HIC 2008 Australia's Health Informatics Conference
Editor: Heather Grain
Publisher: Health Informatics Society of Australia Ltd (HISA)
ISBN 978 0 9805520 0 3

Engaging the Clinician: The Value of Rapid Prototyping of 'Lightweight' Electronic Health Records

Parle A[1], Lassere MN

[1]Consultant, Computers and Information Technology, Gladesville, NSW
[2]Department of Rheumatology, St George Hospital. University of NSW, Sydney, NSW

Abstract

Objective:

To apply rapid prototyping of 'lightweight' applications to clinical re-engineering.

Background:

A 6 month project was funded to trial an electronic patient-held portable health file (PHF) carried by the patient between different health-care providers.

Methods:

Software, interface and functionality decisions required close co-operation between programmer and clinician-user. The process was 'iterative, simple and usable' rather than initial perfection. These considerations drove technical specifications. Many features were rejected because they complicated the interface or were 'programmer' functions rather than 'doctor' functions.

Results:

The record was implemented as a file (encrypted XML in an extensible schema) to be read and modified using a custom program and carried on a USB drive. Java was chosen for rapid development its support on many platforms. The program used a 'click and type' interface. This facilitated 'out of the box' use. Functions such as doctor authentication, audit trail, non-repudiation and saving of changes were built-in and automated. The XML schema was simple and generic rather than based on any existing medical models. The semantics relating these is supplied by the program rather than driven by the data. Records could be viewed but not modified when USB drive was not present. The system was designed as 'moderately secure', as loss of access to data was considered a more significant threat than unauthorized access. The system used a central authority to issue pre-configured USB drives to patients including the software, and install software and authentication information on computers of participating health-care providers and patients.

Conclusions/Significance:

Doctors and programmers think and work in different modes. The success of this program relied critically on the close co-operation of doctor and developer and keeping the result simple. Although the implementation of national e-health technology requires integrated complex systems, rapid prototyping to initially solve niche problems may be an efficient alternative.

Keywords:

Clinical re-design, user-interface, lightweight vs. complex applications.

HIC 2008 Australia's Health Informatics Conference
Editor: Heather Grain
Publisher: Health Informatics Society of Australia Ltd (HISA)
ISBN 978 0 9805520 0 3

Introduction:

The electronic medical record or Personal Health File was conceived as part of a project to improve the care of rheumatology patients by sharing patient information between medical practitioners. It was originally based on a paper diary which would be held by the patient and presented to the doctor on each visit, containing recent tests, known conditions, blood pressure readings, etc. In addition, as part of the original study, the record would include a checklist of quality indicators specific to rheumatology to be filled out by the practitioner at each visit.

Specification:

The paper diary acted as the initial informal functional specification for the electronic version, indicating the information which would be useful. There were also many non-functional requirements which arose from initial discussions between the developer and clinician. The record had to be usable with little or no training, and physically small enough to be carried almost anywhere. It was desirable that the patient should have access to the record as well as medical practitioners, and to be able to record their own observations and information. The patients would be from older groups and not necessarily computer literate. The information held on the record would be used in making clinical decisions, which required an adequate level of security and integrity. The software supporting the record would have to run on a variety of equipment in doctor's premises and it would need to be ready for distribution in a very short time frame.

The initial proposal was to carry the record itself on a USB flash drive, which were coming into widespread use at the time and with rapidly falling prices. The software would be written in Java and developed using rapid prototyping. This decision was taken with the platform independence of Java in mind, and with the thought of executing the software as a Java applet within a browser, but in the end it was simpler to run as a stand-alone program. This in turn required the installation of the Java run-time environment on the doctor's computer and on the computers of any interested patients. The patient data, program, Java runtime, and installation scripts could all be stored on the one USB stick.

One important issue which was specifically not addressed in this project was integration with other medical record systems and applications.

Phase 1 Development:

The first development iteration would adapt the paper version of the patient diary to electronic form and to create a test-bed for further elaboration. At this stage there were many aspects of the system which were not known and whose importance were not recognized. As this iteration was experimental, the developer was given great discretion in making design decisions with the understanding that feedback from the clinical staff might require later changes. On the other hand, when the clinical personnel had a chance to use the prototype and obtained a better understanding of what was possible with the technology and how well (or badly) various ideas worked out, the concepts driving the development would also change.

Until the prototype was in the hands of the clinicians, all discussions between the two sides were purely theoretical. There was little point in expending time to make design decisions which were liable to be changed later on, so the software development was based around the Simplest Thing Which Can Possibly Work. At the same time, some design options were so fundamental that they had to right, or nearly right, from the start because later changes would involve a prohibitive effort. Those key decisions were identified by the developer and discussed with the clinical lead before implementation. Those decisions were:

1) The system would aim for "moderate" security, as its purpose was to share information rather than conceal it. Security efforts were directed primarily to ensuring data integrity and authenticity, rather than to prevent unauthorized access.

HIC 2008 Australia's Health Informatics Conference
Editor: Heather Grain
Publisher: Health Informatics Society of Australia Ltd (HISA)
ISBN 978 0 9805520 0 3

2) The USB copy of the patient record would be automatically copied (if so configured) onto the computer's hard drive whenever the file was opened in the program. Any changes to the record contents would update both copies of the file. The USB copy was the "master", and the hard drive copy (also called the "database copy) could not be modified unless the USB copy was present.

3) Changes made to the record would automatically be copied to the file without further user intervention (no "save" command).

The first two of these decisions were interrelated. The paper diary, which was subject to loss or theft, and readable by anyone who picked it up, formed the original inspiration for the electronic medical record. The EMR could enhance this with password protection, encryption, backups, etc. and still have only moderate security by current standards, so that was thought to be the appropriate access protection level.

Automatic backup to the local hard drive serves to prevent data loss and to allow the doctor or patient to review, but not change, the record when the USB stick is not present. In case the original USB is lost, a new one can be created using the copy held on the most recently accessed computer.

This system also lends itself to various extensions – for example, to allow a doctor to add notes when reviewing the record (these notes to be held in the database on his computer) and these to be linked to the record so they are not copied to the USB and so are not visible to the patient or other doctors. These were not implemented in the trial, however.

The decision to automatically detect changes and write them to the record files was taken because of the users – patients and doctors – who were the intended users of the system. It was considered critically important to minimize the learning requirements on such users, and so the paper diary again served as a model. We could rely on the user finding the right diary page to read it; and to find the right boxes on the page to add new information. On the electronic record, the user could find the right page (well-labeled tabs were visible to be clicked on at all times), use the mouse to select the appropriate box, and type the information. No separate operation was needed to save the information in the paper diary; and so none should be required in the electronic version.

Initial Prototype:

With these considerations and with the paper diary as a guide for screen layouts, the prototype (v1) electronic medical record was returned to the clinical lead in five weeks.

During this phase the important thing was to get a running prototype as fast as possible so that the all-important feedback from the clinical staff directing the project could be obtained. The earlier this feedback happens, and the more responsive the developers are in responding, the better the final results will be. This development methodology (sometimes called "evolutionary prototyping") is different from the classic "waterfall" where the final result is specified in the beginning, and is particularly appropriate for experimental projects such as this one which is driven by concepts rather than by fixed and known requirements.

Development speed comes at a price, and in this phase this is in quick-and-dirty solutions to issues that may need to be fixed later, hopefully when more is known about what is required. An example is the v1 file format, which was implemented just as the native serialization of the Java object representing the patient information. It was known that this would need to be changed, and it was known that to get it right would be a long and difficult process which could not be undertaken until there was a better idea of exactly what information would need to be stored. So the work on the final file format was simply postponed until phase 2, and Java Serialized Objects were used in the meantime.

The v1 Electronic Medical Record software had a four-panel layout with the top two panels containing patient and doctor identification, the lower left listing all accessible patient records on inserted USB sticks or the

HIC 2008 Australia's Health Informatics Conference
Editor: Heather Grain
Publisher: Health Informatics Society of Australia Ltd (HISA)
ISBN 978 0 9805520 0 3

local hard drive, and the main, lower right panel containing the selected patient's medical record in tabbed panels, one panel per page of the paper diary. Pages included: the patient's address and contact information; details of all medical practitioners seen by the patient; any medical alerts; vaccinations; current and past therapies with complementary therapies on a separate page; blood pressure readings; known or diagnosed medical conditions; test and procedures. As one of the major justifications for the project was the evaluation of a checklist of quality indicators for the treatment of rheumatoid arthritis, a page was developed for this purpose.

Phase 2 Development:

In this phase, the clinical staff were using the prototype to determine how it should be changed to meet the needs of the project. It was characterized by frequent conferences between the clinicians and developer, with decisions taken and updates to the software with a short turn-around time. Unlike the first phase which was developer driven with clinical guidance only on critical issues which could not be delayed, phase 2 was driven by the driven by the clinician, with the developer advising on what was possible, how long each new feature or change would take, and otherwise making suggestions. The end point of the phase 2 development was the v2 electronic medical record software which was used in the trial.

The first discovery made in this phase is that clinicians and programmers do not think in the same way. Features which seemed "obvious", "logical", or even "intuitive" to one party often did not appear that way to the other. This was particularly true in the user interface. Once this was recognized, the development methodology was adjusted so that on issues the clinicians considered of critical importance, such as user interface issues, the final decision always lay with the clinician. The programmer's responsibility was to estimate the time and feasibility of various options and to make suggestions.

The turn-around time for feature changes was fairly short, often two or three days. Because the individual changes were usually fairly small, there was the freedom to try and discard approaches which were not working without compromising the project as a whole. Since the clinical staff were using each iteration of the application as it became available, bugs were found fairly rapidly and the software quality of the application improved over time.

During this phase the user interface evolved quite rapidly. One reason for this is the elementary observation that a computer program is not all that like a paper diary. In the paper record, the user (doctor or patient) can expect to understand it and use it, if not perfectly. If the information is written in the wrong column or extends into the margin, the next person to read it will probably be able to figure it out. The electronic version has its own advantages – nearly infinite storage capacity for one, and the ability to erase or fix minor errors – but it did require the user to actually perform actions such as, say, add a record for a new test.

In some cases, it was thought to be simple enough for the user to simply type into an empty box (eg. blood pressure readings). In others cases, such as recording a new medication or changing or stopping an existing one, a definite action by the user was required which had to be made as "intuitive" as possible.

In most computer applications, menus are the method used to initiate actions. However, it was decided for the electronic medical record that menus should not be used as they require more thought and possibly training by the doctor to use. Instead, buttons were placed on the page in close proximity to the place where the doctor would be typing the information, with instructions which were always visible. To try and make things easier, the button would always be visible but would become inactive (greyed out) when the associated operation was not applicable. Immediate visual cues for the reader would be provided where possible. Only advanced functions were to be accessed by the menus provided.

In some areas where the paper way of doing things made less sense for the computer, the paper method was retained in order to make direct comparison of the paper diary and electronic medical record easier for the purpose of the study. One example is the page describing existing conditions. In the paper diary, this

ISBN 978 0 9805520 0 3

page was a list of conditions (weighted towards those of interest in the management of arthritis) which could be checked off. Obviously an electronic record could go to town with this information – more open-ended conditions, when diagnosed, by whom, any related treatments, all cross-linked with the other information held in the record – but in this phase the same layout and list of conditions was used.

One design feature which was implemented in phase two was an audit trail based on who entered data and when. Part of the original design was a file installed on the participating doctor's computer by project staff which uniquely identified the doctor. Program installation from the USB stick, as on to the patient's computer, would not have this file. Changes made when the USB stick was plugged into the doctor's computer – such as a new medication – would by default use this to identify the doctor as the one prescribing the medication (although this could be overwritten, as when a doctor recorded a new medication by a non-participating doctor after being informed by the patient).

In phase two, this notion of the doctor record was expanded into a full audit trail. Every session which modifies the record held on the USB stick will also append information to the audit trail, containing not only the doctor identification if present, but also the date, computer user and host name, and description of the changes made. The audit trail is stored as plain text and may be viewed through one of the tabs.

Much thought went into the v2 file format. The v1 file format using Java Serialized Objects was very quick to implement, but was only ever indented as a prototyping solution. Apart from being dependent upon programs running in Java, any changes to the serialized data structures could make different versions incompatible. This is a generic problem in file formats, particularly in applications such as the electronic medical record where the character of the information to be stored could be expected to change over time.

The solution which was adopted in v2 was based on XML using a generic schema which would not need to change even when the stored data changed. In this schema, there are only two entities: data elements (with a name, type, and value) and object elements (with a name, type, and containing data and object elements). The only fundamental types for data elements were strings, numbers, and dates. Using only these elements, arbitrary data structures can be generated.

In this case, the trade-off was in discarding the power of XML to describe the application data structure while requiring the application code itself to be able to interpret the objects.

During the project, the XML file was compressed by the application more as a simple security measure than for reasons of saving space. The files generated tended to be fairly small, only a few tens of kilobytes. A production version of the application would be able to encrypt the file and use other security measures as well.

Conclusion:

The rapid development methodology was able to produce the electronic medical record software for the project trial within four months from concept to delivery and with a good set of features and quality. This was a particular challenge because the application itself was experimental, and what it would do and how it would do it could only be worked out during the development process itself. Involvement of the clinical staff at all stages was critical in this success, particularly in setting a strong direction on requirements as they became apparent and giving rapid feedback on implementation decisions.

Doctors and programmers think and work in different modes. The success of this program relied critically on the close co-operation of doctor and developer and keeping the result simple. Although the implementation of national e-health technology requires integrated complex systems, rapid prototyping to initially solve niche problems may be an efficient alternative.