. 2011

Miesto: D; FIIT STU

Prítomní: Pedagóg: Ing. Peter Kapec

Členovia tímu: Bc. Tomáš Hurban, Bc. Milan Laslop, Bc. Jaroslav Prokop,
Bc. Matúš Juhas, Bc. Michal Masliš

Zapísal: Bc. Michal Masliš

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

D	Úloha	Stav	Dátum dokončenia	Zodpovední
----------	--------------	-------------	-------------------------	-------------------

2.1	Implementácia podpory vnorených grafov	Rozpracované	1	16.3.201	J.P.
3.1	Implementácia podpory hyperhrán	Rozpracované	1	16.3.201	M.M.
3.2	Implementácia obmedzovača layoutovacieho algoritmu	Dokončené	1	16.3.201	M.L., T.H.
4.1	Preskúmať možnosti pre inštancionalizáciu triedy Graph	Rozpracované	1	16.3.201	J.P.
4.2.	Premysliť vyššiu úroveň reprezentácie grafov (všeobecný graf)	Rozpracované	1	16.3.201	J.P., M.L.
4.3.	Podpora RSF parseru pre hyperhrany	Rozpracované	1	16.3.201	M.J.
5.1.	Podpora uloženia viacerých grafov z jedného súboru v databáze	Rozpracované	1	16.3.201	T.H.

Priebeh stretnutia:

- kontrola úloh
- diskusia o vytvorení špeciálnej triedy, ktorá bude určovať vizuálne vlastnosti
- diskusia o zobrazovaní vnorených grafov v guli, pôsobenie síl
- diskusia o zobrazení, umiestnení a natočení roviny – ovládané používateľom
 - Možnosť využiť niečo z [bakalárskej práce](#) Bc. Miroslava Mikuláša
- Diskusia o vyššej úrovni – univerzálnom grafe

Úlohy, ktoré vplynuli zo stretnutia

D	Úloha	Vypracovať do	Zodpovední
2.1	Implementácia podpory vnorených grafov	23.3.201 1	J.P.
3.1	Implementácia podpory hyperhrán	23.3.201 1	M.M.
4.1	Preskúmať možnosti pre inštancionalizáciu triedy Graph	23.3.201 1	J.P.
4.2.	Premysliť vyššiu úroveň reprezentácie grafov (všeobecný graf)	23.3.201 1	J.P., M.L.
4.3.	Podpora RSF parseru pre hyperhrany	23.3.201 1	M.J.
5.1.	Podpora uloženia viacerých grafov z jedného súboru v databáze	23.3.201 1	T.H.

5.17 Zápis zo stretnutia č. 17

Tím č. 16

Dátum: 23. 3. 2011

Miesto: D; FIIT STU

Prítomní: Pedagóg: Ing. Peter Kapec

Členovia tímu: Bc. Tomáš Hurban, Bc. Milan Laslop, Bc. Jaroslav Prokop,
Bc. Matúš Juhas, Bc. Michal Masliš

Zapísal: Bc. Tomáš Hurban

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

D	Úloha	Stav	Dátum dokončenia	Zodpovední
2.1	Implementácia podpory vnorených grafov	Rozpracované		J.P.
3.1	Implementácia podpory hyperhrán	Rozpracované		M.M.
4.1	Preskúmať možnosti pre inštancionalizáciu triedy Graph	Dokončené	23.3.2011 1	J.P.
4.2.	Premyslieť vyššiu úroveň reprezentácie grafov (všeobecný graf)	Dokončené	23.3.2011 1	J.P., M.L.
4.3.	Podpora RSF parseru pre hyperhrany	Dokončené	23.3.2011 1	M.J.
5.1.	Podpora uloženia viacerých grafov z jedného súboru v databáze	Dokončené	23.3.2011 1	T.H.

Priebeh stretnutia:

- kontrola úloh z predchádzajúceho cvičenia
- je potrebné dokončiť všetky rozpracované časti projektu a tieto časti spojiť do jednej (master)
 - databáza
 - vnorené grafy
 - parsery pre importer
- taktiež je potrebné doplniť dokumentáciu – napísať, čo sme tento semester spravili
- diskusia o tom, na čom chceme v ďalšej časti semestra pracovať
 - vytváranie grafov priamo v aplikácii
 - vytvorenie exportera, aby sme mohli vytvorené grafy aj ukladať
 - pridanie možnosti spájania a rozklikávania uzlov

Úlohy, ktoré vyplynuli zo stretnutia

D	Úloha	Vypracovať do	Zodpovední
2.1	Implementácia podpory vnorených grafov	30.3.201 1	J.P.
3.1	Implementácia podpory hyperhrán	30.3.201 1	M.M.
7.1	spojiť všetky dokončené vetvy projektu do master	30.3.201 1	všetci
7.2	písať časť dokumentácie - návrh	30.3.201 1	všetci
7.3	doplniť do GUI tlačidlo na pridávanie hrán do grafu	30.3.201 1	M.J.
7.4	doplnenie vytvárania nových grafov priamo v programe	30.3.201 1	M.J.
7.5	doplniť GUI pre pridávanie layoutovania do gule	30.3.201 1	M.L.
7.6	návrh – spájanie a rozklikávanie uzlov v grafe	30.3.201 1	T.H.
7.7	doplnenie funkcií importera	30.3.201 1	M.M.
7.8	implementácia exportera (pre jeden formát)	30.3.201 1	M.M.

5.18 Zázpis zo stretnutia č. 18

Tím č. 16

Dátum: 30. 3. 2011

Miesto: D 003; FEI STU

Prítomní: Pedagóg: Ing. Peter Kapec

Členovia tímu: Bc. Tomáš Hurban, Bc. Milan Laslop, Bc. Jaroslav Prokop,
Bc. Matúš Juhas

Zapísal: Bc. Milan Laslop

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

	Úloha	Stav	Dátu	Zodpo
--	--------------	-------------	-------------	--------------

D			m dokončenia	vední
2.1	Implementácia podpory vnorených grafov	Dokončené	30.3. 2011	J.P.
3.1	Implementácia podpory hyperhrán	Dokončené	30.3. 2011	M.M.
7.1	spojiť všetky dokončené vetvy projektu do master	Dokončené	30.3. 2011	všetci
7.2	písať časť dokumentácie - návrh	Rozpracované		všetci
7.3	doplniť do GUI tlačidlo na pridávanie hrán do grafu	Dokončené	30.3. 2011	M.J.
7.4	doplnenie vytvárania nových grafov priamo v programe	Rozpracované		M.J.
7.5	doplniť GUI pre pridávanie layoutovania do gule	Rozpracované		M.L.
7.6	návrh – spájanie a rozklikávanie uzlov v grafe	Dokončené	30.3. 2011	T.H.
7.7	doplnenie funkcií importera	Rozpracované		M.M.
7.8	implementácia exportera (pre jeden formát)	Rozpracované		M.M.

Priebeh stretnutia:

- zhrnutie stavu úloh a možností pokračovania v úlohách
 - podpora reprezentácie a zobrazovania vnorených grafov je hotová, aj so škálovaním (uzly sú postupne menšie podľa úrovne vnorenia)
 - oddelené vetvy boli znovu pripojené do vetvy master
 - do GUI bolo pridané tlačidlo na pridávanie hrán, treba pridať možnosť pridávať nové uzly a vymazávať uzly alebo hrany
- diskusia o problémoch s layoutovaním a obmedzovaním
 - pozícia uzlov pravdepodobne neudáva pozíciu ich stredu
 - layoutovač sa zastavuje, ak nenastávajú žiadne zmeny – tie môžu prísť po pridaní obmedzenia (napr. do gule), vtedy treba znovu spustiť layoutovač cez GUI
 - layoutovač musel byť zmenený tak, aby na seba vplývali iba uzly z rovnakého grafu (aby napr. uzly vnoreného grafu nevplývali na uzly nadradeného grafu)
- testovanie layoutovania vnorených grafov do gule
 - vnorené uzly vytvoria guľu, ale mimo nadradeného uzla – treba skontrolovať výpočty obmedzení a pôsobenie layoutovacieho algoritmu na usporiadanie
- zmeny v pridávaní obmedzení layoutu
 - použiť OSG Manipulator na definovanie obmedzení (namiesto pomocných uzlov)
 - vykresľovanie obmedzovačov (pomôže aj pri testovaní)

- diskusia o možnostiach GITu

Úlohy, ktoré vyplynuli zo stretnutia

D	Úloha	Vypracovať do	Zodpovední
7.2	písať časť dokumentácie - návrh	3.4.2011	všetci
7.4	doplnenie vytvárania nových grafov priamo v programe	6.4.2011	M.J.
7.5	doplniť GUI pre pridávanie layoutovania do gule – pridať zobrazenie gule	6.4.2011	M.L.
7.7	doplnenie funkcií importera	6.4.2011	M.M.
7.8	implementácia exportera (pre jeden formát)	6.4.2011	M.M.
8.1	pridávanie uzlov cez GUI, pridávanie a vymazávanie hrán	6.4.2011	M.J.
8.2	test layoutovania do gule, riešenie problémov s layoutovaním	6.4.2011	M.L.

5.19 Zázpis zo stretnutia č. 19

Tím č. 16

Dátum: 6. 4. 2011

Miesto: D 003; FEI STU

Prítomní: Pedagóg: Ing. Peter Kapec

Členovia tímu: Bc. Tomáš Hurban, Bc. Milan Laslop, Bc. Jaroslav Prokop,
Bc. Matúš Juhas

Zapísal: Bc. Matúš Juhas

Vyhodnotenie plnenia úloh z predchádzajúcich stretnutí

D	Úloha	Stav	Dátum dokončenia	Zodpovední
7.2	písať časť dokumentácie - návrh	Rozpracované	13.4. 2011	všetci
	doplnenie vytvárania nových grafov	Rozpra		M.J.

7.4	priamo v programe	cované		
7.5	doplniť GUI pre pridávanie layoutovania do gule	Rozpracované		M.L.
7.7	doplnenie funkcií importera	Rozpracované		M.M.
7.8	implementácia exportera (pre jeden formát)	Rozpracované		M.M.
8.1	pridávanie uzlov cez GUI, pridávanie a vymazávanie hrán	Rozpracované		M. J.
8.2	test layoutovania do gule, riešenie problémov s layoutovaním	Rozpracované		J. P.

Priebeh stretnutia:

- zhrnutie stavu úloh a možností pokračovania v úlohách
 - rozkliknutie a zakliknutie viacerých hrán a uzlov do jedného bodu je implementované
 - bola doplnená časť dokumentácie k časti návrh
 - do GUI boli pridané tlačidlá pre pridanie uzla a vymazávanie uzlov a hrán
- diskusia o problémoch s vykresľovaním pri zakliknutí a pri vymazaní uzlov, vykresľovanie do gule
 - pri vykresľovaní (presune) uzlov, hrán do jedného bodu bol pri layoutovaní zistený problém zlého umiestnenia, z tohto dôvodu sa hrany a uzly vykreslili na novej pozícii
 - pri vymazaní uzlov z grafu sú tieto uzly stále vykresľované, možným problémom môže byť synchronizácia medzi OSG a Data
 - pri vykresľovaní gúl ako obmedzenia priestoru nie je vidno obsah danej gule,
- diskusia o rozsahu projektu do konca semestra

Úlohy, ktoré vyplynuli zo stretnutia

D	Úloha	Vypracovať do	Zodpovední
9.1	Rozklikávanie uzlov a hrán do a z jedného bodu implementovať cez plynulú animáciu	13.4.201	T. H.
9.2	Príprava testovacích dát		všetci

6 Preberacie protokoly

V tejto kapitole sa nachádzajú preberacie protokoly potvrdzujúce odovzдание dokumentácie k vypracovanej dokumentácii časti analýza a časti riadenie.

Preberací protokol

Odovzdávajúci subjekt: **GraFIT (tím č. 16)**

Preberajúci subjekt: **Ing. Peter Kapec (vedúci tímu)**

Predmety prebrania a ich popis:

Predmety prebrania sú: dokumentácia analýzy problému, špecifikácia požiadaviek, návrh riešenia, dokumentácia k riadeniu.

Poznámky:

Odovzdané v tlačenej podobe a v elektronickej podobe vo formáte PDF na USB kľúči.

.....
podpis zástupcu odovzdávajúceho subjektu

.....
podpis zástupcu preberajúceho subjektu

V Bratislave, dňa 13.4.2011