

A Survey of Software Engineering Practices in the Development of Electronic Patient Record Systems

Claudio G. Alves da Costa, MD¹; Renato M. E. Sabbatini, PhD²

¹ Biomedical Engineering Dept, State University of Campinas (Unicamp) and Associate Researcher of the Center for Biomedical Informatics. claudiog@nib.unicamp.br

² Associate Professor, Faculty of Medical Sciences ; R&D Coordinator of the Center for Biomedical Informatics of Unicamp, Brazil. sabbatin@nib.unicamp.br

Introduction and Aims

In order to assure quality, electronic patient record systems (EPRS) must be developed using robust Software Engineering (SE) methodologies. The extent of use of this technology is not well documented in many projects around the world. In order to evaluate this, we have designed and carried out a systematic survey based on an extensive WWW-based interactive questionnaire. Among other things, it asks the respondents (EPRS developers) about SE practices and tools which were used. This survey is part of a masters dissertation project of one of the authors and is available at: <http://home.nib.unicamp.br/~claudiog/surveyemr.htm>.

Methodology

The survey was carried out by filling out an electronic form, available in Portuguese and English versions. The form had a total of 29 questions about the development process, divided into six parts: General information about the Project, Software Engineering, Implementation, Standards and Security. The survey was available from June 2000 to June 2001. It was announced to the medical informatics communities by means of discussion lists and direct email messages sent to professionals and institutions. The parsed contents of answer forms were converted to a database and analysed statistically using Epi Info 2000.

Results

Results reported here are based on 70 respondents, of which 62.9% came from companies and private institutions, 17.1% from not-for-profit institutions and 20% from public institutions. Approximately 34.3% of the projects were supervised by health professionals and 14.3% were supervised by experts in medical informatics. Object oriented approaches were used by 42.9% of the projects, and structured analysis by 37.1%. Regarding SE technology, 71.4% of the respondents declared to

use it. Particular methodologies used were: CASE tools (25.7%), software metrics (11.8%), UML (20.6%), software quality assurance (51.4%), feasibility study (65.7%), formal test methods (51.4%) and software configuration management mechanism (51.4%). C/C++, Visual Basic, Delphi and Java predominated as programming languages; and SQL Server, Oracle and MySQL were the most used databases. XML was the most used standard for EDI (90%). About security, only 7% of the systems have a strong mechanism for user authentication, and 58.8% allow that the saved data be after changed. Use of a Web interface was used by 65.7% of the projects.

Conclusions

Although the number of collected surveys is still small, it is sufficient to derive some conclusions about the use of software engineering methodologies in EPRS projects. Despite the inherent complexities of this kind of project, there is a low usage of systematic approaches based on SE, CASE tools, software metrics and modeling languages. Moderate use, although less than desired, was made of software quality assurance methods, feasibility studies, formal test methods and software configuration management mechanisms. There was also a less than ideal participation of medical informatics experts in the the supervision of projects; which may compromise quality, applicability and relevance. EPR are systems which deal with important and critical patient information for the health care process. Error-free operation and high quality must be striven for in the development process in order to achieve this aim. The present work suggests that ideal approaches based on SE tools and techniques have not been used in the part of projects thus far surveyed. The authors find this a worrying trend and alert the medical informatics community to look for ways to improve the situation.