Combining Scrum and Model Driven Architecture for the development of the EPICAM platform

Azanzi Jiomekong, Hippolyte Tapamo, Gaoussou Camara

Combining Scrum and Model Driven Architecture for the development of the EPICAM platform

Azanzi Jiomekong[1], Hippolyte Tapamo[1], Gaoussou Camara[2]

[1]UMMISCO, Faculty of Sciences, University of Yaounde I, Cameroon
[2]EIR-IMTICE, University Alioune Diop de Bambey, Sénégal

*E-mail : fidel.jiomekong@facsciences-uy1.cm

Abstract

The problem of failed software developed for epidemiological surveillance is often the result of an unsystematic transfer of business requirements to the implementation. However, empirical research shows on the one hand that, in certain conditions, Model Driven Architectures (MDA) are more effective than code-centric approaches for the development and the maintenance of software. On the other hand, Agile Processes such as Scrum are more effective than Structured. Processes when requirements are subject to frequent change. In this paper, we show how Scrum and MDA were used to develop EPICAM, a platform for epidemiological surveillance of Tuberculosis in Cameroon.

Keywords

EPICAM, Tuberculosis; Epidemiological surveillance; Scrum; Model Driven Architecture

---

I INTRODUCTION

Epidemiological surveillance systems enable the collection, analysis, and interpretation of data, closely integrated with the dissemination of these data to stakeholders for preventing and controlling disease and injury. To be usable, epidemiological surveillance systems use surveillance software to provide timely, accurate, comprehensive and update information to stakeholders [5, 10]. Depending on the data gathered on the field, epidemiologists may need to collect a new parameter in order to explain a phenomenon. For example, in the case of Covid-19, data can be collected in order to determine if people of a certain race are more at risk. This task may be done by using supplementary materials such as paper forms or spreadsheet software and lead to the problem of data integration [2]. To collect new data, another solution consists of updating the software used for the surveillance. This solution is expensive and may lead to the problem of software regression [12] because bugs can be introduced during the source code update. These problems occur because of the problem of failed software - often caused by an unsystematic transfer of business requirements to the implementation [7, 9].

The problem of unsystematic transfer of business requirements to the implementation can be avoided if the system is established using a well-defined framework/architecture such as Model Driven Architecture, permitting the rapid development/update of the surveillance software. Model Driven Techniques in general relies on models and model transformations for the development, maintenance and evolution of software. Through the architectural separation of concerns, the Model Driven Architecture (MDA) in particular assumes the evolutivity, portability, interoperability and reusability [6, 15]. Graphical modeling permits non-informatics experts to be able to build graphical models and automatically generate their software. Empirical research shows that the main motivations for companies to use Model Driven techniques are cost saving, rapid development / update / maintenance, improvement of productivity, software quality, and the maintaining of the architecture consistency from analysis to implementation [6]. In the health domain, many authors have proven that Model Driven techniques is an effective approach for the development of health software [9, 11, 13].

On the other hand, translating requirements to a solution using either code-centric or MDA approach requires the use of a software development process. Software development process defines a scheme to structure and manage the various aspects of software development. Software development processes are grouped into Structured Processes such as the waterfall or V-model and Agile Processes such as Scrum [8]. With Structured Processes, specifications are fixed in advance and the software is developed and delivered thereafter. In the health domain, empirical research demonstrated that software developers experience difficulties when following Structured Processes [7, 14]. In the particular case of a pandemic, it can be very difficult to capture all the requirements of its epidemiological surveillance system at such an early stage of a project. However, Agile software development processes such as Scrum [8] experience great success in software development organizations. Many empirical studies report significant benefits being gained from utilizing Agile practices such as reducing costs, reducing time to market and increasing quality [14].

Concerning Cameroon, given the problem of unsystematic transfer of business requirements to the implementation, the EPICAM[1] (Epidemiology in Cameroon) project was set up. This is a project of academic (University of Yaounde I in Cameroon[2]), clinical (fifty hospitals in Cameroon), epidemiological (Centre Pasteur du Cameroun[3] - Epidemiology and Public Health department and NTCP), and industrial (MEDES[4] in France) partners. It aims to provide methodologies, methods, models and tools for improving the development of epidemiological surveillance of infectious diseases. Firstly, a pilot project consisting to set-up the epidemiological surveillance of Tuberculosis (TB) was put in place. In effect, given the burden of TB in Cameroon, the government has put in place the National Tuberculosis Control Program (NTCP). Thereafter, the NTCP has set-up epidemiological surveillance. Even if some tools like Microsoft Excel are used, this system is mainly manually managed. Given the limit of the manual management [3, 13], the NTCP expressed the need for an electronic system. Thus, the purpose of the pilot project was to describe the methodologies, methods and tools for the development of epidemiological surveillance of TB. This paper presents how we proceed to develop the EPICAM platform for epidemiological surveillance of TB in Cameroon. In the remainder of this paper, we present the research design in Section II, research results in Section III and conclusion in Section IV.

II RESEARCH DESIGN

The research design was inspired by empirical research methods in software engineering [4, 6, 14]. On the question of how to develop epidemiological surveillance software, evidence shows

[1] http://www.medes.fr/fr/nos-metiers/la-e-sante-et-l-epidemiologie/la-tele-epidemiologie/projet-epicam.html
[2] https://uy1.uninet.cm
[3] https://www.pasteur-yaounde.org
[4] http://www.medes.fr

on the one hand that Agile Processes are more effective than Structured Processes when the specification cannot be fixed at the beginning of the project [14]. On the other hand, MDA is more effective than code-centric approach during the development and the update/maintenance of the software [6]. Then, our intuition is that: "meshing Scrum and MDA improves the development of epidemiological surveillance software". This led to the following research question: how to mesh Scrum and MDA to develop epidemiological surveillance software? We have used the term meshing as the common term for combining, mixing, and coexisting.

Our aim in this work was to do intensive research on the development of epidemiological surveillance software in order to provide a deeper understanding. Then, we choose the case study method because it is suited for understanding the complex real-world setting [14]. The selection of the case of epidemiological surveillance of TB was made because of the NTCP's interest to replace manual management with software management. Thus, our aims were to portray and provide a deeper understanding of the development of epidemiological surveillance of TB using MDA and Scrum Process.

To understand the current system, we thought it necessary to investigate the opinions of the stakeholders about the current system and its replacement with a software approach. To this end, a survey was conducted. This was a qualitative study during which open-ended questions were asked to the participants in order to allow them to furnish all the information required to build the system (more detail is given in Appendix A).

Stakeholders allowed us to find a real problem that health software generally faced: this is the problem of failed software mainly caused by the fact that requirements cannot be fixed. In this situation, we completed our methodology with action research [4]. In effect, we were having the NTCP willing to collaborate to both identify the problem, and engage in an effort to solve it.

Globally, the research methodology was composed of the (1) Pre-Intervention phase during which the problems at the NTCP are identified, the method to solve these problems and the list of interventions to implement are defined. (2) Intervention phase during which the solution to the problem is developed. This is a cyclic process during which the list of interventions identified during the Pre-Intervention Phase are implemented. These interventions are executed in iterations. Each iteration is composed of the planning, the implementation, the validation, the reflection. The iteration nature of the process provides us opportunities for improvements to the knowledge, methods, and better understanding of the approaches and methods. (3) Post-Intervention phase during which, all the tasks to implement (deployment, training end users, vulgarization, etc.) in order to make the solution available to the end users take place.

To collect data, we used two data collection techniques: semi-structured interviews and observations. Additional sources of data includes: supports such as pictures of paper data collection forms, Excel documents, etc.; Email which was the online channel of information exchange, decision discussion and sometimes, decision making; Meetings with stakeholders during which we took useful notes. After the data collection, the analysis consisted of labeling and organizing the data in order to identify codes, themes and relationships between them. At the end of each phase, a report was prepared and shared by email and during the meetings. The summary of the collection and analysis method is given by Tab. 1 in the Appendix B.

In this research, we choose the MDA approach, Scrum process, dia (diagramming software) and Imogene tools. (1) MDA allows us to define the conceptual models (Platform Independent Model - PIM and Platform Specific Model - PSM), which are used thereafter to automatically generate the executable source code. (2) Dia[5] is used to manually construct the Platform Independent Model, represented by a class diagram. (3) Imogene[6] was closer to the features expected by the NTCP. Our proximity with the MEDES, the company which developed Imogene motivated us to adopt it. (4) Scrum was used as the software development process.

Globally, the PIM was designed using Dia. Thereafter, the PIM was transformed into the PSM. Once the PSM is constructed, the corresponding applications are generated. In this research, given that there was not a tool for direct transformation of PIM to PSM, this task was done manually. The table 2 in the Appendix B presents an overview of the transformation from the PIM into the PSM and from the PSM into the source code.

Complementary information on the study design is given in Appendix A.

III RESULTS

As the result of this research, we portrait in this section the development of the EPICAM platform. Thus, Section 3.1 presents the results of the survey beside the health workers, Section 3.2 presents the results obtained during the implementation of the approach and Section 3.3 presents the results of the evaluation of the approach by people who participated to this research.

3.1 Users survey

The survey of health workers involved 8 (53%) health professionals working in four hospitals where TB patients are treated, 1 (7%) health regional delegate, 6 (40%) managers at the National TB Control Program. All the managers have worked in hospitals before being managers. During the survey, a number of data collection tools (such as registers, statistics templates, data collection sheets) were collected at the hospital, regional and central levels. Given the small sample size, we did not produced the statistics based on the questionnaire, but have summarized the findings per questions asked:

- On the question: "How do you manage your data?" We noted that at the level of hospitals, only manual management was used. At the regional and the central level, manual management was combined with software management using Microsoft Excel.
- On the question: "What are the main problems you generally have?" Many problems were reported, the main are: (1) The problem of manual collection and management of data. (2) The problem of manual analysis of data. This problem causes many other problems such as completeness, promptitude and erroneous statistics. (3) The problem of transmission of data using land transport. The main consequences of this problem noted were the late arrival of data collection tools, data collected and epidemiological reports at their destination. At the Regional level, the promptness was estimated less than 70%. The rapid exchange of data between doctors is done using phone calls (for instance, during a transfer/referral). The awareness of patients and the general population was done using SMS. (4) Another problem not directly linked with the system such as power and Internet outage were reported.

  Only one health personnel reports that he did not have any problem with data management. We noted however that he treated less TB patients per week (generally, less than thirteen), compared to the other hospitals.

[5] www.dia-installer.de
[6] https://github.com/medes-imps/imogene

- On the question: "What is your knowledge on the use of IT tools?" Six health personnel affirm that they have basic knowledge on IT tools like Word and Excel. One of the six affirms that she already used an electronic health record. All the people from the Regional and the NTCP affirm that they have good knowledge on the use of Word and Excel, but they prefer to hire IT experts to build the Excel files they need for their statistics.
- On the question "What do you expect from the use of a software for epidemiological surveillance?" Only one health worker estimated that he did not see what would be the benefit of the electronic system to his work. From the others, if we want to improve their work with an electronic tool, this tool must: (1) Solve the previous problems and be close as possible to the current tools (data collection sheets and reports) to facilitate the ease of use; (2) Consider data management such as patient registration and following, sharing patient information to entitled users, automatic generation of patient cards, easy identification of lost sigh patients, easy identification of all health centers where TB patients are treated; (3) Allow the production and distribution of basic statistics automatically.

3.2 Implementation of EPICAM

In this section, we present the outputs of the interventions implementation. This constitutes the results of the action research applied in this study. In effect, the combination of Scrum and MDA is deemed successful when its effectiveness can be demonstrated. The successful implementation of interventions is a testimony to the success of the approach. In the next sections, we present these results organized into Pre-Intervention, Intervention and Post-Intervention.

3.2.1 Pre-Intervention

The aim of the Pre-Intervention was to produce a preliminary list of interventions (the summary is presented by the table 3 in the Appendix B) for the NTCP to assess, enhance and approve.

Lesson learned. The first lesson we learned was that epidemiological surveillance systems requirements cannot be fixed and change generally. Then, we should be careful of the approach and tools to use for the development of related software.

3.2.2 Intervention

The Intervention phase was organized into six Sprints. During the development, Scrum boards were used to display the status and progress of each sprint. Ad hoc meetings were also used when some issues were identified. During the Scrum meetings, the developers found it easy to define a common goal and commit to achieve the goal with the product owner on the one hand, and the NTCP staff on the other hand. The following paragraphs give the details of the execution of each Sprint.

Sprint 1: development of a first prototype. The goal of the first Sprint was to design the platform and to develop the first prototype. This Sprint took four iterations of one week each. All the interventions involved were not completely implemented successfully. The table 4 in the Appendix B presents a summary of the interventions implemented in this Sprint.

Lessons learned many lessons were learned during this Sprint:

- We should be careful when using a tool for the complete generation of the software: we remarked that it is difficult to find a tool that fits the requirements of a specific context.
- Domain experts are not generally available: NTCP staff were involved in the modeling and the generation of the software. However, we noted during the Scrum meetings that they were not generally attending.

After a deep reflection during ad hoc meetings between the researchers, the MEDES engineer, the NTCP staff and CPC expert it was decided to develop and update the Imogene platform with the user management module that fit to the NTCP needs.

Sprint 2: implementing user management module generator. The interventions involved in this Sprint comprised on the one hand the specification, analysis, design, development and integration of the user management module in the Imogene platform and on the other hand, the modeling and generation of the first prototype. These interventions were done by the researchers supervised by the MEDES engineers. It lasted 4 months with 8 Sprints. The table 5 in the Appendix B presents the summary of the list of interventions implemented during this Sprint.

Tests and experimentation. The second Sprint permitted us to obtain a first working version of the system that the end users in the field can use in a real environment. After the tests and the Scrum review meeting with the working group, we validated the developed platform with eight health workers in six hospitals.

Lesson learned globally, two lessons were learned at the end of this Sprint: (1) beyond the Scrum team, the end users must be integrated in the validation of the software; (2) the immaturity of the MDA tool can slow down the development of the software system. Then, it can be a serious barrier to its adoption.

Sprint 3: Updating the application. The third Sprint consisted of the updating of the application by the integration of the end users remarks and the generation of another prototype. This was done in only one week and the Scrum weekly meeting allowed us to validate the new version.

Sprint 4: Developing the reporting and mapping modules. During the fourth Sprint, the reporting and the mapping modules were developed (by the developers supervised by MEDES engineers) in two months, organized in 5 Sprints. The reporting module was developed using Business Intelligence and Reporting Tool (BIRT[7]). The mapping module was developed using Open Street Map (OSM[8]). During the Scrum meetings with the working group, the users validated these modules. Then, they were tested, validated at the NTCP and integrated in the code generator. A new version of the platform was validated by the previous users in the field.

Sprint 5: Developing the SMS module. The fifth Sprint consisted of the development of the SMS module in one month with 3 Sprints. To this end, we used Spring Rest API[9]. Actually, to send SMS using the platform, we decided to use a service provided by an SMS provider. The SMS module was tested, validated at the NTCP and integrated in the code generator. The new version of the epidemiological surveillance software generator containing a new version of user management module, reporting module, mapping module and SMS module were named EPICAM[10] and hosted on github. The PIM was transformed into PSM and a platform that we also named EPICAM was generated for the epidemiological surveillance of TB in Cameroon.

As a code-centric developer, we found the development of additional modules arduous than the development of code-centric application. In effect, it was question to reverse engineer Imogene in order to add new elements to generate in the code generator and to link these elements to the graphical interface for the modeling.

[7] https://www.eclipse.org/birt/
[8] https://www.openstreetmap.org/
[9] https://spring.io/guides/tutorials/rest/
[10] https://github.com/UMMISCO/EPICAM

Sprint 6: system update by non-informatic experts. In this research, we wanted a method that allows non-informatic experts to be engaged in the development/updating of their system. Thus, after the development of the platform, the goal of the current Sprint was to prepare the health workers to take charge of the updating. Then, it involved the simulation of the interventions summarized by the table 6 in the Appendix B.

Lessons learned: we learned that counting on health workers to update the system is not advisable. In effect, generally, they have a full time employment and their commitment to contribute to implement interventions is during their sparse time.

Although these can be classified as failed interventions, the experience has some merits. It gave insights on what should be done to make non-informatic experts update or even develop their system: develop a drag and drop modeling tool.

3.2.3 Post-intervention

The Post-Intervention consisted of training end users in the use of the software developed for epidemiological surveillance of TB, deploying the final platform and disseminating the outcomes of the project.

The final version[11] were experimented in 25 pilot sites across the country. Given the success of this pilot phase, the NTCP has adopted the software as its electronic epidemiological surveillance software and extended it in twenty new health centers during the years 2016 and 2017.

3.3 Perceived ease of use and perceived usefulness

The developers and the NTCP staff involved in the development were invited to give their opinion on the combination of MDA and Scrum to develop epidemiological surveillance systems. The analysis of the case data using the Technology Acceptance Model [1] showed that the developers have found it easy and useful to combine agile and MDA for the development of epidemiological surveillance software. However, even if non-informatic experts easily handled and adopted Scrum practices, they found the approach difficult to use because of the lack of skills in modeling and the use of the modeling tool. Table 7 in Appendix B provides an overview of the results.

IV DISCUSSION AND CONCLUSION

In this research, we show how to combine Scrum and MDA to develop epidemiological surveillance software. At its end, we found that if the MDA tool is mature enough, the combination of Agile and MDA have positive effects on programmer productivity and satisfaction, cost-effectiveness, timelines and customer satisfaction.

The generalization of this research can be seen at the level of the case study research. In fact, the selection of a case must be done so that if the research works for this case, it is likely to work for many other cases. But we just chose the epidemiological surveillance of TB because the NTCP were available to collaborate with us in order to find a solution to their problems. We can think that this may hamper the generalizability at the level of epidemiological surveillance systems. But not really because the head of the epidemiological service at Centre Pasteur du Cameroon were involved in the research. This service is specialize in the monitoring of several infectious diseases whose rabies, HIV, Influenza, meningitis, etc.

[11] Source code generated: `https://github.com/ummiscolirima/epicam_tb`

REFERENCES

Publications

[1] F. D. Davis. "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology". In: *MIS Q.* 13.3 (Sept. 1989), pages 319–340. ISSN: 0276-7783.

[2] M. Lenzerini. "Data integration: A theoretical perspective". In: *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. 2002.

[3] Mapatano and Piripiri. "Quelques erreurs courantes d'analyse d'un syst'eme d'information sanitaire(RD Congo)". In: *Santé Publique 2005*. Volume 17. 2005.

[4] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian. "Selecting Empirical Methods for Software Engineering Research". In: *Guide to Advanced Empirical Software Engineering*. Edited by F. Shull, J. Singer, and D. I. K. Sjøberg. London: Springer London, 2008, pages 285–311. ISBN: 978-1-84800-044-5.

[5] B. C K Choi. "The Past, Present, and Future of Public Health Surveillance". In: *Scientifica* 2012 (Aug. 2012), page 875253.

[6] Y. Martínez, C. Cachero, and S. Meliá. "Empirical study on the maintainability of Web applications: Model-driven Engineering vs Code-centric". In: *Empirical Software Engineering* (2014). ISSN: 15737616.

[7] A. Hajou, R. S. Batenburg, and S. Jansen. "An Insight into the Difficulties of Software Development Projects in the Pharmaceutical Industry". In: *Lecture Notes on Software Engineering* (2015). ISSN: 23013559.

[8] R. S. Pressman and B. R. Maxim. *Software Engineering : A Practitioner's Approach, Eighth Edition*. 2015. ISBN: 2008048802.

[9] H. Schlieter, M. Burwitz, O. Schonherr, and M. Benedict. "Towards Model Driven Architecture in HealthCare Information System Development". In: *Wirtschaftsinformatik*. Mar. 2015, pages 497–511.

[10] C. L. Richards, M. F. Iademarco, D. Atkinson, R. W. Pinner, P. Yoon, W. R. M. Kenzie, B. Lee, J. R. Qualters, and T. R. Frieden. "Advances in Public Health Surveillance and Information Dissemination at the Centers for Disease Control and Prevention". In: *Public Health Reports* 132.4 (2017), pages 403–410.

[11] B. Atanasovski, M. Bogdanovic, G. Velinov, L. Stoimenov, A. S. Dimovski, B. Koteska, D. Jankovic, I. Skrceska, M. Kon-Popovska, and B. Jakimovski. "On defining a model driven architecture for an enterprise e-health system". In: *Enterprise Information Systems* 12.8-9 (2018), pages 915–941.

[12] Bin, N. Ali, E. Engström, M. Taromirad, M. R. Mousavi, N. M. Minhas, D. Helgesson, S. Kunze, and M. Varshosaz. "On the search for industry-relevant regression testing research". In: *Empirical Software Engineering* (2019). ISSN: 15737616.

[13] A. Jiomekong and G. Camara. "Model-Driven Architecture Based Software Development for Epidemiological Surveillance Systems". In: *Studies in health technology and informatics* 264 (Aug. 2019), pages 531–535. ISSN: 0926-9630.

[14] L. T. Heeager and P. A. Nielsen. "Meshing agile and plan-driven development in safety-critical software: a case study". In: *Empirical Software Engineering* (2020). ISSN: 15737616.

[15] OMG. *MDA*. `https://www.omg.org/mda/`. 2021.

A DETAILS ON STUDY DESIGN

The research team consisted of two persons playing the role of researchers and software developers. Both had a strong background and experience in Scrum and Model-Driven Engineering Techniques and tools. The target population of the research was a big stakeholder group including: the product owner working at NTCP, the NTCP Staff who is the technical staff at the NTCP, end users who are the health workers at the different levels of the health system and two epidemiologists - one epidemiologist from the NTCP and one epidemiologist from Centre Pasteur du Cameroun (CPC). The NTCP represented by the Product Owner led the recruitment of the NTCP staff and the end users. Globally, all the people at the NTCP willing to participate in the project were accepted. At the hospital level, the NTCP have chosen the ones that furnish greater amounts of data in the center region for the survey, in the Center region and Littoral Region for the tests.

To understand the current system, we collected the opinions of the final users at all levels of the NTCP. To this end, we conducted the survey. The main goal of this survey was to find out what the stakeholders think of the replacement of manual management by software management. The specific objectives were: Understand the current network, Know how stakeholders collaborate, Collect the specific needs of users in the field to complete the specifications already gathered, Determine the computer skills of users of the current surveillance network, Know the opinion of final users about an electronic network. These objectives gave rise to the following questions:

1. How do you manage your data?
2. What are the main problems you generally have?
3. What is your knowledge on the use of IT tools?
4. What do you expect from the use of a software for epidemiological surveillance?

During this research, we collaborated with the NTCP which were using manual tools for epidemiological surveillance in order to identify the problems they have and the challenges to effectively replace this manual management by the software approach. To this end, we needed a method and a process that fully engaged the stakeholders. That is why we set-up an action research method in which we as researchers and the NTCP as the problem owner were engaged collaboratively to identify the problems and develop effective solutions. This process is constituted by a set of consecutive actions and reflections forming a collaborative approach to research whereby the researcher partners with the subject being studied to achieve a particular outcome. It consisted of a set of aggregated interventions to develop epidemiological surveillance software of TB. These interventions involved a series of actions taken during the development of the epidemiological surveillance of TB in order to come up with an effective system. It is organized into the Pre-Intervention, the Intervention and the Post-Intervention phase.

Pre-intervention phase. During the Pre-Intervention, we identified the problems and proposed solutions. These solutions are divided to give a list of interventions. Thereafter, we conduct an empirical investigation of the research environment by interviewing health workers at all the levels of the health system. This empirical investigation allows us to identify problems and issues related to the current system. At the end, we identified a shared practical problem that was presented to the NTCP staff.

Intervention phase. Involves the development of the solution to the problem. It is cyclic, implemented through iterations of planning, implementing, validation, and reflecting. During the Intervention Phase, the list of interventions identified during the Pre-Intervention Phase are implemented. These interventions are executed in iterations. Each iteration is composed of the planning, the implementation, the validation, the reflection. The iteration nature of the process provides us opportunities for improvements to the knowledge, methods, and better understanding of the approaches and methods.

Post-Intervention Phase. Once the Intervention Phase is completed, Post-Intervention takes place. It includes all the tasks to implement (deployment, training end users, vulgarization, etc.) in order to make the solution available to the end users.

To collect data during the project, we used two data collection techniques: semi-structured interviews and observations. The data were recorded by taking notes in structured form in a notebook. These notes were supplemented with pictures (for instance, the photos of data collection forms). After the data collection, the analysis consisted of labeling and organizing the data in order to identify codes, themes and relationships between them (the summary is given in table 1).

B LIST OF TABLES

	Research method	Data collection	Data analysis
Pre-Intervention	Survey and Action research	Interviews	Open-coding
Intervention	Case study and Action research	Observation and Interview	Thematic coding
Post-Intervention	Case study and Action research	Interview	Thematic coding

Table 1: Data collection and analysis methods

Class diagram	**Imogene model**	**Source code**	**Comments**
Class	Card Entity	Form for data collection/consultation and table in the database	The source code generated by the imogene generator involved the form containing the class attributes as field and a table containing class attributes as column names in the database.
Attributes	form fields	Attributes of Java classes	Each card entity contains many fields and will correspond to attributes of Java classes. These fields may have different types.
Class relations	Relation fields between two card entities	One-to-one, one-to-many, many-to-one and one-to-many relations between classes	The source code corresponds to the relation between classes as prescribed by the Object relational mapping[12]

Table 2: Some mapping elements between PIM, PSM and source code

Interventions	**People involved**	**Scrum practice applied**	**Status**	**Comments**
Establishment of the working group	Researchers and product owner	Face-to-face	Complete	The researchers and the product owner select health workers to form the NTCP staff of the project.
Problem identification	Researchers and NTCP staff	Face-to-face and Scrum meetings	Complete	The NTCP staff meet regularly to discuss the problems NTCP have during the epidemiological surveillance of TB.
Users survey	Researchers and Epidemiologist	Scrum meeting	Complete	The survey were conducted by the researchers and the epidemiologist (product owner). At its end, a Scrum meeting was organized to share the results at NTCP.
Specifications and Analysis and Product Backlog	Researchers and NTCP staff	Product Backlog, Sprint, Scrum meetings, Scrum iteration	Complete	After the definition of the product backlog, the definition of the specifications, analysis and product backlog goes in Scrum iteration. During this process, Scrum meetings allowed the NTCP to make remarks and improve the product.

Table 3: Summary of the list of interventions planned and implemented during the Pre-Intervention

Intervention	**People involved**	**Scrum practices**	**MDA practice**	**Status**	**Comments**
System design	Researchers and NTCP staff	Sprint, Iterations, Scrum meetings	Modeling PIM	Completed	Beside the other tasks, the staff were involved in the modeling and the validation of the PIM.
Prototype implementation	Researchers and NTCP staff	Sprint, Iterations, Scrum meetings	Transforming the PIM to the PSM and the PSM to the source code	Incomplete	The health workers participated in the validation of the PIM and the PSM. However, the user management feature proposed by Imogene did not correspond to the features expected by the users.
Product backlog update	Researchers and Product owner	Scrum meetings	None	Complete	Given the remarks on the users' management features, the product backlog is updated with new features to develop and integrate in the Imogene generator.

Table 4: List of interventions implemented during the first Sprint

Intervention	People involved	Scrum practice	MDA practice	Status	Comments
Training	Developers and MEDES engineers	Scrum meetings	None	Completed	Capacity building of developers.
Specifications	Researcher and NTCP staff	Scrum meetings	None	Completed	Specifications of how the user management module should work.
Development	Developers and MEDES engineers and Product owner	Scrum meetings, Scrum iteration, Sprint	Constructing the User Model	Completed	Construction of user model and its use to complete the imogene model.
Validation	Researchers and NTCP staff	Scrum meetings	None	Completed	Validation of the user management module by the NTCP staff.
Prototyping	Researchers and NTCP staff	Scrum meetings	Modifying the PIM and the PSM	Completed	Updating the PIM and the PSM with user management module and generation of a first prototype.
Prototype validation	Researchers and NTCP staff and end users	Scrum meetings	None	Completed	Validation of the prototype by the end users.

Table 5: List of interventions implemented during the second Sprint

Intervention	Status	Comments
Product Backlog definition	Complete	The health worker define the list of tasks to execute
Sprint organization	Complete	The list of tasks were organized into 2 Sprints of one week each
Organizing Scrum meetings	Complete	The simulation of the organization of Scrum meetings were done by the health personnel
Modeling the PIM	Complete	The health worker used Dia to model a small application
Installing MDA tool	Complete	This intervention was the most arduous. The health worker found the installation of Eclipse and the different plugins out of his competence. However, with the help of the developers, he finally installed it
Construction of the PSM from the PIM	Incomplete	The installation difficulties discourage the health worker from continuing the experience.

Table 6: List of interventions implemented during the sixth Sprint

Theme	Characteristics	Developer	Non-experts	Comments
Perceived ease of use	Ease of use	Agree (100%)	disagree (100%)	The Non-informatic experts found the MDA tool out of their competences
Perceived usefulness	Programming productivity	Agree (100%)	Agree (100%)	The modeling tool along with the Scrum practice allowed us to produce the new versions very fastly
	Programming satisfaction	Agree (100%)	Agree (100%)	Combined with MDA, the Scrum process allowed us to gradually build models and generate executable source code.
	Cost-effectiveness	Agree (100%)	Agree (100%)	The reusability of the modeling tool and the update and generation of software allows us to reduce the costs of the development
	Timelines	Agree (100%)	Agree (100%)	The use of models and Scrum practices facilitated the production of the first working versions and the other versions, update given the remarks of the end users
	Customer satisfaction	Agree (100%)	Agree (100%)	The customer were more close to the development and the use of models facilitated the communication. The early deployment allowed us to discover new requirements and facilitate the acceptance of the final product.

Table 7: Overview of the results of the evaluation of the combination of Scrum and MDA by the two developers and three health personnel (representing 27% of the whole team who were involved at the beginning)