

# Simulated Car Racing Competition 2011

---

## *Dokumentácia k riadeniu projektu*



---

<b>Tím číslo:</b>	20	Bc. Maroš Bednár
<b>Vedúci projektu:</b>	Ing. Peter Vilhan	Bc. Adam Brček
<b>Predmet:</b>	Tvorba softvérového systému v tíme	Bc. Marek Briš
<b>Študijný program:</b>	Informačné systémy/Softvérové inžinierstvo	Bc. Marián Florek
<b>Akademický rok:</b>	2010/2011	Bc. Vojtech Juhász
		Bc. Juraj Kosmeľ
		Bc. Ivan Valenčík

## **A)      **Obsah****

---

A)	Obsah.....	A-1
B)	Úvod .....	B-1
	Účel dokumentu .....	B-1
	Prehľad dokumentu .....	B-1
C)	Ponuka.....	C-1
D)	Plány projektu .....	D-1
	Tvorba plánov.....	D-1
	Hrubý plán projektu na zimný semester .....	D-1
E)	Úlohy členov tímu .....	E-1
	Dlhodobé roly členov tímu .....	E-1
	Krátkodobé úlohy členov tímu .....	E-1
	Autori jednotlivých častí dokumentácie k riadeniu .....	E-2
F)	Podporné prostriedky a metodiky práce .....	F-1
	1. Práca s SVN.....	F-1
	2. Komunikácia .....	F-3
	3. Pravidlá písania a komentovania zdrojového kódu .....	F-4
	4. Manažovanie chýb .....	F-7
	5. Tvorba plánov.....	F-8
G)	Zápisnice zo stretnutí .....	G-1
H)	Preberacie protokoly.....	H-1

## B) Úvod

---

### Účel dokumentu

Predložená dokumentácia obsahuje všetky dokumenty súvisiace s riadením projektu v predmete Tvorba softvérového systému v tíme na tému Simulated Car Racing Competition 2011. Dokumenty boli vytvárané priebežne s cieľom sprehľadniť prácu a zdokumentovať vynaložené úsilie pri práci na projekte.

### Prehľad dokumentu

Tento dokument obsahuje nasledujúce časti:

- **Ponuka:** je to dokument, ktorým sme sa uchádzali o pridelenie témy na riešenie v rámci tímového projektu na tému: *Simulated Car Racing Competition 2011*. Ponuka obsahuje predstavenie jednotlivých členov tímu, motiváciu pre výber danej témy a návrh možného riešenia.
- **Plán projektu:** v tejto sekcii uvádzame plán projektu vypracovaný na začiatku zimného semestra, tzv. hrubý plán na zimný a letný semester. Do tejto sekcie sa postupne dopĺňajú jednotlivé zjemnené krátkodobé plány v chronologickom poradí.
- **Úlohy členov tímu:** v tejto časti uvádzame rozdelenie manažérskych funkcií jednotlivých členov tímu ako aj krátkodobé úlohy členov tímu s dátumom predpokladaného ukončenia.
- **Podporné prostriedky a metodiky práce:** táto časť obsahuje popis podporných prostriedkov pre manažment projektu, **ako aj** metódy, postupy a šablóny na tvorbu dokumentov na ktorých sa tím dohodol.
- **Zápisnice zo stretnutí:** obsahuje zápisy zo spoločných tímových stretnutí. Každý jeden zápis obsahuje vyhodnotenie úloh z predchádzajúceho stretnutia ako aj opis priebehu stretnutia a pridelenie úloh na ďalší týždeň pre jednotlivých členov tímu.

## **C) Ponuka**

---

Slovenská technická univerzita

Fakulta informatiky a informačných technológií  
Ilkovičova 3, 842 16 Bratislava 4

---

# **Ponuka**

## **Tímový projekt**

---

Autori: Bednár Maroš, Bc.  
Brček Adam, Bc.  
Briš Marek, Bc.  
Florek Marian, Bc.  
Juhász Vojtech, Bc.  
Kosmeľ Juraj, Bc.  
Valenčík Ivan, Bc.

Kontakt: brcek.a@gmail.com

Predmet: Tímový projekt  
Dátum: 24.9.2010  
Akad. rok: 2010/11

# 1. Členovia tímu

---



**Bednár Maroš, Bc.** Absolvent bakalárskeho študijného programu Informatika na FIIT STU, ktorý završil bakalárskou prácou na tému Multimediálny výučbový systém, v ktorom boli využité technológie ASP.NET, MsSQL, JavaScript, Flash, Ajax. Na škole získal vedomosti o objektovom programovaní (C++, JAVA, C#) a znalosti vo všetkých fázach vývoja softvérových systémov. Vo firme zastáva pozíciu Web Developera využívajúci jazyky PHP, SQL, jQuery, Ajax, kde vývoj je zameraný na webové aplikácie pre firemný CMS systém.



**Brček Adam, Bc.** Absolvent bakalárskeho programu Informatika na FIIT STU. Počas prvostupňového vysokoškolského štúdia si zlepšil svoje vedomosti v procedurálnom programovaní, po naučení objektovo-orientovanej paradigmy však pri riešení úloh v prevažnej miere využíva jazyk JAVA. Počas inžinierskeho štúdia sa okrem softvérového inžinierstva plánuje venovať najmä umelej inteligencii (Neurónové siete, Evolučné algoritmy, Kognitívna veda), ktorou sa zaoberal aj vo svojej bakalárskej práci Konštrukcia diagnózy chybného správania sa zložitých systémov.



**Briš Marek, Bc.** Absolvent bakalárskeho študijného programu Informatika na FIIT STU, počas ktorého získal skúsenosti s programovacími jazykmi C, JAVA, s relačnými databázovými systémami ako MySQL, s webovými technológiami HTML, CSS a modelovacím jazykom UML. V bakalárskej práci Vytváranie prezentácií si osvojil pravidlá interakcie človeka s počítačom a uplatnil objektovo-orientované programovanie. V rámci rôznych študijných projektov sa podieľal na všetkých fázach životného cyklu vývoja softvéru.



**Florek Marian, Bc.** Absolvent bakalárskeho programu Informatika na FIIT STU. Počas štúdia využíval v prevažnej miere objektovo orientovaný prístup programovania s preferovaným programovacím jazykom JAVA. Okrem jazyka JAVA sa oboznámil s prácou aj v iných jazykoch ako C/C++, HTML, PHP, JavaScript, značkovacom jazyku XML, modelovacím jazykom UML a relačnými databázami. Štúdium završil bakalárskou prácou s názvom "Media server pre VoIP komunikáciu" v ktorej preukázal znalosti potrebné počas vývoja softwaru, od analýzy, cez návrh, k samotnej implementácii a testovaniu.



**Juhász Vojtech, Bc.** Prvý stupeň vysokoškolského vzdelania absolvoval na FIIT STU kde sa oboznámil s programovacími jazykmi, ako sú Java, C/C++, databázové systémy a jazyk SQL. Počas štúdia získal poznatky z vytvárania softvérových systémov zastrešujúc analýzu, návrh, dokumentáciu, implementáciu, testovanie softvéru a jeho prezentáciu. Oboznámil sa s nástrojmi podporujúcimi rôzne fázy vývoja softvéru, napríklad jazyk UML a rôzne podporné nástroje (IBM Rational Software Architect, Magic Draw UML) i rôzne vývojové prostredia (Eclipse, NetBeans, MS Visual Studio). V posledných rokoch pôsobí vo firme ktorá je lídrom na trhu informačných technológií v Španielsku a v latinskej Amerike. Pri svojej práci najviac používa jazyky Java, SQL, HTML, JavaScript, XML. V praxi sa podieľal na vývoje viacerých softvérov a webových stránok či portálov.



**Kosmel' Juraj, Bc.** Je absolventom bakalárskeho štúdia na FIIT STU. Počas štúdia na tejto fakulte si rozšíril svoje vedomosti z procedurálneho programovania o jazyk C a nadobudol nové vedomosti v oblasti objektovo orientovanej paradigmy v jazyku Java, ktorý využil pri riešení väčšiny zadání. Vo svojej bakalárskej práci sa venoval návrhu a implementácii „Podporného prostriedku pre hodnotenie a overovanie vedomostí“ pre predmet Dátové štruktúry a algoritmy, ktorý realizoval v skriptovacom jazyku PHP v kombinácii s HTML, CSS, JavaScript a databázovým serverom MySQL. Na rôznej úrovni ovláda aj Pascal, Assembler či notáciu UML (softvér IBM Rational Architect).



**Valenčík Ivan, Bc.** Je absolventom bakalárskeho štúdia v odbore Informatika na FIIT STU, ktoré ukončil s diplomom Magna Cum Laude. Vo svojej bakalárskej práci, za ktorú dostal pochvalu od dekana, vypracovanej na tému Vytvorenie inteligentného agenta subsymbolickým prístupom umelej inteligencie sa zaoberal použitím umelej neurónovej siete na riadenie robota v simulovanom prostredí Robocode. V priebehu štúdia a v doterajšej praxi sa oboznámil s širokým spektrom technológií, ale hlavne s programovacími jazykmi Java, C# a C/C++, počítačovými sieťami, PKI a postupmi softvérového inžinierstva. Okrem doteraz nadobudnutých vedomostí pri práci určite využije aj vedomosti z pri projekte paralelne študovaných predmetov, medzi ktorými sú napr. neurónové siete alebo grafy.

## 2. Simulated Car Racing Competition 2011

---

Inteligentné agenty majú široké využitie na plnenie úloh v rôznych oblastiach. Z tohto dôvodu je užitočné čo najlepšie pochopiť a zlepšiť ich správanie. Na tento účel je vhodné využiť simulované prostredie, pričom za zaujímavé možno považovať hlavne také prostredia, ktoré majú snahu svojim fyzikálnym modelom simulovať reálny svet. Jedným z nich je The Open Racing Car Simulator (TORCS), ktorého doménou sú automobilové preteky.

TORCS framework je výnimočný tým, že poskytuje premyslený fyzikálny model zahŕňajúci prostredie pre interakciu plne autonómneho autopilota s okolím, sieť sensorov a takmer neobmedzené možnosti návrhu autopilota, od jednoduchého, staticky naprogramovaného s uzavretou množinou schopností, cez autopilota so schopnosťou natrénovať sa, až po dynamicky sa zdokonaľujúceho v reálnom čase.

Pri vytváraní inteligentného agenta si je teda možné vyskúšať rôzne prístupy umelej inteligencie na veľmi praktickom probléme, akým riadenie vozidla určite je. Mnohé podproblémy budú zrejme efektívne riešiteľné klasickým symbolickým prístupom, avšak pre zložitejšie situácie bude určite vhodné uvažovať aj nad použitím subsymbolického prístupu k UI, hlavne neurónových sietí. Takto vytvorená problémová oblasť pokrýva záujmy celého tímu a sľubuje zaujímavú prácu na celú dobu trvania tímového projektu. V etape rozvrhovania zastávok pilota v boxoch a nastavenia auta môžeme využiť dlhoročný záujem niektorých členov tímu o motoristický šport po teoretickej a taktickej stránke.

Na danej úlohe je lákavá aj možnosť overiť a porovnať kvalitu vytvoreného riešenia na súťaži s účasťou tímov z prostredia mimo našej univerzity, čo je motiváciou k dosiahnutiu čo najlepšieho výsledku a dobrej reprezentácie svojej fakulty. Vedomie, že človek je pri práci porovnávaný s inou relevantnou prácou je totiž často veľmi dôležité.

## Koncepcia riešenia

Cieľom projektu je vytvorenie čo najúspešnejšieho autopilota, táto vlastnosť však v sebe zahŕňa pomerne odlišné aspekty v závislosti od výberu súťaže, ktorej sa náš robot zúčastní. V krátkosti preto prezentujeme našu koncepciu pre obidve podujatia.

### ***Championship***

V prvej etape je potrebné pilota naučiť jazdiť na neznámej trati. Tu je potrebné zvládnuť základné činnosti ako akceleráciu, brzdenie, radenie a riadenie. Vznikajúce problémy ako pretáčavosť auta či prešmykovanie kolies je možné odstrániť zapracovaním fyzikálnych zákonov (odstredivá sila a iné) do pravidiel. V tejto fáze by pilot určitou konštantnou rýchlosťou, dostatočne nízkou na absolvovanie všetkých zákrut, prešiel celý okruh a ukladal si získané informácie o trati.

V druhej časti bude snahou nájsť ideálnu stopu, čiže takú trajektóriu, ktorá by umožnila dosiahnuť najvyššiu priemernú rýchlosť na jeden okruh. Po jej nájdení je možné ľahko priradiť optimálne rýchlosti pre jednotlivé úseky trate.

Tretie fáza predstavuje preteky, pri ktorých musíme zvládnuť hlavne vyhýbanie sa kolíziám a predbiehanie. Toto chceme zabezpečiť spočiatku základnými úkonmi ako korekcia smeru jazdy a zníženie rýchlosti, neskôr práve tu je možné zapracovať dynamické učenie pomocou konfrontácie s väčším počtom súperov slúžiacej na rozpoznanie vhodných situácií na uskutočnenie predbiehacích manévrov.

### ***Demolition Derby***

Pre tento typ súťaže sa na prvý pohľad javí byť vhodná kombinácia event-driven architektúry s prednastaveným správaním. Vozidlo sa bude na trati pohybovať podľa vopred určených pravidiel, napr. nebude zastavovať, aby sa nestalo ľahkým cieľom, bude sa vyhýbať stredu plochy, aby nepútalo zbytočne pozornosť, ale nebude sa ani pohybovať príliš pri kraji, aby malo možnosť uniknúť protivníkom atď.

Pritom bude rozoznávať výskyt určených situácií, ktoré môžu byť rozdelené do troch skupín: útočné, obranné a strategické. V prípade, že je rozoznaný výskyt viacerých situácií, tak sa vozidlo bude správať podľa tej s najvyššou prioritou (tie bude zrejme vhodné určiť experimentálne, vysokú hodnotu budú mať asi už prebiehajúce útoky alebo úhybné manévry). Význam útočných a obranných akcií je samovysvetľujúci, pri strategických akciách ide o presun do vhodnej pozície alebo o vyhnutie sa nebezpečnej pozícii.

Na ovládanie agenta v tomto druhu súťaže bude pravdepodobne možné aplikovať aj viacero prístupov známych z prostredia Robocode, s ktorým má skúsenosť Ivan Valenčík, ktorý sa v ňom zaoberal súbojom inteligentných agentov.



### **3. Model používateľa pre jeho identifikáciu**

---

V súčasnosti je vo všetkých sférach spoločnosti a najmä v oblasti informačných technológií sústredené obrovské množstvo údajov. Tieto údaje sú často veľmi citlivé a preto sa do popredia dostávajú otázky ich ochrany proti zneužitiu. Ochrana a zabezpečenie údajov je dnes diskutovanou témou na rôznych fórach a konferenciách. Doteraz používané spôsoby zabezpečenia údajov sú často nepostačujúce a je potrebné hľadať nové techniky a možnosti zabezpečenia.

Každý používateľ je jedinečný, a to jednak svojimi biometrickými charakteristikami, ako aj jedinečným správaním sa a používaním informačných systémov. Tieto charakteristiky iniciovali vývoj softvérových a hardvérových prostriedkov umožňujúcich ich rozlišovanie, získavanie a následné využitie pri identifikácii používateľa. Na druhej strane informácie o správaní sa používateľa v systéme umožňujú vytvoriť model používateľa, a na jeho základe predvídať správanie a prispôsobiť systém predpokladaným požiadavkám používateľa resp. ponúknuť používateľovi adekvátne návrhy.

Spôsoby zabezpečenia využívajúce na identifikáciu používateľa biometrické charakteristiky sú ešte málo rozšírené a bežný používateľ s nimi príde do kontaktu len veľmi zriedkavo. Radi by sme sa s naším tímom podieľali na rozvoji tejto oblasti, čo by nám realizácia tohto zadania umožnila. Je pre nás výzvou zúžitkovať svoje doteraz získané vedomosti pri riešení takéhoto projektu a zároveň ich obohatiť o nové schopnosti a zručnosti.

Členovia tímu ovládajú na rôznej úrovni viaceré programovacie jazyky a nástroje využívané v rámci životného cyklu softvéru. Svoje vedomosti využili pri riešení zadaní počas štúdia, ale aj v rámci pôsobenia na trhu práce. Toto dáva nášmu tímu veľmi dobré predpoklady na úspešné zvládnutie tohto zadania.

# Koncepcia riešenia

## ***Funkcionalita***

Na základe zadania a doposiaľ získaných informácií z tejto oblasti je možné realizovať riešenie viacerými spôsobmi. Chceli by sme vytvoriť podporný prostriedok, ktorý by umožňoval identifikovať používateľa a mohol by byť použitý v kombinácii so súčasným spôsobom zabezpečenia prístupu k službám a aplikáciám prostredníctvom prihlasovacieho mena a hesla. Ďalšiu možnosť identifikácie vidíme v aplikácii na monitorovanie správania sa používateľa. Táto aplikácia by umožňovala zozbierať údaje o práci a správaní sa používateľa v systéme a ich následné využitie pri tvorbe analýz, štatistík resp. pri tvorbe modelu používateľa.

## ***API***

Základom by bolo vytvorenie API na zisťovanie dynamiky používania klávesnice a myši, t.j. čas stlačenia klávesy, interval medzi stlačením dvoch kláves, medzi dvojklikom myši a pod. Toto API by umožňovalo aj ďalším tímom v budúcnosti nadviazať a pokračovať vo vývoji systému, prípadne využiť ho pri vývoji vlastných riešení.

## ***Architektúra***

V rámci nášho konceptu by sme realizáciu rozdelili do dvoch častí, a to modul identifikujúci používateľa na základe dynamiky písania na klávesnici pri prihlasovaní a modul na monitorovanie správania používateľa v systéme a vytvorenie modelu používateľa.

## ***Databáza***

Tieto dva moduly by boli podporované relačnou databázou, ktorá by uchovávala údaje o jednotlivých používateľoch. Tieto údaje by boli východiskom pre tvorbu modelov používateľov.

## 4. Portál pre časopis

---

Náš tím pozostáva z členov s rôznymi zameraniami, ale veríme, že na každú úlohu v rámci tímového projektu vieme nájsť toho najlepšieho. Keďže vývoj webového portálu sa nezačína a nekončí pri HTML – PHP – MySQL, myslíme si, že máme všetko k dispozícii na zvládnutie tohto projektu. V našom rozsiahlom tíme môžeme nájsť web dizajnéra a programátora i členov zameraných viac na analýzu alebo dokumentáciu softvéru.

Dôležitým faktom je aj skúsenosť, ktorú tento projekt so sebou prináša. Domnievame sa, že ak tento projekt úspešne vyriešime, odborníci využívajúci náš systém budú tou najlepšou referenciou v budúcnosti. Nezanedbateľnou motivačnou silou je aj hrdosť, že sa budeme môcť podieľať na ďalšom vývoji ACM Slovakia Chapter.

### Návrh riešenia

Existuje viacero spôsobov na riešenie projektu takéhoto typu, ale vzhľadom na naše skúsenosti s technológiami sme zvolili DHTML podporovanú zo servera pomocou PHP a MySQL. Ďalším dôvodom na voľbu spomenutých technológií je ich dostupnosť.

Systém bude poskytovať nasledujúce funkcionality:

1. web prezentácia časopisu
  - a. zobrazenie jednotlivých článkov
  - b. zobrazenie najviac čítaných a najnovších článkov
  - c. možnosť vyjadriť sa k jednotlivým článkom formou pridávania príspevkov
  - d. možnosť hodnotenia článkov na základe rôznych kritérií (či pomohol používateľovi, či bol aktuálny, atď. )
  - e. možnosť nahlásenia plagiátorstva
2. podpora odovzdania článkov
  - a. vloženie článkov ako súborov rôzneho formátu (pdf, doc, html)
  - b. podpora pridania abstraktu vo viacerých jazykoch
  - c. podpora uvádzania zdrojov v rôznych formátoch
  - d. možnosť posielania správ autorovi o posudkoch alebo príspevkoch pridaných k článku
  - e. možnosť písania článku alebo príspevkov priamo na portáli v textovom WYSWYG editore.
3. podpora posudzovania článkov
  - a. možnosť zadávania posudkov
  - b. možnosť písania posudkov priamo na portáli v textovom WYSWYG editore
  - c. možnosť vedenia diskusie s autorom alebo autormi k nejakému článku
4. elektronická knižnica
  - a. možnosť prehľadávania v publikáciách na základe viacerých kritérií (autor, názov, kľúčové slová, atď.)
  - b. zoraďovanie výsledkov vyhľadávania na základe relevantnosti voči dotazu používateľa alebo na základe iných kritérií (dátum vloženia, autor, názov atď.)
  - c. zobrazenie publikácie on-line, bez stiahnutia samotných súborov

Pri návrhu systému sa budeme snažiť vytvoriť aplikáciu čo najmodulárnejšiu tak, aby sa jednoducho dalo pridať ďalšie moduly alebo nahradiť existujúce novšími alebo efektívnejšími.

### **Používatelia systému**

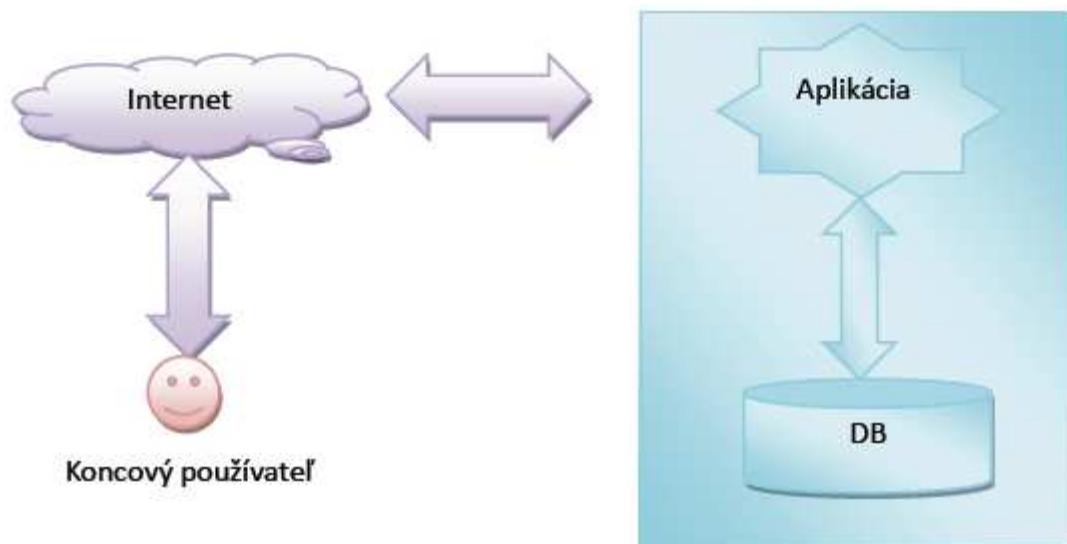
Systém je vytvorený pre širšiu verejnosť, pričom musíme rozlišovať rôzne typy používateľov. Presné určenie práv používateľov bude vyžadovať ďalšie analýzy a konzultácie s vedúcim projektu. Predbežne sme identifikovali nasledujúce skupiny používateľov:

1. Anonymný používateľ – portál bude využívať ako zdroj zverejnených dát
2. Autor – používateľ, ktorý môže odovzdať články a môže ich aj posudzovať.
3. Administrátori – používatelia, ktorí kontrolujú pribúdajúce články, príspevky alebo posudky.

### **Architektúra systému**

Systém pozostáva z troch hlavných častí:

1. Prehliadač používateľa – na jeho počítači vzdialeného od servera
2. Komunikačný kanál – Internet
3. Server – v sebe zahŕňa dátovú (databáza) a aplikačnú vrstvu, je jadrom celého systému.



### **Používateľské rozhranie**

Aj z vlastných skúseností vieme, že zobrazenie nadbytočných prvkov (neaktívne položky v menu, deaktivované polia vo formulároch) môže frustrovať používateľov, ako aj to, keď jednotlivé operácie sú hlboko vnárané v rôznych menu. Práve preto bude používateľské rozhranie pozostávať zo spoločných položiek a z položiek špecifických pre rôzne typy používateľov.

Z hľadiska dostupností informácií, stránka bude dvojúrovňová:

1. verejná úroveň
2. privátna úroveň – vyžaduje autentifikáciu používateľov

## 5. Príloha A: Poradie tém podľa preferencií

---

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

## 6. Príloha B: Aktuálny rozvrh členov tímu

		7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00
Pondelok	Bednár Maroš								Pokr. databáz. tech.		Tímový projekt I		Wyskum soft. syst.			
	Brček Adam										Tímový projekt I		Wyskum soft. syst.			
	Bríš Marek							Pokr. databáz. tech.	Zákl. kryptografie		Tímový projekt I		Wyskum info. syst.			
	Florek Marian							Pokr. databáz. tech.	Zákl. kryptografie		Tímový projekt I		Wyskum info. syst.			
	Juhász Vojtech								Pokr. databáz. tech.	Pokr. databáz. tech.	Tímový projekt I		Wyskum info. syst.			
	Kosmel Juraj							Pokr. databáz. tech.	Zákl. kryptografie		Tímový projekt I		Wyskum soft. syst.			
	Valenčík Ivan										Tímový projekt I		Wyskum soft. syst.			
Utorok	Bednár Maroš										Manažment. projektov softvérových a informačných systémov					
	Brček Adam										Manažment. projektov softvérových a informačných systémov					
	Bríš Marek										Manažment. projektov softvérových a informačných systémov					
	Florek Marian										Manažment. projektov softvérových a informačných systémov					
	Juhász Vojtech										Manažment. projektov softvérových a informačných systémov					
	Kosmel Juraj										Manažment. projektov softvérových a informačných systémov					
	Valenčík Ivan										Manažment. projektov softvérových a informačných systémov					
Streda	Bednár Maroš															
	Brček Adam															
	Bríš Marek															
	Florek Marian															
	Juhász Vojtech												Dejiny dizajnu	Dejiny dizajnu		
	Kosmel Juraj															
	Valenčík Ivan					Neurónové siete *										
Štvrtok	Bednár Maroš															
	Brček Adam															
	Bríš Marek															
	Florek Marian															
	Juhász Vojtech															
	Kosmel Juraj															
	Valenčík Ivan															
Piatok	Bednár Maroš															
	Brček Adam															
	Bríš Marek															
	Florek Marian															
	Juhász Vojtech															
	Kosmel Juraj															
	Valenčík Ivan															

\* cvičenia z predmetu Neurónové siete by som v prípade potreby mohol presunúť o dve hodiny skôr

Belasou farbou sú vyznačené školské povinnosti.

Zlatou farbou sú vyznačené časy mimoškolských aktivít.

Zelenou farbou sú vyznačené najvhodnejšie časy na stretnutia.

## D) Plány projektu

---

### Tvorba plánov

Hrubý plán projektu sa vytvára od tretieho týždňa zimného semestra, po získaní konkrétnej témy tímového projektu. Po vytvorení hrubého plánu sú každý týždeň pridelené úlohy jednotlivým členom tímu a zároveň je kontrolované splnenie úloh z predchádzajúceho týždňa. Plnenie plánu je kontrolované každý štvrtý týždeň, t.j. v siedmom a v jedenástom týždni zimného semestra.

### Hrubý plán projektu na zimný semester

#### **3. týždeň:**

- Analýza problémovej oblasti. Podrobná analýza TORCS framework-u a oponentov
- Vytvorenie web prezentácie tímového projektu
- Vytvorenie plagátu a loga tímu

#### **4. týždeň:**

- Analýza tímov z predchádzajúcich rokov
- Analýza alternatívnych riešení na určenie optimálnej trate

#### **5. týždeň:**

- Vyhodnotenie analyzovaných riešení
- Návrh architektúry auto-pilota
- Určenie funkcie jednotlivých navrhnutých modulov

#### **6. týždeň:**

- Implementácia základného auto-pilota
- Integrácia modulu obnovenia do základného pilota
- Návrh modulu na vytvorenie modelu trate

#### **7. týždeň:**

- Implementácia základného modulu na vytvorenie modelu trate
- *Odovzdanie dokumentácie k riadeniu*
- *Odovzdanie dokumentácie obsahujúcej analýzu problému, špecifikáciu požiadaviek a návrh riešenia*
- Kontrola splnenia plánu zostaveného v treťom týždni, zostavenie modifikovaného plánu

#### **8. týždeň:**

- Vypracovanie spôsobu určenia zákrut a rovných úsekov na základe modelu trate
- Kontrola integrácie a prípadné doladovanie modulu obnovenia

- Návrh modulu detekcie oponentov
- Revízia plánu a vytvorenie zjemneného plánu na nasledujúce štyri týždne

### **9. týždeň:**

- Implementácia modulu detekcie oponentov
- Návrh modulu Defense na základe navrhnutého modulu detekcie oponentov
- Kategorizácia zákrut a určenie stratégie prechádzania cez ne

### **10. týždeň:**

- Návrh modulu predbiehania na základe navrhnutého modulu detekcie oponentov
- Implementácia modulu asistenta trate (Line Assist)
- Implementácia modulu Defense

### **11. týždeň:**

- Integrácia a prípadné ladenie asistenta trate
- Integrácia a prípadné ladenie modulu Defense
- Kontrola splnenia plánu zostaveného v siedmom týždni, zostavenie modifikovaného plánu

### **12. týždeň:**

- Testovanie riešenia: asistent trate, modul detekcie oponentov, modul Defense, modul predbiehania
- Revízia plánu a vytvorenie hrubého plánu na letný semester
- Zhodnotenie stavu vývoja auto-pilota

### **13. týždeň:**

- odovzdanie prototypu vybraných častí systému spolu s dokumentáciou



## E) Úlohy členov tímu

---

### Dlhodobé roly členov tímu

Na spoločnom stretnutí sme si rozdelili jednotlivé manažérske pozície v rámci tímu a to nasledovne:

- **Maroš Bednár:** zástupca vedúceho tímu, web dizajnér
- **Adam Brček:** vedúci tímu, pridelenie úloh a kontrola ich splnenia
- **Marek Briš:** manažér podporných činností
- **Marian Florek:** manažér vývoja
- **Vojtech Juhász:** manažér plánovania, tvorba dlhodobých a krátkodobých plánov
- **Juraj Kosmeľ:** manažér kvality, definovanie štandardov
- **Ivan Valenčík:** manažér rizík

### Krátkodobé úlohy členov tímu

Meno	Úloha
Maroš Bednár	tvorba web prezentácie tímu
	aktualizácia web prezentácie tímu
	analýza pilotov z predchádzajúcich rokov: Luigi a Onieva&Pelta
	návrh modulu ESP
	dokumentácia návrhu riešenia
Adam Brček	návrh sofistikovaného modelu trate
	analýza možností využitia setup-u v championship-e
	analýza a návrh metódy na zistenie najvyššej bezpečnej rýchlosti
	dokumentácia analýzy TORCS
Marek Briš	analýza pilotov z predchádzajúcich rokov: AUTOPIA a Cardamone
	implementácia modulu obnovenia
	vytvorenie loga tímu
	metodika SVN
Marián Florek	analýza pilotov z predchádzajúcich rokov: COBOSTAR a Mr Racer
	návrh modulu obnovenia
	dokumentácia analýzy minuloročných tímov
Vojtech Juhász	dokumentácia k riadeniu
	analýza pilotov z predchádzajúcich rokov: Joseph Alton a Jorge Munoz
	metodika plánovania
	metodika manažovania chýb
Juraj Kosmeľ	analýza pilotov z predchádzajúcich rokov: Tim Alford a Neil Clarke
	dokumentácia špecifikácie riešenia
Ivan Valenčík	analýza súťaže <i>demolition derby</i>
	návrh pamätania si trate
	metodika písania zdrojového kódu

## **Autori jednotlivých částí dokumentácie k riadeniu**

### ***Úvod***

Vojtech Juhász – 100%

### ***Ponuka***

Maroš Bednár – 11%; Adam Brček – 11%; Marek Briš – 11%; Marián Florek – 11%; Vojtech Juhász – 21%; Juraj Kosmeľ – 10%; Ivan Valenčík – 25%;

### ***Plány projektu***

Vojtech Juhász – 100%

### ***Úlohy členov tímu***

Vojtech Juhász – 100%

### ***Zápisnice zo strenutí***

Maroš Bednár – 20%; Adam Brček – 40%; Marek Briš – 20%; Juraj Kosmeľ – 20%;

### ***Podporné prostriedky a metodiky práce***

Marek Briš – 25%; Vojtech Juhász – 25%; Juraj Kosmeľ – 25%; Ivan Valenčík – 25%;

## F) Podporné prostriedky a metodiky práce

---

### 1. Práca s SVN

#### **Manažment verzií**

Jednou z podporných činností, ktoré sa aktívne zúčastňujú na vývoji softvéru je manažment verzií. Manažment verzií je potrebné použiť vtedy, keď sa predpokladá práca viacerých používateľov (vývojárov) na jednom, spoločnom projekte. Toto je aj náš prípad, pretože zdrojové kódy, ktoré sú predmetom implementácie potrebujeme mať neustále k dispozícii a samozrejme v aktuálnej podobe. Nasledujúca metodika popisuje prácu so zvoleným klientom.

#### **TortoiseSVN**

Z dostupných riešení sme sa rozhodli použiť jeden z klientov *Subversion* a to konkrétne *TortoiseSVN*. Oplýva bohatou funkcionalitou, ktorá ponúka prinajmenšom:

- Správu súborov a adresárov uložených v centralizovanom úložisku (v našom konkrétnom prípade sa centrálnym úložiskom stal server `code.google.com`)
  - Aktualizácia lokálneho úložiska (angl. Update)
  - Odovzdanie modifikovaných súborov centrálnemu úložisku (angl. Commit)
  - Zamykanie a odomykanie súborov
- Pamätanie si každej zmeny (vhodné pre detekovanie dobrých a horších zmien)
- Návrat k predchádzajúcim verziám (v prípade, že sa neuberáme správnou cestou)

#### **Centralizované úložisko**

Server: <https://carsimulationraceteam20.googlecode.com/svn/>

#### **Štruktúra úložiska:**

Nasledovná štruktúra predstavuje úložisko, základných 5 adresárov, v ktorých sú ďalšie súbory či podadresáre

- src (súbory .java)
  - champ2010client (prevzatá kostra)
  - controller (naša implementácia)
- bin (súbory .class)
- Doc
  - JavaDoc (vygenerovaný)
- Documents
  - .pdf
  - .doc
  - .png
  - ...

- OthersSolutions (ostatné riešenia)
  - ChinHiongTan
  - DiegoPerez
  - LeonardKinnaird-Heether
  - ...

### ***Aktualizácia lokálneho (pracovného) úložiska – Update***

Prečo nepodceniť aktualizáciu lokálneho úložiska:

- Cieľom je pracovať vždy s aktuálnou verziou centrálnemu úložiska
  - Čiastočne predídeme kolíziám
  - Neaktuálne lokálne úložisko je hrozbou, pretože nami vykonané zmeny často úzko súvisia so zmenami, o ktorých žiaľ nevieme

### ***Odovzdanie modifikovaných súborov centrálnemu úložisku – Commit***

- Zmeny menšieho či radikálnejšieho charakteru je potrebné odovzdať čo najskôr
- Odovzdanie vykonať najlepšie po každej dokončenej zmene
  - cieľom je, aby bolo centrálnu úložisko v čo najaktuálnejšej podobe

### **Samotné odovzdanie zmenených súborov**

V prípade dokončenia úprav je potrebné odovzdať ich centrálnemu úložisku zvolením *Commit*. Každé odovzdanie si vyžaduje aj určitú správu o tom, aké zmeny nastali a čoho sa dotýkajú.

### **V správe uviesť:**

- Stručné zhrnutie zmien uvedené ako „názov“ správy
  - maximálne jeden riadok
- Podrobnejšie zhrnutie zmien
  - podrobnejšie o zmenách (ak je to potrebné)
  - v odrážkach
- Ktorý problém rieši (angl. Issue)
  - napr.: Toto rieši problém #55.

### **Neuvádzať: (explicitne uvedie TortoiseSVN)**

- Autora
- Dátum odovzdávania
- Číslo revízie
- Súbor, ktorých sa zmeny dotýkajú

### **Na formátovanie textu použiť:**

- `*text*` -> pre **tučné**
- `_text_` -> pre podčiarknutie
- `^text^` -> pre *kurzivu*

### ***Zamykanie a odomykanie súborov***

- Nepoužívať!
- Server nepodporuje túto funkcionálnosť

## **2. Komunikácia**

Pri realizácii nášho projektu je potrebné diskutovať a riešiť rôzne problémy, zdieľať a archivovať zdrojové kódy ako aj rôzne materiály použité na štúdium a analýzu danej problematiky a zároveň plánovať a kontrolovať vykonávanie úloh členmi tímu. Okrem osobnej komunikácie na tímových stretnutiach využívame v našom tíme viaceré podporné prostriedky:

- E-mail
  - najmä komunikácia väčšieho rozsahu
  - notifikácia o pridelených úlohách a ich stave
- ICQ
  - interaktívna komunikácia
  - zväčša medzi dvoma členmi
  - okamžité riešenie problémov a získanie informácií
- Google Groups
  - diskusné fórum
  - zdieľanie menších súborov
  - spoločný tímový mail
- Google Code
  - riadenie tímu
  - pridelovanie úloh
  - verziovanie zdrojových kódov
  - wiki
- Google Calendar
  - prehľad dôležitých termínov

Služba Google Groups je využívaná najmä ako diskusné fórum, keď je potrebné vyjadrenie všetkých členov tímu na nejakú tému ohľadom riešenia projektu. Taktiež je výhodou tejto služby spoločný mail, ktorý slúži na komunikáciu tímu ako celku.

Google Code slúži na plánovanie a riadenie činnosti nášho tímu a taktiež na správu, verziovanie a archiváciu vytvorených zdrojových kódov. Poskytuje prehľad o vykonanej práci tímu ako aj plánovaných aktivitách.

### **3. Pravidlá písania a komentovania zdrojového kódu**

Účelom tejto metodiky je zadefinovanie pravidiel pre komentovanie zdrojového kódu za účelom generovania technickej dokumentácie pomocou nástroja Javadoc. Jej obsahom sú pravidlá a odporúčania, ktoré by mali byť dodržané vo všetkých vytvorených súboroch obsahujúcich java zdrojový kód. Metodika predpokladá, že pri vývoji je používané prostredie Eclipse, avšak toto nie je podmienkou.

Dodržanie metodík, ktoré sú označené ako pravidlá alebo sú uvedené slovesom "musieť", je považované za nevyhnutné. Metodiky, ktoré sú označené ako odporúčania alebo sú uvedené slovesom "odporúča sa" alebo "malo by byť", nie je nutné naplniť. Rozhodnutie o tom, kedy odporúčanie nasledovať, je zodpovednosťou programátora zodpovedného za danú časť kódu. Účelom odporúčaní je zvýšenie prehľadnosti kódu pri práci s ním, nie tvorba dokumentácie.

#### ***Použité pojmy***

- Eclipse – vývojové prostredie
- Java - programovací jazyk navrhnutý spoločnosťou Sun Microsystems (v súčasnosti vlastnená spoločnosťou Oracle Corporation)
- Javadoc - generátor dokumentácie v HTML z java zdrojového kódu od spoločnosti Sun Microsystems

#### ***Zodpovednosť za kód***

Každý súbor s java kódom musí mať svojho vlastníka, ktorý je zodpovedný za udržiavanie kódu. Pod udržiavaním sa myslí hlavne udržiavanie úpravy kódu, jeho čitateľnosti, konzistentnosti a správnej formy dokumentácie. Zodpovednosť za kód nemusí byť nutne formálne zaznamenaná. Pokiaľ súbor nie je označený, tak zodpovednou osobou je autor prvej verzie súboru, resp. programátor, ktorý kód prvý prebral z externého zdroja. Pri prenose zodpovednosti by mal byť upovedomený celý tím. Vedúci tímu a programátori, medzi ktorými je zodpovednosť prenášaná, musia byť s touto skutočnosťou oboznámený nepodmienečne.

Zodpovednosť za kód môže byť zaznačená v komentári na prvom riadku súboru. Tento riadok obsahuje názov súboru za ktorým nasleduje meno zodpovedného programátora. Príklad: `// source.java Ferdinand Carrot`

#### ***Tvorba dokumentácie***

Dokumentácia je tvorená autorom zdrojového kódu, ktorý ju musí priebežne vytvárať. Pokiaľ sú v kóde vykonávané zmeny, tak úpravu dotknutej dokumentácie musí spraviť autor zmien. Zmeny v dokumentácii by mali byť skontrolované aj autorom pôvodnej verzie kódu.

Dokumentácia môže byť vygenerovaná s rôznym stupňom viditeľnosti. Nezávisle od tejto skutočnosti stále platí pravidlo, že zdokumentované musia byť všetky prvky, nezávisle od použitého stupňa viditeľnosti.

## ***Názvy***

Všetky názvy tried, metód a atribútov musia byť v anglickom jazyku a čo najvýstižnejšie pomenovávať daný objekt. Na zápis názvov je použitý prístup camelCase, čiže názvy zložené z viacerých slov sa oddeľujú veľkým písmenom každého nasledujúceho slova a medzi slovami nie sú medzery. Skratka je považovaná za jedno slovo.

### ***Názvy tried***

Názov triedy vždy začína veľkým písmenom a musí byť podstatné meno vhodne popisujúce obsah danej triedy. Príklad: SampleClass

### ***Názvy rozhraní***

Názov rozhrania vždy začína veľkým písmenom a musí byť prídavné meno vhodne popisujúce účel rozhrania. Príklad: Exemplary

### ***Názvy metód***

Názov metódy sa musí začínať malým písmenom a prvé slovo metódy je sloveso. Názov metódy vhodne popisuje účel metódy. Je odporúčané, aby názvy metód s podobným účelom odrážali ich príbuznosť, napr. všetky metódy slúžiace na výpočet údaju budú začínať "compute", nikdy nie synonymom, napr. "calculate". Pri nasledujúcich typoch metód je použitie určitého slovesa na začiatku metódy pravidlom:

- metódy vracajúce typ boolean, ktorý vraví, či objekt má istú vlastnosť začínajú "is"
- metódy nastavujúce hodnotu zapuzdreného atribútu začínajú "set"
- metódy slúžiace na získanie hodnoty zapuzdreného atribútu začínajú "get"

### ***Názvy atribútov***

Názov atribútu sa musí začínať malým písmenom a mal by byť dostatočne popisný. Názov popisuje na čo slúži daný atribút, pričom je možné použiť aj skratky. Názov konštanty sa skladá iba z veľkých písmen. V prípade viacslovného názvu sú slová oddelené podtržníkom. Príklad: counter, i, CONST\_ARG

### ***Názvy balíkov***

Názov balíku sa musí skladať iba z malých písmen. Názov zodpovedá triedam, ktoré obsahuje. V prípade, že je to možné, by mal názov obsahovať doménu prináležiacu autorovi kódu, aby sa predišlo možným konfliktom. Príklad: sk.stuba.fiit.project

## ***Komentáre***

Okomentované musia byť všetky java zdrojové kódy. Zdokumentované musia byť všetky triedy, rozhrania a metódy, a to pomocou štýlu komentárov určeného pre Javadoc. Všetky komentáre sú písané v slovenskom jazyku bez použitia diakritiky.

### ***Komentáre tried***

V komentároch k triedam musí byť zdokumentovaný účel danej triedy. Ďalej musí byť uvedený spôsob práce s triedou, pokiaľ nie je zrejmý z jej metód alebo uvedeného účelu. Silno odporúčané je uviesť aj jej špeciálne vlastnosti, napr. či je immutable.

V komentári tried môže byť uvedený autor, resp. autori, a to použitím alebo opakovaným použitím tagu "@author". Využitie tagu "@version" sa vo všeobecnosti neodporúča, správu verzií má na starosti externý program. Použitie tohto tagu môže požadovať programátor zodpovedný za daný kód. V tomto prípade je verziovanie povinné a jeho pravidlá určí zmienený zodpovedný programátor.

Pri komentovaní triedy, ktorá je implementáciou rozhrania alebo abstraktnej triedy vytvorenej v projekte je možné použiť dedenie komentáru zo spomenutého rozhrania alebo abstraktnej triedy. Nový komentár je nutné poskytnúť, iba ak sa funkcionality odkláňa od pôvodne uvedenej.

### ***Komentáre rozhraní***

V komentároch k rozhraniam musí byť zdokumentovaný účel daného rozhrania triedy. V komentári rozhrania môže byť uvedený autor, resp. autori, a to použitím alebo opakovaným použitím tagu "@author". Využitie tagu "@version" sa vo všeobecnosti neodporúča, správu verzií má na starosti externý program. Použitie tohto tagu môže požadovať programátor zodpovedný za daný kód. V tomto prípade je verziovanie povinné a jeho pravidlá určí zmienený zodpovedný programátor.

### ***Komentáre metód***

V komentároch metód musí byť opis činnosti, ktorú metóda vykonáva. Opísané musia byť všetky jej argumenty (pomocou tagu "@param"), návratová hodnota (pomocou tagu "@return") a výnimky, ktoré metóda vyhadzuje (pomocou tagu "@throws"). Pre argumenty musí byť zrejmé, čo má byť obsahom vstupných dát. Z popisu návratovej hodnoty musí byť jasné čo predstavuje a aká je v prípade možného neúspechu metódy. Komentár metódy obsahuje aj popis okolností, za ktorých dôjde k vyhodneniu možných výnimiek.

### ***Komentáre atribútov***

Účel atribútov triedy by mal byť zrejmý z ich názvov, takže ich komentovanie nie je v tomto prípade nutné. Komentovať atribút prostredníctvom javadoc komentára sa odporúča len v prípade, že je jeho použitie nejasné.

Konštanty, ktoré boli určené experimentálne alebo je ich pôvod nejasný (napr. fyzikálne konštanty, ale nie hodnota Pi), musia byť riadne okomentované.

### ***Iné komentáre***

Okrem technickej dokumentácie vygenerovanej prostredníctvom Javadoc má dokumentačnú hodnotu aj samotný kód. Okrem komentárov využívajúcich javadoc štýl komentovania sa od



programátora očakáva, že bude komentovať aj jednotlivé bloky kódu pomocou jednoriadkových, resp. viacriadkových komentárov.

Je odporúčané doplniť každý úsek kódu o krátke vysvetlenie, čo je jeho účelom. Ďalej je silno odporúčané pri blokoch kódu, v ktorých sa nachádzajú matematické výrazy, goniometria, fyzikálne výpočty a pod., podrobnejšie uviesť čo je vstupom a výstupom danej operácie a pokúsiť sa stručne opísať jej priebeh.

V časti kódu, na ktorej je potrebná ďalšia práca alebo nie je zatiaľ implementovaná, je vhodné v popise uviesť označenie TODO nasledované krátkym popisom nedostatku. Príklad: `// TODO rozšíriť funkcionality`

V časti kódu, kde bola zistená chyba, je vhodné v popise uviesť označenie FIXME nasledované krátkym popisom nedostatku. Príklad: `// FIXME nezvláda veľké vstupy`

#### 4. Manažovanie chýb

Pri tvorbe jednoduchého programu v rámci zložitého softvérového systému si môžeme byť istí, že počas vývoja sa budeme stretávať s chybami. Tieto chyby je potrebné správne manažovať. Rozlišujeme dva typy chýb, ktoré sa nahlasujú nasledovne:

1. *Defect report from user*: chyby zistené pri testovaní, ktoré vedú k neočakávaným výsledkom, ale neindikujú chybový stav (t.j. padanie programu)
2. *Defect report from developer*: chyby zistené pri vývoji pri ktorých dochádza k chybovému stavu

Jednotlivé chybové správy musia obsahovať nasledujúce informácie:

- a. Popis chyby: čo sa očakávalo (stav objektu alebo hodnota výstupu) a čo nastalo (chybová správa, logy programu)
- b. Presný popis miesta výskytu chyby: názov súboru resp. objektu, názov funkcie, číslo riadku
- c. Popis okolností: vstupné parametre funkcie resp. stav objektu
- d. Ako problém reprodukovateľ

Každá chyba sa nahlasuje cez ticketovací systém tímu na `code.google.com` vedúcemu tímu. Vedúci tímu následne túto správu vyhodnotí a vzniknutú úlohu rozdelí medzi členov tímu.

Životný cyklus spracovanie vzniknutých chýb v ticketovacom systéme:

1. Hlásenie chyby vedúcemu tímu
  - a. stav správy: *New*
  - b. štítky správy (labels): *Type-Defect*, priorita podľa dôležitosti
2. Vedúci tímu vyhodnotí chybu
  - a. stav správy: *Accepted*
3. Vedúci tímu rozdeľuje úlohy na riešenie chyby

- a. stav správy: *Started*
  - b. stav nových úloh: *New*
  - c. štítky nových úloh: *Type-Defect*, priorita podľa dôležitosti
4. Členovia tímu začnú pracovať na riešení chýb
    - a. stav nových úloh: *Accepted*
  5. Členovia tímu dokončili svoju úlohu
    - a. stav nových úloh: *Done*
  6. Každá čiastková úloha je dokončená
    - a. stav správy: *Fixed*

Vedúci tímu sa môže rozhodnúť, že nahlásenú chybu nie je potrebné riešiť. V tomto prípade sa životný cyklus chyby v ticketovacom systéme skončí krokom 3, pričom nie sú vytvorené nové úlohy pre členov a stav správy je modifikovaný na *WontFix*.

## 5. Tvorba plánov

Plánovanie je dôležitou súčasťou každého projektu. Plány slúžia ako smernice pri organizovaní práce v tíme. Zostavenie plánu sa robí spoločne zohľadnením názorov a odhadov jednotlivých členov tímu. Pri zostavovaní plánov sa postupuje nasledovne:

1. vyhodnotenie splnenia plánu za uplynulé obdobie
2. zistenie dôvodov prečo plán nebol splnený
3. odhad zostávajúceho času pre jednotlivé nesplnené úlohy
4. určenie nových úloh s odhadom ich náročnosti
5. preplánovanie
  - a. naplánovanie nesplnených úloh na ďalšie obdobie
  - b. naplánovanie nových úloh na ďalšie obdobie

## G) Zápisnice zo stretnutí

---

### Zápis 1. stretnutia tímu č. 20

Vedúci pedagóg: Ing. Peter Vilhan	
Pritomní členovia tímu: Bc. Maroš Bednár Bc. Adam Brček Bc. Marek Briš Bc. Marian Florek Bc. Vojtech Juhász Bc. Juraj Kosmeľ Bc. Ivan Valenčík	Dátum: 29. 9. 2010 Miestnosť: Softvérové štúdio Čas: 13:00 – 13:30 hod.
Chýbajúci:	Zápis vypracoval: Bc. Adam Brček

#### Téma stretnutia:

Úvod, zoznámenie členov tímu, bližšie oboznámenie sa s témou

#### Vyhodnotenie úloh z predchádzajúceho stretnutia

1. Neexistujú žiadne úlohy, prvé stretnutie

#### Opis stretnutia

1. Zoznámenie členov tímu s vedúcim
2. Zhrnutie informácií o súťažiach
3. Oboznámenie sa s formálnymi požiadavkami
4. Rozdelenie úloh na ďalšie stretnutie

## Úlohy do ďalšieho stretnutia

Číslo úlohy	Popis úlohy	Kto	Termín ukončenia
1.1	Spraviť návrhy pre logo a názov tímu	Všetci	1.10.
1.2	Spraviť návrhy pre plagát	Všetci	3.10.
1.3	Rozdelenie tímových rolí	Všetci	6.10.
1.4	Sprevádzkovanie vývojového prostredia	Všetci	6.10.
1.5	Návrh webovej prezentácie	Maroš, Vojtech	6.10.
1.6	Sprevádzkovanie podporných prostriedkov (SVN, tickety)	Maroš	6.10.

## Zápis 2. stretnutia tímu č. 20

Vedúci pedagóg: Ing. Peter Vilhan	
Pritomní členovia tímu: Bc. Maroš Bednár Bc. Adam Brček Bc. Marek Briš Bc. Marian Florek Bc. Vojtech Juhász Bc. Juraj Kosmeľ Bc. Ivan Valenčík	Dátum: 6. 10. 2010 Miestnosť: Softvérové štúdio Čas: 13:00 – 14:30 hod.
Chýbajúci:	Zápis vypracoval: Bc. Adam Brček

### Téma stretnutia:

Diskusia k analýze existujúcich riešení z minulých rokov.

### Vyhodnotenie úloh z predchádzajúceho stretnutia

Číslo úlohy	Popis úlohy	Kto	Stav
1.1	Spraviť návrhy pre logo a názov tímu	Všetci	Splnené
1.2	Spraviť návrhy pre plagát	Všetci	Splnené
1.3	Rozdelenie tímových rolí	Všetci	Splnené
1.4	Sprevádzkovanie vývojového prostredia	Všetci	Splnené
1.5	Návrh webovej prezentácie	Maroš, Vojtech	Splnené
1.6	Sprevádzkovanie podporných prostriedkov (SVN, tickety)	Maroš	Splnené

### Opis stretnutia

1. Prezentácia vytvoreného loga a plagátu.
2. Rozdelenie rolí v tíme:
  - o Maroš – zástupca vedúceho a manažér testovania
  - o Adam – vedúci tímu
  - o Marek – manažér podporných činností
  - o Marian – manažér vývoja
  - o Vojtech – manažér plánovania
  - o Juraj – manažér kvality
  - o Ivan – manažér rizík
3. Všetkým členom sa podarilo bez problémov nainštalovať vývojové prostredie TORCS aj minuloročný patch pre súťaž Championship.

4. Budeme vytvárať autopilotov pre obidve súťaže – Championship aj Demolition Derby.
5. Dohodli sme sa na ďalšom postupe – prvoradá je analýza minuloročných riešení, porovnanie ich výhod/nevýhod, na jej základe zvolíme koncept nášho riešenia.
6. Vyskytol sa problém ohľadne informácií o kategórií Demolition Derby, z vlaňajšieho prvého ročníka nie sú známe žiadne informácie o účastníkoch.
7. Dohodli sme sa na vytvorení modulárneho riešenia – pilot bude rozdelený do modulov s cieľom využitia niektorých z nich v oboch súťažiach.
8. Na záver sme si rozdelili minuloročných účastníkov na analýzu.

## Úlohy do ďalšieho stretnutia

Číslo úlohy	Popis úlohy	Kto	Termín ukončenia
2.1	Analýza doterajších riešení	Maroš, Marek, Marian, Vojtech, Juraj	13.10.
2.2	Analýza možností nastavenia auta a prenos informácií medzi jednotlivými časťami preteku	Adam	13.10.
2.3	Získanie informácií o Demolition Derby	Ivan	13.10.

## Zápis 3. stretnutia tímu č. 20

Vedúci pedagóg: Ing. Peter Vilhan	
Pritomní členovia tímu: Bc. Maroš Bednár Bc. Adam Brček Bc. Marek Briš Bc. Marian Florek Bc. Vojtech Juhász Bc. Juraj Kosmeľ Bc. Ivan Valenčík	Dátum: 13. 10. 2010 Miestnosť: Softvérové štúdio Čas: 13:00 – 14:30 hod.
Chýbajúci:	Zápis vypracoval: Bc. Maroš Bednár

### Téma stretnutia:

Diskusia k analýze existujúcich riešení z minulých rokov.

### Vyhodnotenie úloh z predchádzajúceho stretnutia

Číslo úlohy	Popis úlohy	Kto	Stav
2.1	Analýza doterajších riešení	Maroš, Marek, Marian, Vojtech, Juraj	Splnené
2.2	Analýza možností nastavenia auta a prenos informácií medzi jednotlivými časťami preteku	Adam	Splnené
2.3	Získanie informácií o Demolition Derby	Ivan	Splnené

### Opis stretnutia

1. Všetci členovia od prezentovali analýzy minuloročných účastníkov.
2. Analyzovanie nastavenie auta spoločne s voľbou ideálnej stopy.
3. Potvrdilo sa, že kategória Demolition Derby sa neuskutočnila minulý rok, čo malo byť po prvý raz.
4. Predbežne sme sa dohodli na určitej voľbe časti riešenia, riadenie spoločne s predbiehaním využiť od tímu COBOSTAR.

## Úlohy do ďalšieho stretnutia

Číslo úlohy	Popis úlohy	Kto	Termín ukončenia
3.1	Hĺbková analýza doterajších riešení, zdrojové kódy pilotov.	Maroš, Marek, Marian, Vojtech, Juraj	20.10.
3.2	Analyzovať výpočet ideálnej stopy	Adam	20.10.
3.3	Získanie informácií o fyzickom modeli	Ivan	20.10.
3.4	Návrh architektúry pilota	Všetci	20.10.



## Zápis 4. stretnutia tímu č. 20

Vedúci pedagóg: Ing. Peter Vilhan	
Prítomní členovia tímu: Bc. Maroš Bednár Bc. Adam Brček Bc. Marek Briš Bc. Marian Florek Bc. Vojtech Juhász Bc. Juraj Kosmeľ Bc. Ivan Valenčík	Dátum: 20. 10. 2010 Miestnosť: D 109 Čas: 13:00 – 16:00 hod.
Chýbajúci:	Zápis vypracoval: Bc. Marek Briš

### Téma stretnutia:

Diskusia ohľadom modulárnej architektúry a návrh jednotlivých modulov.

### Vyhodnotenie úloh z predchádzajúceho stretnutia

Číslo úlohy	Popis úlohy	Kto	Stav
3.1	Hĺbková analýza doterajších riešení, zdrojové kódy pilotov.	Maroš, Marek, Marian, Vojtech, Juraj	Splnené
3.2	Analyzovať výpočet ideálnej stopy	Adam	Splnené
3.3	Získanie informácií o fyzickom modeli	Ivan	Splnené
3.4	Návrh architektúry pilota	Všetci	Splnené

### Opis stretnutia

1. Optimálna trajektória analyzovaná z dostupného dokumentu, vyšla ako nepoužiteľná v našom konkrétnom prípade.
2. Potvrdené, že API a pravidlá sa budú modifikovať len minimálne.
3. Všetci členovia prezentovali svoje návrhy architektúr.
4. Následne sme aplikovali „brainstorming“ pri vytváraní modulárnej architektúry.
5. Špecifikovali sme niekoľko základných modulov a to (posledný je na najvyššej úrovni):
  - Recovery – zabezpečuje návrat na trať po jej opustení
  - Driving – zabezpečuje základné akcie ako plyn, brzda, radenie, riadenie, spojka
  - Line-assist – uchováva informácie o segmentoch trate, slúži na optimalizovanie trajektórie
  - Overtake – modul, ktorý zabezpečuje stratégiu predbiehania

- Defense – tzv. obranný modul, ktorý sa snaží eliminovať kolízie pri návrate na trať či predbiehaní
  - Strategic – rozhoduje o tom, či je potrebné ísť do boxov, či je kritické poškodenie alebo málo paliva
6. Riadenie sa bude zakladať na hodnotách najdlhšieho senzora.
  7. Na záver sme si rozdelili nové úlohy.

## Úlohy do ďalšieho stretnutia

Číslo úlohy	Popis úlohy	Kto	Termín ukončenia
4.1	Implementovať riadenie, pamätanie si trate	Ivan	27.10.
4.2	Analyzovať a navrhnuť „recovery“ modul	Marian, Juraj	27.10.
4.3	Implementovať „recovery“ modul	Marek	27.10.
4.4	Navrhnuť a implementovať ESP	Maroš	27.10.
4.5	Špecifikovať max. povolenú rýchlosť	Adam	27.10.
4.6	Vytvoriť sofistikovaný model trate (mapa)	Adam	27.10.
4.7	Napísať dokumentácia k riadeniu projektu	Vojtech	27.10.

## Zápis 5. stretnutia tímu č. 20

Vedúci pedagóg: Ing. Peter Vilhan	
Prítomní členovia tímu: Bc. Maroš Bednár Bc. Adam Brček Bc. Marek Briš Bc. Marian Florek Bc. Vojtech Juhász Bc. Juraj Kosmeľ Bc. Ivan Valenčík	Dátum: 27. 10. 2010 Miestnosť: Softvérové laboratórium Čas: 13:00 – 16:00 hod.
Chýbajúci:	Zápis vypracoval: Bc. Juraj Kosmeľ

### Téma stretnutia:

Diskusia k splneným úlohám (riadenie, pamätanie trate, recovery modul) – možnosti ďalšieho vylepšenia. Riešenie problematiky a algoritmu vytvárania modelu trate. Diskusia k vypracovaniu dokumentácie - analýzy, špecifikácie, návrhu a riadenia.

### Vyhodnotenie úloh z predchádzajúceho stretnutia

Číslo úlohy	Popis úlohy	Kto	Stav
4.1	Implementovať riadenie, pamätanie si trate	Ivan	splnené
4.2	Analyzovať a navrhnuť „recovery“ modul	Marian, Juraj	splnené
4.3	Implementovať „recovery“ modul	Marek	splnené
4.4	Navrhnuť a implementovať ESP	Maroš	splnené
4.5	Špecifikovať max. povolenú rýchlosť	Adam	splnené
4.6	Vytvoriť sofistikovaný model trate (mapa)	Adam	rozpracované
4.7	Napísať dokumentáciu k riadeniu projektu	Vojtech	rozpracované

### Opis stretnutia

1. Zhodnotenie úloh z posledného stretnutia, prezentácia výsledkov.
2. Návrh a prototyp recovery modulu hotový – potrebné doladiť detaily, optimalizovať rýchlosť.
3. Riadenie, maximálna rýchlosť, „učenie“ zákrut implementovaný prototyp – dohodli sme sa na určení bodov vjazdu a výjazdu zo zákrut, možnosť spojenia za sebou idúcich zákrut – optimalizácia prejazdu zákrutami.

4. Model trate – rozpracovaný. Ďalej sme pracovali na algoritme vytvárania modelu trate, ktorý nereprezentoval trať správne.
5. Dokumentácia – dohodli sme sa dopracovať doteraz vykonané analýzy, vytvoriť špecifikáciu a dopracovať predbežný návrh riešenia, dokončiť dokumentáciu k riadeniu – skompletizovať dokumentáciu na odovzdanie.
6. Rozdelili sme si úlohy na ďalší týždeň.

## Úlohy do ďalšieho stretnutia

Číslo úlohy	Popis úlohy	Kto	Termín ukončenia
5.1	Zdokumentovať analýzu TORCS (Championship, Demolition derby).	Adam	03.10.2010
5.2	Zdokumentovať analýzu minuloročných účastníkov súťaže.	Marián	03.10.2010
5.3	Dopracovať dokumentáciu k riadeniu.	Vojtech	03.10.2010
5.4	Vypracovať metodiku plánovania.	Vojtech	03.10.2010
5.5	Vypracovať metodiku komunikácie.	Juraj	03.10.2010
5.6	Vypracovať metodiku používania SVN.	Marek	03.10.2010
5.7	Vypracovať metodiku písania zdrojového kódu.	Ivan	03.10.2010
5.8	Vypracovať špecifikáciu riešenia.	Juraj	03.10.2010
5.9	Zdokumentovať návrh riešenia.	Maroš	03.10.2010
5.10	Pokračovať v optimalizácii prechádzania zákrut.	Ivan	03.10.2010
5.11	Pokračovať v práci na reprezentácii trate.	Adam	03.10.2010
5.12	Doladiť chyby recovery modulu.	Marek	03.10.2010

## **H) Preberacie protokoly**

---

# TÍMOVÝ PROJEKT

2010/2011

## PREBERACÍ PROTOKOL

### Predmet:

Tvorba informačného systému v tíme

Tvorba softvérového systému v tíme

### časť:

### počet strán:

Dokumentácia k riadeniu

\_\_\_\_\_

Projektová dokumentácia

\_\_\_\_\_

Poskytovateľ – za tím č. 20: Ready2Win

\_\_\_\_\_

Preberajúci

\_\_\_\_\_

Dolu podpísaný preberajúci týmto potvrdzuje, že prevzal predmet od poskytovateľa v uvedenom rozsahu.

\_\_\_\_\_  
poskytovateľ

\_\_\_\_\_  
preberajúci