Vladimir Filipović

Expiriences in defining projects for the course Software Development at Faculty of Mathematics in Belgrade

Outline

- About Software Engineering curricula guidelines and capstone projects
- About course Software development
- Projects in course Software development
- Analysis of Projects in course Software development at Faculty of Mathematics in Belgrade

Software Engineering

Guidelines for creating Software Engineering curriculum

[1] *Curriculum designers and instructors must have sufficient relevant knowledge and experience and understand the character of software engineering.*

[2] *Curriculum designers and instructors must think in terms of outcomes.*

[3] *Curriculum designers must strike an appropriate balance between coverage of material, and flexibility to allow for innovation.*

[4] *Many SE concepts, principles, and issues should be taught as recurring themes throughout the curriculum to help students develop a software engineering mindset.*

Software Engineering

Guidelines for creating Software Engineering curriculum

[5] *Learning certain software engineering topics requires maturity, so these topics should be taught towards the end of the curriculum, while other material should be taught earlier to facilitate gaining that maturity.*

[6] *Students must learn some application domain (or domains) outside of software engineering.*

[7] *Software engineering must be taught in ways that recognize it is both a computing and an engineering discipline.*

[8] *Students should be trained in certain personal skills that transcend the subject matter.*

Software Engineering

Guidelines for creating Software Engineering curriculum

[9] *Students should be instilled with the ability and eagerness to learn.*

[10] *Software engineering must be taught as a problem-solving discipline.*

[11] *The underlying and enduring principles of software engineering should be emphasized, rather than details of the latest or specific tools.*

[12] *The curriculum must be taught so that students gain experience using appropriate and up-to-date tools, even though tool details are not the focus of the learning.*

Software Engineering

Guidelines for creating Software Engineering curriculum

[13] *Material taught in a software engineering program should, where possible, be grounded in sound research and mathematical or scientific theory, or else widely accepted good practice.*

[14] *The curriculum should have a significant real-world basis.*

[15] *Ethical, legal, and economic concerns, and the notion of what it means to be a professional, should be raised frequently.*

[16] *In order to ensure that students embrace certain important ideas, care must be taken to motivate students by using interesting, concrete and convincing examples.*

Software Engineering

Guidelines for creating Software Engineering curriculum

[17] *Software engineering education in the 21st century needs to move beyond the lecture format: It is therefore important to encourage consideration of a variety of teaching and learning approaches.*

[18] *Important efficiencies and synergies can be achieved by designing curricula so that several types of knowledge are learned at the same time.*

[19] *Courses and curricula must be reviewed and updated regularly.*

Software Engineering

Software Engineering Capstone Project

Capstone project course is essential in a software engineering degree program. The capstone course provides students with the opportunity to undertake a significant software engineering project, in which they will deepen their knowledge of many Software engineering areas.

This means development of significant software system, employing knowledge gained from courses. Includes development of requirements, design, implementation, and quality assurance. Students may follow any suitable process model, must pay attention to quality issues, and must manage the project themselves, following all appropriate project management techniques. Success of the project is determined in large part by whether students have adequately solved their customer’s problem.

Software Engineering

Software Engineering Capstone Project

Students should be expected to deliver one or several iterations of a software system, along with all artifacts appropriate to the process model they are using. These would likely include a project plan (perhaps updated regularly, and containing cost estimations, risk analysis, division of the work into tasks, etc.), requirements (including use cases), architectural and design documents, test plans, source code, and installable system.

- It is anticipated that this course will not have formal lectures, although students would be expected to attend progress presentations by other groups.
- The objective of the project would be to solve the customer's problem, and the customer would therefore assist the instructor in evaluating the work.
- It is strongly suggested that students work in groups of at least two, and preferably three or four, on their capstone project. Strategies must be developed to handle situations where the contribution of team members is unequal.

Course Software Development

Software development is elective course, that is held on study program Informatics, in last year of undergraduate bachelor studies.
It is structured like Software Engineering capstone project course – central activity is concentrated around projects.

Course goal : Mastering in software development basic activities, as complex teamwork.

Course content: Introduction to software evolution. Software requirements and software specification. Projecting and development of the software. Software maintenance. Management of the software projects. Human-computer interface. GUI design, building and programming. Event-driven programming. Using of the API. Software tools and development environments. Software verification. Debugging. Testing. Test-driven development. Using RDBMS in programs. Application interfaces. Object-relational mapping.

Course Software Development

Literature*:

1. Horstmann Cay, Cornell Gary: Core Java 2, Volume II – Advanced Features, Sun Microsystems Press - A Prentice Hall Title, 2005.
2. Crawford William, Farley Jim: Java Enterprise in a Nutshell, 3rd Edition, O'Reilly, 2005.
3. Evjen Bill, Hanselman Scott, Rader Devin, Muhammad Farhan, Sivakumar S. Srinivasa : Professional ASP.NET 2.0 Special Edition, Wrox Press, Birmingham, 2006.
4. Robinson Steven: C# 2nd Edition, Wrox Press, Birmingham, 2003.
5. McConnel Steve: Code Complete 2, Microsoft Press, 2004

Classes: 2+3+0

ECTS: 6

Grade: 40% group project presentation, 20% short quizzes, 40% oral exam (includes individual presentation of student’s work on the project).

Course Software Development

Software development course is built around the projects

- Real project for real client who intends to use the software in production.
- Students select their own project, any branch of software development
- Each project should create feasibility study and plan
- Each project should create group presentations and reports
- At the end, project will be evaluated as a whole, and each member of the team will explain his/her part
- Project will be evaluated together with the client representative.

Course Software Development

Contact potential clients

- Gain idea of their expectations
- Estimate scope and complexity of the project
- Discuss business decisions
- If teacher is client, student can expect that expectations that client have are a little bit higher than expectations of regular client

Assemble project team

- Post message to all students in the (in this case yahoo) mailgroup
- Advertise at the beginning of class

Course Software Development

Projects

- Target must be a production system (not research)
- Client should be one or two designated people -- client should be prepared to meet with you regularly and attend the presentations

Team

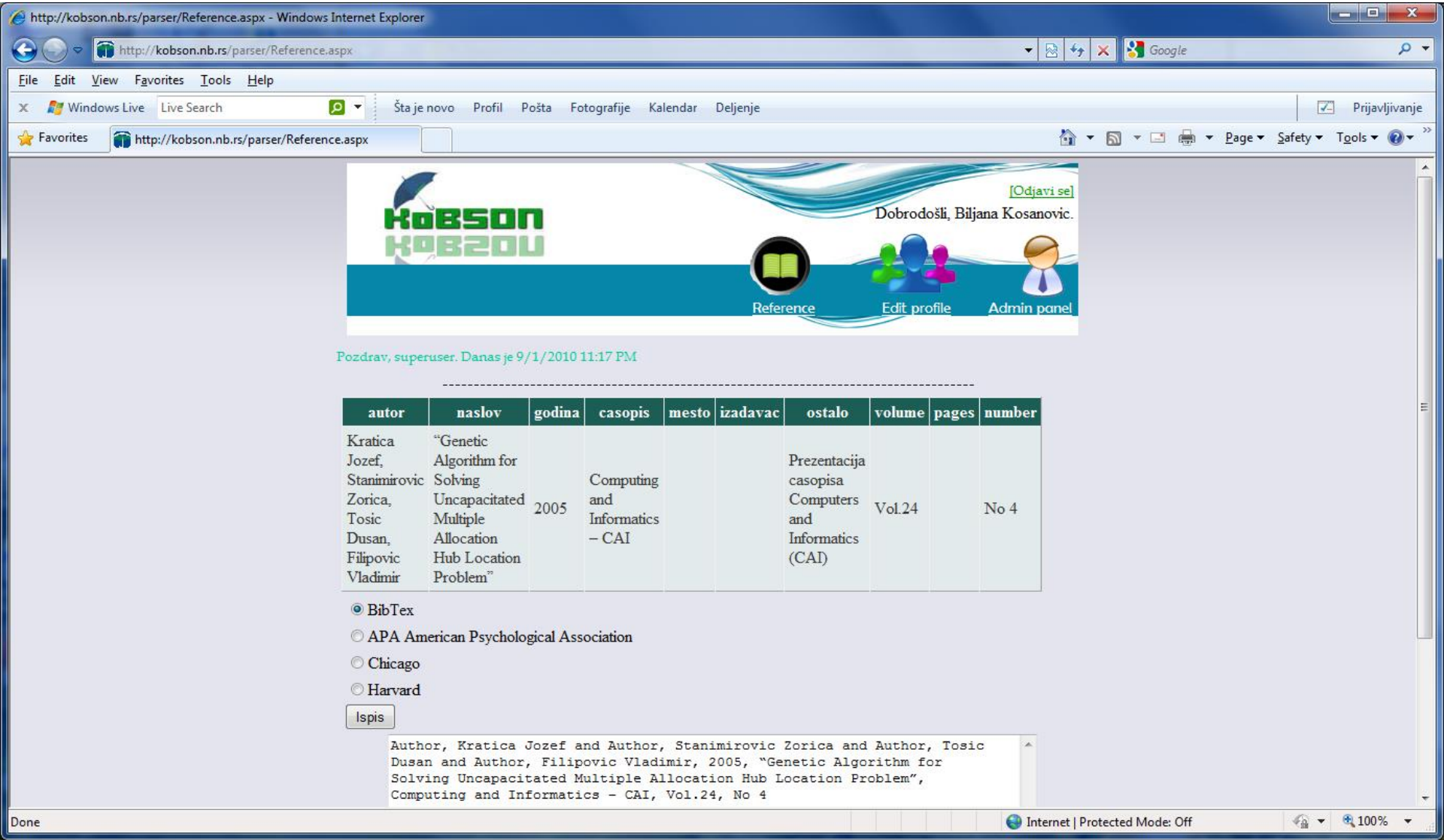
- Teams need many strengths -- organizational, technical, writing, etc.
- Consider appointing a leader to coordinate the effort

Examples of projects

1. Helpdesk System for checking network infrastructure
2. Bibliographic reference parser
3. Terrorist and Organized Criminal Search (TOCs)
4. Document flow and archive system – EOS
5. Informational system supporting orientiring competitions
6. Advanced Windows Performance Monitor for more frequent sampling program execution data
7. Informational system for organizing PR and presentation activities on Faculty of Mathematics
8. Accounting system for small companies
9. Informational system for supporting small tourist agency
10. Informational system for supporting tractor service
11. Informational system for evidencing items in storages
12. Informational system for supporting scheduling medical appointments in the hospital

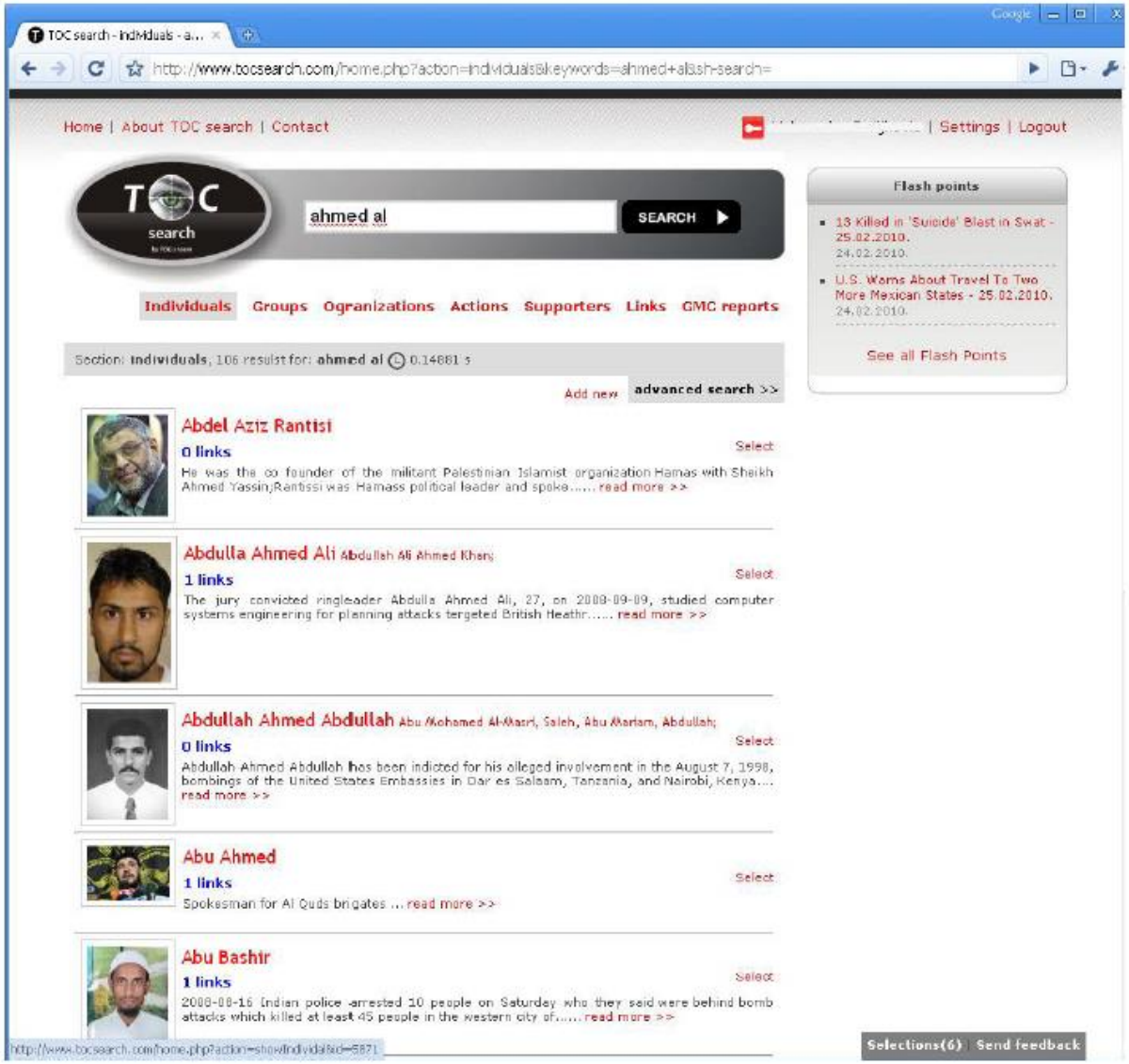
Examples of project (1)

Examples of project (2)



autor	naslov	godina	casopis	mesto	izadavac	ostalo	volume	pages	number
Kratica Jozef, Stanimirovic Zorica, Tosic Dusan, Filipovic Vladimir	"Genetic Algorithm for Solving Uncapacitated Multiple Allocation Hub Location Problem"	2005	Computing and Informatics – CAI			Prezentacija casopisa Computers and Informatics (CAI)	Vol.24		No 4

Examples of project (3)



Examples of project (4)

File Edit View Bookmarks Widgets Tools Help

New tab | Transfers | Speed Dial | ASP.Net Web Application... | Agilno arhiviranje

http://localhost:1754/del/dispatchPages/AgileArchiving.aspx

početna | o nama | portofolio | kontakt | pomoć

Dobrodošli, lennen
Logout

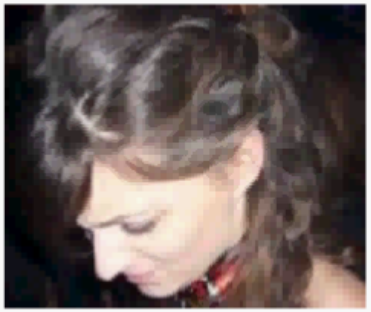
- Lična
- **Administracija**
  - **Korisnici**
    - Registracija Novog
    - Pretraživanje i izmenjivanje
  - **Kategorija podataka**
    - Nova Kategorija
    - Pretraživanje i izmenjivanje
  - **Sektori**
    - Novi Sektor
    - Izmena ili brisanje
- **Delovođa**
  - Arhiviranje
  - Pretraga arhive
  - Agilno arhiviranje
- **Korisnici**
  - **Dokumenti**
    - Pretraga dokumenata
    - Unos novog dokumenta
  - **Predmeti**
    - Pretraga po predmetima
    - Unos novog predmeta
  - **Poslovi**
    - Lista aktivnih poslova
    - Dodeli posao svojim podređenima
  - **Izveštaji**
    - Štampanje izveštaja
- Promena lozinke

Korisnicka stranica

Kruna Matijevic

Zvanje	*student informatike*
Titula	*black*
Sektor	*administracija*
Nadređeni	*Aleksandar Kartelj*
Prava pristupa	*dispatcher*
Telefon	*0953403*

Lisni poslovi *Dispecerski poslovi*



Final remarks

Curriculum guidelines for Software engineering that are marked with red color highly interfere with presented organization of the Software development course.

During his lectures, teacher is concentrated on project-management aspects of the course and teamwork coordination and assistant is concentrated to coding aspects.

Project are developed on same programming platform. If some team choose platform that is different to proposed one, that team is, to some extent, “on its own”. Team members that chose different programming platform also have obligation to know everything that assistant explains on labs.

Course lasted one semester, but teams have almost a year to finish their work.

Final remarks

Projects, project progress and project activities are discussed with teacher at regular intervals. Teacher also decided what should be done in the case of serious problem (e.g. client doesn't support project anymore, team members start to fight among themselves, some team members decided to drop course at the end of semester etc.)

Student from Informatics like that way of teaching. They create various software, that really help their clients.

Teacher should put more effort to properly quantify work of team (sizes and characteristics of real projects are very different).

Teacher should put more effort to properly quantify work of team members within team.

Thank you!

My e-mail address is vladaf@matf.bg.ac.rs.