

An Adaptive Scrum Model for Developing Disease Registries

Hatem Bellaaj^{1,3}, Afef Mdhaffar^{2,3}, Mohamed Jmaiel³ and Bernd Freisleben⁴

¹Department of Mathematics and Computer Science, IPEIS, University of Sfax, Sfax, Tunisia

²Department of Mathematics and Computer Science, ISSAT, University of Sousse, Sousse, Tunisia

³Digital Research Center of Sfax, Sfax, Tunisia

⁴Department of Mathematics and Computer Science, University of Marburg, Marburg, Germany

Keywords: Disease Registry, Agile, Scrum, Sprint, Transparency, Inspection, Adaptation.

Abstract: This paper presents an adaptive model for developing disease registries. The proposed model is based on the Scrum methodology. It can be used to draw a road map to identify priorities, inputs, outputs, team members and exchanges for all tasks required to develop disease registries. Our model has been applied to real cases from several Tunisian hospitals where it has improved the efficiency of the team members. The developed disease registries are currently used in Tunisia. They allow medical researchers to identify origins of diseases, establish new protocols, perform surveys and compute morbidity.

1 INTRODUCTION

Disease registries are used to collect and analyze epidemiological information related to the frequency and distribution (i.e., incidence and prevalence) of a specific disease. In addition to the detection of known symptoms and diagnosis parameters of diseases, statistics obtained from disease registries can help doctors to discover new risk factors. These are important to assess the medical situation and to provide framework conditions, preventive measures and management plans. A clinical information system refers to the organization of clinical data from medical records of patients to coordinate the delivery of interventions and self-management support activities for the entire population. Building a disease registry is crucial for managing a population of patients suffering from chronic diseases (McCulloch et al., 1998).

A disease registry can be considered as a medical information system (MIS) and as an instance of a general information system (IS). The development of this kind of software system is a resource-intensive process that involves different groups of people in an organization. Several software development models have emerged. Among them are: the systems-development life cycle (SDLC), rapid application development (RAD), and agile methodologies. Agile methodologies are widely used for software project management, and Scrum is the most common agile method (Schwaber and Sutherland, 2001).

In this paper, we present a new model for developing disease registries based on Scrum. The model is adaptive in the sense that it provides support for incorporating recommendations that satisfy specific requirements of this domain. The proposed model has been applied to real cases from several hospitals in Tunisia, such as the Tunisian Fanconi Anemia Registry, the Tunisian Gaucher Disease Registry and the Tunisian Non-Hodgkin's Lymphoma Registry. In these projects, the use of our model has (a) accelerated the establishment of paper-based disease forms as functional specifications, (b) established a strong relation with the doctors, (c) minimized losses due to changes in fields' data types by fixing the sprint duration between 7 to 15 days, (d) found the right balance between documentation and discussion, (e) increased chances that the final product is as originally specified.

The paper is organized as follows. Section 2 introduces disease registries. Section 3 discusses related work. In Section 4, we summarize Scrum. Section 5 presents the proposed methodology. Applications are described in Section 6. Section 7 concludes the paper.

2 DISEASE REGISTRIES

A disease registry stores medical information related to the same disease. This allows medical researchers

to (1) review and study health records of many individuals, (2) answer questions about diagnostic disease parameters and (3) build a knowledge database for each disease.

A disease registry is a tool for tracking the clinical care and outcomes of a defined patient population. Most disease registries are used to support care management for groups of patients suffering from one or many chronic diseases, such as diabetes, coronary artery disease and asthma. In contrast to paper-based registries that have been used to track patients suffering from chronic diseases, digital registries provide users with an automated way to (1) store data, (2) create, sort, and display lists of patients and (3) extract data used for planning, quality improvement, reporting, and direct care delivery (Hummel, 2000).

Digital disease registry functionality is included (i.e., available as an add-on) in electronic health record (EHR) products. In addition, stand-alone options can be implemented and are usually simpler to set up than EHRs. Based on their priorities, some organizations may choose to implement digital disease registries as an interim step prior to implementing a more comprehensive EHR system. Disease registries can also provide more flexibility for reporting and aggregating data from multiple data sources. To implement a digital disease registry, an organization needs to analyze and adjust practice workflows to support new data collection requirements and to integrate the new information from their registry into decision making and planning (Hummel, 2000).

3 RELATED WORK

3.1 The Process

A software development methodology is used to structure, plan, and control the process of developing an information system. Several methodologies to develop software have been proposed, such as Agile Software Development, Crystal Methods, Dynamic Systems Development Model (DSDM), Extreme Programming (XP), Feature Driven Development (FDD), Joint Application Development (JAD), Lean Development (LD), Rapid Application Development (RAD), Rational Unified Process (RUP), Scrum, Spiral, and Systems Development Life Cycle (SDLC). Three often used methods are discussed below.

3.1.1 Systems Development Life Cycle (SDLC)

SDLC is considered to be the oldest project execution framework. It can be used to manage large soft-

ware projects, associated with corporate systems running on mainframes. It is a structured methodology, designed to manage complex projects that implicate many programmers and systems, having a high impact on the organization (Bourgeois, 2016).

3.1.2 Rapid Application Development (RAD)

RAD is a software development methodology that favors rapid prototyping on complex planning. In RAD, modules are developed in parallel as prototypes. Later, they are merged to the final product for faster delivery (Vickoff, 2000). The RAD method presents a secured and short development cycle following different phases: framing, design, building with fixed duration. Each phase takes about 90 to 120 days. RAD includes methods, techniques and tools to achieve four potentially contradictory objectives: budget, deadlines, technical quality, functional quality and visibility (Vickoff, 2000). The most important aspect for this model to be successful is to make sure that developed prototypes are reusable.

3.1.3 Agile Methods

Agile methods aim to reduce the life cycle of software development by creating a minimal version and then integrating functionality by an iterative process based on a customer listening and tests throughout the development cycle. The origin of agile methods is linked to the instability of the technological environment and the fact that the client is often unable to define his or her needs exhaustively from the beginning of the project. The term "agile" thus refers to the ability to adapt to changes in the context and changes in specifications occurring during the development process. It was first coined in 2001 in the Manifesto for Agile Software Development (Agile Manifesto) (Beck et al., 2001). Now, agile refers to any process that follows concepts of the Agile Manifesto. There are four main points in the Agile Manifesto:

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

The Agile Manifesto lists 12 principles to guide teams on how to execute with agility. The principles are described below.

1. *Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.*

2. *Welcome changing requirements, even late in development. Agile processes harness change for the customers competitive advantage.*
3. *Deliver working software frequently, from a couple of weeks to a couple of months, with preference to the shorter timescale.*
4. *Business people and developers must work together daily throughout the project.*
5. *Build projects around motivated individuals. Give them the environment and support their need, and trust them to get the job done.*
6. *The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.*
7. *Working software is the primary measure of progress.*
8. *Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.*
9. *Continuous attention to technical excellence and good design enhances agility.*
10. *Simplicity: the art of maximizing the amount of work not done – is essential.*
11. *The best architectures, requirements, and designs emerge from self-organizing teams.*
12. *At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly (Beck et al., 2001).*

3.2 The Product

The basic architecture of a digital disease registry consists of four layers. They are described below.

3.2.1 Data Layer

Data is stored in a specific electronic health record (EHR), with data corresponding to parameters related to the diagnosis of the disease, such as responses to a treatment. The included data respects specific properties, such as persistence, accuracy (Staroselsky et al., 2008) and validity (Carroll et al., 2007). There are three main ways to populate the registry with data:

1. directly through the fields in the disease form.
2. importing data using standard interoperability mechanisms, such as HL7 or CCD. Data is automatically inserted into the database.
3. combining these two approaches.

The data layer must respect the international standards. Among them are:

- ISO 13119:2012 describes metadata that relates to resources including medical knowledge. This standard applies mainly to digital documents, such as WWW resources, accessible from databases or file transfers. It can also be applied to paper documents, such as articles in the medical literature (ISO 13119, 2012).
- ISO 13606-1:2008 describes the communication of some or all of the electronic health record (EHR). The record of a patient is identified between the DSEs, or between this latest and a centralized repository. It can be used to communicate an EHR system or repository with clinical applications or middleware components that need to access or provide EHR data (ISO 13606-1, 2008).

3.2.2 Security Layer

We distinguish between different kinds of users: system administrator, group administrator (who establishes the disease registry form, statistics parameters, therapy protocol, etc.), reference doctors, participant doctors, analysts, developers, patients, simple users, etc. A set of rules that defines the behavior of each kind of users must be specified at the start of the project. Different security norms are defined for such systems. Two of them are described below:

- ISO / TR 11633-1, 2009 concerns remote maintenance services (RMS) for information systems in healthcare facilities. It presents an example of a risk analysis that protects the information in the medical information system (ISO/TR 11633-1, 2009).
- ISO / TS 14441, 2013 examines the electronic patient records systems at the point of clinical care that are also interoperable with EHRs. It treats their protections in terms of security and privacy through a set of security and privacy rules. It includes guidelines and practices for conformity assessment (ISO/TS 14441, 2013).

3.2.3 Dashboard Layer

The dashboard layer is the main goal of disease registries. It shows morbidity, mortality, survey curves, treatment responses and illness origins. The plurality of dashboard components is associated with a plurality of types of health-care content and are based on parameters received from a user.

3.2.4 Interoperability Layer

Interoperability consists of communication between healthcare organizations. The medical information

system is specific for each organization. Building a global medical information system for many organizations is a complex task. Indeed, there is an array of healthcare-related applications that supports multiple needs but remains isolated or incompatible. Thus, interoperability is a challenge (Hammami et al., 2014). Different interoperability norms are defined for such systems. Some of them are presented below.

- ISO / TR 16056-1, 2004 introduces the interoperability of telehealth systems and networks. It also defines telehealth and related terms.
- ISO EN 13606: Medical record communication is based on a two-model approach using paradigms. This ensures that the data collected from heterogeneous systems are correctly interpreted. The dual model provides the basis for a stable architecture. It separates information from knowledge.

4 SCRUM

Scrum originates from the sporting term rugby meaning: melee. Like this technical aspect of the game, the methodology asks its actors to be united in the accomplishment of a project, in achieving a goal. A melee is not a unique process. This is a part of the game that is often found to move the team forward. In the same concept Scrum uses a procedure that we name sprint. Each iteration or sprint provides a functional part of product. Three pillars uphold every implementation of empirical process control: transparency, inspection, and adaptation (Schwaber and Sutherland, 2001). Scrum includes definitions, descriptions, concepts and methods for better running projects. It defines details for: the Scrum team, the product owner (PO), the development team, the Scrum master, the Scrum events, the sprint, the daily Scrum, the sprint review, the sprint retrospective, the artifacts, the product backlog, the sprint backlog and the artifact transparency.

5 AN ADAPTIVE SCRUM MODEL FOR DISEASE REGISTRIES

This section presents all adapted pillars, mechanisms, and concepts for developing disease registries using the Scrum model.

5.1 Three Pillars

5.1.1 Transparency

Transparency insurance is difficult between people who do not share the same discourse. Thus, we propose to build:

- a medical dictionary, including term definitions comprehensible by all players
- a crossing table between the variables, specifying: mathematical equations, if they exist and logical relations between values, e.g., it is not logical to find the symptom S1 true while the variable V1 is normal
- an effective way to present a variable: selection between alphanumeric value, checkbox, text field to be specified

5.1.2 Inspection

There are three levels of inspections that can be done sequentially and at the same time according to the iteration in question:

- functional validation by computer scientists
- medical validation of the distribution of the fields to be entered, the logical and mathematical links of the different variables
- validation of the possibility of statistical interpretation of the different variables

The second type of validation often leads to the addition, deletion or modification of the type of presentation of some fields. Development constraints cause developers to modify a type of representation or an operating logic that can essentially lead to poor statistical quality. Inspection becomes foolish and can greatly slow down the development process.

5.1.3 Adaptation

Scrum prescribes four formal events for inspection and adaptation: sprint planning, daily scrum, sprint review and sprint retrospective. The fact that the project is carried out by people of different specialties, a misunderstanding of need can delay and burden the concept of adaptation. Reports of daily meetings at a frequency of maximally three days can be sent to the head of the medical study group.

5.2 The Scrum Team

The Scrum team consists of a product owner (PO), the development team, and a Scrum master. Scrum

teams are self-organizing and cross-functional. In the following, the needed skills, the specific mission, relations and input / output of each of them are detailed.

For a disease registry, three teams should be established, which will work together and simultaneously: the doctors who are members of the study group, statisticians and developers. The PO leads these three teams and takes decisions after meetings including all the teams or representative members of each of them. Therefore, (s)he is the first person responsible for the product backlog.

The team model in Scrum is designed to optimize flexibility, creativity, and productivity. For disease registries, flexibility is guaranteed by trust, designing, and good communication. Trust means the capability of achieving a good job, possibly by young employees. Designing consists of giving priority to dispatching the members of the team and not the tasks. Productivity is increased by communication (e.g., regular meetings), tools (e.g., management version applications) and documentations.

Scrum teams deliver products iteratively and incrementally, maximizing opportunities for feedback. An average incremental delivery period can be defined as about 10% of the number of fields for form and dashboard pages. For the user management module, 25% of the total number of user groups is appreciated. For other modules, these values must be defined at the start of the project but cannot exceed two weeks as a period to increase profitability.

5.2.1 The Product Owner

The product owner is in charge of maximizing the value of the product and the work of the development team. The PO is the sole person responsible for managing the product backlog (Sverrisdottir et al., 2014).

For disease registries, it is recommended that the product owner communicates periodically the updated backlog to the leader of the disease study group, discusses and may change some priorities.

The PO insures that the product backlog is visible, transparent, and clear to all, and shows what the Scrum team will work on in the next step. The draft version of backlog should be validated at the last meeting with doctors before the kick-off of the project.

5.2.2 The Development Team

The development team consists of professionals who are in charge of delivering a feasible increment of the done product at the end of each sprint. Only members of the development team create the increment.

In Scrum, it is recommended that the development team includes 3 to 9 members to insure efficiency and global efficacy. For a disease registry, we have adopted this structure: 1 designer, 1 tester, 1 architect, 1 to 2 developers for security management and 3 to 9 Java developers.

5.2.3 The Scrum Master

The Scrum master is responsible for ensuring that Scrum is understood and implemented. It will play exactly the same roles as defined in the Scrum guide. Thus, it will ensure the availability of tools necessary for the good understanding and adherence to Scrum for the product owner and the development team.

Ideally, the Scrum master is a member of the development team. He or she is supposed to master notions of statistics. In his or her profile, conducting research in medical informatics or participating in a similar project is appreciated.

5.3 The Scrum Events

In Scrum, we attempt to fix regular events and to minimize the need for unplanned meetings. All events are timed, so that each event has a maximum duration. Once a sprint starts, it can not be shortened or lengthened. Events end each time the associated goal is reached. Appropriate time must be allocated for each sprint to avoid having waste in the process.

5.3.1 The Sprint

The heart of Scrum is a sprint. It is a time block resulting in an increment of the potentially deliverable product. It lasts for one month or less, and a deliverable product increment is created. A new sprint starts immediately after the conclusion of the previous sprint (Schwaber and Sutherland, 2001).

Sprints contain and consist of the sprint planning, daily Scrums, the development work, the sprint review, and the sprint retrospective (Schwaber and Sutherland, 2001). For a medical disease registry, the average duration of a sprint is ideally between 7 and 15 working days.

5.3.2 Sprint Planning

One of the first methods spread around the world is the V-Cycle method. It is a project organization logic that limits the problem of reactivity in the event of an anomaly, limiting the return to the preceding steps. It is represented by a V whose descending branch contains all the phases of the design of the project, and the rising branch all the stages of tests of the project.

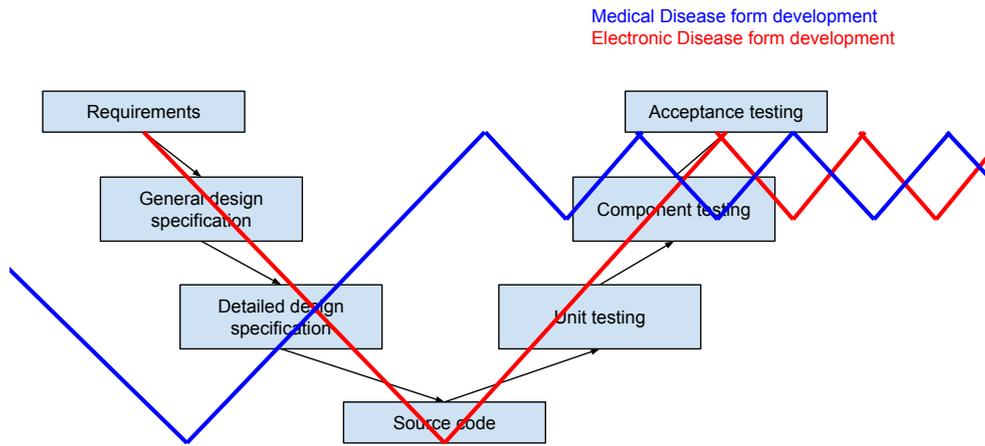


Figure 1: The W Model for disease registry development.

The tip of the V represents the stage of realization of the project. Each phase of the descending branch is related to a phase of the rising branch.

In disease registry development, we have adopted, for the first time, this kind of project organization method. The same thing is done with the medical team to establish the disease form and the list of included elements in a dashboard (statistics). For this purpose, two V cycles (i.e., a W cycle) are established with intersection points (see Figure 1). These points represent meetings between members of medical and development staff. In addition to the complicated problems of return in the cycle for both teams, the meeting management presented by the intersection points in Figure 1 must be handled.

In disease registry development, we recommend that a sprint duration is between 1 and 2 weeks. The reduction of this value increases the profitability of the team, but may make the management of the different tasks complicated. In this stage, it is important to take into account the source version management process. Two methods can be used: distribution according to functionalities or according to teams. The first is strongly recommended for disease registries.

5.3.3 Daily Scrum

The daily scrum is a 15-minute time-boxed event for the development team to synchronize activities and create a plan for the next 24 hours (Schwaber and Sutherland, 2001). It is recommended that a member of the study group participates in the daily scrum. Since doctors are usually solicited by their patients, it is recommended to establish these meetings at the end of the day. Thus, a meeting should answer the following questions:

- What did I do today that helped the development

team meet the sprint goal?

- What will I do tomorrow to help the development team meet the sprint goal?
- Do I see any impediment that prevents me or the development team from meeting the sprint goal?

5.3.4 Sprint Review

A sprint review is held at the end of the sprint to inspect the increment and adapt the product backlog if needed. During the sprint review, the Scrum team and doctors (two or three doctors who have different specialties), and members of the study group collaborate about what was done in the sprint.

For the first six months, the sprint review should be done twice a month. After that, the frequency can be decreased to once a month. The meeting can take between 15 minutes and 1 hour.

In addition to the elements defined in the Scrum guide, the sprint review includes (1) the team problems to understand field types and relations; (2) doctors' comments and validation about cognitive workload; (3) the steps to do by participating doctors and (4) discussion of technical issues, system interoperability, privacy, confidentiality and lack of health information data standards.

5.3.5 Sprint Retrospective

The purpose of the meeting is to improve the process for the next sprint. The entire Scrum team participates in the meeting. The retrospective takes place just after the sprint review, and the speakers who have attended can remain for the retrospective as observers (Schwaber and Sutherland, 2001). The retrospective sprint clarifies the points of interference with the doc-

tors; a sharp intervention by them to verify the understanding may be planned for the next sprint.

5.4 Scrum Artifacts

Scrum artifacts represent information that the Scrum team needs to insure inspection and adaptation. They help the team to understand the product under development, the activities done, and the activities being planned in the project. Scrum defines the following artifacts: product backlog, sprint backlog, increment and burn-down chart. For a disease registry, the modification proposal is limited to the three first artifacts (Schwaber and Sutherland, 2001).

5.4.1 Product Backlog

The Scrum product backlog is a prioritized feature list, containing short descriptions of the functionality desired for the product. In a disease registry project, the product backlog includes:

- Disease form management operations: insert, modify, list and delete. The deletion must be logic. A module for logging the various actions carried out by the user must be set up. Data verification algorithms may be required.
- User management operations: insert, modify, list and delete. A group right must be clearly defined to access data and to consult statistics. In the general case, the doctor should consult and modify only forms that (s)he has introduced.
- Data mining includes histogram presentations, pie chart, or curves for:
 - enumerating parameters (example: consanguinity)
 - numerical parameters (example: weight), with a possibility of zooming on particular zones
- Security requirements: who can do what?

5.4.2 Sprint Backlog

The sprint backlog is created during the sprint planning event, which is the first event in a sprint. A critical point in a disease registry project is the establishment of the detailed plan for delivery of the items and realization of the sprint goal during the sprint. Some distributions may occur due to insufficient explanations of requirements by the doctor for mathematical or logical relations between fields. This detailed plan will continue to be updated during the sprint.

5.4.3 Increment

An increment represents items made during the current sprint and those that preceded it. At the end of a sprint, the tasks included in the new increment must be performed. This means it must be finished (i.e., developed and tested) and meet the Scrum teams definition of done. In a disease registry, a task is done when it is validated by at least one doctor.

6 APPLICATIONS

Three disease registries were developed based on the proposed approach: The Tunisian Fanconi Anemia Registry (TFAR), The Tunisian Non-Hodgkin Lymphoma Registry (NHLR) and the Tunisian Gaucher Disease Registry (TGDR).

The TFAR was developed within one year. The disease form has more than 200 fields. Doctors from 5 medical areas participated: hematology, oncology, biology, pediatrics, cytogenetics. They belong to 10 hospitals and institutes. The project was developed by 8 members: 5 developers, 1 statistician, 1 human machine interface designer and 1 product owner. Scrum daily meetings included one doctor and usually took about 1 hour. The first backlog sprints contain explanations of data fields and relations between them.

The NHLR was developed, in cooperation with MDSOFT¹, during three years. The statistics module is still in the works. The disease form includes more than 1000 fields. Doctors from 2 medical areas participated: hematology and oncology. They belong to 6 hospitals and institutes. The project was developed by 7 members: 4 developers, 1 statistician, 1 human machine interface designer and 1 product owner. The duration of Scrum daily meetings is about a half hour, with the participation of one doctor. A particular difficulty was experienced during Scrum planning. The disease form is divided into sections. Each section contains several input fields. During the merging operation of the source code of the different sections, we have encountered difficulties especially in the case where there are relationships between the fields that they contain. The sprint retrospective was complicated due to the new requests raised by the doctors after each trial of the product.

The second edition of the TGDR was established in November 2016 in collaboration with the MD-SOFT team. The disease form includes more than 250 fields. Doctors from 5 medical areas participate: hematology, pediatrics, internal medicine, rheumatology and gastroenterology. They belong to 6 hospitals

¹<http://www.mdsoft-int.com>

and institutes. The project was developed by 6 members. The product backlog has been updated several times due to the instability of both medical and developer teams.

During these projects, our new methodology has improved the following indicators: dedication (additional spontaneous working hours reached up to 80%), focus (90% of the tasks were completed within deadlines), openness (the rate of misunderstanding between team members was less than 5%), audacity (the team did not hesitate to name things, to ask questions and to propose new solutions), and continuity (despite major changes in team composition, the projects did not have any delays in delivery). Moreover, the use of our methodology has reduced the risk of failure by 95% in the case of TFAR.

7 CONCLUSION

In this paper, we have presented a new methodology based on Scrum for disease registry development. Several actions have been proposed to improve team performance: (a) minimize the time of different iterations, (b) facilitate code retrieval in the majority of iterations, (c) clarify the descriptions and interactions between different fields, (d) maximize collaboration between the different teams and specialists involved, namely doctors, computer scientists and statisticians. Our methodology has been applied to the Tunisian Fanconi Anemia Registry, the Tunisian Non-Hodgkin Lymphoma Registry and the Tunisian Gaucher Disease Registry. The developed registries are currently used in several hospitals in Tunisia.

In the future, we plan to define an effective methodology for managing source code and deliverables to improve team profitability.

ACKNOWLEDGEMENTS

This work is supported by the German Academic Exchange Service (DAAD) (Transformation Partnership: Theralytics Project).

REFERENCES

Beck, K., Beedle, M., Bennekum, A., Alistair, C., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. (2001). The agile manifesto. *Agile Alliance*, 2001.

- Bourgeois, D. (2016). *Information Systems for Business and Beyond*. Textbook Equity Edition.
- Carroll, N., Ellis, J., Luckett, C., and Raebel, M. (2007). Improving the validity of determining medication adherence from electronic health record medications orders. *J. Am. Med. Inform. Assoc.*, 18(5):717–720.
- Hammami, R., Bellaaj, H., and Kacem, A. (2014). Interoperability for medical information systems: an overview. *Health and Technology*, 4(3):261–272.
- Hummel, J. (2000). Building a computerized disease registry for chronic illness management of diabetes. *Clinical Diabetes*, 18(5).
- ISO 13119 (2012). Health informatics – clinical knowledge resources – metadata. Standard, International Organization for Standardization.
- ISO 13606-1 (2008). Health informatics – electronic health record communication – part 1: Reference model. Standard, International Organization for Standardization.
- ISO/TR 11633-1 (2009). Health informatics – information security management for remote maintenance of medical devices and medical information systems – part 1: Requirements and risk analysis. Standard, International Organization for Standardization.
- ISO/TS 14441 (2013). Health informatics – security and privacy requirements of EHR systems for use in conformity assessment. Standard, International Organization for Standardization.
- McCulloch, D., Price, M., Hindmarsh, M., and Wagner, E. (1998). A population-based approach to diabetes management in a primary care setting: early results and lessons learned. *Effect. Clin. Pract.*, 1(1):12–22.
- Schwaber, K. and Sutherland, J. (2001). The Scrum guide.
- Staroselsky, M., Volk, L., Tsurikova, R., Newmark, L., Lipincott, M., Litvak, I., Kittler, A., Wang, T., Wald, J., and Bates, D. (2008). An effort to improve electronic health record medication list accuracy between visits: patients' and physicians' response. *Int. J. Med. Inform.*, 77(3):153–160.
- Sverrisdottir, H., Ingason, H., and Jonasson, H. (2014). The role of the product owner in Scrum - comparison between theory & practice. *Procedia - Social & Behavioral Sciences*, 19(3):257–267.
- Vickoff, J. (2000). RAD, le developpement rapide d'applications. *RAD.fr*.