Combining Scrum and Model Driven Architecture for the development of an epidemiological surveillance software

Jiomekong Azanzi, Hippolyte Tapamo, Gaoussou Camara

Combining Scrum and Model Driven Architecture for the development of an epidemiological surveillance software

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

Epidemiological surveillance systems evolve with time, depending on the context and the data already collected. Then, the software used must evolve in order to meet requirements. However, introducing new requirements in order to update the software takes time, is expensive and may lead to the problem of software regression. The problem of failed software developed for epidemiological surveillance are often the result of an unsystematic transfer of business requirements to the implementation. This problem can be avoided if the system is established using a well-defined framework/architecture permitting the rapid development/update of the surveillance software. Empirical research shows on the one hand that Model Driven Techniques such as Model Driven Architecture (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. Researchers demonstrated that developers of medical software such as epidemiological surveillance software are experiencing difficulties when following Structured Processes and code-centric approaches. The main goal of this empirical study was to apply the combination of Scrum and Model Driven Architecture for the development of epidemiological surveillance of tuberculosis. During this research, we found the approach ease of use and very useful when the MDA tool can generate the complete source code. It has had positive effects on programmer productivity and satisfaction, cost-effectiveness, timelines and customer satisfaction. In addition, we learned that to involve non-informatic experts in the development/update, the modeling user interface must be as simple as possible.

Keywords

Tuberculosis ; Epidemiological surveillance systems ; Agile ; Scrum ; Model Driven Engineering ; Model Driven Architecture ; EPICAM

I INTRODUCTION

Over the centuries, communicable diseases have always been one of the main problems for human health because they spread rapidly and result in high mortalities. Currently, respiratory infectious diseases such as Influenza, Coronavirus disease, Tuberculosis (TB) are in the top of the most killers globally. Effective management of these diseases requires to put in place epidemiological surveillance systems. Epidemiological surveillance systems enable the collection, analysis, and interpretation of data, closely integrated with the dissemination of these data to

the public health practitioners, clinicians, policy makers and the general population 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 [28, 49, 54–56]. Depending on the data gathered on the field, the epidemiologists may need to collect a new parameter in order to explain a phenomenon. For example, collect the data on the height of the patients in order to calculate their body mass index and understand why some of them are dying the most. This task may be done by using supplementary materials such as paper forms or spreadsheet software and lead to the problem of data integration [6]. In fact, at the end of data collection, the data obtained using supplementary materials must be put together with the other data. 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 [61] because bugs can be introduced during the source code update. These problems occur because of the problem of failed software developed for epidemiological surveillance.

The problem of failed software in the health domain is often the result of an unsystematic transfer of business requirements to the implementation [44, 47]. In the case of epidemiological surveillance, this problem can be avoided if the system is established using a well-defined framework/architecture permitting the rapid development/update of the surveillance software by non-informatics experts such as health workers.

Model Driven Techniques in general relies on models and model transformations for the development, maintenance and evolution of software [40, 48, 64]. Through the architectural separation of concerns, the Model Driven Architecture (MDA) in particular assumes the evolutivity, portability, interoperability and reusability [10, 15, 68]. 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 the architecture consistency from analysis to implementation [32, 33, 39, 59]. Model Driven Techniques experience incredible success in many domains such as automotive, business process engineering, and embedded systems [64]. In the health domain, many authors has proven that Model Driven techniques is an effective approach for the development of health software [11, 14, 20, 47, 53, 57, 62].

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 the development which are requirements elicitation, design, implementation, verification, maintenance. Software development processes are grouped into Structured Processes such as the waterfall or V-model and Agile Processes such as Scrum [46]. 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 [31, 38, 44, 65]. In the particular case of epidemiological surveillance of diseases like Covid-19, it can be very difficult to capture all the requirements at such an early stage of a project. In addition, any change introduced once a project is underway can create cost and budget overruns. However, Agile software development processes such as Scrum [9] experience great success in software development organizations [29, 42, 45]. Many empirical studies report significant benefits being gained from utilising Agile practices such as reduce costs, reduce time to market and increase quality [31, 34, 38, 65].

Many authors have written about the benefits of deploying Agile [21, 27, 30, 31, 34] on the one hand and Model Driven Techniques [11, 14, 20, 24, 53, 57, 62] on the other hand to develop health software. But, unfortunately, the percentage of empirical studies be them surveys, experiments, case studies or postmortem analyses that provide data to illustrate the effects of the combination of Agile and MDA approaches over different quality characteristics such as productivity of the development team and satisfaction of stakeholders is still very low, which hampers the generalizability of the results. Thus, our main contributions are:

- to provide empirical data that contributes to giving a scientific answer to the following research question: how to combine Scrum and MDA to develop and maintain epidemiological surveillance software?
- provides a framework based on MDA that can be used for the rapid development of any epidemiological surveillance system.

The remainder of the paper is organized as follows: Section II presents the research context; Section III presents the MDA approach and Scrum Process; Sections IV and V present respectively the research design and the research results; Sections VI and VII present respectively the threats to validity and related works. We end with the conclusion in Section VIII.

II RESEARCH CONTEXT

The EPICAM[1] (Epidemiology in Cameroon) project aims to improve Cameroon's health system by strengthening the Health Information System. Given the problem of failed software developed in the health domain [44, 47], the 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 were set-up. 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 [13, 62], 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 and to investigate factors considered as important for its adoption.

Given the limits of the code-centric approach [39] and the structured process [31, 38, 44, 65] for software development, Scrum as the process which aims to describe step by step the development of epidemiological surveillance software and the MDA approach which aims to define models and automatically generate the software were adopted.

III MDA AND SCRUM

Epidemiological surveillance software can be developed on the one hand using a code-centric or a MDA approach. With the code-centric approach, generally, the developers use languages like Unified Modelling Language (UML) and methods like Unified Process (UP) or Merise in the

[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

analysis and design steps. The conceptual model such as class diagram/Entity Relational model is usually designed at once and used to develop the software [10]. With the MDA approach, the conceptual model such as Platform Independent Model (PIM) and Platform Specific Model (PSM) are used to automatically generate the executable source code [8, 15]. Then, we choose MDA in this research because related research [39] shows that it improves programmers productivity and satisfaction, shortens development time, improves software quality, and promotes reuse of tools and models and saving costs.

On the other hand, whether code-centric or MDA approaches are used during the development of a software, there is a need for a software development process [31, 38, 65]. Given that epidemiological surveillance systems evolve faster, the recommended process is Agile. While several agile development processes have existed for some years, a large number of software organizations have implemented agile processes such as Scrum [65]. Then, in this research we used Scrum.

In the next subsections, Model Driven Architecture (Subsection 3.1) and Scrum (subsections 3.2) are presented.

3.1 Model Driven Architecture

Model Driven Engineering (MDE) is a software development approach based on the use of models to develop software artifacts. Its goal is to improve the productivity and maintainability of software by raising the level of abstraction from source code to high-level domain-specific models such that developers can concentrate on application logic rather than implementation details [8, 33, 48, 64]. It is identified by several paradigms whose Model Driven Architecture (MDA). MDA [8, 10, 15, 40] is promoted by the the Object Management Group (OMG)[5] as an architecture and a framework for software development based on models.

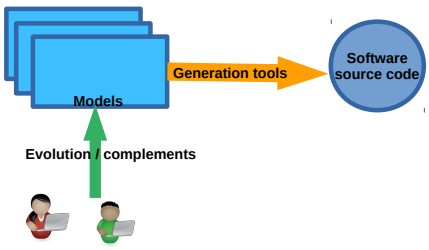

Figure 1: MDA Software development approach

Globally, the MDA approach (Figure 1) is an approach to software specification, design, and implementation which provides guidelines for structuring software specifications that are expressed as models. It focuses on forward engineering in which the executable source code is (semi)automatically generated from abstract, human-elaborated modeling diagrams such as a class diagram [20, 47, 53, 57]. A class diagram describes the structure of a domain by identifying the domain classes (e.g., patient), their attributes (e.g., age, sex), their operations (e.g., calculate a body mass index) and the relationships amongst classes (e.g., the relation between a patient and his/her appointments at the hospital) [14, 18]. Models are used throughout the development life cycle of an application, that is, from the application specification to code

[5]https://www.omg.org/

generation. For the generation task, the MDA approach proposes to provide developers with automation tools [10, 23]. When adopting the MDA approach, a natural counterpart is the framework which is the body of code that implements the aspects that are common across an entire domain. This framework allows us to link the business level and the information technology [15, 47].

The two key concepts of MDA are models and transformations. All the models are well defined by a modeling language, itself defined by syntax and semantics in a meta-model. The transformation from one model (e.g., requirements) to another model (e.g., design) is a systematic process explicitly defined by transformation rules. During the development of applications using the MDA approach, the business models as a result of the discussion between domain experts and modeling experts are (partially) automatically transformed through the three models layers ($CIM \longrightarrow PIM \longrightarrow PSM$) [14, 15]:

- **Computational Independent Model (CIM):** presents exactly what the system is expected to do. It hides all the information technology related specifications to remain independent of how that the system will be (or currently is) implemented;
- **Platform Independent Model (PIM)**: obtained from the CIM, the PIM is exactly what its name denotes: a model that is free of specific hardware or computing platform requirements and constraints. An example of a PIM is the class diagram modeling the domain and presenting the entities and the relationships between these entities;
- **the Platform Specific Model (PSM):** The PIM constructed can be transformed (manually or automatically) into one or more PSM for different target platforms by integrating platform specific details to the PIM;
- **Executable source code:** The PSM is used to generate the software solutions which are application source code, configuration files, component descriptor files, deployment files, build scripts, documentation, etc. Depending on the maturity of the MDA tool-set, code generation varies from significant to substantial or, in some cases, all the source code is generated.

Model Driven Techniques make a greater impact in the software engineering domain. Researchers claim several advantages over code-centric approaches among which [32, 33, 39, 59, 64]:

- Improve productivity and shorten development time by increasing automation (generating code and other artifacts from models) and promoting reuse. For instance, Martínez et al. [39] demonstrate that maintaining Web applications using Model Driven Techniques improves the performance of developers.
- Improve software quality by: earlier detection of bugs, maintaining the architecture consistency from analysis to implementation, improving communication and information sharing among stakeholders, and enabling traceability throughout software development artifacts.

In the domain of health, the literature review shows that the MDA approach has been widely used [11, 14, 20, 47, 53, 57, 62]. To cite a few:

- Hajou et al. [44] showed how Model Driven techniques can be used to overcome the difficulties of adapting pharmaceutical software to the standard by proposing a rapid update of pharmaceutical products;
- Schlieter et al. [47] showed how an MDA approach can be used in order to build a method, fundamental for a systematic creation of an application system. Their use-case is a software for IT based workflow support for an interdisciplinary stroke care project;

- Curcin et al. [36] presented a work on the use of MDA to build tools which allow non-IT experts to model their requirements and to generate data collection tools for medical organizations;
- Blobel and Pharow [16] presented a work done in Germany consisting of the establishment of a health telematics platform combining card enabled communication with network based interoperability;
- Jones et al. [11] presented the use of MDA to develop m-health (Mobile-health) application providing an integrated set of personalized health-related services to users;
- Rayhupathi and Umar [24] proposed the use of MDA to develop a system capable of tracking patient information.

Concerning the domain of epidemiological surveillance, we found some tools such as Imogene[6], Magpi[7], Open Data Kit(ODK)[8] generally used for the modeling and the generation of data collection software. However, epidemiological surveillance software involved more features.

3.2 Scrum

Agile software development Processes started officially when the agile movement presented the agile manifesto in 2001. The agile manifesto contains the central values which proposes to place more emphasis on people, interaction, working software, customer collaboration, and change, rather than on processes, tools, contracts and plans during software development [4, 29, 42, 45].

Agile Processes rely on an iterative, incremental and adaptive development cycle which considers that the needs cannot be fixed and proposes to adapt the development to the changes. It's intended to support early and quick production of working code. In the particular case of health information, it allows fast deployment and adoption [21, 30, 31, 34, 44, 53, 65]. It consists firstly of setting a short-term goal and of getting into the project without delay. Once the first objective is attained, we take a short break and adapt the path according to the situation of the moment, and so on, until we reach the final destination. This helps us to give more visibility, involving the client from the beginning to the end of the project and adopting an iterative and incremental process. In a broader sense, an Agile Process can be seen as a process consisting of an initialization step, iterative steps and incremental steps [4, 7]. While several agile development processes have existed for some years, a large number of software organizations have implemented agile processes such as Scrum [65].

The first reference of the term "Scrum" originates from Japan and refers to an adaptive, quick, and self-organizing product development. Scrum describes how the team members should be organized in order to develop a system using Agile principles [9].

Scrum Process includes three steps: Pre-Development, Development and Post-Development.

- **Pre-Development:** the Pre-Development step includes the planning and the design. The planning is done based on the Product Backlog, in collaboration with the customer. The requirements are prioritized, the resources (team, tools, etc.) needed for their implementation are estimated, risks assessed, identification of training needs and verification management approvals made. The design phase is done by defining the architecture based on the Product Backlog List; preliminary plans for the contents of releases are proposed. The

[6] https://github.com/medes-imps/imogene
[7] www.magpi.org
[8] https://opendatakit.org/

design meeting permits us to make decisions about the implementation and preliminary plans for the contents of releases are prepared;
- **Development:** during the development steps, the system is built in Sprints during which the functionalities are developed or enhanced to produce new increments;
- **Post-Development:** this phase is reached when the requirements are completed. The system is ready for the release and tasks such as system integration and documentation are made.

Systematic literature review papers demonstrate the relevance of Agile in industry [4, 29, 42, 45]. Many empirical studies report significant benefits being gained from utilizing Agile practices such as reduce costs, reduce time to market and increase quality [31, 38, 65]. In addition, in October 2012, the Association for the Advancement of Medical Instrumentation (AAMI) released a Technical Information Report (TIR) entitled "Guidance on the use of agile practices in the development of medical device software [27]". The AAMI recognised the shift in the non-regulated software development industry towards more agile practices and the evidence presented from successful adoption of agile practices in medical device software development organizations.

The use of Agile Processes in the health domain is an old practice. Many authors presented its adoption for the design and implementation of health software [21, 30, 31, 34, 44, 53, 65].

However, these studies use Agile to develop health software with the code-centric approach. In this study, we choose to use Agile to develop software using the MDA approach. The next section presents the research design.

IV RESEARCH DESIGN

The research design was inspired by papers on empirical research methods in software engineering [17, 66]. They describe a number of empirical methods available for software engineering research, examine the goals of each method and analyze the types of questions each best addresses. Theoretical stances behind methods, practical considerations in the application of the methods and data collection are also reviewed. Taken together, these information provide us with a suitable basis for both understanding and selecting a research method from the variety of methods applicable to empirical software engineering research. These works allowed us to design our study. Thus, we designed the research question (Section 4.1), stakeholders involved in the research (Section 4.2) the research methods (Section 4.3), data collection and analysis (Section 4.4), and how the models were constructed and transformed 4.5).

4.1 Research question

On the question on how to develop epidemiological surveillance software, evidence shows 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 [31, 38, 65]. On the other hand, MDA is more effective than code-centric approach during the development and the update/maintenance of the software [32, 39, 59]. Then, given the problem of failed software, 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 and maintain epidemiological surveillance software? We have used the term meshing as the common term for combining, mixing, and coexisting.

4.2 Stakeholders involved

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, he is the member of the Scrum Team responsible for the prioritization of the features to develop. He also plays the role of interchange belt between the software team and the others stakeholders;
- **The NTCP Staff:** this is the technical staff at the NTCP who are going to participate in the development and the maintenance of the surveillance software. They assists to the Scrum meeting and will be trained on the update of the system;
- **End users:** they are the health workers at the different levels of the health system. They are the final users of the platform developed. They used and validate the different versions developed before the release of the final product;
- **Two epidemiologists:** one epidemiologist from the NTCP and one epidemiologist from Centre Pasteur du Cameroun (CPC). It should be noted that the Service of Epidemiology of CPC monitored many diseases including influenza, HIV, rabies, etc.

The NTCP represented by the Product Owner led the recruitment of the NTCP staff and the end users.

4.3 Research methods

Our aim in this research as presented by the research context (Section II) was to do intensive research on a specific case of 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 [22, 25, 65, 66]. 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. However, to understand the current system, we thought it necessary to investigate the opinions (through survey) of the stakeholders about the current system and its replacement with a software approach. 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. In effect, action research is used in real situations where major challenges cannot be studied without implementing them, and where implementation implies a long term commitment because effect may take time to emerge [17, 35, 63, 66]. In addition, we were having the NTCP willing to collaborate to both identify the problem, and engage in an effort to solve it. Combining empirical research methods is largely accepted in empirical software engineering research [3, 5, 17, 66]. The research methodology is presented by the Fig. 2.

This methodology is composed of the Pre-Intervention Phase, the Intervention Phase and the Post-Intervention Phase. During the Pre-Intervention, the problems at the NTCP are identified, the method to solve these problems and the list of interventions to implement are defined. During the Intervention, the solution to the problem is built and during the Post-Intervention, this solution is deployed. In the following paragraphs, we present the different research methods and how they were used.

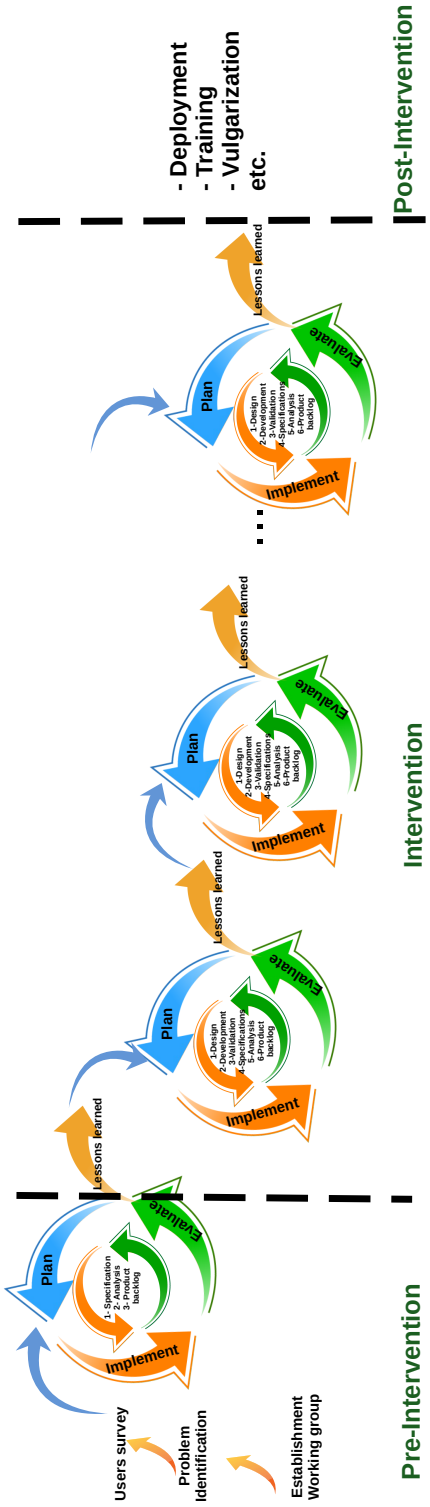

Figure 2: Research methodology

4.3.1 Case Study

The aim of studying the case of the development of epidemiological surveillance software was to portray this development so that it can be reproduced to other epidemiological surveillance. Then, we choose the case study method because it is suited for understanding the complex real-world setting [22, 25, 65, 66]. The selection of the case of epidemiological surveillance of TB was made because of the NTCP interest to improve epidemiological surveillance and the researcher's interest in understanding the effects of development approach and process during the development. Thus, our aims were to portray and provide a deeper understanding of the development of epidemiological surveillance of TB using a software development approach (either code-centric or MDA) and a software development process (either Structured or Agile). Section 4.5 provides the justification elements of the choice of Scrum and MDA.

4.3.2 User survey

Putting in place an electronic surveillance network requires the collection of user needs, a good analysis of the existing network, information and knowledge of the users in the use of Information Technology (IT) tools. Therefore, to put in place the electronic surveillance of TB in Cameroon, we collected the opinions of the final users at all levels of the NTCP. To this end, we conducted the survey to better document the problem so that the system deployed will be accepted.

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.

It should be noted that survey research is accepted in empirical software engineering research [2, 17, 66]. In the next section, we present action research.

4.3.3 Action Research

Action research is an iterative process in which plans are created, implemented, revised, then implemented, lending itself to an ongoing process of reflection and revision. During this process, findings emerged as interventions are implemented. It is used where major challenges cannot be studied without implementing them, and where implementation implies a long term commitment because effect may take time to emerge [12, 17, 35, 66]. It was successfully used by Adam Alami et al. [63] to study how to implement changes in Free and Open Source Software (FOSS) communities. Thus, the researchers joined with the FOSS community and proposed solutions for improving and enhancing FOSS practices. The study included interviews with FOSS developers in order to seek solutions to their problems.

Like in the case of epidemiological surveillance, there was a clearly identified group of stakeholders that suffered problems. Thus, 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 [12].

As presented by the Fig. 2, the action research we applied 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. Globally, the Pre-Intervention is composed of the following interventions (see Fig. 2):

1. Establishment of the working group: this is a group of people involved in the whole research;
2. Problem identification: this is the identification of the problems of the current system;
3. Users survey: this is the survey of users of the current system;
4. Specification: this is the description of users needs;
5. Analysis: after the description of users' needs the analysis allow us to well understand these needs;
6. Identification of the list of interventions: once the system is well understood, the list of tasks to implement in order to provide the solution is defined.

Intervention phase. Involves the development of the solution to the problem. It is cyclic, implemented through iterations of planning, implementing, validation, and reflecting [12, 35]. During the Intervention Phase, the list of interventions identified during the Pre-Intervention Phase are implemented. These interventions are executed in iterations (see Fig. 2). Each iteration is composed of the planning, the implementation, the validation, the reflection.

1. The planning consists of selecting a list of intervention and planning their execution. The planning is designed and enhanced iteratively. It is obtained from requirements, precedents implementations, problems encountered, etc. Once these elements are identified, the list of interventions is updated and the planning is updated.
2. The implementation consists to implement all the interventions planned during the planning.
3. The validation consists of the confirmation that the result of the implementation corresponds to the outcome. During the implementation and the validation of the solution, new learning may emerge.
4. The reflection consists of drawn lessons, revise and prioritize the list of interventions.

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.

	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

4.4 Data collection

To collect data, 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). It should be noted that in case study research, many authors acknowledge the use of interviews for data collection [19, 25, 65]. Interviews and observations were used for data collection during the action research conducted by Adam Alami et al. [63]. Survey data are collected either using quantitative data or qualitative data such as interviews [3, 5, 17]. In the next paragraphs, we present the data collection during the different phases of the research.

Pre-intervention. The data collection during the Survey was made using semi-structured interviews. The participants were health workers involved in the epidemiological surveillance of TB at the hospital, regional and NTCP levels. These interviews were conducted by the two researchers and one epidemiologist. Each interview lasted between 30 to 45 min and was recorded in a notebook. The text and conclusions were presented to the NTCP staff.

The following open-ended questions were asked to participants:

- How do you manage your data?
- What are the main problems you generally have? What are the solutions you generally use to solve these problems?
- What is your knowledge on the use of IT tools?
- What do you expect from the use of a software for epidemiological surveillance?

Alongside the data collected from the health workers, other sources of data used in the research were the supports that are used at NTCP for TB surveillance.

To analyze this data, we started from scratch and created codes based on the data. Thus, we go through the transcript of every interview and highlight everything that jumps out as relevant or potentially interesting.

Intervention. During the Intervention phase, the data were gathered by watching how the combination of Agile and MDA is being used during the development of epidemiological surveillance of TB. Thus, data from daily observation of the implementation of the interventions was collected in order to gain an understanding of the software development phases and decisions. These observations were documented by note taking in notebook. In some cases, we used semi-structured interviews to investigate a phenomenon. For instance, we observed that only one health worker investigated himself in the use of the modeling tools. To understand this phenomenon, a semi-structured interview was conducted.

Then, through self-assessment, we determined how things went and how they can be improved. To this end, after each Sprint, the following questions were answered:

- What went well? - we determine if what has been done can be replicated in another context.

- What didn't go well? - we acknowledge mistakes and pledge improvements.
- What can we do to improve? - We look at what was done during the Sprint and if there is a better way to do it.

The responses to these questions were used to measure the acceptance of the approach using Technology Acceptance Model (TAM) as proposed by Fred Davis [1]:

- Perceived ease of use: measure the extent to which the developers and the non-informatic experts involved in the development think that using the combination of agile and MDA to develop their system is easy.
- Perceived usefulness: measure the extent to which the user (the developer and the non informatic-user) thinks that using the methodology enhances his job performance. This is measured using the following indicators: programmers productivity and satisfaction, development cost-effectiveness, timelines, customer satisfaction, absence of problems.

Other sources of data during the Intervention Phase were:

- Email: this was the online channel of information exchange, decision discussion and sometimes, decision making;
- Meetings: notes were taken during meetings.

At the end of each step, a report was prepared and shared to all the team by emails and during meetings.

Post-Intervention. The post-intervention phase consists of many activities such as the deployment of the solution and its vulgarization. In the case of the EPICAM project, after the intervention phase, a meeting was organized by the NTCP. During this meeting, we have presented the approach in order to have the opinion of the users. After the presentation, the questions, remarks about the approach were noted.

4.5 Model construction and transformation

In this research, the PIM is constructed using UML class diagram. To construct UML class diagram, many tools are proposed. We choose to use Dia Diagram Editor[9] because we master its use to build UML models. On the other hand, Dia is used to build large diagrams spanning multiple pages that can be saved in XML format and be printed.

The assessment of tools used for complete code generation in epidemiological surveillance (presented in Section 3.1) reveals that Imogene were more close to the features expected by the NTCP. Our proximity with the MEDES, the company which developed Imogene motivated us to adopt it.

Imogene is a platform to model and generate integrated data collection information systems. Thus, the class diagram (PIM) designed using Dia is transformed into the imogene model (Imogene). Thereafter, the model is used to generate the application. Imogene is developed using the Eclipse framework, the Eclipse Modeling Framework including Ecore meta model (to describe model) and the plugins furnished by Eclipse for code generation from the model.

As the Eclipse EMF, the Imogene platform provides graphical tools to model business entities and generation tools to generate a set of applications based on the model. The modeling phase consists of defining the forms (with their fields) which will be used to collect/manage data. These forms contain the fields, their types and specific constraints; and the relations with other forms.

[9] www.dia-installer.de

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[10]

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

Additional information such as the application description, the description of each form (card entity) and each field (field entity) may be added using the imogene editor.

To manage access to the data, Imogene editor provides a particular card entity named actor. It's used to generate the application for user management. In our case, the user rights to the data is defined in the class diagram and converted in the imogene model using actor card entity.

Once the PSM is constructed, the corresponding applications are generated. In our research, given that there was not a tool for direct transformation of PIM to PSM, this task was done manually. The table 2 presents an overview of the transformation from the PIM (class diagram) into the PSM (Imogene model) and from the PSM into the source code.

V RESULTS

Globally, this research allowed us to demonstrate that the combination of Scrum and MDA allows effective development of epidemiological surveillance software when the MDA tool is mature enough to generate the entire source code. In effect, Scrum practices allow users to participate in the entire process of modeling and generation, which make the solution acceptable by end users. The details of the research results are given in the following subsections: Section 5.1 presents the results of the survey beside the health workers, Section 5.2 presents the results obtained during the implementation of the approach, and Section 5.3 presents the results of the evaluation of the approach by people who participated to this research. The results obtained, the finding and the lessons learned allowed us to provide an answer to the research question in Section 5.4.

5.1 Users survey

The survey of health workers involved 8 (53%) health professionals working in four hospitals where TB patients are treated, 1 (40%) health regional delegate, 6 (4%) 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 - these elements are added as additional materials to this paper) 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 hospital level that only manual management was used. In effect, at the central level, the data collection tools used are constructed using Microsoft Word, printed and distributed to health personnel. At the regional and the central level, manual management was combined with

software management. Excel was generally used to put all the data collected in the hospital together and produce statistics. For instance, at the regional level, all the data from the hospitals of the region are compiled in an Excel file to obtain the overall situation of the region. At the NTCP level, all the data at the national level are compiled in an Excel file in order to have the overall situation in the country. Paper data is generally transmitted using the land transport from the hospitals to the Region. From the Region to the NTCP, they used USB keys and email.

- 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. In effect, everything at the hospital level is manual, which causes many difficulties to the health workers: difficulties to verify and correct the errors in the data collection tools, difficulties to guide the patients to the health center near to his home, difficulties to recall patients who have not come to their appointments, difficulties to update data collection (which is modify each years), difficulties to integrate data (for instance, children data) with the entire data. In addition, the use of many data collection tools (patient card, treatment register and laboratory register) to collect the same data (information on patient) is a tedious task; archiving paper data takes too many places and makes it difficult to find information. (2) The problem of manual analysis of data. This problem causes many other problems such as stock-outs of medicine, completeness, promptitude and erroneous statistics. (3) The problem of transmission of data using land transport. The main consequences of this problem noted at the Regional and the NTCP 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%. Generally, it is advised to transmit $\frac{2}{3}$ reports and to send the rest soon it is available. 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 is 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.
- 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 epidemiological surveillance software?" 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 easy use; (2) 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. They report to be aware that the power and Internet outage will be the weakness of this system compared to the existing system and that the training of health personnel on the use of the electronic tool is the key to its success.

The table 3 presents the summary of the results.

Question	Participants	Code	Comments
How do you manage your data?	Health center	Manually	Data are collected using data collection forms and analyzed by counting by hand and filling the reporting form.
	Regions	Manually and with software	The paper data arrive from the health center. These data are aggregated in an Excel form. Some data is collected at our level manually.
	Central	Manually and using software	The data comes from the region in Excel form or paper form. We aggregate all these data in an Excel sheet. Some statistics are used in the data form.
What are the main problems you generally have?	Health center	Errors in data and tedious tasks	The fact that the data is in the paper form make it easy to make errors. In addition, collecting and analyzing data using paper is a tedious task.
	Regional	Errors and tedious task	Some data comes from the health centers (paper forms) with errors and we have to contact the health personnel to correct these errors
	Central	Data integration	When the data comes from the Region in an Excel sheet, we should aggregate these data in order to have a global and local view of the burden of the disease.
What are your knowledge on the use of IT tools?	Health center	Basic knowledge	Almost all the health workers in the health centers have basic skills (use computers only for typing or searching information on Internet) in the use of IT tools.
	Regional	Basic and Advanced	Some health personnel from the region have basic knowledge on the use of IT tools. However, others master the use of spreadsheet for form design and data analysis
	Central	Basic and Advanced	At the central level, we found some health personnel having intermediate and advanced level in the use of IT tools such as Word, Excel and the navigation on Internet
What do you expect from the use of a software for epidemiological surveillance?	Health center	Friendly UI	They proposed to develop the UI so that it's as close as possible to the tools they currently use for data management
	Regional	Automatic production of reports	The main concern at this level was the automatic production of reports
	Central	Automatic production of reports	As at the Regional level, the main concern was the automatic production of reports

Table 3: Summary of the survey results

At the end of this survey, the users confirmed that the electronic system will enhance their work as long as they are trained on the use of computers and the problem of power/Internet outage is considered.

5.2 Implementation of the approach

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.

5.2.1 Pre-Intervention

The aim of the Pre-Intervention was to produce a preliminary list of interventions for the NTCP to assess, enhance and approve. Then, the following activities were planned: Establishment of a working group, identification of the problem faced by the current system, stakeholders survey to gain more insights in the problem, reflection on the solution, software specification and analysis. Specification, analysis, product backlog definition and validation were implemented by following the Scrum iteration practice. Then, three iterations of one week each were conducted. The work was divided into tasks (for instance, writing the specifications of data collection, identification of actors) and the Scrum meetings make it easy to define common goals and commit to achieving the goal with the product owner. The face-to-face communication during these meetings allowed us to discuss, define and validate intervention outputs. At the end of each week, the Scrum Weekly Meeting was held at the NTCP to evaluate the comprehension of the needs by the researchers. The table 4 presents a summary of the Pre-Intervention.

Interventions	People involved	Scrum practice	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 4: Summary of the list of interventions planned and implemented during the Pre-Intervention

Establishment of the working group. The aim of this intervention was to establish the group of people involved in the development of the system. This was successfully implemented and a team composed of two persons acting as researchers and developers, a product owner, a staff at NTCP, an epidemiologist from Centre Pasteur du Cameroun and end users were put in place. A mailing list group was put in place as a means to facilitate the collaboration and a channel to disseminate the decisions.

Problem identification. We observed that epidemiological surveillance at NTCP is managed manually. According to the staff at NTCP, this manual management caused many problems such as low rates of promptitude and completeness, poor statistics, etc. Because of these problems, information may arrive to decision makers late.

Users survey. The aim of this intervention was to have the opinion of the end users at all the levels (health centers, district, region central levels). This was successfully implemented and the results are presented in section 5.1.

Specification. The aim of this intervention was to gather and document the most requirements of the epidemiological surveillance system of TB. A study of existing software used for epidemiological surveillance and the survey of the end users made previously allowed us to identify the main functionalities of epidemiological surveillance systems. These are: data collection, data analysis, and information dissemination to stakeholders.

This intervention was particularly difficult because in addition to the data that the NTCP demands to collect, many health centers collect additional data. Then, these features have to be integrated in the specifications. On the other hand, the data to be collected may vary every year. The presence of the product owner impacted positively on the validation of the specifications. Then, the first version of the specification went through an approval process with the working group team and was validated.

Scrum iterations were very useful during the specifications and analysis. In effect, in order to understand very well what was really needed, interviews were made, transcribed by the researchers and validated during the Scrum meetings with the working group. This was a cycle composed by the software specification (or update), analysis (or update), validation of specification and analysis (or update) and product backlog definition (or update). The positive effect that we observed was the openness that the Scrum meetings give to us in order to communicate with the working group.

The involvement of the NTCP staff was very useful to avoid poorly written requirements specifications. This allowed us to avoid severe problems presented by [65] which are: starting over with the product specification and restructuring it, difficulties to understand the requirements, and rewriting requirements.

The presence of the product owner, who was one of the users of the system at the NTCP level, has had positive effects on the Scrum practice. In effect, customer participation in the definition of product backlog prioritization of the backlog items, planning meetings and sprint review meetings make it easy to define common goals and commit to achieve this goal.

Analysis. This intervention consisted of determining the actors, use cases and to describe most important use cases. It was implemented successfully. The main actors identified were at the central level, all the staff of the NTCP; at the regional level, the regional delegate and the personnel in charge of TB surveillance; at the district level, the head district and the personnel in charge of TB surveillance; at the health center level, the nurses, the doctors, the laboratory technicians and the pharmacists in charge of TB management. Each user of the system has a particular role and collects data corresponding to their activities. After the identification of the actors, we grouped use cases by actors and at the end, we described the most important use cases.

Product backlog definition. Once the needs of the NTCP were well understood, the developer team transformed requirements into a product backlog. This product backlog was prioritized with the help of the product owner. Globally, the product backlog comprises the following backlog items:

- Platform design: this consists to design the software architecture, determine the MDA tool to use and design the user interface;
- Development of user management module: this module is used to grant data access to users according to their rights;
- Development of data management module (patient and medical data): data management module involves the forms used for data registration, data searching;
- Development of the reporting module: the reporting module allows users to have a synthesize version of their data (in form of statistics);
- Development of the mapping module: the mapping module allows health workers to have a map containing the different hospitals involved to the system;
- Development of the sensitization module: this module allows the NTCP to sensitize the population by SMS.

The Sprint Review Meetings permitted us to adopt the software specification, analysis and design with the whole staff at the NTCP. We designated the user management module and the data management module as our initial list of interventions which we presented to the NTCP for discussion and deliberation.

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.

This lesson led to a deep reflection by the software team on the choice of the development approach to use. Given the limits of the code-centric approach presented in the introduction, insight was then to model and generate the software instead of coding it. Then, the combination of Scrum that were already chosen with MDA were adopted. Thus, software development was

divided into Sprints and with the first Sprint composed of the following interventions: software design, development of user management module and development of data management module.

5.2.2 Intervention

Intervention phase were organized in 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 achieving 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 4 iterations of one week each. All the interventions involved in this Sprint were not completely implemented successfully. In effect, the user management feature generated by the MDA tool was not corresponding to how the NTCP managed their users. The table 5 presents a summary of the interventions implemented in this Sprint. The details about these interventions are given below.

Intervention	People involved	Scrum practices	MDA practice	Status	Comments
System design	Researchers and NTCP staff	Sprint, Iterations, Scrum meetings	Modeling PIM	Completed	During this intervention, beside the other tasks, the product researchers and the two health workers (including the product owner) were involved in the modeling of the PIM with Dia. Their role was to validate the different elements and relations between them.
Prototype implementation	Researchers and NTCP staff	Sprint, Iterations, Scrum meetings	Transforming the PIM to the PSM and the PSM to the source code	Not complete	This intervention involved the researchers and two health workers. The health workers participated in the transformation of the PIM to the PSM using Imogene. The production of this version was presented to the NTCP staff during Scrum meetings. However, the user management feature proposed by Imogene did not correspond to the features expected by the users. We observed that the two health workers involved have had great difficulties in modeling with UML.
Product backlog update	Researchers and Product owner	Scrum meetings	None	Complete	Given to the remarks on the users' management features, the product backlog is updated with new features to develop and integrate in the Imogene generator.

Table 5: List of interventions implemented during the first Sprint

System design. Given the system specification and analysis, the architecture, the user interfaces and the Platform-Independent model were made.

1. **Architecture design.** The architecture design intervention consisted of designing an architecture (given in appendix A) which suits an environment like Cameroon where access to the Internet is not always available.
2. **Choice of MDA tool:** to choose the MDA tool that is more suitable to our problem, an overview of existing tools used in epidemiological surveillance were made. Either these tools were develop using code-centric approach (DHIS[11], OpenRMS[12]), either they were

[11] https://dhis2.org
[12] https://openmrs.org

specialized in the generation of mobile-based data collection tools (ODK[13], Magpi[14]). Only Imogene[15] were found to cover most of our requirements. Then, a closed collaboration between the University of Yaounde I and MEDES allowed us to build good skills in the modeling and the generation of data collection applications with Imogene. A deep analysis of Imogene by the researchers shows that it permits to model and generate only two modules of the TB surveillance software: the data management module and the user management module.

3. **User Interfaces (UI) Design:** during this intervention, a deep discussion with the working group and the constraints of the Imogene platform on the design of UI allowed us to decide how the elements on the UI will be organized to resemble the manual management tools.
4. **Platform Independent Model design:** this intervention consisted of determining the entities, their properties, their relationships. Once these elements were determined, we set-up a UML class diagram using Dia. The working group were involved in this intervention. In fact, given that our main goal was to allow them to be in charge of the update of the system, they were invited to install Dia and use it. However, only 2/11 (18%) installed the tool. 3/11 (27%) estimated that the tools were out of their domain of expertise and 6/11 (55%) were not available. The final version of the class diagram validated by the staff is presented in appendix B.

Prototype implementation. Imogene was used to build the Platform Specific Model in only one week. This intervention involved the researchers and two NTCP personnel. The whole activities were executed by the researchers and followed by the health workers. In fact, the passage from the PIM to the PSM was manual following the rules given in the table 2 of the Section 4.5. Additional information such as field description, field name in French and English were added to the PSM using the Imogene editor. The screenshot of the epicam model obtained during this intervention is given in appendix C. This model was used to generate the first prototype of the application. This first prototype consisted of:

- **An administration application:** Used to manage health workers information, their roles, and their access rights on data;
- **A Web application:** Containing the main user interfaces needed. It is used for the registration of patients and their follow-up;
- **Offline application:** The offline application has the same functionalities as the Web application, but is used in an asynchronous mode. When Internet connection is not available, the system uses the local database as storage system and the local database is synchronized with the server when the Internet is available;
- **Synchronization application:** Used to synchronize the client with the server (updating local or remote database).

Testing and updating the Product Backlog. During this Sprint, six review meetings took place with the working group. The goal of these meetings was to evaluate the users and data management modules. During these meetings, the missing fields and some field names errors were noted. The missing fields information were used to update the imogene model in order to make the source code correspond to the requirements. Error fields were corrected. It was also noted that some users from the central level can access personal patient information. The NTCP staff warned us that only the health workers in charge of the patient must have access

[13] https://opendatakit.org
[14] https://www.magpi.com
[15] https://github.com/medes-imps/imogene

to personal information of patients. Users' remarks permitted us to integrate the specification, analysis and design of user rights to the data in the Product Backlog.

Next iterations. After the development of the data collection module, five iterations were conducted. These iterations permitted us to integrate all the information that was not present in the PIM and correct some typology errors. This involved the design, the development, and the testing. The design consisted of the updating of the PSM, the development consisted of the generation of a new version of the platform.

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. It should be noted that most health workers in the NTCP are nurses, physicians, lab technicians, epidemiologists. They have full time employment and their commitment to the project was during their spare time. This constrains the implementation of the interventions.
- Modeling can be a difficult task for domain experts: among the domain experts that were involved in the modeling, only 2/11 (18%) installed the modeling environment and 3/11 (27%) estimated that installing and using the modeling environment is out of their competencies.

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. This new intervention was put as a main priority. Given that medical data is very sensitive, it was decided to prioritize the development of this module. Then, the second Sprint was dedicated to the development of the generator of user management modules that fulfill epidemiological user management.

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 Product Backlog contains the following interventions: training on the development technologies, specifications, analysis, design, development, validation. The iterations processes involved the specification, analysis, development, validation. The table 6 presents the summary of the list of interventions implemented during this Sprint.

Only the developers (supervised by the MEDES engineers) and the product owner were involved in the development of this module. The main role of the product owner was to verify that the user management module allowed us to generate the user management system of TB surveillance at the NTCP.

It was noted during the specifications and analysis that users have different roles. In some hospitals where there is not enough staff, a user may be the person who consults the patient and gives its medication. With the user management module, when the administrator creates a user, she/he allows her/him to read/modify the data. When a user login, she/he can access only the data she/he is entitled to. The appendix E presents a screenshot of the UML model and the user interface allowing the creation of a role and the attribution of rights to users.

Intervention	People involved	Scrum practice	MDA practice	Status	Comments
Training	Developers and MEDES engineers	Scrum meetings	None	Completed	The developers were trained to the software to be used to develop new modules.
Specifications	Researcher and NTCP staff	Scrum meetings	None	Completed	The NTCP staff explained to the developers how users should be managed in their system.
Development	Developers and MEDES engineers and Product owner	Scrum meetings, Scrum iteration, Sprint	Constructing the User Model	Completed	This intervention consisted to constructed the user model, use it to complete the imogene model so that the user management feature generated will fit to the NTCP needs. During this intervention, the product owner verified during Scrum meetings that we are developing what they expected.
Validation	Researchers and NTCP staff	Scrum meetings	None	Completed	After the development, we present the interface that is generated for user management to the NTCP staff.
Prototyping	Researchers and NTCP staff	Scrum meetings	Modifying the PIM and the PSM	Completed	The PIM and the PSM is modify in collaboration with the two health workers involved in the development
Prototype validation	Researchers and NTCP staff and end users	Scrum meetings	None	Completed	Once validated by the NTCP staff, the product is presented to end users in the field for validation.

Table 6: List of interventions implemented during the second Sprint

We used the same tools used to develop Imogene to develop this module: the programming language was Java, the Integrated Development Environment was Eclipse Kepler for Java EE, the Google Web Toolkit to develop User Interface, the Spring 3 framework, the Spring security framework, code management with Maven and git, the modeling environment was Eclipse Modeling Framework and the code generation environment was composed of oAW (involved Xtext, Xpand, MWE components). Before the development of this module, the developers were trained on these technologies. After the development, this module was integrated in the source code generator of Imogene.

Once the generator of the user management module was developed and integrated into Imogene, the epicam model (appendix C) was updated manually and used to generate a new version of the prototype. It should be noted that this version of the prototype allowed us to manage users and TB data.

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 decided to validate the developed platform in the field. Then, we deployed it in six hospitals in Douala and Yaounde (which are the largest cities in Cameroon). Eight health personnel were involved in this validation which lasted one month. The goal were to evaluate the perceived usefulness and the perceived ease of use of the end users:

- **Perceived usefulness:** All the users found the tool useful for data management. They estimated that this tool will allow them to obtain data of good quality and improve their work. They also noted that the tool may not be suitable in certain hospitals. In effect, the scenario presented by the prototype considers only the registration of patients when they are tested positive to TB. However, in certain scenarios, the patient is registered in the system when they are suspected to have TB. We noted also that amongst the TB exams that the NTCP generally uses, some hospitals also use radiology exams. The system was not considering the registration of this exam. New requirements were noted as a new intervention and the product backlog was updated.

- **Perceived ease of use:** the users estimated that the use of the system was intuitive. All the eight personnel involved found that the forms resemble the paper forms they are using in the hospital.

Lesson learned globally, two lessons were learned at the end of this Sprint:

- We learned during this Sprint that beyond the Scrum team, the end users must be integrated in the validation of the software. After a deep reflection with the working group, it was decided to invite at least 50 users around the country for the validation before the deployment;
- We learned that the maturity of the MDA tool can slow down the development of the software system. Then, it can be a serious barrier to adoption. In effect, the development of the user management module has decelerated the execution of interventions and have had a negative impact on the timeline of the development.

Initially, 11 health personnel were involved in the project. However, after the first and the second Sprint, this number dropped to 5 (0.45%) with 3 (27%) completely engaged. Most participants justified their absence by their job. In addition, the product owner got another job and left the project. Unfortunately, he was among the personnel accustomed with the modeling. He was replaced by one person among the NTCP staff.

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. Then, a Scrum planning meeting was done, followed by the updating of the model and the generation of the new version of the system. The Scrum weekly meeting allowed us to validate the new version.

It should be noted that involving the NTCP staff in the development allowed 3 (27%) to grow more accustomed to the Scrum practices such as Product Backlog definition, organization of Scrum Sprint, and Scrum meetings. They found Scrum advantageous in the development of their system.

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[16]). The mapping module was developed using Open Street Map (OSM[17]). 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 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. 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[18] and hosted on github.

[16] https://www.eclipse.org/birt/

[17] https://www.openstreetmap.org/

[18] https://github.com/UMMISCO/EPICAM

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	Not complete	The installation difficulties discourage the health worker to continue the experience.

Table 7: List of interventions implemented during the sixth Sprint

We found the development of additional modules arduous than the development of code-centric application. In effect, it was questioned 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. During this development, we observed that Scrum practices, particularly face-to-face, Product Backlog, Scrum iterations and Scrum meetings were very helpful to organize the work and communicate among the developers, with the developers and the MEDES engineers and the developers and the stakeholders. It should be noted that all the fields used to automatically generate the mapping module were already presented in the PIM, which was transformed into the PSM.

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. As presented before, only three NTCP staff followed the development to the end, only two were involved in the modeling task and one of them moved to another job. The main goal of this Sprint was to allow the health personnel to update the system using Scrum and MDA. Then it involved the simulation of the following interventions: building the product backlog and organization into Sprints, validation of the product backlog by the working group, organizing and conducting Scrum meetings (daily and review), modeling the PIM with Dia, installing the MDA tool, manual transformation of the PIM into PSM, generation of the new version of the software, deployment on a local machine and validation by the NTCP. These interventions were planned to be executed by the health personnel, supervised by the researchers. The summary of their implementations is presented by the table 7.

The team conducted two-weeks of iterations, trial, errors and learning. During the first week, the modeling environment was installed and the product backlog was defined. This product backlog contained the following interventions: adding new forms with new fields for data collection, constructing the PIM, transforming the PIM into PSM. Since the EPICAM platform (used for the modeling and the generation) is based on the Eclipse platform, we observed that installing these tools by non-informatics experts is not an easy task. This did non encourage the latter to continue with the experience and he stopped after the second week.

The interview of this participant gave more insights on his motivation to participate in the development and update of the surveillance platform. Actually, he characterized himself as an advanced computer user that works with Microsoft Excel, and Epi Info [19]. He used Epi Info for modeling (with drag and drop) and generating data collection forms. He also had experience with the use of Microsoft Excel to create forms. We learned from him that the only barrier for

[19] www.cdc.gov/epiinfo

the adoption of the approach by non-informatic experts was the MDA tool because Scrum was very intuitive. He stated that if the UI of the modeling software reassembles to the drag and drop UI of Epi Info, the modeling tool and the approach can be more accepted by the health personnel. This view was strongly supported by two other participants. However, developing this new feature was beyond the mandate of the developers. The problem was the time that this drag and drop UI can take to develop and the technologies that the developers must master in order to develop it.

We approached the two other staff members at NTCP to have their opinion about the approach. They characterized themselves as basic users of computers. They used computers mainly to type text and to navigate on the Internet. They had no experience with modeling and Scrum before joining the project. They found that Scrum is a good tool for work organization and for communication with developers. Some of them affirmed that they have developed good skills in the Scrum process and are using it in their own work. We asked the question "Why are they reluctant to participate in the experience". The most common reasons were the lack of expertise. They stated that it took a long time to feel like they understood the graphical models we were presenting during the Scrum meetings. They found the tools too much oriented towards the engineer. They stated that the complexity is high, because of the installation, the interface of modeling and also, they found UML very complex.

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. In addition, the update by non-informatic experts declined due to the lack of knowledge on modeling and the difficulty to install the MDA tool.

This lesson has had some implications at the NTCP level. They decided to hire a person with strong skills in Scrum and MDA to accompany them in the update of their system.

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.

5.2.3 Post-intervention

The Post-Intervention consisted of training end users in the use of the software developed for epidemiological surveillance of TB and to disseminate the outcomes of the project.

To disseminate the outcome of the project, a demo session of the approach was presented to the stakeholders coming from the whole country (around 60) during a meeting organized by the NTCP. During this meeting, some participants said that the NTCP played a passive role in the software development before, this is a new experience for them and they have to rethink their participation model to software development.

Fifty health personnel involved in the epidemiological surveillance of TB at the hospital, regional and NTCP levels were invited for a two days experience at the university of Yaounde I. Our main goal was to assess the tool before the deployment in the whole country. Conscious of the level of the entire health workers on the use of IT tools (found during the deployment of the prototype), we decided to train them before on the use of IT tools. Then, the first day consisted of getting all the health workers familiar with the use of the computer. For the perceived usefulness, the users found the tool useful to improve their job. However, they noted some typology errors at the User Interface level - these errors were corrected and a new platform generated.

The users also reported that the system was easy to use. They particularly appreciate that all the data on the patient treatment is in one form (the patient form is given in appendix D).

The final version[20] were experimented in 25 pilot sites across the country. Around 3900 patients which represents 15.6% of the annual number of TB cases in Cameroon were registered and followed. 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.

5.3 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 approach. The analysis of the case data using the Technology Acceptance Model presented in section 4.4 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.

5.3.1 Perceived ease of Use

To measure perceived ease of use, we considered the developers evaluation and the stakeholders evaluation. We measure the degree to which these participants believe that using the combination of Scrum and MDA is free of effort.

As developers, globally, we can say that the application of Scrum and MDA for the development of epidemiological surveillance software is easy to use. In effect, once the features were validated and integrated in a Sprint Backlog, it generally took us a maximum of one week to update the model and generate a new version of the software.

The integration of Scrum in the development was very easy. In effect, once the different roles were introduced, the NTCP designed the product owner. The researchers explained his tasks and he conducted them very well without any difficulties.

After the production of the final version, we involved two health workers in the simulation of the update of the system using the combination of Scrum and MDA. Non-informatic experts constituted by the NTCP staff found Scrum simple, easy to learn, easy to handle and use. For them, it does not require a lot of mental effort. Thus, it was rapidly adopted. They found it easy to learn through experiences, the self organization when some members leave the team were very appreciated. Thus, during the project, they grew more accustomed to the Scrum practice. However, they did not get accustomed to the MDA. For them, the main reason was the lack of knowledge in modeling and the use of the MDA tool.

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

5.3.2 Perceived Usefulness

As we have done above, in the following paragraphs, we present the developers and non-informatic experts views of perceived usefulness.

Developers found the combination of Scrum and MDA very useful in our work of software development. In effect, combined with MDA, the Scrum Process permitted us to gradually build models and generate the executable source code which are updated with the gradual expansion of the epidemiological surveillance. This has a significant impact on qualities such as: programming productivity and satisfaction, development cost-effectiveness, timelines and customer satisfaction.

- **Programming productivity and satisfaction:** we found the approach useful because it allowed us to develop/update as fast as possible the software. The use of models allowed us to reduce complexity in the development and the Scrum practices allowed us to better organize the work between us and the communication with the customers.
  In effect, to develop the system, all the tasks to perform are prioritized and consigned in the product backlog. Depending on the size of the product backlog, one or many Sprints can be put in place. The Scrum meetings and the face-to-face communications were very useful to clear up any misunderstanding and facilitate the discussion and the decisions during the development. During the Sprints, the developers tasks involved the modeling and generation of the software. The use of Scrum iterations allowed us to repeat during any Sprint until all the tasks planned are completed. For instance, once the first model was built and the software generated, during the users evaluation, modifications to apply were identified and the cycle resumed. The repetition of iterations helped us to learn from the errors and reinforce what went well, achieving continuous improvement. The use of Scrum increment and scrum meetings with the NTCP in the requirements specifications and analysis have had a great positive effect on the understanding of the system by the developers.
  To update the system: Once the product backlog is defined, the system is updated in one Sprint of maximum one week. We found the one week iteration very useful because it helps us to review the modification of the model and generate the new working version as fast as possible.
  In conclusion, the combination of Scrum and MDA have had a positive impact on programming productivity and satisfaction. We found a positive effect on:
  - long-term productivity improvement because of automating repetitive tasks especially adding/modifying new forms, report, fields, etc.;
  - satisfaction because the organization of the tasks and communication using Scrum practices allowed us to agree with the NTCP with the continuous improvement of the solution. Whatever the new requirements were added, the Scrum team refined the product backlog and the new software was generated. Consequently, all the feature sets were completed during the project.

  We found that to have a great positive effect on programmer productivity and satisfaction, the MDA tool must be mature enough to generate the code covering all the features demanded. In our case, the immaturity of the MDA tool has also had a negative effect in the development because the development of additional modules was hard (we have to be trained in new technologies) and took time.
- **Development cost-effectiveness:** we also found that the approach has paid off by reducing the complexity of software development. In effect, the reusability of the generation tool and model allowed us to reduce the costs of the development and update. The use

of Scrum practices such as Product Backlog and Scrum meetings allowed us to facilitate the development of the solution. The modification of the model and generation of the new software have made it more valuable. In addition, the approach reduced costs by saving the time that may be used to debug a system because of the problem of software regression.

It should be noted that the EPICAM platform can be reused for the development of several epidemiological surveillance software.

We concluded that the combination of Scrum and MDA has a positive effect on development cost effectiveness.

On the other hand, the development of additional modules for the generation tool was time consuming and has had a negative effect on the cost-effectiveness.

- **Timelines:** the Scrum planning meetings, Scrum daily meetings facilitated the organization and prioritization of tasks. The Product Backlog allowed us to define the different tasks and prioritize them so as to deliver the first working version as soon as possible. The one week duration of iterations were useful to ensure that the development is advancing in a good way. The development of this first working version was greatly facilitated by the use of models. In effect, once the requirement and the analysis were defined, the transformation into product backlog with prioritized and estimated tasks allowed us to have insights on the main features of the software and develop the first working version of the platform as soon as possible.

  In conclusion, the approach presented in this research has a positive impact in the timelines of the development.
- **Customer satisfaction:** Scrum practices and MDA approach have had positive effects in customer satisfaction. In effect, the involvement of the stakeholders in the development allowed the improvement of short time visibility of common goals. For instance, the scrum meetings and the use of models to present the evolution of the work during these meetings have brought the customers closer to the development and revealed core requirements. Exchanges between the developers and the NTCP staff have had a positive effect in the development. The opinions of the stakeholders revealed that this was a positive way to keep people aware of the project. The Sprint planning meetings were very useful for the Scrum team. It allows the product owner to explain to the team the most valued requirements and the prioritized backlog items to be delivered to the customer at the end of the Sprint. Through these meetings, any misunderstanding between the developers and the customers were solved. Different Scrum Meetings with the NTCP staff and the end users allowed us to build a product which was as close to the users as possible. At the end, we observed that the software was accepted by the end user. Some of them found the user interface close to the tools they were using in their work.

  For the development phase, during the Scrum meetings, when the customer or the end users proposed to add or modify something, these tasks were added in the product backlog and the modifications were planned. These modifications and the testing took one Sprint of less than one week because in most time, it was question to modify the models and regenerate the updated solution. The stakeholders were very satisfied because during the Scrum weekly meetings, they found their remarks integrated in the solution. Sprint reviews were very positive, because we could also get a view of what other types of requirements the stakeholders had.

  As prescribed by Scrum, the goal of producing a working version of the system for iteration reviews was a good way of noticing and communicating integration problems that would otherwise be communicated only after the overall implementation. This practice

has a positive effect because the end users and the customer no longer wait until the end of the project before they have the results.
The fact that the customer commenced to use the software was useful to the development. The sprint review meeting shows the customer the new features and changes implemented in the Sprint. The Scrum team inspected the work done by the end users. Then, we found the approach very useful to reduce time-to-market. The end users started using the tool and got accustomed to it as soon as possible. This helps to realize if changes are necessary. In fact, sooner the changes are detected, the less its impact and cost.
After the success of the deployment in 25 pilot health centers, the NTCP extended the deployment in twenty new health centers.
In conclusion, the use of Scrum and MDA allows us to deliver a regularly working version of the software. This provided positive impacts in customer satisfaction.

Even if the approach has been found useful, we also found that the immaturity of the MDA tool can have negative effects in the development. It can slow down the programmer productivity and satisfaction, augment the development cost because new generators must be developed, and be costly in time.

Non-informatic experts also found the combination of Scrum and MDA useful. They were impressed by the rapidity with which new versions of the software were proposed every time where the changes were necessary. However, they found that they don't have good knowledge in the use of models and modeling tools to use the approach to update the system. The interview of the user and a deep observation on the use gave more insights on the non adoption of the method. The main bottleneck of the adoption was the installation of the tool and the UI. Developing tools based on Eclipse and Eclipse plugins such as EMF, etc. have had a negative effect on the acceptance of the approach by the stakeholders.

5.4 Conclusion

The lessons learned in this research and the main finding allow us to provide a scientific answer to the research question, independently to TB surveillance. This is required because the pilot project presented in Section II aims to study the epidemiological surveillance of TB in order to propose solutions that can be generalized to any other cases. We found that to mesh Scrum and MDA for the development and maintenance of epidemiological surveillance software, the following points should be considered:

- MDA framework (such as EPICAM for epidemiological surveillance) should be mature enough to generate the entire source code. In effect, the development of additional modules was not an easy task on the one hand because of the reverse engineering of the Imogene platform and on the other hand, because it is not easy to build a tool that should generate source code for all the surveillance systems. The additional development has had a negative impact on programmer productivity.
- Scrum practices such as involvement of stakeholders in the project, Product Backlog definition, Sprint definition, Scrum iteration, Scrum meetings must be used to build models, make these models validated by the domain experts and make the application developed validated by end users.
- Use the Scrum process involving the pre-development, the development, and the post-development:

1. Use the pre-development step to put in place the Scrum team, well document the problems so as to lift any ambiguity and misunderstanding between the developers and the users. In addition, write the application specifications, analysis and produce the first version of the product backlog. During the pre-development step, the Scrum iteration practice can be used and different iterations will be used to lift any misunderstanding.
2. During the development step, organize the development into Sprints. The first sprint consists of the design of the system by firstly justifying the adoption of the MDA approach and secondly identifying the most relevant tool to be used. If there is no adapted tool, the team should evaluate the cost of developing a new tool or updating an existing tool compared to the cost of developing a new system using the code-centric approach. It is during this sprint that the MDA approach is adopted. If there is not an adaptable tool, thus, the team should consider to develop the new ones during the sprints 2 to sprint n. Once the tool is adopted, the team should develop a first version of the system during the sprint n+1 by building the PIM, transforming into PSM, generating and deploying the software. In our case, it was not too difficult to pass from the PIM to the PSM manually. But, the automatic transformation can be a better solution to augment the programmer productivity. The first version of the system should be validated by the product owner who is part of the team and deployed to some end users. From the sprints n to the sprint m, the models are updated using the users feedback and the most complete solution is developed.
3. The post-development consists of training and making the software available to all the intended users.

VI THREATS TO VALIDITY

We found in this research 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. In this section, we evaluate under which conditions our research is applicable and offers benefits and under which circumstances it might fail. Section 6.1 presents threats to internal validity and section 6.2 presents threats to external validity.

6.1 Threats to internal validity

Threats to internal validity refers to whether the study supports the finding [17]. A number of threats to internal validity may surface at the Pre-Intervention Phase and the Intervention Phase.

At the Pre-Intervention phase, the main threat came from the survey of health personnel during the Pre-Intervention Phase. In fact, we choose the qualitative technique (data collection with interview) in order to lead to richer information on the subject of the study. The small number of subjects in a study does not allow any claim that the finding is representative of the organization in general [17, 66]. In effect, when the number of subjects is small, this may hamper the identification of all the problems of the system and give rise to the nonacceptance of the solution. This was remarked during the validation with health personnel where other patient management scenarios and additional features were proposed to be integrated.

During the Intervention Phase, the data were collected using direct observation by watching how the combination of Agile and MDA is being used. Collecting data with observation may lack clarity [17, 66]. On the other hand, we were the developers and the researchers who

developed and evaluated during the Intervention phase. It would have been better to have an external researcher who observes and evaluates everything.

In spite of the threats presented in this section, we have reasons to assume that they had only limited impact on the research. Firstly, the threats brought by the survey were corrected during the evaluation of the first prototype by end users. Secondly, interviews were used to solve the problem of lack of clarity of the observation. For instance, the health personnel involved in the updating of the platform using our approach were interviewed to know why they were reluctant. This allows us to discover the lack of competencies in modeling and in the use of modeling tools.

6.2 Threats to external validity

Threats to external validity refers to whether the finding can be generalized [17]. In this research, the generalization can be seen at two levels: at the level of epidemiological surveillance systems and at the level of the industrial systems.

The first threat to external validity came from 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 [17, 66]. However, 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. This may hamper the generalizability at the level of epidemiological surveillance systems and industrial systems. It should be noted that 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.

The second threat came from the case study and action research. In fact, the generalizability of results is limited to the domain of health where the population shares similar characteristics with the sample involved. In Empirical Research in Software Engineering, to generalize an approach, one should replicate it to many other settings [17, 66]. Then, to generalize the approach we propose in Section 5.4, to the other industrial cases, it needs to be replicated with different languages, tools, applications and application size.

The fourth threat came from the developers. Only two developers were involved in the study. This can hamper the generalizability of the study to other developers.

VII RELATED WORK

The problem of failed software does not only concern the domain of epidemiological surveillance. In fact, software developers of today face the problem of increasingly complex software as clients demand richer functionalities in shorter timescales. This has given rise to the software development question and dilemma: how to provide the customers with the software and software artifacts that fulfill their expectations in time and within the budget [41]? To respond to this question, related work proposes approaches and tools amongst which Model Driven Techniques and Agile Processes (refer to Section III. To write this related work section on the combination of Agile and MDE, we used Scopus, Semantic Scholar and Google Search engine to search for papers related to the development of: (1) any software, (2) health software, and (3) epidemiological systems. As demonstrated by Hessa and Kevin [52] in a review article published in 2022, a lot of work was proposed on the combination of Agile methodologies and MDA for the development of software systems in general. However, we found only two works [50, 57] proposing the combination of Agile and MDA for the development of health software

in general and zero work for the development of epidemiological surveillance systems. In the following paragraphs, we present the two works and a table comparing these works with our work.

Kannan et al. [50] proposed the combination of Agile with Model Driven Development for the development of electronic health records (EHR) in the United States. On the one hand, the authors highlighted the problems when developing EHR which are: misunderstanding requirements, frequent product change requests, product defects, the developments are costly and time consuming. On the other hand, the authors have the challenge to develop EHR tools across 40 speciality academic medical center practices over one year. Thus, they thought it necessary to use a rapid yet high quality development approach such as the combination of Agile and MDD. Agile artifacts that were used involve User Stories, User Interface Storyboards, and Use Case Text. Concerning the MDD tools, the authors used the UML language to build diagrams to facilitate the communication amongst different stakeholders involved in the project. They used Class Diagram, Object Diagram, Use Case Diagrams, Decision Threes, Swimlane Workflow diagrams, and State Diagrams. They did not adopt any tool that is used to model and automatically generate the software source code. As results during the year of 2015, 58 EHR were developed, 111 EHR-based data collection tools built and 16.000 patients on registries. The authors found that the combination of Agile and MDD allows rapidly gaining consensus, acceptance, clarification on requirements and removing ambiguity. They recognized useful design patterns for re-use on subsequent projects.

Blagoj et al. [57] presents the national e-health system implemented in Serbia as "MojDoktor" and in Macedonia as "MojTermin". These systems are based on the same integrated health information platform. For its fast development and adoption, its architecture was designed with Agile methodologies. However, this system suffers problems such as: its extensibility, soundness, interoperability and standardization. To solve these issues, the authors proposed a formalization of the system design and its implementation as a Model Driven Architecture. The iterative and gradual development agile practice was adopted for the development. The models proposed involve a set of formal models on several abstraction levels, a model for data transformation that provides interfaces for interoperability between systems. The system were modeled by adopting the CIM, PIM and PSM:

- Computational Independent Model: consists of the workflow diagrams used to describe the business logic, and the document description.
- Platform Independent Model: on the one hand, the authors used several tools to model the system: class diagrams, abstract state machine to capture behavior formalism. On the other hand, they proposed a framework for describing the software architecture that is based on creating models, meta-models and constraints between meta-models. The models layer contains models that describe specific elements of the system (e.g., services, entities, process). The meta-model layer defines generic models for specific elements (e.g., services, entities, process).
- Platform Specific Model: this layer involves the source code that implements the different elements modeled by the PIM. For each model of the PIM, a platform specific implementation exists. These implementations must conform to the models, and the models must conform to the meta-models.

Using this organization, the authors demonstrated how the model built can be used to model and generate RESTful APIs. The authors used Visual Paradigm for UML for code generation. The code generated was REST API and API documentation. The main benefits highlighted by

System	Domain	Problem	Agile practice	MDE tools	Code gen.
Kannan et al. [50]	EHR	Misunderstanding requirements, Frequent product change requests, Product defects, Costly and time consuming development	User Stories, User Interface Storyboards, Use Case Text	Class Diagram, Object Diagram, Use Case Diagrams, Decision Threes, Swimlane Workflow diagrams, State Diagrams	No
Blagoj et al. [57]	e-health	Extensibility, Soundness, Interoperability, Standardization	Iterative and gradual development	CIM, PIM, PSM - Workflow diagrams, Class diagram, Abstract state machine diagram, Visual Paradigm	Yes (partial code generation)
Our work	Epi. Surv.	Requirements cannot be fixed and change generally	Sprint, Scrum boards, Scrum meetings, Scrum Team, Product Backlog, Iterative and Incremental	CIM, PIM, PSM - Class Diagram, Eclipse, Imogene	Yes

Table 8: Comparison of existing systems to our system

this new system are the abstraction of the complexity of the system through the MDA model, the standardization of the system to make it easy to exchange information with other systems, and the reduction of the development costs regarding the future improvements of the system.

We summarized this related work section in table 8. It presents the key comparison points of the two related work papers to our system. Given to this table, our work is the only work that proposes a tool for the automatic generation of a whole system. On the other hand, the work we proposed is the only work exploiting the whole potential of the agile methodologies and presenting step by step how a health information system can be modeled and generated using an Agile process and a MDA framework.

VIII CONCLUSION

The main objective of this research was to answer the research question: "how to combine Scrum and MDA to develop and maintain epidemiological surveillance software?" Thus, we put together three empirical methods in software engineering to investigate the combination of Scrum and MDA for the development of epidemiological surveillance software and provide an answer to this research question (see Section 5.4). The survey was used to collect information on the system in place and how the solution should look like. Using Case Study research, we portrayed the development the software for epidemiological surveillance of tuberculosis so that it can be reused for other diseases. Action research allowed us to resolve the problem of failed software developed for epidemiological surveillance by involving health workers in the process of finding the solution.

Globally, we found that when the MDA tool is mature enough to generate the entire source code, the combination of Scrum and MDA allows effective development of epidemiological surveillance software. In effect, combining Scrum and MDA with an appropriate MDA tool has a positive impact on programmer productivity and satisfaction, development cost-effectiveness, timelines, and customer satisfaction. Scrum practices and models developed showed to be efficient practices to validate the evolution of the development and, therefore, facilitates more productive software development because customers are integrated and misunderstanding problems are solved as soon as possible. This was particularly observed during the validation of the Platform Independent Model by the National Tuberculosis Control Program staff and the validation of the software by end users. Thus, to combine Scrum and MDA to develop epidemiological surveillance software, Scrum practices (Involvement of stakeholders, Product Backlog definition, Sprint definition, Scrum iteration, Scrum meetings) must be used to build models and generate related software. The primary novelty of this work is the ability to use Scrum Process

in the modeling and the generation of software. To our knowledge, the combination of Scrum and MDA for the complete development of health software has not been attempted previously.

The direct perspectives to this work are: (1) the application of the approach presented in Section 5.4 into many other settings in order to come up with a generic approach that can be applied in any epidemiological surveillance and other industrial settings. To this end, we are currently working with nutritionists on a system for nutrition (overnutrition and undernutrition) surveillance; (2) generally, many people whose domain experts and computer scientists intervene in the development of epidemiological surveillance systems. We are currently developing a Web version that can allow the collaborative modeling of the system (the same idea as Google Docs). On the other hand, we are working on the automatic generation of the PSM (imogene model) from the PIM (class diagram) and vice versa. To this end, we are exploring solutions such are srcML[21] and papyrus[22].

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] B. Kitchenham, L. Pickard, and S. Pfleeger. "Case studies for method and tool evaluation". In: *IEEE Software* 12.4 (1995), pages 52–62.

[3] D. Damian, A. Eberlein, M. Shaw, and B. Gaines. "Using different communication media in requirements negotiation". In: *IEEE Software* 17.3 (2000), pages 28–36.

[4] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta. "Agile Software Development Methods: Review and Analysis". In: *CoRR* abs/1709.08439 (2002).

[5] L. Bratthall and M. Jørgensen. "Can You Trust a Single Data Source Exploratory Software Engineering Case Study?" In: *Empirical Softw. Engg.* 7.1 (Mar. 2002), pages 9–26. ISSN: 1382-3256.

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

[7] C. Larman and V. R. Basili. "Iterative and Incremental Development: A Brief History". In: *Computer* 36.6 (June 2003), pages 47–56. ISSN: 0018-9162.

[8] F. David and P. John. *The Mda Journal: Model Driven Architecture Straight From The Masters*. Meghan Kiffer, 2004. ISBN: 0929652258.

[9] K. Schwaber. *Agile Project Management With Scrum*. Redmond, WA, USA: Microsoft Press, 2004. ISBN: 073561993X.

[10] X. Blanc. *MDA en action, Ingénierie logicielle guidée par les mod'eles*. EYROLLES, 2005.

[11] V. Jones, A. Rensink, and E. Brinksma. "Modelling mobile health systems: an application of augmented MDA for the extended healthcare enterprise." In: *Ninth IEEE International Enterprise Distributed Object Computing Conference (EDOC 2005), 19-23 September 2005, Enschede, The Netherlands*. 2005, pages 58–69.

[12] S. Kemmis and R. Mctaggart. "Participatory Action Research: Communicative Action and the Public Sphere." In: *Handbook of qualitative research* (Jan. 2005).

[21] https://www.srcml.org/

[22] https://www.eclipse.org/papyrus/

[13] Mapatano and Piripiri. “Quelques erreurs courantes d’analyse d’un syst‘eme d’information sanitaire(RD Congo)”. In: *Santé Publique 2005*. Volume 17. 2005.

[14] K. S. Rubin, T. Beale, and B. Blobel. “Modeling for Health Care”. In: *Person-Centered Health Records: Toward HealthePeople™*. Edited by G. A. Demetriades James E.and Christopherson and R. M. Kolodner. New York, NY: Springer New York, 2005, pages 125–146.

[15] F. Truyen. “The Fast Guide to Model Driven Architecture The Basics of Model Driven Architecture”. In: (Jan. 2006).

[16] B. Blobel and P. Pharow. “A model driven approach for the German health telematics architectural framework and security infrastructure”. In: *International journal of medical informatics* 76.2-3 (2007), pages 169–175. ISSN: 1386-5056.

[17] 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.

[18] J. Mathenge Kanyaru, M. Coles, S. Jeary, and K. Phalp. “Using visualisation to elicit domain information as part of the Model Driven Architecture Approach”. In: *CEUR Workshop Proceedings* 376 (Jan. 2008).

[19] M. Pikkarainen, J. Haikara, O. Salo, P. Abrahamsson, and J. Still. “The impact of agile practices on communication in software development”. In: *Empirical Software Engineering* 13 (June 2008), pages 303–337.

[20] W. Raghupathi and A. Umar. “Exploring a model-driven architecture (MDA) approach to health care information systems development”. In: *International Journal of Medical Informatics* 77.5 (2008), pages 305–314. ISSN: 1386-5056.

[21] P. A. Rottier and V. Rodrigues. “Agile development in a medical device company”. In: *Proceedings - Agile 2008 Conference*. 2008. ISBN: 9780769533216.

[22] R. K. Yin. *Case Study Research: Design and Methods (Applied Social Research Methods)*. Sage Publications, 2008. ISBN: 1412960991.

[23] D. Gasevic, D. Djuric, and V. Devedzic. *Model Driven Engineering and Ontology Development*. 2nd. Springer Publishing Company, Incorporated, 2009.

[24] W. Raghupathi and A. Umar. “Integrated Digital Health Systems Design”. In: *International Journal of Information Technologies and Systems Approach* 2 (June 2009), pages 15–33.

[25] P. Runeson and M. Höst. “Guidelines for conducting and reporting case study research in software engineering”. In: *Empirical Software Engineering* (2009). ISSN: 13823256.

[26] R. Matinnejad. “Agile model driven development: An intelligent compromise”. In: *Proceedings - 2011 9th International Conference on Software Engineering Research, Management and Applications, SERA 2011*. 2011. ISBN: 9780769544908.

[27] AAMI. *Guidance on the use of agile practices in the development of medical device software*. AAMI, 2012. ISBN: 1570204454.

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

[29] T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe. “A Decade of Agile Methodologies”. In: *J. Syst. Softw.* 85.6 (June 2012), pages 1213–1221. ISSN: 0164-1212.

[30] M. McHugh, F. McCaffery, and V. Casey. “Barriers to Adopting Agile Practices When Developing Medical Device Software”. In: *Software Process Improvement and Capability Determination*. Edited by A. Mas, A. Mesquida, T. Rout, R. V. O’Connor, and A. Dorling. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pages 141–147.

[31] M. Mc Hugh, O. Cawley, F. McCaffcry, I. Richardson, and X. Wang. "An agile V-model for medical device software development to overcome the challenges with plan-driven software development lifecycles". In: *2013 5th International Workshop on Software Engineering in Health Care, SEHC 2013 - Proceedings*. 2013. ISBN: 9781467362825.

[32] P. Mohagheghi, W. Gilani, A. Stefanescu, and M. A. Fernandez. "An empirical study of the state of the practice and acceptance of model-driven engineering in four industrial cases". In: *Empirical Software Engineering* (2013). ISSN: 13823256.

[33] P. Mohagheghi, W. Gilani, A. Stefanescu, and M. A. Fernandez. "An empirical study of the state of the practice and acceptance of model-driven engineering in four industrial cases". In: *Empirical Software Engineering* (2013). ISSN: 13823256.

[34] T. Özcan, S. Kocak, and P. Brune. "Agile Software Development with Open Source Software in a Hospital Environment – Case Study of an eCRF-System for Orthopaedical Studies". In: *Web Engineering*. Edited by F. Daniel, P. Dolog, and Q. Li. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pages 439–451.

[35] E. Stringer. *Action Research*. SAGE Publications, 2013. ISBN: 9781483320731.

[36] V. Curcin, T. Woodcock, A. J. Poots, A. Majeed, and D. Bell. "Model-driven approach to data collection and reporting for quality improvement". In: *Journal of Biomedical Informatics* 52 (2014). Special Section: Methods in Clinical Research Informatics, pages 151–162. ISSN: 1532-0464.

[37] U. Eliasson, R. Heldal, J. Lantz, and C. Berger. "Agile model-driven engineering in mechatronic systems - an industrial case study". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2014). ISSN: 16113349.

[38] H. C. Estler, M. Nordio, C. A. Furia, B. Meyer, and J. Schneider. "Agile vs. structured distributed software development: A case study". In: *Empirical Software Engineering* (2014). ISSN: 15737616.

[39] 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.

[40] G. Mussbacher, D. Amyot, R. Breu, J.-M. Bruel, B. H. C. Cheng, P. Collet, B. Combemale, R. B. France, R. Heldal, J. Hill, J. Kienzle, M. Schöttle, F. Steimann, D. Stikkolorum, and J. Whittle. "The Relevance of Model-Driven Engineering Thirty Years from Now". In: *Model-Driven Engineering Languages and Systems*. Edited by J. Dingel, W. Schulte, I. Ramos, S. Abrahão, and E. Insfran. Cham: Springer International Publishing, 2014, pages 183–200.

[41] B. Perisic. "Model Driven Software Development – State of The Art and Perspectives". In: *INFOTEH-JAHORINA*. Volume 13. Mar. 2014, pages 1237–1248.

[42] D. Salah, R. F. Paige, and P. Cairns. "A Systematic Literature Review for Agile Development Processes and User Centred Design Integration". In: *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. EASE '14. London, England, United Kingdom: ACM, 2014, 5:1–5:10. ISBN: 978-1-4503-2476-2.

[43] F. P. Basso, R. M. Pillat, F. Roos-Frantz, and R. Z. Frantz. "Combining MDE and scrum on the rapid prototyping of web information systems". In: *International Journal of Web Engineering and Technology* (2015). ISSN: 17419212.

[44] 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.

[45] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband. "A Systematic Literature Review on Agile Requirements Engineering Practices and Challenges". In: *Comput. Hum. Behav.* 51.PB (Oct. 2015), pages 915–929. ISSN: 0747-5632.

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

[47] 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.

[48] A. R. da Silva. "Model-driven engineering: A survey supported by the unified conceptual model". In: *Computer Languages, Systems Structures* 43 (2015), pages 139–155.

[49] D. A. Henderson. "The Development of Surveillance Systems". In: *American Journal of Epidemiology* 183.5 (Feb. 2016), pages 381–386.

[50] V. Kannan, J. C. Fish, and D. L. Willett. "Agile model driven development of electronic health record-based specialty population registries". In: *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*. IEEE. 2016, pages 465–468.

[51] H. Alfraihi and K. Lano. "The integration of agile development and model driven development: A systematic literature review". In: *MODELSWARD 2017 - Proceedings of the 5th International Conference on Model-Driven Engineering and Software Development*. 2017. ISBN: 9789897582103.

[52] H. A. A. Alfraihi and K. C. Lano. "The integration of agile development and model driven development: A systematic literature review". In: *The 5th International Confrence on Model-Driven Engineeing and Software Development* (2017).

[53] B. Atansovski, M. Bogdanović, G. Velinov, L. Stoimenov, D. Sahpaski, I. Skrceska, M. Kon-Popovska, D. Janković, and B. Jakimovski. "Transforming an Enterprise E-Health System from Process Oriented to Model Driven Architecture". In: *7th International Conference on Information Society and Technology ICIST*. May 2017, pages 159–162.

[54] H. Bagherian, M. Farahbakhsh, R. Rabiei, H. Moghaddasi, and F. Asadi. "National Communicable Disease Surveillance System: A review on Information and Organizational Structures in Developed Countries". In: *Acta informatica medica* (Dec. 2017).

[55] H. Bagherian, M. Farahbaksh, R. Rabiei, H. Moghaddasi, and F. Asadi. "National Communicable Disease Surveillance System: A review on Information and Organizational Structures in Developed Countries". In: *Acta Informatica Medica* 25 (Dec. 2017), pages 271–276.

[56] 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.

[57] 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.

[58] I. Essebaa and S. Chantit. "Model driven architecture and agile methodologies: Reflexion and discussion of their combination". In: *Proceedings of the 2018 Federated Conference on Computer Science and Information Systems, FedCSIS 2018*. 2018. ISBN: 9788394941970.

[59] F. Ricca, M. Torchiano, M. Leotta, A. Tiso, G. Guerrini, and G. Reggio. "On the impact of state-based model-driven development on maintainability: a family of experiments using UniMod". In: *Empirical Software Engineering* (2018). ISSN: 15737616.

[60] F. J. Azanzi, G. Camara, and M. Tchuente. "Extracting ontological knowledge from Java source code using Hidden Markov Models". In: *Open Computer Science* 9.1 (2019), pages 181–199.

[61] 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.

[62] 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.

[63] A. Alami, P. Nielsen, and A. Wasowski. "A Tailored Participatory Action Research for FOSS Communities". English. In: *Empirical Software Engineering* 25.5 (Sept. 2020), pages 3639–3670. ISSN: 1382-3256.

[64] A. Bucchiarone, J. Cabot, R. F. Paige, and A. Pierantonio. "Grand challenges in model-driven engineering: an analysis of the state of the research". In: *Software and Systems Modeling* (2020). ISSN: 16191374.

[65] 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.

[66] P. Ralph, N. b. Ali, S. Baltes, D. Bianculli, J. Diaz, Y. Dittrich, N. Ernst, M. Felderer, R. Feldt, A. Filieri, B. B. N. de França, C. A. Furia, G. Gay, N. Gold, D. Graziotin, P. He, R. Hoda, N. Juristo, B. Kitchenham, V. Lenarduzzi, J. Martínez, J. Melegati, D. Mendez, T. Menzies, J. Molleri, D. Pfahl, R. Robbes, D. Russo, N. Saarimäki, F. Sarro, D. Taibi, J. Siegmund, D. Spinellis, M. Staron, K. Stol, M.-A. Storey, D. Taibi, D. Tamburri, M. Torchiano, C. Treude, B. Turhan, X. Wang, and S. Vegas. *Empirical Standards for Software Engineering Research*. 2020.

[67] A. S. I. G. on Software Engineering. *Empirical Standards*. 2020.

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

[69] A. Jiomekong. *EPICAM supports*. July 2022.

[70] A. Jiomekong, H. Tapamo, and G. Camara. "Combining Scrum and Model Driven Architecture for the development of the EPICAM platform". In: *CARI 2022*. Yaounde, Cameroon, Oct. 2022.

A SYSTEM ARCHITECTURE

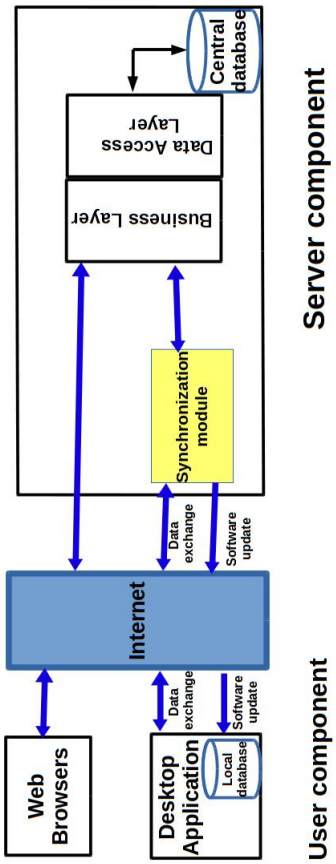

Figure 3: Epidemiological surveillance platform architecture

B PLATFORM INDEPENDENT MODEL

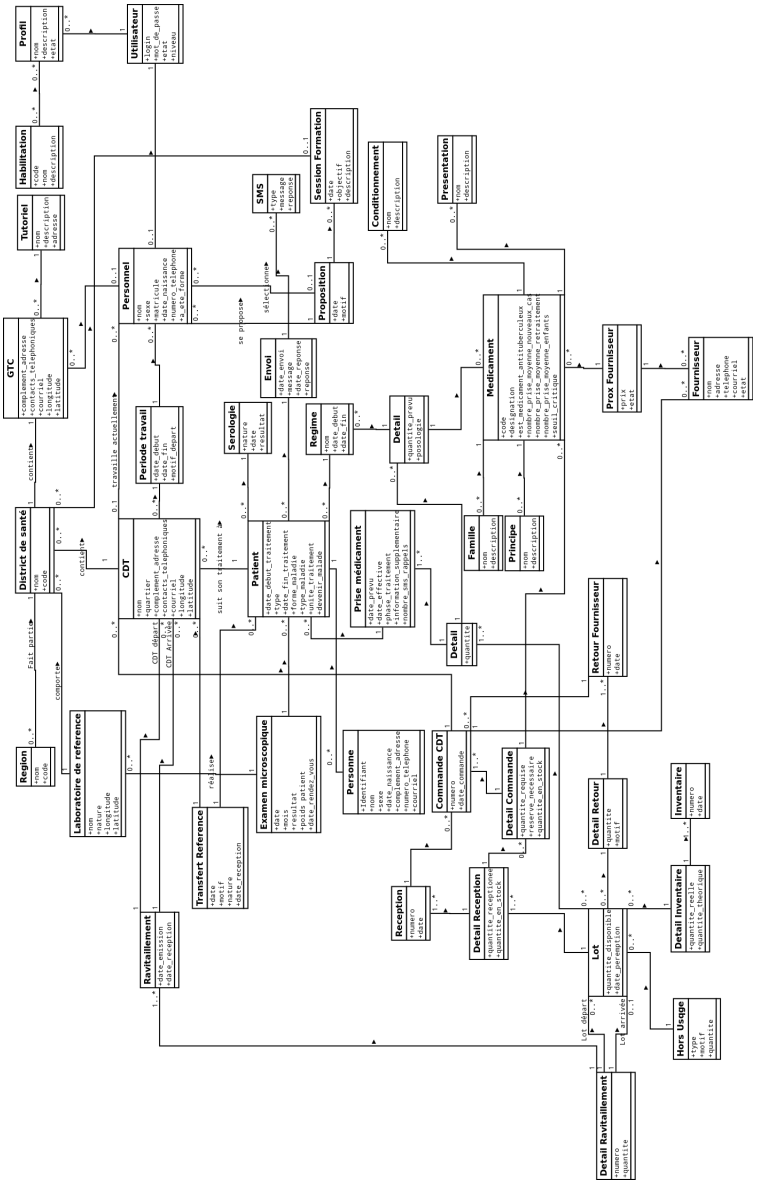

Figure 4: The Platform Independent Model of EPICAM

C PLATFORM SPECIFIC MODEL

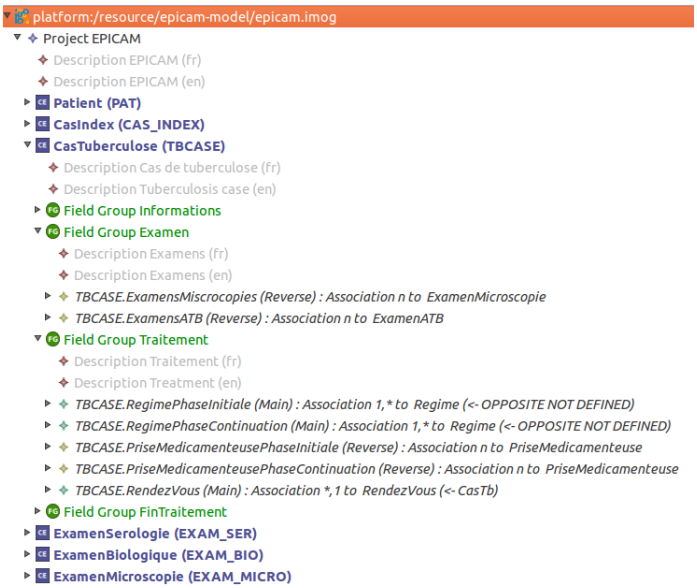

Figure 5: The Platform Specific Model of EPICAM constructed using the PIM

D SOME SCREENSHOTS OF THE APPLICATION GENERATED

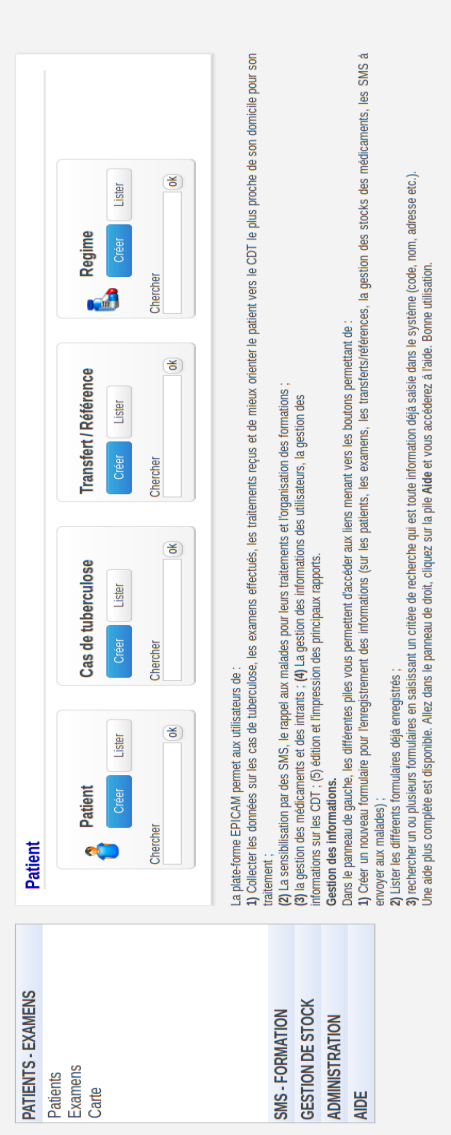

Figure 6: The entry point of the EPICAM platform

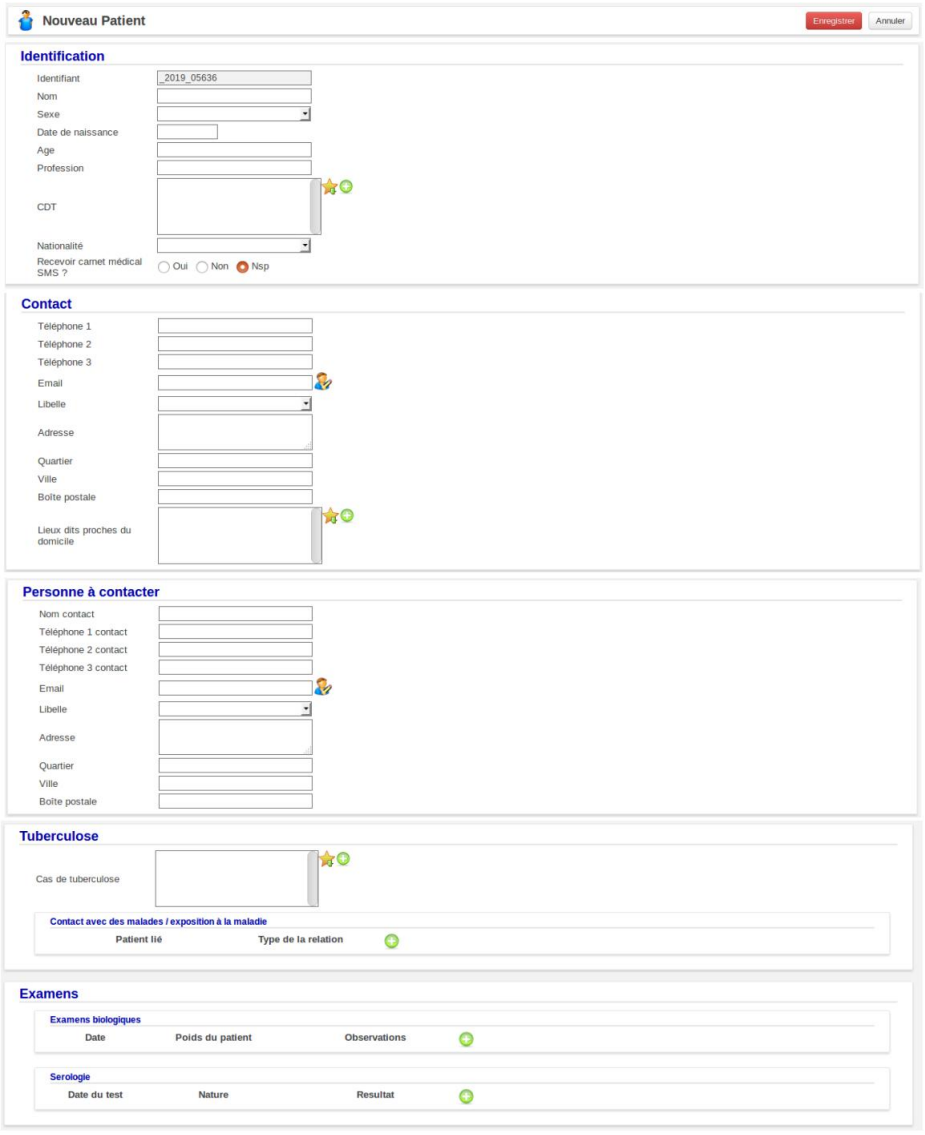

Figure 7: The patient registration form

E SCREENSHOTS FOR USER MANAGEMENT INTERFACE

The following figures presents the screenshots of the UML model of user management and the user interface generated thereafter by the Imogene platform.

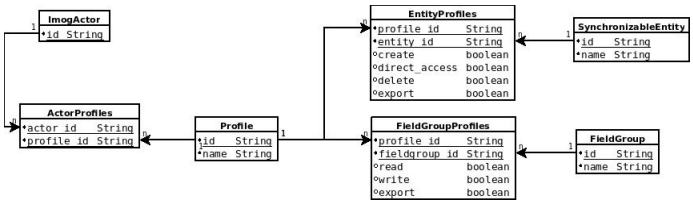

Figure 8: The class diagram of users management in the EPICAM platform

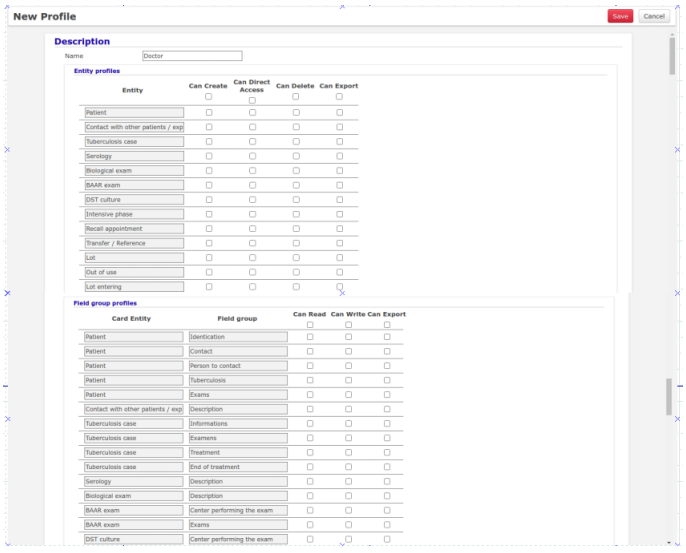

Figure 9: Interface presenting how rights are assigned to users

F RESOURCE AVAILABILITY STATEMENT

This paper is the extended version of the paper [70] accepted at CARI'2022. Additional material [69] that supports this research is available on Zenodo.

G ACKNOWLEDGEMENTS

The EPICAM project was partially supported by grants from PRODESO (Programme Franco-Camerounais pour un Développement Solidaire).

We would like to thank all the health workers for their feedback and constructive comments during the experimentation and the use of the EPICAM platform. Beside, our gratitude goes also to Dr. Nolna Désiré and Dr. Abena Jean Louis from the National Tuberculosis Control Program in Cameroon; Dr. Gaetan Texier from Centre Pasteur du Cameroon; Pr. Maurice Tchuente, from UMMISCO in Cameroon; Mr. Laurent Braak from MEDES in France and finally, all those who contributed to the EPICAM project.

This research was selected as an innovation at "Forum Afrique - 100 innovations pour un développement durable[23]."

[23]Results of selected innovations available at www.diplomatie.gouv.fr/IMG/pdf/Document-Multimedia-FR_cle0857a7.pdf