Medical Informatics Europe 2002
G Surján et al. (eds)
IOS Press 2002

Software Engineering in Medical Informatics: the academic hospital as learning environment

H PRINS[a], R CORNET[a], FM van den BERG[b], R van der TOGT[c], A ABU-HANNA[a,c]

[a]*Dept Medical Informatics,* [b]*Dept Pathology,* [c]*Educational Board Medical Informatics, Academic Medical Center, PO Box 22660, 1100 DD Amsterdam, The Netherlands*

Abstract. In 2001, the revised course Software Engineering has been implemented in the Medical Informatics curriculum at the Academic Medical Center, Amsterdam. This 13 weeks, full-time course consists of three parts: internship, theory and project. All parts are provided in problem-oriented manner with special attention for relevant skills such as project management, documentation and presentation. During the internship, students observe how health care professionals at several hospital wards work and how information supply is organized. In the theory part, students study concepts and methods of software engineering by means of case descriptions and self-directed learning. During the project, they apply their acquired knowledge to an observed, clinical information problem and complete several stages of the software engineering process. Evaluation by inquiry showed that, compared to other courses, students spent more time, and distributed their time more evenly, during the whole period of the course. In conjunction with theory, a combination of internship and project in a hospital seems to provide a surplus value compared to a practical in a computer laboratory. The integration of software theory, clinical practice and problem-based approach, contributed to the enthusiastic, intensive and realistic way students learned in this important topic that might be chosen as a future profession.

1. Introduction

Software engineering is a very important activity as it determines the quality of computerization, which penetrates almost every aspect of human life [1]. In health care, this activity requires a large degree of insight in the medical domain and professional skills in informatics [2]. Therefore, in the 80's a Medical Informatics curriculum was developed at the Faculty of Medicine of the Academic Medical Center, Amsterdam [3]. This curriculum intends to provide students with insight in biomedicine, medical practice and health care systems on the one hand, and knowledge and skills in information processing on the other hand. In the latter, emphasis is put on methods from computer science, epidemiology and the social sciences for analyzing problems and designing and implementing solutions. The aim is to educate students who can, in their later profession, contribute effectively to analysis and solution of information problems within health care. The four-year, full-time program is increasingly internationally oriented [4] and leads to a Masters degree. Software engineering is a course taught in the 2nd year.

In the year 2000 it was decided to revise the Software Engineering course. Until then, the course consisted of theory and a 30-hours practical in a computer laboratory and took 7 full-trime weeks. The theory was given according to the problem-based learning method [5]. The reason for the revision was an internal assessment and an external audit of the curriculum as a whole [6, 7]. The recommendations that formed the basis for the revision

were: 1) extend knowledge of the functioning of health care services and professionals, 2) integrate more extensively clinical and informatics issues within courses, 3) accentuate and extend the informatics content, and 4) pay more attention to academic development, which means: train critic-analytical attitude, problem-solving methods and communication skills.

2. Module Software Engineering

By software engineering we mean the concepts and methods of development of software for information systems. Software is defined in a broad sense: all not-tangible products for development, operation, use and maintenance of information systems, e.g. project plan, requirements specifications, design, code and user manuals. It is important to realize that students taking the software engineering course, already were educated according to a reformed curriculum with an extension of informatics content [8]. They followed, among others, courses in computer science, electronic patient record, logic, set theory, linear algebra, programming, data structures, databases, cell biology, milieu-interieur and neoplasm.

The renewed, 13 weeks full-time software engineering course consists of three parts: internship, theory and project. The three parts are closely related. During internship, students observe clinical practice and the support by clinical information systems. In the theory part they learn how to analyze and realize desired information systems, and during the project acquired knowledge is applied in practice. The idea behind this combination of theory and practice is that it is important for students to experience what it means to develop software for, and within, health care settings. Through practical experience the abstract theory of software engineering becomes more vivid and is better understood.

The three parts have as common features the problem-oriented approach, clinical domain and attention for social skills. Teachers have more the role of a coach than a lecturer and have to motivate students to initiative and self-reliance.

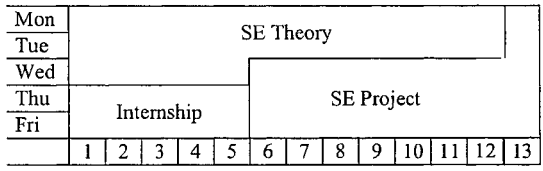

Figure 1: Interrelationship between parts in time.
(horizontal: week numbers, vertical: day of week)

Figure 1 shows how the parts interrelate. The internship jump-starts the software project, and the content of theory is given ahead to its project application.

2.1. Internship

Goal of the internship is to bring students into contact with, and to provide insight in, routine clinical practice on the floor and operational management of hospital wards. Furthermore, students gain experience in analyzing existing and required information facilities for professionals and managers. During the internship, students visit several wards, like operation room and outpatient clinic. The acquaintance takes place by means of observation on the ward, accompanying health care workers and interviews. Students completed the internship by writing a report directed at bottlenecks and requirements regarding the information supply.

2.2. Theory

Goal of the theory part is to provide insight into the reason behind the software engineering process and methods, into several techniques for different phases of the process and into important aspects as project management and safety assurance (see Table 1). Goal is also to train skills in structured problem analysis and solution, and acquiring knowledge autonomously. This part is given by the problem-based learning method [5]. This means that written cases are analyzed and solved in groups. Each case description depicts a problem about information supply in health care. By aiming to solve the problem within the field of their study, students are stimulated in self-directed learning. During preliminary discussion of a case, present knowledge is identified. Subsequently, gaps in knowledge are converted into learning goals. By self-directed learning an attempt is made to fulfil the learning goals and individual findings will be reported in the next group meeting. Each group meets one or two times a week for 2 two hours. During a meeting, one of the students plays the role of chair and another plays the role of minutes secretary. The tutor observes and guides the content and process of the meeting. Besides the meetings, guests give some lectures, especially about their practice experiences.

Examination takes place by written exam with 20 open questions. The compulsory subject matter for the examination concerns the formulated learning goals. The student has to select 17 questions that have to be graded. This is done to meet the variability of studied material.

Table 1: Topics in the theory part.

Topics	**Subtopics (some)**
Introduction	*Software Development Life Cycle, development methods*
Object Orientation	*OO, UML: static, dynamic and implementation models*
Project Management	*Planning techniques (PERT-CPM), project organization, metrics*
Requirements Engineering	*Acquisition and specification techniques*
Reference Information Models	*EHCR, HL7, role in SE*
Design	*Software design, modular design*
Quality	*Process- en product quality; validation, verification, testing, evaluation*
Implementation in organization	*Socio-technical issues in health care organizations*
Reuse	*Component Based Development,design-patterns*
Safety-Critical Software	*Risk-analysis, fault-tree analysis*
Evolution	*Maintenance: corrective, adaptive; configuration management*

2.3. Software Engineering Project

Goal of the project is to gain experience in developing software for a clinical setting. Students select relevant theory and apply this in practice. Besides these cognitive skills, also skills are gained in working together, project management, documentation and presentation. In each project, four students work together under supervision of a clinician and a medical informatician. Per project, goals have to be set, method of working to be determined, a planning to be made, tasks to be divided and progress to be assured. Regular consultation with client and end-users are organized, documentation has to be taken care for and presentations are given about methods and results. The project is examined on basis of process (partly peer-review), products (documents & system) and presentation.

2.4. Implementation

The course was implemented in spring 2001 and followed by 26 students. In the theory part 14 cases were handled. The burden of study was planned at 456 hours in total: 80 for

internship, 204 for theory and 172 for the project. Of the 204 hours for theory, 146 were planned for self-directed learning (72%), and 28 and 30 respectively to lectures and case meetings. An electronic available manual formed the guide through the course. Study material was formed by a reader, two books [1, 9] and electronically available articles and documents.

During course, a computer room was continuously available with Together/J®, MS-Project® and ZOUGA® installed. With Together/J® UML-models can be made on the basis of which rudimentary JAVA-code can be generated. MS-Project® is an application for project management. ZOUGA® is a JAVA-tool developed in the AMC for an integrated presentation of patient data from different sources.

Students did their internship and projects at seven hospital wards, e.g. Neonatology, Radiology and Anesthesiology where they worked respectively on a problem registry in a patient data management system, information supply for outpatient mamma-clinic and database for quality of care. Students completed several stages of the software engineering process, mostly from analysis to prototype.

3. Evaluation

After the course, an inquiry was conducted with 73 multiple-choice questions. Students were asked after their opinion about internship, theory, project, coherence of parts, content, study-burden and examination. Results were discussed with three student representatives. Under 26 students, 22 inquiries were distributed, of which 16 were fully completed (response 73%).

Of all respondents, 69% thought that the internship contributed to a better understanding of clinical practice and the role of information supply. 31% was not pleased with the supervision from the hospital. It is assumed that unfamiliarity of clinical supervisors with the study is the main reason for this.

With regard to the theory, 75% stated to have spent more time on self-directed learning than in the other, more traditionally organized, courses. Likewise, 63% said to have studied more regularly and 56% more goal-oriented. Reporting to peers was found advisable by all in order to learn how to chair a meeting and how to explain. Not all topics were found to be suitable for reporting back during the student meeting sessions, especially UML models. The level of content was found reasonable (75%) and difficult (25%) due to the amount of time spent to prepare for the meetings.

The project was classified as quite demanding (50%) and 38% found the time planned for the project insufficient because of two reasons. First, the students felt as to be thrown in at the deep end. Second, students were too optimistic in their choice of project goals.

Table 2: Percentage of students that agree with statements on coherence of parts

Statement	Agreed
Coherence between parts was good	87 %
Internship was useful introduction to project	87%
Project led to better understanding of theory	81%

Table 2 shows results on the coherence between the parts and speaks for itself.

Notably, the mean number of hours spent to the course was 40 weekly of which 30 were spent to self-directed learning and self-activity. In many other full-time courses, a mean number of about 30 hours is spent in total. The appeal made to self-activation was found pleasant (94%). The course was experienced as labour-intensive as the result of the binding character of the group meetings and projects. The alternation of the parts was found

stimulating. All agreed that they acquired a better insight in their future work as medical informatician. The chosen learning methods led also to uncertainty, especially with regard to project method and content of examination.

4. Discussion

Our course provides unique facets because of the integration of learning methods and its embedding in the hospital. However, parts of the course resemble courses elsewhere, for example the education program in Heidelberg/Heilbronn also integrates clinical and informatics issues [10] and Maastricht applies problem-based learning [5]. In 2002 we will use a virtual learning environment for disseminating course material and support of group meetings and projects. Moreover, a guest lecturer from Heidelberg/Heilbronn will participate.

It was difficult to compare the revised course to former versions because of the many changes: the simultaneously reform of the whole curriculum, extension of the course from 7 to 13 weeks, inclusion of UML, introduction of internship and project instead of solely relying on a practical. Moreover it was impossible to construct a control group or a meaningful before-after measurement.

Nevertheless, we conclude that, in conjunction with theory, a combination of internship and project in a hospital has more surplus value for teaching software engineering in a medical informatics curriculum than a practical in computer laboratory. It contributed to the enthusiastic, intensive and realistic way students learned in this important topic that might be chosen as a future profession. We believe that the combination of internship, theory and project in a clinical setting, provides a unique opportunity to pass on to young adults the ideas and skills in software engineering in health care.

References

[1] Pfleeger SH. Software Engineering: theory and practice. (second ed.) Upper Saddle River, New Jersey: Prentice Hall, 2001.

[2] IMIA. Recommendations of the International Medical Informatics Association (IMIA) on education in health and medical informatics. Methods of Information in Medicine 2000;39(3):267-77.

[3] Bouman LN, Zwetsloot-Schonk JH, Jaspers MW, Louter GL, Timmers T. The graduate training in medical information sciences in the Academic Medical Centre at the University of Amsterdam. International Journal of Medical Informatics 1998;50(1-3):151-7.

[4] Jaspers MW, Gardner RM, Gatewood LC, et al. IPHIE: an International Partnership in Health Informatics Education. Studies in Health Technology & Informatics 2000;77:549-53.

[5] Hasman A, Boshuizen HP. Medical informatics and problem-based learning. Methods of Information in Medicine 2001;40(2):78-82.

[6] Jaspers MWM, Limburg M, Ravesloot JJ. Medical Informatics in Amsterdam: research and education. In: Haux R, Kulikowski C, eds. Yearbook of Medical Informatics 2001:digital libraries and medicine. Stuttgart, New York: Schattauer, 2001:117-123.

[7] Vereniging Samenwerkende Nederlandse Universiteiten. External Audit of Medicine and Health Care Sciences Curricula (in Dutch). VSNU, 1997.

[8] Abu-Hanna A. The Medical Informatics curriculum at the AMC-University of Amsterdam (in Dutch). Informatie en Zorg 2001;30(3):78-84.

[9] Stevens P, Pooley R. Using UML; Software Engineering with objects and components. (updated ed.) Harlow, England: Pearson Education Limited, 2000.

[10] Haux R, Leven FJ. The medical informatics curriculum at the University of Heidelberg/School of Technology Heilbronn: new developments in its 5th revision. Studies in Health Technology & Informatics 2000;57:30-9.