Supporting Competence Management in Software Projects

Jürgen Dorn[1], Markus Pichlmair, Kathrin Schimper[2], Hilda Tellioğlu[3]

[1]*Vienna University of Technology Institute for Software Technology and Interactive Systems, Favoritenstraße 9-11, A-1040 Wien, Austria, juergen.dorn@ec.tuwien.ac.at*

[2]*Vienna University of Technology, Institute for Information Systems, Favoritenstraße 9-11,A-1040 Wien, Austria, {pichlm@dbai.tuwien.ac.at,* kathrin.schimper@gmail.com}

[3]*Vienna University of Technology, Institute of Design and Assessment of Technology, Argentinierstraße 8/187, A-1040 Wien, Austria, hilda.tellioglu@tuwien.ac.at*

Abstract

Competence management is used today for different purposes: company and project staffing, competence detection and competence development for single persons, whole companies or virtual organisations. We have developed a university competence management system enabling students to plan and control their further study based on their current competence profile. The final profile can be used for job applications. In this paper we describe experience made in a special university course on software project management capturing and developing students' competencies relevant in this area. This approach would be applicable also in commercial software projects. We present the first approach, its implementation and evaluation before we propose an extended version.

Keywords

Competence management, software project management, competence capturing

1 Introduction

Human resource management as a function of enterprises has changed evolved due to changes in our society, which is becoming more and more knowledge-based. Society and companies expect more flexibility from human resources. Human resources are not anymore life-long bound to single companies and often especially high-skilled people work as "single-person" companies. In the past, a person was expected to pass some professional education and then to work for a single company. Thus it was sufficient to have some certificate stating that one has passed successfully a professional education. Today it is expected that employees have certain competencies to achieve tasks but moreover, these competencies of employees are seen as a strategic asset of a company (Prahalad and Hamel, 1990).

In companies competencies are used in several ways. If tasks have to be performed or staff is selected for a project, the competencies of the staff can be used to find the appropriate person. Required competencies can also be used to specify new personnel to be hired. Sometimes companies set the salary of employees based on their competencies they actually apply (Tucker and Cofsky, 1994). Competencies required are often derived from the organisation's strategy. A company may derive by means of a gap analysis which competencies are missing. The explicit reasoning about human resources and the systematic development of their competencies is called competence management (Lindgren et al., 2004). Competence management is also required in virtual organisations or projects where several companies are involved (Dorn, 2006). An inter-organisational project may be initiated because competencies of several organisations are required and thus the competencies of an organisation and its employees should be disclosed to other organisations.

Competence management covers four main tasks (North and Reinhard, 2005):

- Representation (competencies of the staff and the core competencies of the company have to be documented),
- Reflection (competencies have to be captured, measured and evaluated),
- Sharing (competencies referred to different departments or processes have to be available at the right location at the right time) and
- Development (competencies have to be developed to be more efficient in carrying out current work and to be prepared for the future).

Another important task in competence management are difference analysis (including gap analysis between present and future competencies on personnel and organisation level).

One of the most difficult tasks in competence management is to evaluate competencies (Fletcher, 2001). First, there must be some common understanding what a certain competency means. Many classifications and competence catalogues are available. Typically (hard) technical competencies (e.g. database management) and soft skills (e.g. communication skills) are distinguished. Second, there must be some convention how a competency is measured. These may be a metric scale or some behavioural indicators as proposed in (Tucker and Cofsky, 1994). Furthermore, we may distinguish evaluation by experts and by peers. Third, the evaluation procedure must be trustworthy. Therefore supervisors with experience in evaluation are often asked to take the responsibility of evaluation. Unfortunately, Especially this is not always possible, especially very difficult in project-oriented work.

We have developed a first prototype of a competence management system for universities (Dorn and Pichlmair, 2007). The objective is to store students' competency profiles and to update these profiles when a student has successfully passed a university course. A student's competence profile can also be used to decide whether s/he has the required competencies to participate in a course. In this paper we consider a special course where students shall learn theoretical knowledge and practical skills in software engineering. Since the course is organised similar as a commercial software project, participants take over different roles requiring also different competencies, e.g. project management.

In the following we describe briefly the state of the art in competence management systems and project management. The third section describes our approach in measuring competencies for software projects. A first evaluation in the course was performed in last winter term. Some results and open problems with the distribution of roles in the course are shown in the fourth section. Finally, we conclude with a discussion how these problems may be solved in future.

2 State of the Art

To support competence management many companies use competence management systems implemented on different detail levels. Competence management systems have to fulfil several requirements concerning the content (how detailed the competency model is developed), the technical implementation (which data is kept when and how data are kept up to date) and the organisational implementation (who implements the system, how are people motivated to use the system and to keep data up to date). Draganidis and Mentaz (2006) give an overview of existing competence management systems. In the following we sketch only three examples.

Microsoft implemented a competence management system by defining more than 300 competencies in four categories (foundation, local and unique, global, and universal skills). Basic, working, leadership, and expert are the terms for different competency levels. Staff members are rated by themselves and by their superiors within an iterative process. Microsoft expects better matching of employees to jobs and work teams. Moreover, they expect that employees will know better what competencies are required and thus are better consumers of educational offers (Davenport and Prusak, 1998).

Before introducing a competence management system at *Ericsson*, they had individual solutions in each country, sometimes paper-based or based on databases or spreadsheets. They extended the existing SAP R/3 Personal Management Module and implemented a competency catalogue. For the grades there exists a scale from 0 (not assessed) to 9 (excellent). The values are evaluated by using the period of time a person has used the competency (Hustad and Munkvold, 2005).

Zelewski et al. (2005) describe a competence management system designed for supporting the product development process in engineering networks consisting of different organisations. To support the understanding of competencies and their development between organisations an ontology is used. They match tasks that require certain competences with those of humans.

Dorn and Pichlmair (2007) have developed a prototype of a university competence management system. The system is based on an ontology defining competencies, evidence types and jobs in information systems and computer science domain. The system distinguishes competencies in knowledge and an experience aspect. To decide on a competency, evidences such as passed examinations, books read by the person, trainings, assessments, project work, e-learning courses and more are stored. Each competency can be evaluated on three levels: beginner, advanced and expert and on each of these levels a finer scale exists with real values between 0 and 1. The competence of a person is computed by the system at the point of time when it is required and all evidences until this moment are used for the computation. A gap analysis algorithm determines differences between actual profiles and desired goal profiles (Pichlmair, 2008). The computed gap is used to recommend further courses to a student. This system was also used in the described experiments in this paper.

Software projects are projects in which software applications are produced to solve current problems of users. Besides the distinction between system, programming and application software, one may categorise software systems by considering their target group. They may be implemented for a specific customer or user group, or as a standard tool. In most cases, the high complexity of the system or limited personnel, temporal and financial resources require team working. Members must have certain competencies in order to be able to carry out different tasks in all phases of the design process, like analysis, coding, evaluation and deployment. In software engineering there are several process models like waterfall, iterative and incremental (like spiral, prototyping, V-model, unified process), agile development (extreme programming, Scrum), test-driven or model-driven development etc. Processes, logical and temporal order of tasks, artefacts created and used during the design and development process, communication and cooperation practices are different in these process models. Depending on the type of the project, project manager normally choose the most appropriate process model, which again depends on the competencies project members have.

Software projects must be managed. This starts by planning human and non-human resources, timeline, (intermediary) results, continues during the project execution until planned results are delivered and the project has successfully been accomplished. Risk management, quality assurance, monitoring, documenting are some other activities necessary for project management. Project managers make use of several methods (like critical path method or Gantt diagrams) and of software systems to ease their work, e.g. they use project management tools or simple spreadsheets. They draw diagrams and time plans to visualise their plan and update the plan data during the project to be able to recognise problem areas in the accomplishment of the project.

Issues concerning competence management and software project management are not well described in the research literature. Rose et al. (2007) investigate software project management in a medium sized company to develop a competency-based view on project management. They use qualitative methods like questionnaires and focus groups to develop a competence pyramid comprising the most important seven main competencies for a project manager.

3 Managing competencies required for software projects

The competence management system is applied and evaluated in a course on software engineering and project management at Vienna University of Technology (Schimper, 2008). The objective is to measure competencies and the enhancement of these competencies in the course by self-assessment, peer-group assessment and assessment by a supervisor. Assessments are implemented by structured Internet questionnaires. As a side effect, we achieved a tool supporting also students and supervisors, by giving them feedback after each questionnaire and a detailed analysis of the project progress and the teamwork at the end of the course. On one hand, this helps professors to grade students. And on the other hand, this supports students choosing their roles at the beginning, and at the end it gives them feedback about which competencies they have enhanced and how the project and the teamwork has been rated by the whole team and the supervisor.

With regard to further use of the method and the large number of students participating in the course, it was necessary to choose a method, which needs little time and effort. Therefore a one-to-one interview could not be considered and a questionnaire has been chosen. Furthermore an online questionnaire was selected, because we assume that all participating students have access to the Internet. The disadvantage is that we cannot interrogate so closely particularly with regard to the behavioural competencies and the teamwork.

In the following, we describe the course, the roles students can choose for the project and typical tasks to be solved in the project. Afterwards we explain which competencies are considered and how students' competencies are measured.

3.1 Course Description

"Software Engineering and Project Management" is a course with 6 ECTS for bachelor students in the curriculum of computer science and information systems at Vienna University of Technology. It is about practicing object-oriented programming in a team of six students with project management and groupwork activities. There are several predefined roles and responsibilities like project management, system architecture, testing, user interface and interaction design, programming, documenting. Each student has to carry out each type of activity in the project by being responsible for one area where s/he has to coordinate this activity in the whole group (see Figure 1). This characterises the project work as a course, which would not be the case in a real software project. There is a supervisor coaching the group. Together with the responsible lecturer s/he tries to manage the groupwork and to keep deadlines and the quality of artefacts and the software system that the group has to implement.

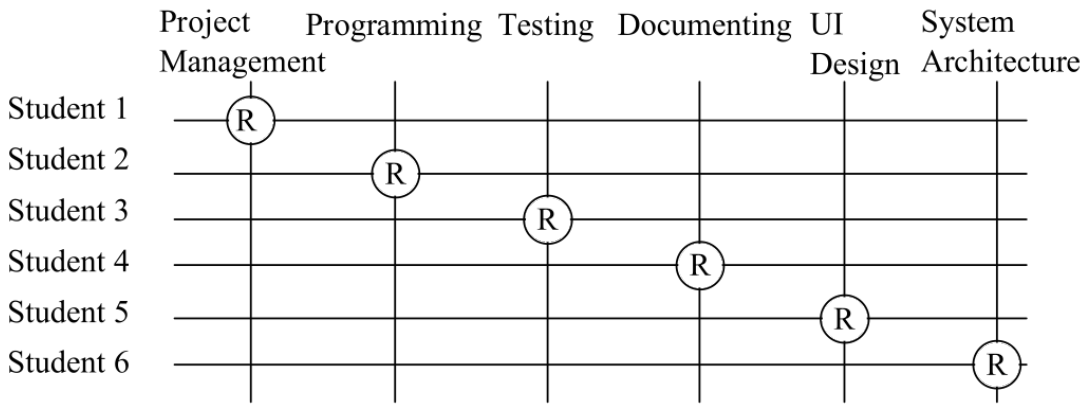

Figure 1: Roles and responsibilities of students in the course.

Guidelines describe how to create the artefacts. In an introduction phase students work alone and proof their ability to program. They have to pass an exam to move to the groupwork phase, which takes three months and has three milestones. The temporal structure is given, the artefacts to deliver are predefined, and the project content is defined by the group itself.

3.2 Capturing and Measuring Competencies

We distinguish functional and behavioural competencies and both have two different values, knowledge and experience (Dorn and Pichlmair, 2007). The main competencies identified for this course and used for the measurement are shown in Table 1.

Functional competencies	Behavioural competencies
Project organisation and management competencies	Communication competencies
Requirement analysis competencies	Team competencies
Data modelling and design competencies	Leadership competencies
Implementation/Engineering competencies	
Test competencies	

Table 1: Evaluated competencies

Evaluation of competencies is based on questionnaires. Students answer a questionnaire at the beginning and another one at the end of the course. Further, a supervisor fills out one at the end of the course. The first form asks for the competencies students already have. After the first form students get a feedback about the level they already hold for every role in regard to the competencies they can improve in this course. This is accomplished by assigning competencies to roles. For example the project manager needs project management, leadership and special communication competencies. A table in the feedback shows the levels the students have in the beginning. The levels are divided in four categories: high, average, low and no competency. They effect only the course internal competencies, that means if a student has the highest level of one role, s/he could not or rather little enhance his/her competencies anymore with this course, if s/he chooses this role. Students could then either choose a role they are good in to upgrade their competencies or choose a role they have no or little experience in acquiring new competencies.

At the end of the course students are asked about their competencies and for the progress of project and teamwork. The final feedback for the students consists of two parts. The first part shows which competencies were improved and which were acquired newly. The second part shows a summary of project progress and teamwork estimated by the team and by the supervisor. The feedback consists of textual information, tables and charts visualising the results.

The course supervisor gets a form at the end of the course with questions about the progress in project and teamwork. The supervisor and the professor also obtain feedback at the end to assist the grading. The feedback gives information about the progress of the project and the teamwork.

Questionnaires	**Time**	**Content**
Questionnaire Students	Start of course	• Functional competencies • Behavioural competencies
Questionnaire Students	End of course	• Functional competencies • Behavioural competencies • Progress of the project • Teamwork
Questionnaire Supervisor	End of course	• Progress of the project • Teamwork

Table 2: Questionnaires in software engineering and project management course

4 Evaluation

For the first evaluation we have chosen one student group. In this course, students have determined one weekday to meet their supervisor and outline the project progress. During the course students answered two questionnaires, one at the beginning and one at the end of the course. Furthermore the supervisor has completed one questionnaire at the end of the course. In the following, the results of the questionnaires and the problems of the method are discussed.

4.1 Results of questionnaires

The result of the first questionnaire demonstrates that the competencies, which the students already own, differ. First of all, it depends on students' interests. Another reason is that some students have already been working before in commercial software projects. So they have already acquired considerable competencies in software projects.

The reporting of both questionnaires regarding the enhancement of the functional competencies has been visualised with bar charts, which show both the competencies in the beginning and their further development. In the following we show an example chart, where one can see the behavioural competencies of three students of one team before the course, after the course and the assessment of the team members (Figure 2). If we take a closer look, we can see that one student had assessed his competencies accurately. Another student had overestimated his competencies and the third student has underestimated his competencies with regard to the assessment of the team members.

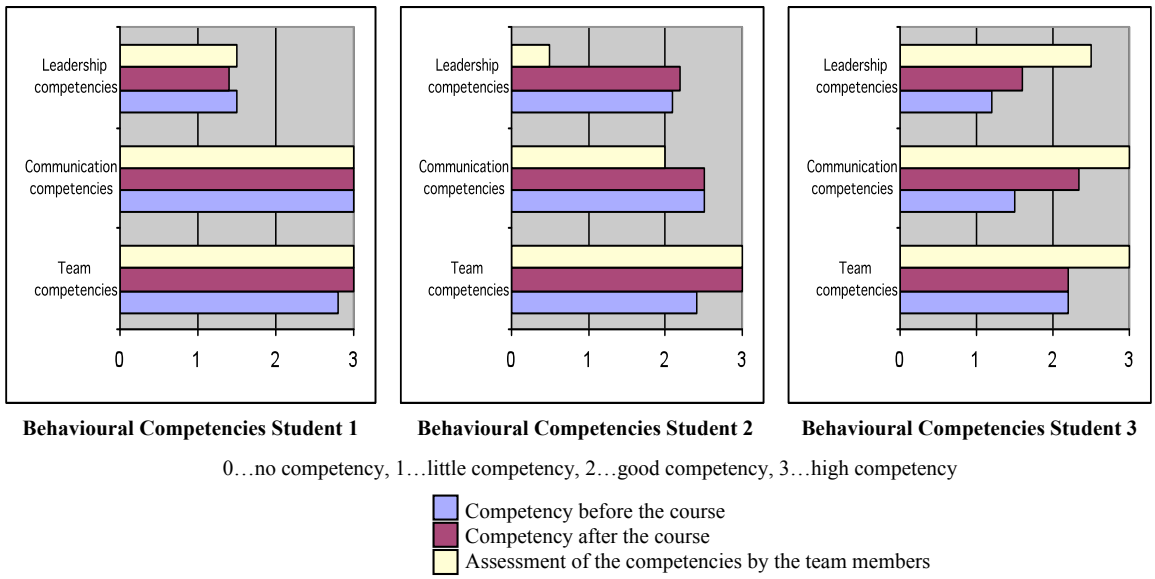

Figure 1: Behavioural Competency Profile of three different students.

The difference between functional and behavioural competencies is that the functional ones are easier to measure. It is more difficult to measure for example the decision-making ability of a student, than measuring his/her Java programming skills. Another important factor is that it is easier for students to assess their programming skills than to assess their decision-making ability. The self-assessment of students is a fundamental and necessary point of our approach. If a student cannot or will not assess his/her competencies appropriately, this will distort the results. This does not require that students lie about their competencies, it means that students do not know exactly how to assess their competencies correctly. The results show that the students enhance their competencies, but also change their minds about the value of their competencies. For example, a student answered the question "Please rate your programming skills in Java" with "excellent" in the first questionnaire. During the course s/he had problems with the implementation in Java and therefore s/he answered the question at the end only with "very good" in the second questionnaire because s/he is not sure about the excellence anymore. One could assume that his/her competencies decreased, but we think that only his/her self-assessment has changed. This is a problem for automated self-assessment, because we cannot inquire as in a personal questionnaire. For this reason, the behavioural competencies have been additionally measured by asking team members, so there are two views on these competencies, which could be compared. This leads to another problem. The assessment of team members not only depends on the objective, realistic view. It has also to do with sympathy, age, gender or culture. Such measurement failures must be considered in an evaluation. Another important aspect is that competencies, in particular behavioural competencies, can change regularly, because of various

influences like workload of students and other experiences. Therefore it is almost impossible to say that one competency has been changed in one special course.

The results of the last two parts of the second questionnaire, progress of the project and team work, show, presented with tables and textual information, two views on these parts, the view of the team and the view of the supervisor. The results of the project part demonstrate how satisfied the students, respectively the supervisor, are with the output of the project, what could be improved and which problems or risks occurred. The evaluation of the team part shows how the students and the supervisor think about the team work, how they think about the performance of each role, how the team members prove themselves in the different phases of the project and if there where any conflicts or communication problems and why. It is possible to see if there were weak, strong or equal members in the team. The results of these two parts give a good overview of the project and the team and all problems become obvious.

To prove all matters, which are exposed above, and to do more comprehensive evaluations, like what competencies and about how much could these be enhanced with this course in general, another evaluation must be accomplished, with more teams respectively more students.

4.2 Problem with persons in wrong roles

Due to the learning aspect of the course it is quite common to have persons with a role that do not fit. For example, if the project manager is not able to manage the team and control meetings, the team could at least loose a lot of time, which they will miss badly at the end of the project. Another example, which occurred in the test team, is about conflict management. If a conflict evolves and the project manager is not able to find the reason, to mediate or to solve the problem, the group can split up or one member could be cold-shouldered. But the more problems occur during the project, the bigger is the learning effect. Furthermore, supervisors accompany the teams along the course to support the team and assist them if any problems occur. So this is a known problem, but it should not be avoided due to the learning aspect.

5 Conclusions and Further Work

We have presented an experiment in measuring competencies in a course on software engineering and project management. The basic purpose is to document the competencies achieved by students in the course and to supply these to a competence management system. At the moment, required competencies are checked in the beginning of the course. With a competence management system storing competencies in every university course this check would be not necessary.

Further advantages in courses could be the controlling of the competence development on a detailed level in the course and scoring by using this system. Also the transparency for students, that they know what we expect from them in a course, is a pro. A difficulty however will be always the subjectivity of evaluations. This can only be overcome by redundant evaluations by different persons and in different courses.

Further advancements are directed to management of problems occurring in a course or in a project. Projects are difficult to plan and we have to react on occurring problems. If for example, a student finishes the course early, other students have to overtake his role. Knowing the competencies of other students better can facilitate this role shift. However, to react on problems as early as possible, we should monitor the behaviour of the group, e.g. by analysing the communication patterns and observing participatory (apply ethnographic methods, i.e. Tellioğlu, 2006) during their meetings, to identify clusters, to recognise the dynamics in the team, to capture them and derive from these concepts soft competencies. Technical methods such as text mining and social network analysis may support additional information.

Our approach is not only valid in a university environment. As claimed in the introduction, also companies manage competencies of their employees and want to know their skills in project

management related competencies. And for the development of competencies they need to plan which competencies could be developed in a project. Also here, our approach can be used as a troubleshooting instrument. This becomes especially important in distributed engineering environments, where on the one hand the history of an artefact can be used to identify the persons, who created or modified that artefact and then we may derive that this person has certain competencies. But this approach is limited, because we also want to identify competent people who were not involved in a certain project, but able to carry out a particular work. To achieve the full benefits of our approach we think about integrating the approach into a current project management tools or into spreadsheets, which are also often used for current project management.

Finally, we must admit that one has to be very careful how to use such a system in a real work environment. If managers use such a system as a measuring tool, the consequences can be very sensitive for employees, like being fired or promoted.

References

Davenport, Th. And Prusak, L. (1998) Working Knowledge: How Organisations Manage what they Know, Haward Business School Press.

Dorn, J. (2006) Competence Management in Virtual Enterprises, in Proceedings of XML-Tage, Berlin.

Dorn, J. and Pichlmair, M. (2007) A Competence Management system for Universities, in Proceedings of European Conference on Information Systems, St. Gallen, pp. 759-770.

Draganidis and Mentzas (2006) Competency based management: a review of systems and approaches in Information Management & Computer Security.

Fletcher, Sh. (2001) Competence-Based Assessment Techniques, Kogan Page.

Hustad and Munkvold (2005) IT-Supported Competence Management: A Case Study at Ericsson in Journal of Information Systems Management.

Lindgren, R., O. Henfridsson, and U. Schultze (2004) Design Principles for Competence Management Systems: A Synthesis of an Action Research Study, MIS Quarterly Vol. 28 No. 3, pp. 435-472.

North and Reinhard (2005) Kompetenzmanagement in der Praxis: Mitarbeiterkompetenzen systematisch identifizieren, nutzen und entwickeln; Wiesbaden.

Pichlmair, M. (2008) Universitäres Kompetenzmanagement, Ph. D. Thesis, Vienna University of Technology, Computer Science Faculty

Prahalad and Hamel (1990) The Core Competences of the Corporation.

Rose et al. (2007) Management competences, not tools and techniques: A grounded examination of software project management at WM-data in Information and Software Technology 49.

Schimper, K. (2008) Erfassung und Messung von Kompetenzen in Software Projekten, Master Thesis, Vienna University of Technology, Computer Science Faculty

Tellioğlu, H. (2006). Searching for Quality, Proceedings of the IADIS International Conference WWW/Internet 2006, Volume 1, October 5-8, Murcia, Spain, pp. 71-78

Tucker, Sh. A. and Cofsky, K. M. (1994) Competency-Based Pay on a Banding Platform, ACA Journal, Spring Vol. 3(1).

Zelewski et al. (2005) Ontologiebasierte Kompetenzmanagementsysteme: Grundlagen, Konzepte, Anwendungen.