MEDINFO 2010
C. Safran et al. (Eds.)
IOS Press, 2010

doi:10.3233/978-1-60750-588-4-585

Training Software Developers for Electronic Medical Records in Rwanda

Rowan P. Seymour[a], Amy Tang[a], John DeRiggi[a], Christian Munyaburanga[b], Rita Cuckovitch[a], Patrick Nyirishema[b], Hamish S.F. Fraser[a]

[a] *Partners In Health, Boston MA*
[b] *Rwanda Development Board – Information Technology, Kigali, Rwanda*

Abstract

In many developing countries, electronic medical record (EMR) systems are being implemented in resource-poor settings. Essential to such implementations are software developers with a high technical capacity, a good understanding of medical informatics and an awareness of local clinical needs. This paper describes a training program which has been run in Rwanda to enable local computer science graduates to play a significant role in that country's forthcoming implementation of a national EMR system. Such a training program is unique in that region of Africa and we discuss the challenges inherent in such an undertaking. We describe the development of the curriculum and the evolution of the teaching methodologies over the course of the year and discuss its potential integration with academic institutions in Rwanda. Finally we propose that training programs of this nature which produce local software developers who are familiar with medical informatics are a requirement for successful and sustainable eHealth implementations in the developing world.

Keywords:

Rwanda, Education, Software, Electronic medical record

Introduction

In the last decade, electronic medical records have become increasingly central to healthcare in developed countries, and such systems are also now being used in many developing countries. This trend has increased markedly in recent years as large projects have been established to treat HIV, Multi-Drug Resistant Tuberculosis (MDR-TB) and Malaria. Funding for these projects, which comes from the Global Fund to Fight Aids, Tuberculosis and Malaria, the US PEPFAR program and others, has supported the development of a range of information systems in Africa, Latin America and the Caribbean. These systems range from spreadsheets and simple databases, to commercial software packages (which may be modified for the new purpose), to sophisticated, custom built software solutions. They are used for a wide variety of purposes from clinical care and telemedicine to reporting, supply chain, accounting and research.

There are many challenges to implementing such systems in resource-poor areas ranging from lack of stable electricity and communication systems to basic problems of heat, humidity and physical security. Additional to the hardware and software costs there are important human costs in terms of training of staff that will be using the system. However, training of local staff is normally limited to end-user training and does not include maintaining or extending the software itself. With the use of closed source propriety medical records software, this is usually not an option anyway. However, with the use of community developed open source software, it is not just an option but an important way of connecting developers with users, thus realizing the benefits of open source by enabling implementers to customize the software to their needs. Such training means that implementations are not dependent on ex-patriot software developers for all improvements and fixes, which improves sustainability and supports broader development goals.

Background

The OpenMRS project, which is co-lead by Partners in Health (PIH), the Regenstrief Institute in Indiana, USA and the South African Medical Research Council, is an open source initiative built by many collaborating groups around the world [1,2]. A key strength of this initiative is that the software is free and developed through the experience of a growing number of different parties, including many in resource-poor areas. To date, OpenMRS is implemented in at least 23 developing countries and is supporting the care of over 1 million patients, more than 150,000 of which are HIV/AIDS patients on antiretroviral treatment. An important aspect of OpenMRS, especially for a developing country, is its modular architecture. This allows developers to independently add their own functionality to the EMR without having to make changes to the main application. Developers in different countries can share and collaborate on such modules. For this reason, OpenMRS is more accurately an extensible platform rather than an application.

In 2006, the Rwandan Ministry of Health (MOH) decided to begin plans for a national roll-out of OpenMRS for the purpose of tracking patient medical information in all health centers and hospitals around Rwanda. The MOH requested that

PIH, who had already implemented the system in six health centers and one hospital in the country [3], provide expertise and support for the project. It also called on the Treatment and Research AIDS Center (TRAC), an institution within the Ministry of Health responsible for treatment and research related to infectious diseases including HIV, tuberculosis, and malaria, to lead the roll-out of such a system throughout the country. Since 2006, PIH has expanded its use of OpenMRS to 12 sites, and the Government of Rwanda has invested heavily in their telecommunications sector, to provide power and internet to hospitals and health centers around the country.

It was determined in 2007 that in order to support the large and ambitious endeavor of a national OpenMRS roll-out and to prepare Rwanda to sustain the system internally, many Rwandan software developers and IT staff would need a considerable amount of further training. Additionally there was the immediate need to fulfill additional functionality requirements that the government had identified for OpenMRS. The principle of Rwandans building and supporting such a national system is consistent with the government's plan to move to a predominantly knowledge-based economy by the year 2020, commonly referred to as "Vision 2020". Therefore, it was vital for Rwanda to build the necessary technical capacity to take ownership of its development agenda, and it was for these reasons that PIH developed the plan for a mentor-driven, software development training course.

Such a course may be unique in the developing world. There are several examples of less developed countries such as South Africa [5] and Brazil [6] where health informatics courses have been established for medical students and professionals. Currently the Government of Rwanda is working to establish a health informatics center of excellence which would provide access to such training in this region of Africa. It is a difficult task given the lack of academics with a suitable background and it will likely require the sharing of resources and curricula with other neighboring countries as proposed in [7]. This training course is however, quite different from those initiatives in that it aims to produce graduates capable of developing the system rather than using it. Despite extensive literature search in the biomedical (Pubmed) and computer science fields, no reports of similar courses were found.

A 3 month pilot training program was started in November 2007 in partnership with RDB-IT (Rwanda Development Board – Information Technology). This pilot consisted of one mentor and six students who were a mixture of recent computer science graduates and existing employees of RDB-IT. The course content consisted of web design, Java programming and some of the technologies used by OpenMRS, and culminated in the development of at least one module that was deployed and used at TRAC. The limited time of the course unfortunately meant that by the end of the course, most of the students were not able develop software for OpenMRS without significant supervision. However, the students did leave with a good understanding of the technologies involved from an implementation point-of-view. One of the students became the EMR manager at TRAC, and has now been tasked by the MOH with overseeing the OpenMRS rollout.

It was clear from this pilot that the level of experience with software development that most of the students had received at their universities was not sufficient to enable them to grasp the advanced programming skills required for developing a large enterprise level software platform such as OpenMRS with only a few months of additional training. This was finding was consistent with experience of other similar programs which considered the use of local software developers such as [4]. Thus plans were made for a more substantial training course which would have the resources and time to produce local software developers capable of independently contributing to the OpenMRS project. Based on the experiences of the pilot program it was decided that the course should:

- Be one year long to give sufficient time to cover all the necessary technologies.
- Have a thorough screening of applicants to ensure that only high quality students were recruited.
- Involve enough mentors to ensure all students received a practical "hands on" experience, with no more than 6 students per mentor.

In addition to these requirements for the actual training of students, it was important to develop a course which was relevant in the context of the government's plans for a national EMR, and adhered to a model that was sustainable and scalable. The following goals were decided:

- Create a cyclical course where some of the students become mentors in subsequent years.
- Develop a comprehensive curriculum which can be shared, peer-reviewed and refined by other institutions and organizations.
- Indentify health management needs within Rwanda, and develop and deploy customized OpenMRS software modules to meet those needs.
- Include teaching of core medical informatics principles and expose students to non-technical issues through site visits and other activities.
- Undertake an evaluation of the technical training program and the software tools developed to assess the effectiveness of the program.
- Document the experiences and lessons of the OpenMRS adaptation process and capacity building effort to provide policy guidance for the scaling up of the initiative beyond the pilot phase in Rwanda and for a wider application in Africa.

Methodology

Different training and academic course models were considered and it was decided that this training course should use the model of a sponsorship program where students receive salaries. This was important because in this initial year of the course, it would not be able to offer a qualification accredited by an academic institution, and so to attract strong candidates

there had to be a financial incentive. This model is also familiar to public institutions in Rwanda where employees are often sent on training courses whilst still receiving a salary. The employee is then contractually obliged to continue working for that public institution for a period of time proportional to the amount of training they received. Such a contract would be essential to meeting the goals of this course, as it would ensure that the training invested in the students benefit Rwanda's health sector. To facilitate such a model the students would be employed by RDB-IT and obliged to work for the MOH for 2 years after completing the course.

For practical reasons the course length was set at 11 months. The first 7 months were split into two teaching sections. The first would be 3 months long and used as a probation period, and thus an opportunity to drop any students who were not performing well enough. The final two months were set aside as a production stage to ensure that there would be sufficient time for students to develop software modules for OpenMRS.

Screening of candidates

One of short-comings of the pilot program was the lack of entrance qualifications of the students before they were accepted onto the course and so a screening exam was developed which would test the candidates':

- Understanding of basic IT concepts.
- Knowledge of the software development lifecycle.
- Ability to construct simple algorithms.
- Awareness of more advanced software development concepts such as object-oriented programming.

In October 2008, RDB-IT selected 40 computer science graduates from applications they had received, and facilitated them undertaking the screening exam. In addition to enabling the selection of the best students, the results also enabled the course mentors to begin creating a curriculum appropriate for the students' understanding of the relevant technologies. The results of the screening exam also showed that although students were generally familiar with theoretical IT concepts, most struggled to construct even basic algorithms. It was evident that students generally had little practical experience of software development, especially with modern object-oriented languages such as Java. From the 40 candidates who sat the screening exam, the top 19 were called for an oral interview. These interviews allowed mentors to examine non-technical criteria such as:

- Students interest in healthcare
- Communication skills
- Ability to mentor others

The final selection of 10 students was based on the combination of their screening exam scores (weighted at 70%) and oral interview scores (30%).

Development of the curriculum

The screening exam results showed that some topics which the students studied at university would have to be covered again in the new training course. So a curriculum was developed that would start with very little assumption about what the students already understand, and then throughout the teaching stages, introduce all of the core technologies required for OpenMRS development. Even where students were familiar with the topics, there would be a new emphasis on best practices. The curriculum was broken down into 8 units to ensure that all of the important topics would be covered (see Table 1), and each of these topics is described briefly in the following sections.

Table 1 – Course curriculum

Unit	Title	Weeks
1	Foundations	2
2	Basic Web Design	4
3	Basic Java Programming	5
4	Advanced Java Programming	7
5	OpenMRS Implementation	4
6	Enterprise Java Programming	6
7	OpenMRS Development	7
8	Medical Informatics	-

Foundations

This was a high level overview of many topics from IT and Medical Informatics. The intention of this short un-assessed unit was not so much to give the students an understanding of the topics, but rather

- To start building a common technical vocabulary for students and mentors. This proved to be very important given that *for most of the students, English was their third language*.
- To ensure that all of the students are familiar with the basic tasks of maintaining their computer, and using its software to make documents and presentations.
- To introduce the students to the broader ideas of eHealth and their role as software developers.

Basic web design

This was to build on the students' prior experience of HTML, introducing XHTML, CSS and JavaScript. It emphasized the importance of separating content from design and developing standards compliant web pages.

Basic Java programming

This was to give students a firm foundation in the Java language. It would cover all the syntax of the language, and introduced the students to object oriented concepts, with which many were not familiar.

Advanced Java programming

This covered more advanced topics in Java programming such as design patterns, multithreaded applications and unit testing.

OpenMRS implementation

A break from development in general, this unit covered OpenMRS from an implementer perspective. This included:

- Basic setup and maintenance of a Linux based server as well as how to manage the necessary server software such as Tomcat and MySQL.
- Infrastructure issues such as power and internet connectivity.
- Staffing and training issues.

Enterprise Java programming

This covered most of the Java Enterprise Edition (J2EE) technologies at the core of OpenMRS such as JSP, JSTL, MySQL and Hibernate.

OpenMRS development

This covered some J2EE technologies not covered in the previous unit (e.g. Spring), and teaches development of OpenMRS modules using the OpenMRS API.

Medical informatics

Unlike the other units of the course, this unit was taught throughout the year. The content was largely provided by guest lecturers, either in person, or through remote lectures using Skype. Topics covered included medical data and coding, EMR systems, decision support systems, pharmacy systems, evaluation of medical information systems. These were taught with a focus on developing country environments.

Course implementation

It was clear from the start that the course needed to emphasize practical experience over theoretical understanding. Each of the students was given a laptop for the duration of the course to ensure that they had an adequate machine for completing course exercises and projects (of the ten students, only a few had their own laptop). Typically each day, a mentor would give a lecture to introduce a new topic, and then set small exercises to reinforce the content of the lecture. At the end of each week, a project would be set which would involve many of the topics taught that week. Students were given a few hours each afternoon to work on these, and also expected to work on them at home.

Assessment

At the end of each unit (except the first), students were given a week long project which would incorporate many of the technologies from that unit. Most of these projects were related to eHealth, and included:

- An online health check website (Basic Web Design)
- A simple console based EMR (Basic Java Programming)
- A virtual OpenMRS server and deployment plan (OpenMRS Implementation)

Students were also given a written exam at the end of each module. Because students were encouraged to help each other with projects, the written exams were generally a more accurate indicator of an individual student's performance.

Module production

The last two months of the course were set aside for development of modules for OpenMRS sites in Rwanda. The EMR managers at TRAC (MOH) and Rwinkwavu hospital (PIH) were asked to compile a list of additional functionality they would like to have implemented at their sites. Course mentors then discussed these requests with some of the OpenMRS programmers, in order to determine which were the most suitable for implementation as modules by the students. A list of 5 potential modules was agreed upon, and then allocated to pairs of students, depending on the complexity and the students' ability (see Table 2). As much as possible these student projects were managed as real software projects. All are now in clinical use.

Table 2 – Modules developed by students

Name	Description
PMTCT	Registers mothers and their children on the Prevention of Mother To Child Transmission program
Patient Alerts	Alerts that notify clinicians of patient data indicating the need for immediate action
Drug Order Export	Allows drug order data and patient regimen histories to be exported
TRAC Data Quality	Produces lists of potential errors or omissions in data entered into the EMR
Data Entry Delay	Provides statistics about the time it takes for encounters between clinicians and patients to be entered into the EMR

Discussion

The curriculum described here and the teaching materials are all available online for other people to download[1]. It is hoped that this will lead to collaboration with other similar training programs around the world, and allow the materials to be peer reviewed and further developed by others. Much of the materials are not specific to OpenMRS and thus a suitable resource for any similar Java-based training course.

Teaching methodologies

One of the advantages of creating a new training course has been the freedom to explore different teaching methodologies, some of which would be difficult to incorporate into a traditional academic course. One method that was found to be particular effective and increasingly used as the course progressed, was asking stronger students to explain complex con-

[1] http://openmrs.org/wiki/EHSDI_Training_Course

cepts to the rest of the class in the language of their choice. This usually led to discussion amongst students - often involving those students who would be less inclined to ask a mentor for clarification. On other occasions all of the students would be assigned a different topic and required to give a presentation to the rest of the class. This had the added benefit of improving their presentation and communication skills.

Institutionalization

The creation of an independent training course was considered necessary in this first year, to satisfy the immediate need for local software developers capable of producing OpenMRS modules. Likewise the sponsorship model used was considered necessary to attract high quality students and to provide a mechanism for placing students in jobs with the MOH. However, moving forward it is important to explore ways of making the course more sustainable and scalable. We have already begun discussions with one of the universities in Rwanda about how this course could be integrated into their postgraduate programs. This presents a challenge because our emphasis on "hands on" mentoring is at odds with the requirements of most post-graduate academic courses. We would also need to reduce the overlap in topics covered by this course and the undergraduate computer science courses, perhaps by finding ways of assisting them in strengthening their courses. Even in an academic institution, it may be possible to maintain the sponsorship provision for some or all of the students. We must continue to partner with the MOH and other groups who are implementing OpenMRS in Rwanda, giving them the opportunity to invest in the training of their staff, and to ensure that the course continues to be relevant to their needs.

Collaborations

We believe that there is much that other organizations implementing eHealth solutions in the developing world can learn from this training course. First and foremost we hope that other organizations are encouraged to initiate similar programs, and can see that it is both possible and beneficial to train local software developers, rather than relying on ex-patriot staff. We are also keen to explore ways of collaborating with similar training programs that OpenMRS is involved in such as Google Summer of Code, which provides grants for students to spend several months working on an open source project and being mentored by a developer. It is hoped that this course will eventually integrate with the health informatics center of excellence that the Government of Rwanda is trying to develop with funding from IDRC and Rockefeller Foundation. It will link this developer training program with training in medical informatics for physicians, nurses and other clinical staff, data manager training, and also include research and evaluation. Another important collaboration linked to this course is the American Medical Informatics Association GPP program to create collaborative training programs in medical informatics in Africa (funded by the Gates Foundation).

Conclusion

This training course was created to provide high-quality Rwandan software developers with the necessary programming skills and grasp of medical informatics to contribute to the further development of OpenMRS in that country. The students developed OpenMRS modules which are being used at the TRAC and Rwinkwavu Hospital sites. They worked closely with the EMR managers from those sites, who were satisfied that the modules were of sufficiently high quality to be deployed on EMR servers. They graduated in October 2009, and in February 2010 the course was restarted with 12 new students. Clearly, it is difficult to objectively measure the success of the course without external assessment of the students, so it is intended that the students will sit exams for Sun certifications for J2EE development. We will also carry out a more detailed review of the modules in the near future.

Acknowledgments

We thank IDRC for the funding and RDB-IT for support.

References

[1] Mamlin BW, Biondich PG, Wolfe BA, Fraser HSF, Jazayeri D, Allen C, Miranda J, and Tierney WM. Cooking Up An Open Source EMR For Developing Countries: OpenMRS – A Recipe For Successful Collaboration. Proc. of AMIA Symposium 2006, pp. 529-533.

[2] Seebregts CJ, Mamlin BW, Biondich PG, Fraser HSF, Wolfe BA, Jazayeri DG, Allen C, Miranda J, Baker E, Musinguzi N, Kayiwa D, Fourie C, Lesh N, Kanter A, Yiannoutsos C, Bailey C. The OpenMRS Implementers Network. Intl. J. of Med. Informatics. 2009, pp. 711-720.

[3] Allen C, Jazayeri D, Miranda J, Biondich P, Mamlin B, Wolfe B, Seebregts C, Tierney W, Fraser HSF. Experience in implementing the OpenMRS medical record system to support HIV treatment in Rwanda. Proc. of Medinfo '07, pp 382-386.

[4] Braa J, Humberto M. Building collaborative networks in Africa on health information systems and open source software development – Experience from the HISP/BEANISH network. 2007.

[5] Hanmer LA. The postgraduate medical informatics programme at the University of Cape Town. Methods Inf Med 1989; 28(4): 321-3.

[6] Marin HF, Massad E, Marques, EP, Ohno-Machado L. International training in health informatics: a Brazilian experience. Medinfo 2004 11(2) pp 898-902.

[7] Kouematchoua G, Rienhoff O. Sustainable Development of Medical Informatics in Africa: Need for a Health Informatics Curriculum. HELINA2007; Bamako, Mali 2007.

Address for correspondence

Partners in Health, 643 Huntington Ave, Boston, MA 02115-6019
Email: rseymour@pih.org