

# Slovenská technická univerzita

Katedra informatiky a výpočtovej techniky, Fakulta elektrotechniky a informatiky  
Ilkovičova 3, 812 19 Bratislava

---

## Posudzovanie projektov v prostredí internetu

### Posudok dokumentácie

#### **Autori dokumentácie:**

##### **Tím č. 2 – *Danamythos***

Bc. Ján Antala  
Bc. Rastislav Klč  
Bc. Michal Kolarovič  
Bc. Juraj Mlynár  
Bc. Filip Pucher  
Bc. Kristián Vedrůdy

#### **Autori posudku:**

##### **Tím č. 4 – *YonBan***

Bc. Peter Blšták  
Bc. Pavol Kiša  
Bc. Martin Mačica  
Mgr. Marek Pisch  
Bc. Martin Ševčík

---

**Odbor:** Softvérové inžinierstvo

**Predmet:** Tímový projekt

**Šk. rok:** 2002/2003

# Obsah

<b>1. ÚVOD</b> .....	<b>1</b>
<b>2. FORMÁLNA STRÁNKA</b> .....	<b>2</b>
<b>3. OBSAHOVÁ STRÁNKA</b> .....	<b>3</b>
3.1. DOKUMENTÁCIA K SOFTVÉROVÉMU SYSTÉMU .....	3
3.1.1. <i>Analýza</i> .....	3
3.1.2. <i>Špecifikácia</i> .....	4
3.1.3. <i>Hrubý návrh architektúry</i> .....	6
3.2. DOKUMENTÁCIA K RIADENIU PROJEKTU .....	7
<b>4. ZHODNOTENIE</b> .....	<b>9</b>

# 1. Úvod

Obsahom tohto dokumentu je posúdenie dokumentácie prvej etapy práce na tímovom projekte<sup>1</sup> tímu č.2 – *Danamythos* v zimnom semestri. Tím *Danamythos*, podobne ako náš tím, pracuje na projekte *Posudzovanie projektov v prostredí internetu*. Posudzovaná dokumentácia je rozdelená do dvoch samostatných celkov: *Dokumentácia k softvérovému systému* a *Dokumentácia k riadeniu projektu*. Prvý celok opisuje výsledky práce na samotnom projekte (analýza, špecifikácia, hrubý návrh architektúry) a druhý organizáciu a riadenie práce na projekte (ponuka, plán projektu, úlohy členov v tíme, zápisy zo stretnutí).

Dokumentácia je posúdená z pohľadu formálneho i pohľadu obsahového. V posudku nerozoberáme dopodrobna všetky detaily, ale snažíme sa zhodnotiť celkový dojem z dokumentácie.

---

<sup>1</sup> Tímový projekt je absolvovaný v rámci inžinierskeho štúdia na Katedre informatiky a výpočtovej techniky FEI STU

## 2. Formálna stránka

Dokumentácia je na prvý pohľad rozsiahla. Po prelistovaní pôsobí dobrým dojmom, bez rušivých estetických vplyvov. Členenie dokumentu je prehľadné. Nachádza sa v ňom množstvo obrázkov a schém, čo priaznivo vplýva na celkovú čitateľnosť.

V dokumentácii je vidieť snahu o jasnosť a jednoznačnosť. Každá zavedená notácia je prehľadne a stručne vysvetlená. No táto snaha nie je dotiahnutá do optima. Autori pozabudli na niektoré vysvetlenia a opis názvoslovia. Z tohto vyplynula nekonzistencia v názvoch.

Napr. v rámci jedného odseku sa hovorí o hodnotení projektu a o posudzovaní projektu, pričom z kontextu sa dá vycítiť, že ide o tú istú činnosť (kap. *Riešenie a hodnotenie projektu*).

V dokumente je dosť nevhodné použitie výrazu „pavučina“. Pre čitateľa je to máťúce, pretože niektorých kapitolách autori píšu o „svetovo rozsiahlej pavučine“ (väčšina dokumentu), v iných o „Internet“ (str. 23). V kapitole hrubého návrhu sa s tým istým významom používa aj výraz „www“ (str. 51).

Pri čítaní niektorých častí dokumentu sme mali pocit, že nie všetky vety sú správne sformulované. V niektorých prípadoch by zmena slovosledu pomohla jasnejšie vyjadriť myšlienku, ktorú autori chceli prezentovať.

Za nie veľmi vhodné považujeme v dokumente *Dokumentácia softvérového systému* odvolávanie sa na kap. *Ponuka* (str. 21) alebo na iné kapitoly z dokumentácie k riadeniu projektu (str. 6).

V kap. *Význam použitých skratiek* je dva krát uvedená skratka JSP, pričom tam ale chýbajú niektoré iné skratky (napr. *PWS*, str. 23).

Kvôli prehľadnosti by sme odporúčali prílohy členiť tak, aby vždy začínali na novej strane.

## 3. Obsahová stránka

### 3.1. Dokumentácia k softvérovému systému

#### 3.1.1. Analýza

Kapitola *Analýza* obsahuje opis a analýzu súčasnej správy projektov, existujúcich systémov a technológií.

Kapitola *Súčasná správa projektov* obsahuje zjednodušený logický model údajov súčasného stavu i slovný popis entít a relácií. V tejto časti je občas miesto slova „stav“ použité slovo „systém“. Nevieme posúdiť, či to bol zámer autorov, alebo je to len chyba. Čitateľovi z toho môže vyplývať, že na sa modeluje nejaký súčasný informačný systém (čo asi nie je pravda).

Logický model je prehľadný a ku každej entite existuje aj relevantný opis. Opis väčšiny entít je výstižný a jasný. Trochu slabšie to je pri entitách *Téma* a *Študent*. Nie je jasné, či sa téma vytvorí zo zadania, alebo zadanie z témy (str. 6). Pri študentovi nám chýba časť o tom, že študent rieši a odovzdáva zadanie resp. riešenie prideleného projektu (str. 7).

V kap. *Analýza existujúcich systémov* autori opisujú viacero systémov venujúcich sa spracovaniu dokumentov, či už vo forme projektov alebo príspevkov do konferencie. Pri opise jednotlivých systémov zvolili jednotnú štruktúru, čo sprehľadňuje orientáciu v texte. Aj keď viacero z týchto systémov bolo už popísaných v dokumentáciách minuloročných tímov, ktoré sa venovali problematike organizácie konferencie, autori zvolili iný pohľad na systémy. Treba vyzdvihnúť to, že sa venovali najmä možnostiam využiteľnosti niektorých vlastností systémov, ako aj hľadaniu paralel daného systému s navrhovaným systémom.

Na konci analýzy každého systému poukázali na vlastnosti, ktoré môžu využiť pri navrhovaní ich projektu. Takisto poukázali na nedostatky alebo chyby v niektorých systémoch, aby sa takýmto chybám vyvarovali pri tvorbe svojho systému.

V kap. *Analýza technológií* autori opisujú niektoré existujúce technológie, ktoré by mohli byť využité pri implementácii riešenia. V úvode kapitoly uvádzajú tvrdenie o „všeobecnej klient/server 3-vrstvovej architektúre“, ktoré je trochu zavádzajúce. Vo všeobecnosti nemusí byť klient/server architektúra 3-vrstvová.

Pre každú logickú vrstvu 3-vrstvovej architektúry (prezentačná, aplikačná, dátová) autori uvádzajú viacero možných technológií. Za popisom technológií nasleduje stručné zhrnutie a výber novej alternatívy pre implementáciu. Výber technológie ešte nie je konečný, čo v tejto etape vývoja softvéru považujeme za veľmi rozumné.

### 3.1.2. Špecifikácia

Kap. *Špecifikácia* obsahuje špecifikáciu rolí používateľov, životný cyklus projektu, špecifikáciu funkcionálnych a nefunkcionálnych požiadaviek.

V kapitole *Špecifikácia rolí* sú spomínané roly používateľov a skupiny používateľov, pričom sa zrejme myslí to isté (nekonzistencia názvov, resp. jedno je vysvetľované pomocou druhého). Z toho však vyplývajú dôsledky ako: koľko *loginov* bude mať niekto, kto patrí do viacerých skupín alebo rolí.

V opise rolí je uvádzané, že *zadávatel' projektu* môže byť aj externý pracovník, čiže má právo na definovanie zadania a i. *Pedagogický pracovník* má zastupovať externého zadávateľa a teda so systémom pracuje namiesto externého pracovníka. Tieto dve vyjadrenia si odporujú.

Taktiež nie je v opise rolí uvedené, či sa roly vzťahujú na konkrétne projekty, tzn. jeden používateľ môže mať rôzne roly v rôznych projektoch, čo je podľa nášho názoru dosť dôležité a nemalo by sa nachádzať „len“ v *špecifikácii nefunkcionálnych požiadaviek*.

V kapitole *Životný cyklus projektu* sa autori zamerali na objasnenie procesov súvisiacich so zadávaním, riešením a hodnotením projektov. Pri tomto opise vychádzali z procesov, ktoré sú vykonávané v tejto súvislosti na KIVT<sup>2</sup>.

Na lepšie znázornenie životného cyklu projektu použili stavový diagram podľa notácie UML – Unified Modeling Language. K diagramu stavového diagramu máme nasledovné pripomienky:

- Zmiešanie stavových diagramov životného cyklu projektu v systéme so stavmi systému, ktoré súvisia s riešením projektu. Čiže ide o zmiešanie dvoch stavových diagramov do jedného. Napríklad:
  - Stav *inicializácia projektu* súvisí s definovaním typov projektu(ov) (je to stav systému) a nie so stavom jedného konkrétneho projektu.
  - Zo stavu *schvaľovanie projektu* (stav projektu) sa prechádza po udalosti *schválený projekt* do stavu *výber a pridelovanie projektu*, pričom pri tomto prechode sa vykonáva akcia publikovanie zadaní. Táto akcia by však nemala súvisieť so schválením nejakého jedného konkrétneho projektu.

---

<sup>2</sup> Katedra informatiky a výpočtovej techniky Fakulty elektrotechniky a informatiky STU

- Celková nekonzistencia diagramu (zrejme vznikla snahou o čo najstručnejšie vyjadrenie veľkého množstva informácií). O stavoch sa hovorí ako o procesoch, čo pripomína skôr DFD – Data Flow Diagram. Toto by sa mohlo dať vysvetliť tým, že v danom stave sa vykonávajú nejaké procesy, ktoré sa vzťahujú k tomuto stavu (ako je aj naznačené v „meta“ opise diagramu na začiatku kapitoly). Pri pohľade na stavový diagram je však veľmi ťažko zistiť, či ide o stavy alebo skutočné procesy alebo niečo iné. V stavovom diagrame a jeho opise sa napr. nachádza stav *riešenie a hodnotenie projektu*, do ktorého je možné dostať sa zo stavov *výber a pridelovanie projektov* ale aj sprostredkovane cez ďalší stav *pridelovanie oponentov*, pričom však stav/proces *riešenie a hodnotenie projektu* vyžaduje, aby bol projekt pridelený pre potreby riešenia (prvý prechod medzi stavmi) a aby boli priradení oponenti pre potreby hodnotenia (druhý možný prechod). Tieto časti stavového diagramu vôbec nie sú zrejme a ide zjavne o miešanie dvoch nezávislých pohľadov na jednu vec – stavy projektu a (paralelné) vykonávanie procesov (funkcií) nad projektom(ami). Napr.:
  - Na to, aby sa procesy v stave *riešenie a hodnotenie projektu* mohli vykonať je potrebné, aby bol projekt niekedy v stave *pridelovanie oponentov*, čo však vôbec nemusí nastať.
- Zo stavového diagramu ďalej vyplýva, že po tom, ako bol projekt pridelený, neexistuje možnosť ukončiť projekt za žiadnych podmienok. Toto odporuje napr. skutočnosti, že projekt môže byť ukončený z dôvodu neprebratia zadania alebo neodovzdania požadovaných častí riešenia projektu v požadovaných termínoch. Čiže podľa stavového diagramu nemusí životný cyklus projektu vôbec skončiť (ani keď nie je vyriešený).
- Táto časť dokumentácie obsahuje aj otázku vo forme poznámky pod čiarou, ktorá by sa zrejme nemala nachádzať v špecifikácii systému. Táto otázka mala byť už v tejto fáze vyriešená.

Až na spomenuté nedostatky, z ktorých všetky podstatné vychádzajú z nejasností súvisiacich so stavovým diagramom, táto kapitola dostatočne opisuje procesy súvisiace so životným cyklom projektu ako aj kontextu tohto cyklu.

V kap. *Špecifikácia funkcionálnych požiadaviek* sú funkcie systému uvedené prehľadne v tabuľkách a zotriedené do skupín. Nemyslíme si ale, že bolo nevyhnutné opisovať znova súčasný stav, pretože ten je opísaný v samostatnej kapitole (*Súčasná správa projektov*).

Je dobré, že ku každej funkcii je priradená aj priorita jej implementácie, resp. jej potrebnosť pre beh systému. Nie je ale uvedené, čo znamená *vstupná podmienka* funkcie a z kontextu je cítiť, že sa ňou nemyslí to, čo sa všeobecne pojmom vstupná podmienka označuje.

Funkcia *Priamy prístup do databázy* je pravdepodobne zbytočná. Administrátor môže takúto funkciu vykonávať priamo prostredníctvom konzoly na serveri. Okrem toho funkcia vystavuje systém riziku útoku, vzhľadom na to, že bude chránená len autentifikáciou na aplikačnej úrovni.

Ďalším zisteným nedostatkom je nekonzistencia v opise tvorby typu projektu. V kapitole *Životný cyklus projektu* je spomenutá inicializácia typu projektu, ktorú má na starosti garant projektu. No garant projektu je priradený k typu projektu pri inicializácii tohto typu projektu. Tzn. garant nemôže vykonávať inicializáciu typu projektu. V ďalších častiach špecifikácie sa už táto drobná chyba neopakuje (pravdepodobne bola spôsobená nepozornosťou).

Z opisu *harmonogramu projektu* je dosť ťažko si predstaviť, ako bude fungovať. Sú špecifikované nejaké tri základné termíny, ale nie je dostatočne opísaná ich funkčnosť. Spomína sa i možnosť definovať ďalšie termíny. Nie je zrejmé, aká je predstava o pridávaní týchto termínov, vzhľadom na to, že na termíny sa vzťahuje aj určitá funkčnosť, resp. stav projektov alebo iné.

Špecifikované funkcie systému uvedené v kapitole *Špecifikácia funkcionálnych požiadaviek* dobre pokrývajú požiadavky na systém.

V kapitole *Špecifikácia nefunkcionálnych požiadaviek* sa zjavne nenachádzajú len nefunkcionálne požiadavky (napr. *správa informácií o používateľovi* je zahrnutá aj v kap. *Špecifikácia funkcionálnych požiadaviek*).

### **3.1.3. Hrubý návrh architektúry**

V poslednej kapitole *Dokumentácie k softvérovému systému* autori uvádzajú hrubý návrh architektúry, ktorú sa rozhodli použiť pre implementáciu. Pozitívnou črtou návrhu je jeho rozdelenie do troch logických vrstiev (prezentačná, aplikačná, dátová).

Obrázok architektúry systému (str. 50) má čitateľovi graficky ozrejmiť rozdelenie komponentov do jednotlivých vrstiev. Tu sa stretávame s nie najvhodnejšou formou zobrazenia:



- v prezentačnej vrstve sa nachádzajú technológie a protokoly (FTP, SMTP, HTTP, XML, HTML)
- v aplikačnej vrstve sa nachádza zoznam modulov, na ktoré je namapovaná funkcionálna špecifikovaná v predchádzajúcich kapitolách (bez uvedenia technológií a protokolov)
- v dátovej vrstve sa nenachádzajú žiadne relevantné údaje.

Z pohľadu v budúcnosti použitých technológií je najlepšie spracovaná časť prezentačnej vrstvy (str. 51), kde sa autori rozhodli pre použitie jazyka XML. Vhodne je tam uvedené aj ošetrenie prípadu, keď klient nebude rozumieť vytvorenému XML dokumentu. Autori spomenuli aj výhody takéhoto riešenia pri práci v tíme.

Pri popise aplikačnej vrstvy (str. 51), bol najväčší dôraz kladený na popis jednotlivých modulov (a k nim prislúchajúcich funkcií) tejto vrstvy. Jednotlivé moduly sú popísané stručne a výstižne. Keďže ide o hrubý návrh, je takýto popis úplne postačujúci a vhodne zvolený. Menej sa autori zamerali na technológie tejto vrstvy. Stroho skonštatovali, že sa rozhodli pre implementáciu pomocou technológie PHP.

Dátovej vrstve (str. 52) je venovaný najmenší priestor. Autori sa rozhodli použiť databázový systém PostgreSQL.

Obsahom projektovej dokumentácie v tomto kontrolnom bode má byť hrubý návrh riešenia (stránka tímového projektu). Nie je špecifikované o aký návrh ide. Autori posudzovanej dokumentácie sa rozhodli pre hrubý návrh architektúry bez akéhokoľvek modelovania dát. Vzhľadom na povahu a zložitosť navrhovaného systému sme očakávali aspoň hrubý návrh dátového modelu, pretože správny a vhodne navrhnutý dátový model považujeme za najdôležitejší dokument v etape vývoja softvéru takéhoto typu.

### **3.2. Dokumentácia k riadeniu projektu**

Kap. *Dokumentácia k riadeniu projektu* obsahuje organizáciu a riadenie práce na projekte. Ostatné kapitoly budú zrejme do dokumentácie zaradené neskôr, no v súčasnej dokumentácii sú uvedené len nadpisy.

Kap. *Ponuka* je dobre rozpracovaná a dá sa z nej jasne vyčítať, akým spôsobom chce tím projekt riešiť. Zarážajúci však je nadpis *Predpokladané nároky produkčného systému*. Autori mali asi na mysli “Predpokladané nároky produkovaného (zamýšľaného) systému”, keďže nevyvíjajú produkčný systém. Napriek tomu oceňujeme, že sa už vo fáze ponuky zhodli na použitej platforme a technológiách (po predchádzajúcej diskusii). Pri prezentácii

svojej ponuky sa tím zamerá najmä na objasnenie jeho predstavy o funkčnosti tvoreného systému.

Kap. *Plán projektu* je rozpracovaná dostatočne podrobne. V poli 11. a 12. týždňa plánu sú chybné dátumy, mali sa tam nachádzať decembrové dátumy.

Kap. *Úlohy členov v tíme* je rozpracovaná podrobne, napriek tomu by riešitelia mohli zvážiť priradenie náhradníkov za každého člena v prípade absencie.

Kap. *Zápisy zo stretnutí* obsahuje zápisy z jednotlivých stretnutí tímu. Zápisy spĺňajú všetky náležitosti. Navyše obsahujú vždy aj dátum predpokladaného ďalšieho stretnutia. Každý zápis zo stretnutia je podporený viacerými dobre prepracovanými prílohami.

Oceňujeme aj definovanie jednotných šablón pre dokumenty ako aj konvenciu názvov súborov a spôsob odovzdávania a zdieľania čiastkovej dokumentácie medzi sebou prostredníctvom web-rozhrania.

## 4. Zhodnotenie

Posudzovaný dokument i práca tímu pôsobí vyspelým dojmom s dobre organizovanou štruktúrou. Pokrýva všetky požiadavky kladené na formálnu i obsahovú stránku dokumentácie k tímovému projektu.

V dokumentácii sa nevyskytli závažnejšie chyby. Práca členov v tíme je podľa vypracovaného výstupu oduševnená.

Túto prácu i celkovú dokumentáciu hodnotíme *výborne*.