Competence Centered Specialization in Web Engineering Topics in a Software Engineering Masters Degree Programme

Peter Dolog, Lone Leth Thomsen, Bent Thomsen, and Jan Stage

Aalborg University, Computer Science Department
Selma Lagerlöfs Vej 300, DK-9220 Aalborg-East, Denmark
{dolog,lone,bt,jans}@cs.aau.dk

Abstract. Web applications and Web-based systems are becoming increasingly complex as a result of either customer requests or technology evolution which has eased other aspects of software engineering. Therefore, there is an increasing demand for highly skilled software engineers able to build and also advance the systems on the one hand as well as professionals who are able to evaluate their effectiveness on the other hand. With this idea in mind, the computer science department at Aalborg University is continuously working on improvements in its specialization in web engineering topics as well as on general competence based web engineering profiles offered also for those who specialize in other areas of software engineering. We describe the current state of the art and our experience with a web engineering curriculum within the software engineering masters degree programme. We also discuss an evolution of topics and problems students are working on while learning web engineering skills.

Keywords: Web Engineering, Competence, Skill, Problem Based Learning, Project Oriented Learning

1 Introduction

The Web has witnessed a tremendous growth from a network of simple Web pages to an environment which hosts quite heterogeneous and large scale applications, middlewares, systems and Web sites. Such complex systems require a disciplined and systematic way of development and maintenance. Skills required for such development can be acquired within Web engineering or related curricula. There have been several proposals for such curricula focusing mostly on the body of knowledge approach (see for example [5, 8]). We argue, that the Web engineering curricula should also be viewed from the competences and skills perspective and not only from the body of knowledge which can be provided by lectures.

Aalborg University features a pedagogical model focused on problem based, project oriented, and group based learning. The study programmes are organized into semester topics. Each topic is explored by students in group projects.

The projects are further supported by concepts learned from lectures with exercises organized around the semester topics. By doing so, the central student learning activities are focused on performing problem solving tasks in the group. This engages the students higher cognitive skills and therefore they should learn deeper.

The software engineering masters programme is one of the three study programmes provided by the computer science department at Aalborg University. It has a nominal duration of two years, split into four semesters. It builds upon the software engineering bachelors programme, which is provided in six semesters. Web Engineering as a theme is assigned to one semester exclusively, but it spans another semester dedicated to mobile and distributed software development. Furthermore, students can pursue Web Engineering during their final research specialization year. It should be noted that the view on web engineering in this education is fairly technical and appears to be a sub-discipline of software engineering; but we also have other educations where web development is viewed differently, e.g. in relation to user interaction and human communication.

In this paper we describe an experience in building up and teaching Web engineering skills as part of the masters programme in software engineering at Aalborg University. The rest of the paper is structured as follows. Section 2 briefly describes the foundations for teaching at Aalborg University — Project Oriented and Problem Based Learning — and its specifics developed at Aalborg University. Web Engineering specialization can be obtained as part of the software engineering masters degree at Aalborg University. This programme is briefly introduced in Section 3. Section 4 describes how and where students can develop their Web Engineering knowledge, competencies and skills and how they can specialize in Web related issues. The paper concludes with Section 5 summarizing our main contributions.

2 Project Oriented and Problem Based Learning

History. The pedagogical principles underpinning Problem-Based Learning (PBL) emerged at the end of the nineteen sixties at McMasters University in Canada. The principles were first applied to a curriculum for medical sciences aimed at training general practitioners with a strong focus on a holistic view of humanity and a critical vision on the pedagogical practices of traditional medical training at universities, where medical science had grown into a collection of highly specialized and often unrelated fields.

PBL emphasises development of analytical, methodical and transferable skills. The main idea behind PBL is the use of different methods and strategies to reach the needed knowledge by identifying and solving (real) problems. The learning process is organised in such a way that the students are actively engaged in finding answers themselves. The success of the outcome clearly depends on the experiences of the problem solver(s) and thus requires training in identifying and solving problems suitable for facilitating reaching the learning goals.

PBL is based on a constructivist perception of learning and teaching. Learning is the student's individual process of constructing knowledge and meaning, and teaching is the "setting up of a situation from which a motivated learner cannot escape without having learned" [4]. In other words: PBL is student-centred learning — not teacher-centred teaching, hence introducing a new teacher role and a new student-teacher relationship. A guiding principle for PBL is that the students have the responsibility for their own learning.

Somewhat independently of PBL the notion of Project Oriented (PO) pedagogical principles started to emerge in engineering education as a response to industry's demand for a new competence profile of engineers. In addition to traditional teaching of specialized fields, industry needed engineers who could work together in teams, often with people from other disciplines, thus PO has a strong focus on team-work and an interdisciplinary understanding of subjects.

Today it can be somewhat difficult to separate the two pedagogical concepts; what one university practices as PO another may practice as PBL and the two pedagogical principles supplement each other emphasizing different aspects of learning. Project Oriented Problem Based Learning (POPBL) is today practiced at a number of universities around the world.

From a student perspective, POPBL means working with real life problems, which meets the interests of students and therefore enhances motivation. Additionally, POPBL further develops the students' ability for critical thinking, their problem solving skills and project management skills, improves communication, negotiation and conflict resolution skills, and strengthens analytical and methodological skills, i.e. transferable skills.

Seen from a teacher perspective, POPBL encourages a closer relation between teaching and research — and often teachers learn together with their students.

Finally, from an institutional perspective, POPBL means more motivated students, leading to lower drop-out rates. It also means more competent graduates and improved interdisciplinary collaboration between staff members, improved collaboration between university and industry, as well as a better match between industry needs and graduate skills.

In Denmark POPBL principles were chosen as the underpinning principles at two new universities created at Roskilde in 1972 and Aalborg in 1974.

Aalborg Model. Aalborg University employs POPBL-based studies in a form that has become known as the Aalborg model. This version of POPBL distinguishes between two types of projects: design-oriented or problem oriented projects. In the design-oriented projects the students deal with know-how problems which can be solved by theories and knowledge they have acquired in lectured course. In the problem-oriented projects the students deal with unsolved problems within science and profession. The project-work has a know-why approach and can be supported by relevant lectures [1].

Most students at Aalborg University undertake a master's degree study, with the option of stopping at the bachelor level, though recently several bachelor degree studies have been introduced. In the first year the students learn to work in project-groups. For the next two years in the undergraduate programmes the

project work is mainly design-oriented. During the last two years in the graduate programmes the project work is mainly problem-oriented. Every term has a particular problem/project theme. The students usually divide themselves into project groups of size 5-8 on the first five semesters of the bachelor programme and the first two semesters of the master programme. The last semester of the bachelor programme and the last two semesters of the master programme the group size is usually 2-4 students, with a few students choosing to undertake individual projects. Each group of 5-8 students is assigned a working room full-time. Each semester is credited 30 ECTS, corresponding to 900 hours of study. Approximately half of the time, the students work in the groups on topics within the chosen problem/project theme. This activity is credited 15-18 ECTS, corresponding to 450-540 hours of work. The other half of the time the students attend courses of a more traditional form, usually 3-5 courses credited with 3 ECTS, corresponding to around 90 hours of work pr. course. About half of the courses are directly relevant to the projects, the rest form part of the general discipline. Each project group produces a scientific report (approx 80-150 pages), supervised by a teacher, and submits after typically $3\frac{1}{2}$- 4 months of work. The students are examined orally and individually based on the report. Most courses are examined with traditional (written or oral) exams. Each project group is assigned a member of scientific staff as supervisor for the project. Scientific staff members serve as both project supervisors and as course lectures, but not necessarily on the same semester.

Students seem to appreciate their studies at Aalborg University. In a study among 90.000 students at 550 institutions in 20 countries, conducted in 2008 by the Trendence Institute, Berlin, 90-100% of the IT and Enginering students asked, assessed Aalborg University to be “Good” or “Very good” which was the highest satisfaction rate of students at other Danish and European universities [2]. In 2005 the dropout rate at Aalborg University after 5 years of study was 23% compared to the national average of 56%, and the average study length for a five year masters degree was 5 years, compared to the national average of 6.7 years. Furthermore, industry appreciates graduates from Aalborg University. In a study conducted by the Danish Engineering magazine ”Ingeniøren” in 2004, more than 90% of Engineering companies rated the overall quality of the education as ”Good and Very Good”.

The idea of combining projects and courses is based on the pedagogical philosophy that learning requires activity. Thus the students learn theories and methods in the courses and apply them in their project. In this sense, the projects serve the same role as exercises do in a traditional university education. The difference is that the students only do a single large project in each semester, and in that project they combine elements from the different courses they are taught in that semester; and in most semesters, the projects are based on a realistic problem, and they work to provide a realistic solution. The experience is that the projects are highly motivating for the students. They are very active in the projects, and spend an incredible amount of time on them. They also make the

students show up every day because of the commitment to the other members of the project group.

The downside is that the majority of students focus too much on the projects at the expense of the courses. The official division of work is 50% on courses and 50% on projects, but in practice the project often accounts for 75% or more of the time a student actually spends on the education. This reflects a considerable focus on the specific challenges experienced in the project whereas general problems dealt with in the courses receive much less attention. Some students also become very instrumentalistic in the sense that they only attend course lectures if there are specific topics, methods and solutions that they see as directly useful for their projects. These also tend to read less than in a traditional education. They are more interested in the specific way a solution can be used in their project rather than the general problem and solution.

In this paper we are particularly concerned with the web development semester of the master education in Software Engineering, and that semester has a clear focus on skills required in a software development context. Therefore, the POPBL-based education model may appear to be closely related to the development of such skills. Yet the same educational model is used on the second year, where the focus is on research skills. In that context, the problem of the project is now a research question that the students work with throughout that year, and the final project report is a classical master thesis with a strong emphasis on research.

3 Software Engineering Masters Programme

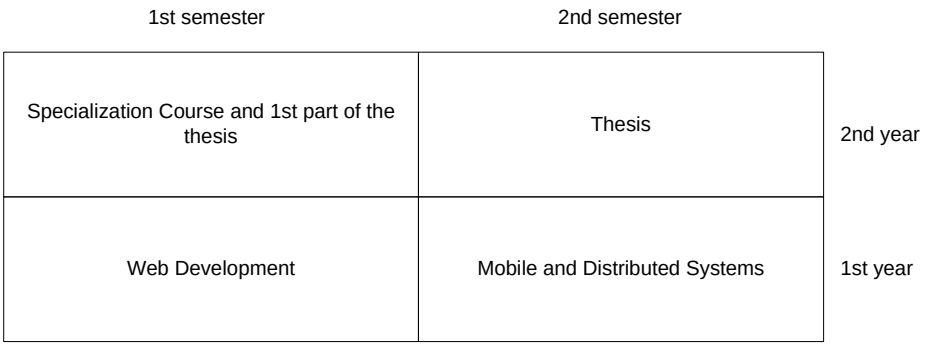

Fig. 1. Software engineering masters programme at Aalborg University

The software engineering masters programme builds upon a bachelors programme in software engineering. During the bachelors programme students are introduced to a wide range of subjects relevant for Web Engineering, including: Object Oriented Programming; Object Oriented Analysis and Design; Software

Architectures; Design, implementation and evaluation of User Interfaces; Test and Verification of Software.

The software engineering masters programme has a nominal duration of four semesters. The topic of the first semester is Web development with the main focus on Web application engineering projects. The second semester is devoted to mobile and distributed applications with the main focus on application building for mobile platforms. The third semester extends to a specialization topic chosen by the student(s) and is performed as a semester project together with literature reading and presentation seminars. The last semester of the programme mainly consists of master thesis research and writing. This scheme is depicted in Figure 1.

There are two main aims of the software engineering masters programme at Aalborg University. The first is to broaden the skills of software engineering graduates mainly towards building more complex systems than they are used to after obtaining a bachelor degree. This is achieved in the first two semesters. These two semesters should also serve as additional input for students to decide about their final research specialization. The second aim is to specialize through research based on the students choice. This is achieved in the last two semesters of the masters degree programme.

4 Web Engineering Topics in the Masters Degree Programme

Motivation. There are two main motivations for having a Web development semester in the software engineering masters programme:

- *Labour market* — Many software applications are developed or are now being ported to the Web so there is a market and a need for software engineers with Web engineering skills;
- *Complexity* — Web applications or Web systems are complex, distributed systems for a diverse audience which brings new challenges to building and maintaining them.

The main outcomes to be achieved at the competence level are that the students are able to:

- demonstrate knowledge and understanding of Internet, Internet technologies, and Internet services
- demonstrate skills in development of an Internet application, agent, or service

Projects and Results. It seems that the market and trends in what is build for customers have been the main drivers of student interests in choosing a problem to be solved in the Web development project. The topics of the projects have ranged from Web shops and enterprise resource planning systems a few years ago to, more recently, intelligent homes controlled over Internet of things middlewares. No matter which of the topics the students have chosen, the projects provided a good foundation for building up their skills in:

– *Data preservation and processing* — different options for storing data in relational database, XML technologies and its use, and so on
– *Business logic in a broader sense* — describing and representing computation even in a distributed fashion in various languages and platforms (java, enterprise java beans, .Net, Web services, middleware, security and privacy, brokers, and so on)
– *Presentation* — designing and implementing advanced user interfaces following established HCI principles and employing and experimenting with new technologies (silverlight, rich Internet applications, Google Web toolkit, java server faces, and so on)
– *Team coordination* — developing advanced applications in an organized manner following modern methodologies (document centric or agile).

As examples of projects we include the students own descriptions of their projects:

Easy Clocking — A system for automatically clocking in and out employees. "Easy Clocking is a system capable of automatically clocking in and out employees based on presence at workstations. Embedded devices are configured and software is developed to detect the presence of butchers at workstations using a localisation technology. We analyse and compare a number of different localisation technologies from which we choose Bluetooth. Its applicability is evaluated by conducting experiments. A Web application is developed in order to view registered location information and to solve conflicts, such as if the butcher is detected at multiple workstations at the same time. Work has been done with emphasis on usability and flexibility of the Web application, making Easy clocking effective to work with and allows for customisation according to the needs of the company deploying it, respectively [7]."

Duelco — Company Web Shop Integration. "This report documents the development of the new Web shop for the company Duelco. The system is based on a 5-tier architecture, and is designed with emphasis on flexibility and usability. The system is based on an analysis of Duelcos current system and their requirements. Regarding flexibility, the tier interfaces are designed using XML Schema, and all communication is done using XML. Furthermore the project focuses on the use of XML and related technologies. Although the use of XML caused some problems, overall it helped the development. Despite software license limitations, the final system ended up as a working prototype thanks to the loosely coupled 5-tier architecture, which meant that only two tiers had to be changed to get a working system [3]."

Analysis of Projects. We have analysed 16 projects out of those which have been completed on the Web development semester. Six were new Web shops or integration between Web shops to achieve a business advantage. Two were Web applications related to retail customers in different ways than Web shop. Three were focusing on games or media. One project was an editor extension for a content management system. Finally, four projects focused on mobile applications

or Internet of things. The projects mostly used .Net technology (11 projects). Four projects used Java related technology, two projects used Ruby or Google Web Toolkit technology respectively, and one project used PhP technology. Regarding the methodology used for the project implementation, the majority of groups chose an agile methodology based on XP or SCRUM or tailored their own methodology with elements of agile and traditional methodologies (11 projects). Seven projects followed a rather traditional document centric methodology.

Furthermore, project based learning has also enabled settings for various studies. Students of two consecutive instances of the semester have been asked to follow the ADRIA [6] methodology to study understandability, applicability and suitability of the methodology for various tasks. Another example of research on that semester is studies of various programming languages for different aspects of the developed systems and algorithms used for different computational problems. The students also most often follow user interface design evaluation based on established HCI principles. They usually use the HCI lab to perform their evaluations, involving real users. The semester projects also featured a number of university-industry collaborations. The focus in these projects was mostly some advanced applications of new technologies which a company wanted to explore on their types of projects, looking at feasibility, consequences of introducing new technology and which features it supports.

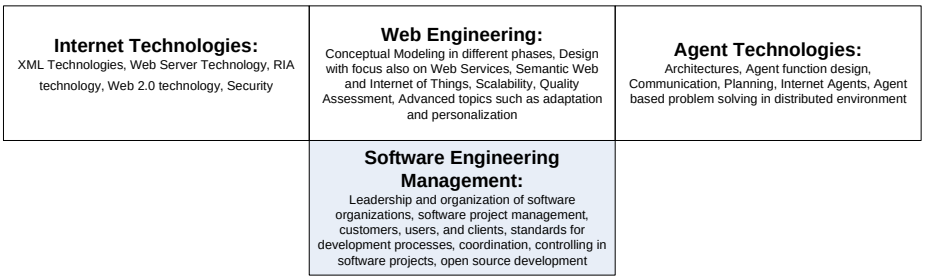

Fig. 2. Web Engineering Semester courses at Aalborg University

Courses. Based on the experiences from teaching on the semester, lectured courses evolved as well. Figure 2 depicts current lecture modules provided for students on that semester. There are three courses which are directly related to the development projects. They provide conceptual foundations for technologies, methodologies, modeling, concepts and agents on the Web. Software Engineering management is shaded because it is a general advanced engineering course which broadens the student's knowledge beyond the project management area. Especially the three project related courses were updated quite frequently according to wishes of the students and their project topics as well as changes in the respective fields of study. The latest additions to the curricula are the Web engineering and agent technologies courses. The motivation for the first course

was to introduce Web specific conceptual, modeling, and methodological issues into the students competences. Besides other things, students are introduced also to service oriented architectures and business process modeling. The slot on advanced topics in the course provides a flexibility for new directions in the area which might in the future include for example cloud computing and other movements. The course was evidently needed from student evaluations from previous and current instances of the semester. The lecture slot was previously provided for more application oriented topics such as ERP systems, especially due to the large market share of software companies in the area and also due to the problems previously chosen in student projects. Instead of the Agent Technologies course, students previously had an opportunity to follow a programming paradigms course, supporting their experimentation with rule based, functional, and aspect oriented languages. Recently, the programming paradigms course has been moved to an earlier semester. Similarly, students previously experimented with usability evaluation and user interface design methods. Therefore, the semester offered a course on design and use of user interfaces to provide the foundation for building Web applications. Recently, this course has also been moved to an earlier semester and replaced with an advanced software engineering management course.

Other Semesters and Web Engineering. The follow up semester on the software engineering masters programme features mobile and distributed applications. It is a natural extension of Web technologies with presentation options and computing realized not only on desktop, server side, or Web browser platforms but also on mobile devices and technologies. The semester features courses such as mobile systems technologies which include programming techniques and languages for mobile platforms as well as usability design and evaluation with mobile platforms. Other courses on this semester are rather of broadening character and do not directly relate to Web engineering.

The Software Engineering bachelor education offers various prerequisite courses necessary for Web engineers. It includes distributed systems, networks, operating systems, software engineering, programming and others which are general and required.

Web Engineering Specialization. At Aalborg University it is possible to specialize in Web engineering when choosing one of the specialization topics related to Web. Specialization is further possible in *programming for the Web* where there are currently a number of problems offered to be solved by master students in their research project in their final year. This includes for example new declarative markup languages, combining object oriented languages with declarative ones, or programming rich Internet applications. Students may choose from the topics offered within *intelligent Web and information systems* which are typically projects solving problems in the areas of adaptation and personalization strategies on the Web, adaptive middleware for service integration, or Web engineering concepts and methods. *Human computer interaction and usability evaluation for Web portals* is another area where students can also specialize within Web engi-

neering. This includes usability methods and studies and advanced user interface design concepts in different Web applications. *Database management on the Web* is the area where students focus mostly on location aware services and connected data management issues as another specialization. *Mobile and embedded systems* specialization offers students a possibility to specialize in formal methods, test and verification and concurrency for such applications even in connection to the Web.

5 Conclusions

We have described how we have incorporated various training on Web engineering related topics into the software engineering masters degree programme at Aalborg University. We have described the context in which this training happens: project oriented problem based learning. We have also described how the Web engineering specialization takes part in other software engineering courses and problems. We argue, that such a model provides a high flexibility, as evidenced by the evolution of the programme, for changes needed either due to change in students interests, software industry interests, or changes in technology. We argue, that due to the inherent properties of problem based learning, this teaching style provides more competence based learning and deeper understanding required by industry and also necessary for those students pursuing a research career.

References

1. http://studyguide.aau.dk/workmethod.
2. http://www.ingenioer.aau.dk/godt+studiemilj%c3%b8/stor+tilfredshed/.
3. Mads Schaarup Andersen, Kristian Bødker, Martin Madsen, Daniel Rødtness Poulsen, and Mads Vøge. Duelco - company web shop integration. Technical report, Deaprtment of Computer Science, Aalborg University, 2005.
4. John Cowan. *On Becoming an Innovative University Teacher*. Buckingham: The Society for Research into Higher Education and Open University Press, 1998. ISBN 0-335-19993-3.
5. Yogesh Deshpande. Web engineering curriculum: A case study of an evolving framework. In Nora Koch, Piero Fraternali, and Martin Wirsing, editors, *ICWE*, volume 3140 of *Lecture Notes in Computer Science*, pages 526–530. Springer, 2004.
6. Peter Dolog and Jan Stage. Designing interaction spaces for rich internet applications with uml. In Piero Fraternali, Luciano Baresi, and Geert-Jan Houben, editors, *ICWE2007: International Conference on Web Engineering*, volume 4607 of *LNCS*, pages 358–363, Como, Italy, July 2007. Springer Verlag.
7. Christian Frost, Casper Svenning Jensen, and Kasper Søe Luckow. Easy clocking - a system for automatically clocking in and out employees. Technical report, Deaprtment of Computer Science, Aalborg University, 2009.
8. E. James Whitehead Jr. A proposed curriculum for a masters in web engineering. *J. Web Eng.*, 1(1):18–22, 2002.