35 Using Correct-by-Construction Software Agile Development

Rafael Augusto Lopes Shigemura, Gildarcio Sousa Goncalves,
Luiz Alberto Vieira Dias, Paulo Marcelo Tasinaffo, Adilson Marques da
Cunha, Luciana Sayuri Mizioka, Leticia Hissae Yanaguya,
and Victor Ulisses Pugliese

Abstract

Disasters and crises, whether climatic, economic, or social are undesirably frequent in everyday lives. In such situations, lives are lost mainly because of inadequate management, lack of qualified and accurate information, besides other factors that prevent full situational awareness, including software failures. The goal of this paper is to report the agile conceptualization, design, build, and demonstration of a computerized system, containing correct-by-construction software, to safely manage critical information, during alerts or crises situations. On this research, the following challenges and requirements were tackled: formal specifications, aerospatial-level reliability, agile development, embedded systems, controlled testability, and product assessment. An Interdisciplinary Problem-Based Learning (IPBL), involving a Scrum of Scrums Agile Framework was adapted for managing the cohesive, productive, and collaborative development team of around 100 undergrad and graduate students remotely working. In addition, the following hardware technologies, for supporting the software development were used: environment sensors, Radio Frequency Identification (RFID), and Unmanned Aerial Vehicles (UAVs). Other software technologies were also used, as well cloud-based web-responsive platforms and mobile applications to geographically manage resources at real-time. Finally, the ANSYS® SCADE (Safety-Critical Application Development Environment) was employed to support the embedded and correct-by-construction module of this system, according to Model-Driven Architecture (MDA) and Model-Driven Development (MDD).

R. A. L. Shigemura (✉) · G. S. Goncalves · L. A. V. Dias
P. M. Tasinaffo · A. M. da Cunha · L. S. Mizioka · L. H. Yanaguya
V. U. Pugliese
Computer Science Department, Brazilian Aeronautics Institute of Technology (Instituto Tecnologico de Aeronautica—ITA), Sao Jose dos Campos, Sao Paulo, Brazil
e-mail: rafael@ita.br

Keywords

Agile software development · Correct-by-construction software · Model-Driven Architecture (MDA) · Model-Driven Development (MDD) · Scrum agile method · Interdisciplinarity

35.1 Introduction

According to the United Nations Office for Disaster Risk Reduction (UNISDR), a disaster is a serious disruption of community or society functions, involving widespread human, material, economic, or environmental losses and impacts, which exceeds the ability of the affected community or society to cope with, by using its own resources [1].

Only in 2016, there were 327 catastrophes worldwide, killing about 11,000 people and bringing an estimated total economic loss of USD 175 billion. Additionally, since the year 2000, there were three disasters with more than 100,000 victims each [2].

Given this huge global impact, several strategies are underway to mitigate its consequences, organized both national and internationally around the World [3].

However, a study published by the Swiss Reinsurance Company shows that the frequency of disasters, both natural and man-made, has been steadily increasing year by year, in the last 40 years, as summarized in Fig. 35.1 [2].

Information Technologies (ITs) can, undoubtedly, provide tools to manage critical information, during alerts and/or crises situations but high reliability is a must [4]. Even high-profile software systems, like the American 911, can suffer from dumb software defects [5]. Also, the ever-changing nature of occurrence demands for fast response and adaptation.

This scenario led some members of the Software Engineering Research Group (*Grupo de Pesquisa em Engenharia de Software*—GPES) of the Brazilian Aeronautics Institute

S. Latifi (ed.), *Information Technology – New Generations*, Advances in Intelligent Systems and Computing 738,
https://doi.org/10.1007/978-3-319-77028-4_35

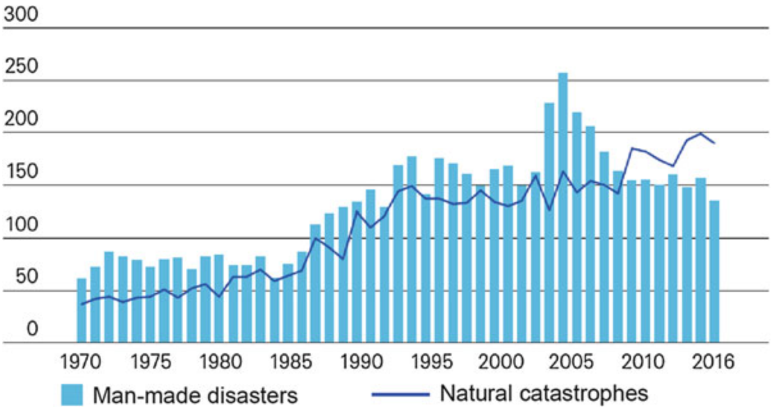

Fig. 35.1 The number of catastrophic events from 1970 to 2016 [2]

Table 35.1 Standards partially used in the RT-ACMIS project, through SCADE KCG compiler

Standards	Descriptions
DO-330 TQL-1	Software tool qualification considerations
DO-331	Model-based development and verification supplement to DO-178C and DO-278A

of Technology (*Instituto Tecnologico de Aeronautica*—ITA) to raise the following essential research questions:

1. Is it possible to bring together and use, productively, the two supposedly divergent approaches: Agile Software Development and Formal Specifications?
2. Assuming agile development together with highly reliable software technologies, how could them be effectively applied to mitigate risks and/or increase community resilience?

Recent works and advances to answer the question 1 suggest that hybrid approaches, using Agile and Formal Methods, can bring 'the best of each' to the software engineering and/or system engineering fields [6].

Aiming to move forward in answering these two questions, this paper reports an academic capstone project of 17-weeks, during the second Semester of 2016, at the ITA, involving the following challenges and requirements: formal specifications, aero spatial-level of reliability, agile development, embedded systems, controlled testability, and product assessment.

An Interdisciplinary Problem-Based Learning (IPBL), involving a Scrum of Scrums Agile Framework [7, 8] was adapted, for managing the cohesive, productive, and collaborative development team of around 100 undergrad and graduate students remotely working and, applying knowledge gathered from the following courses (occurring paralleled to the project): CES-65 Embedded Systems Project; CE-235 Real-time Embedded Systems; CE-230 Software Quality, Reliability, and Safety; and CE-237 Advanced Topics in Software Testing.

35.2 Background

This section describes the following key concepts, methods, and techniques used in the development of the Real-Time Accident and Crises Management Integrated System (RT-ACMIS) project, named in Portuguese *Sistema Integrado de Gerenciamento em Tempo Real de Acidentes e Crises* (SIG-TRAC): Software Quality, Reliability, and Safety; Agile Scrum Method; Agile Testing; the ANSYS® SCADE; and also the involved hardware.

35.2.1 Software Quality, Reliability, and Safety

The RT-ACMIS project was developed, by following quality, reliability, safety, and testability requirements and also the DO-178C [9] and the DO-278A [10] standards.

Through the SCADE KCG compiler [11], other Standards were also partially applied, as shown in Table 35.1. The KCG is a C and Ada code generator from SCADE models. More detailed information will be presented in Sect. 35.2.4.

The software quality, reliability, and safety evaluation were performed throughout a systematic examination of activities during all the sprints. There was used an auditing process by checking the activities compliance with project and standards requirements, previously established, as shown in Fig. 35.2.

Quality never occurs by accident, it is always the result of high intention, sincere effort, intelligent direction, and skillful execution of expected planning [12].

Fig. 35.2 Example of DO-178C Compliance Matrix used in the RT-ACMIS

	Reference	Sprint 1			Sprint 2			Sprint 3			
		US01	US03	US06	US12	US17	US18	US12	US17	US13	US19
QUALITY	DO-178C A 3.6	S	S	S	S	S	S	S	S	S	S
	DO-178C A 3.5	S	S	S	S	S	S	S	S	S	S
	DO-178C A 2.4, A 4.1 e A 4.2	S	S	S	S	S	S	S	S	S	S
	DO-178C A 3.4 e A 4.4	S	S	S	S	S	S	S	S	S	S
	DO-178C A 2.4 e A 4.6	S	S	S	S	S	S	S	S	S	S
RELIABILITY	DO-178C A 2.3	S	S	S	S	S	S	S	S	S	S
	DO-178C A 2.7	S	S	S	S	S	S	S	S	S	S
	DO-178C A 6.1 e 6.3	S	S	S	S	S	S	S	S	S	S
	DO-178C A 2.1 e A 2.4	S	S	S	S	S	S	S	S	S	S
	DO-178C A 9.x	S	S	S	S	S	S	S	S	S	S
SAFETY	DO-178C A 2.2 e A 2.5	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	DO-178C A 7.2	S	S	S	S	S	S	S	S	S	S

35.2.2 The Agile Scrum Method

According to the Scrum Alliance, Scrum is a framework within which people can address complex adaptive problems, while productive and creatively delivering products of the highest possible value [8].

Scrum has three pillars: transparency, inspection, and adaptation, involving team following roles, ceremonies, and artifacts, as presented in Fig. 35.3 [7]:

- Roles—Product Owner (PO), Scrum Master (SM), and Team Developer (TD);
- Ceremonies—Sprint Planning Meetings, Daily Sprints or Weekly Meetings, Sprint Reviews, and Sprint Retrospectives; and
- Artifacts—Product Backlog, Sprint Backlogs, Kanban, and Burndown Charts.

The roles are played by the Product Owners (POs), who represent interests of all stakeholders. They also provide requirements and funds and also accept deliverables; Development Teams (DTs) are responsible for developing and testing these deliverables. Finally, there are Scrum Masters (SMs) who are responsible for managing the Scrum process and the DT, solving any issue; in order to have deliverables complied with requirements, ensuring timely deadlines and high quality of products.

Sprint is a manageable short time period interaction. In general, it takes about 4 weeks. Its goal is to produce a tested and stable deliverable.

At the beginning of a Sprint, the team selects prioritized requirements of the Product Backlog to be developed and compiled in an artifact named Sprint Backlog.

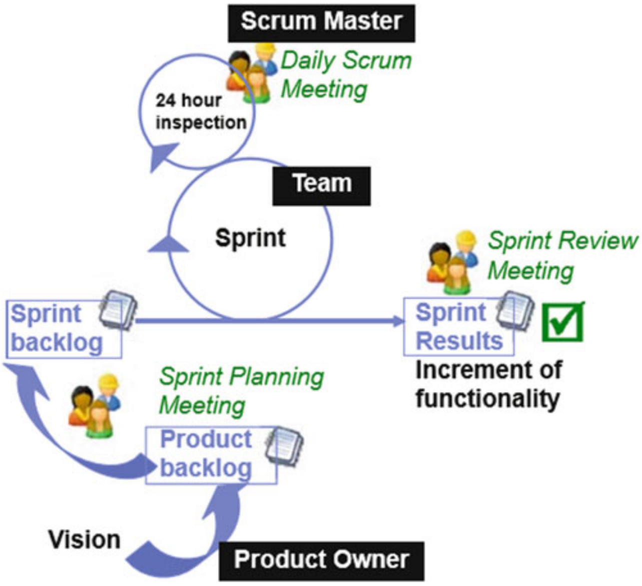

Fig. 35.3 The Scrum roles, ceremonies, and artifacts [13]

At the end of a Sprint, in the Sprint Review ceremony, the team presents Sprint results. The SM and the PO inspect and adapt the project and the product for the next Sprint. The DT and the SM hold a ceremony named Sprint Retrospective, to report any limitation or experience that could affect the team performance.

35.2.3 Agile Testing

Besides Formal Methods, software testing is used to identify possible defects in the software and to check whether the

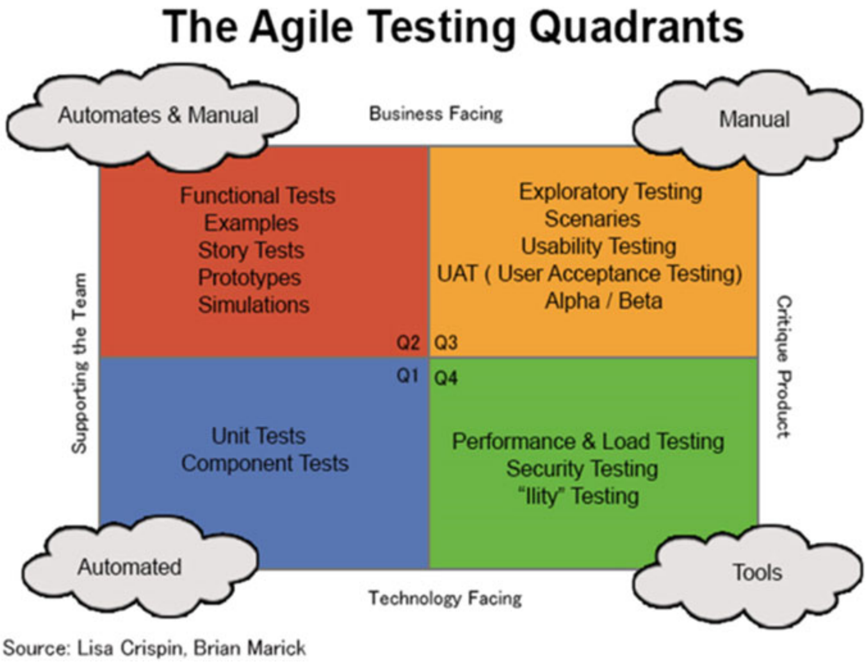

Fig. 35.4 The agile testing quadrants [16]

system complies with customers' requirements, considering effectiveness and use. The software product testing basically involves four steps: test planning, test case design, implementation, and evaluation of test results [14]. In general, these steps are materialized in four test levels: unit, integration, system, and acceptance [15].

Crispin and Gregory define agile testing quadrants, which in this RT-ACMIS project were used to guide the testability assessment activities, as shown in Fig. 35.4 [16].

Test Driven Development (TDD) is used to guide the development, because it must be written before implementing the system. Tests are used to provide project understanding and to clarify what is expected from the code [17, 18]. The TDD cycle is composed by: test adding, code execution and results' analyses, code writing, automated tests execution, and code refactoring.

The creation of unit or component testing is a crucial part of a project. Individual components are tested to ensure proper operations. Each hardware and software component must be independently tested. Components may be single entities such as functions or classes of objects or may be coherent groups of these entities. TDD is usually chosen for the following reasons [19, 20]:

- The programmer is the one who creates the test, while the software code is still very present in his mind;
- The test might be automated, ensuring greater frequency in implementation;
- It must also ensure that it runs with identical results, every time it is used; and
- Not only unit testing brings success to an application, but also it is necessary to test the system as a whole.

According to [21], components are integrated to compose the system. This process is related to the search for defects that result from interactions not provided between system components.

In this project, the agile software testing using Agile Test Quadrants aimed to help coding, significantly reducing problems from the development phase, as well to advance the study of applicability of this model in embedded, real-time and Internet of Things environment, as recommended in [22].

35.2.4 The ANSYS® SCADE

This section presents some theoretical overview of SCADE Suite® I-CASE-E tool and it's Synchronous Programming Model (SPM) of reactive systems. SPM is based on the synchronous concurrency model [23], in which concurrent processes can perform computation and exchange information instantly, at least at theoretical level.

This model is widely accepted and used by automatic control and engineering industry, ranging from hardware circuit design to large scale real-time process control, including embedded systems, drivers, communication protocols, aerospatial and automotive control, amongst others [23].

Reactive Systems Model (RSM), in turn, respond to stimuli from the environment, within a strictly defined period and safety standards, unlike interactive systems, which respond to users' stimuli according to the availability of resources [23]. This property of RSM is essential to safety-critical systems due of its behavioral determinism [23].

SPM and RSM form an ideal pair for development of embedded safety-critical systems, but can become a problem if they need to be coded by hand, reducing agility to change and flexibility. The SCADE Suite®, fully implements SPM and RSM, through a graphical Model-Driven Development (MDD) environment.

The MDD paradigm has been widely used to complex projects in various areas of science and engineering. The ability to perform computer simulations in abstract models can provide hundreds of million dollars savings, instead of generating a single physical prototype test [15].

It also provides the automatic generation of significant amount of source-code in C or Ada languages, through the implementation of native available components standardized according to DO-178C [24], as the SCADE KCG compiler.

SCADE Suite KCG is a C and Ada code generator from SCADE models that has been qualified as a development tool for DO-178B software up to Level A and DO-178C/DO-330 at TQL-1 [11].

This code generator saves verification effort in the coding phase, such as code reviews and low-level testing on the SCADE Suite KCG generated code. This productivity improvement shortens certification and/or modification time and effort. SCADE Suite KCG has successfully passed the qualification procedure on several large programs, and is currently used in production for several programs in Europe, Asia and the Americas [11].

Another highlight worth to mention is the fact that not only code, but all generated artifacts (including documentation) are readily certifiable by the DO-178C standard, and its use significantly reduces the waste of time and other resources in certification process [24].

Finally, SCADE developed models are formally guaranteed to accomplish the requisites, because is submitted to internal model checking formal verification. This check is performed using formal methods, namely Weak Bisimulation Reduction and Temporal Logic, over Finite State Machines [23].

35.2.5 Hardware

The following hardware were researched, implemented, and/or used in the RT-ACMIS project, as a Internet of Things Proof of Concept (PoC):

- Raspberry Pi, used as remote processing unit;
- Arduino, as sensors microcontroller;
- A high sensitive noise sensor;
- A temperature and humidity sensor;
- An inflammable gas and smoke sensor;
- A heartbeats sensor;
- A Radio Frequency Identification (RFID) transmitter/reader, used in this project to store serial numbers for personal, object, or information identifications on microchips;
- A RFID bracelet, a mix of a radio transmission device used to store and retain patient information, having a microchip and an antenna to allow communication to RFID transmitters/readers; and
- A drone Parrot 2.0, a quadcopter managed via WiFi connection, using a Software Development Kit (SDK) programmed with an Application Program Interface (API) in C language, and a Global Positioning System (GPS).

35.3 The Project Development

This section describes the usage of the Scrum method on a project management for academic purposes. It shows the product vision and the assigned mission project. At the end, it addresses the development and the main challenges faced on its four sprints. Figure 35.5 shows the RT-ACMIS project divided into segments and subsystems assigned from ST01 to ST08.

35.3.1 Tailoring the Scrum Agile Method

During the RT-ACMIS project, the Scrum of Scrums technique was adopted. It allowed STs to manage the development of products and services [19].

In this project, only for academic purposes, the following two new roles were created: General Product Owner (GPO) and General Scrum Master (GSM), in order to improve the RT-ACMIS project development management with students.

The project development was divided into eight STs listed below with their macro-functions:

- ST01—CIVIL DEFENSE/Collaboration and Coordination;
- ST02—HEALTH CARE/Medical and Ambulance First Aid;
- ST03—FIRE DEPARTMENT/Search and Rescue;
- ST04—POLICE DEPARTMENT/Police Report and Civilian Security;
- ST05—CIVIL DEFENSE/Communication and Cooperation;
- ST06—HEALTH CARE/Hospital and Intensive Care Unit (ICU);
- ST07—FIRE DEPARTMENT/Rescue and Aftermath; and
- ST08—POLICE DEPARTMENT/Preventive Security and Military Security.

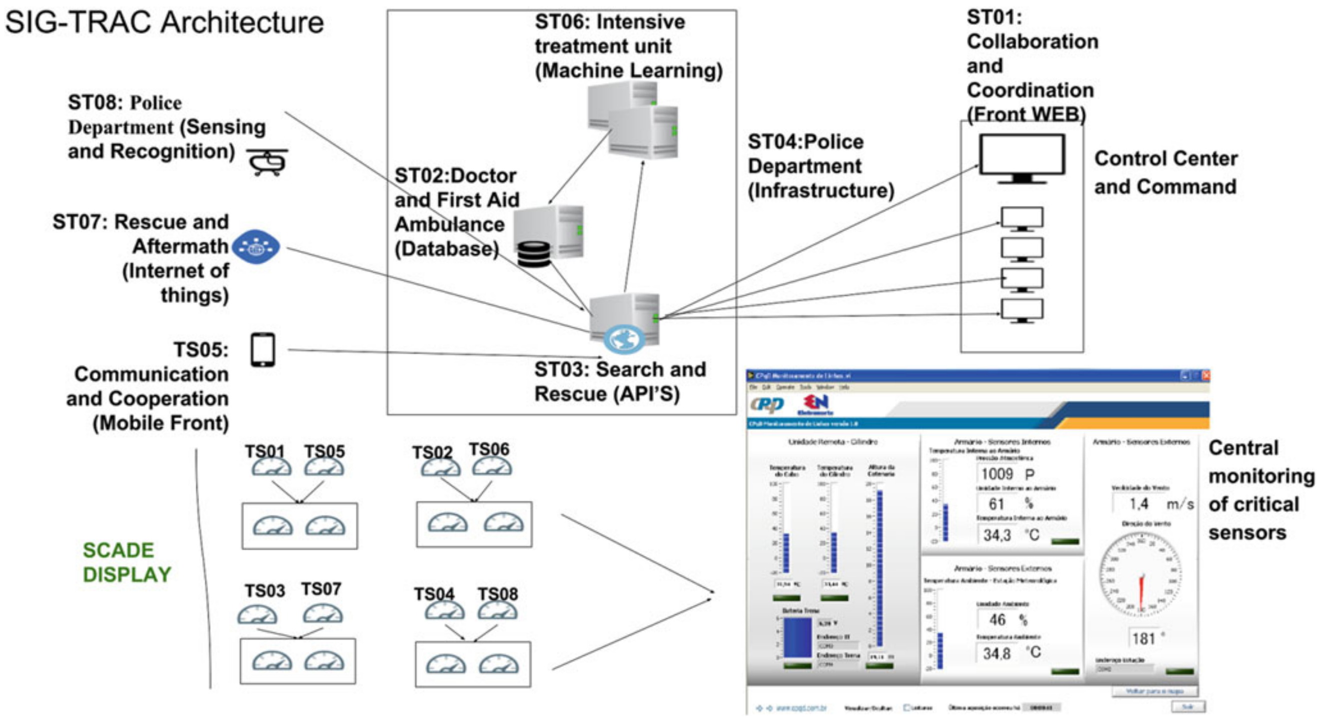
SIG-TRAC Architecture
ST06: Intensive treatment unit (Machine Learning)
ST01: Collaboration and Coordination (Front WEB)
ST08: Police Department (Sensing and Recognition)
ST02:Doctor and First Aid Ambulance (Database)
ST04:Police Department (Infrastructure)
Control Center and Command
ST07: Rescue and Aftermath (Internet of things)
TS05: Communication and Cooperation (Mobile Front)
ST03: Search and Rescue (API'S)
TS01
TS05
TS02
TS06
TS03
TS07
TS04
TS08
SCADE DISPLAY
1009 P
61 %
34,3 °C
46 %
34.8 °C
1,4 m/s
181 °
Central monitoring of critical sensors

Fig. 35.5 The RT-ACMIS (SIG-TRAC) project architecture

35.3.2 The RT-ACMIS Project Pre-game Phase: Product Vision

During the preparation for Sprints, POs of each ST have created the product vision, as an important agile artifact, used to identify and describe the RT-ACMIS project focusing on the product to be developed, as follows:

> *"**For** public or private organizations involved in monitoring, warning, or accidents and/or crises preventions motivated by adverse events of any nature, including terrorist attacks, **who** require and/or wish to explore Real-time Embedded Systems, Big Data, Remote Sensing, Machine Learning, among other related emerging technologies, **the** RT-ACMIS project **is** a Real-time Computer System (involving Hardware and Software) for analysis and issuance of automatic alerts. **Differently** from existing products from Universities, Research Institutes, Government Agencies and/or Public or Private Enterprises, **this product** was developed in just 17 weeks in an academic environment, at a distance and as a part time project, using the best agile practices and cutting-edge technologies with quality, reliability, safety, and testability."*

35.3.2.1 The Assigned Mission

For the final analysis and evaluation of the RT-ACMIS project, the students had to show the accomplishment of a fictitious crisis assignment mission, demonstrating the management of it through a system operation Proof of Concept (PoC). This mission was composed by three main phases:

1. Preparation—Crisis alert and activation of firefighter segment;
2. Action—Segments acting on crisis; and
3. Reconstruction—Police acquiring images from the local of accident and closes the crisis.

35.3.3 The RT-ACMIS Project Development: Game Phase

This section tackles the academic project development and its challenges.

35.3.3.1 Sprint #0: Preparation for Sprints

The Sprint 0 was designed to provide basic training for all students in different subjects to enable their roles as TDs, POs and SMs, using the SCADE and several other tools.

The training was conducted in an agile, collaborative, and interdisciplinary way, through books, lectures, hands-on exercises, and classes. In the end, those involved were able to learn the embedded systems fundamentals in real-time.

35.3.3.2 Sprint #1

On this Sprint #1, both ST01 and ST05 (from the Civil Defense Segment) were able:

- To create a registration screen, named Civil Defense Community Centers (CDCC), with the ability to provide basic information about accidents and/or crises;
- To register and use the other segment resources; and

- To develop a service registration of accident and/or crisis report.

ST02 and ST06 (from the Health Segment) were able:

- To develop scripts, for calculating the number of doctors per ambulances;
- To develop the initial web portal layout;
- To develop a screen in SCADE for the ambulance;
- To develop some research on patient care protocols for crises and disasters; and
- To create an exhibition screen bed management for Intensive Care Unit (ICU) also using the SCADE.

ST03 and ST07 (from the Firefighter Segment) were able:

- To develop a victims layout classification;
- To establish constant contacts with real-life firemen to study, investigate, and better understand the used procedures of search and rescue and path location to the nearest firefighters; and also
- To study and investigate the integration of SCADE applications with some relevant database.

ST04 and ST08 (from the Police Segment) were able:

- To implement a preliminary modeling of the RT-ACMIS project database; and also
- To implement an agile software testing, applying TDD techniques in a prototype of a Cockpit Display System (CDS), aiming: to access the project database, by using the SCADE Suite; to define coordinates (latitude and longitude) and displacements of Drones; and also to create mechanisms for modeling and data availability.

35.3.3.3 Sprint #2

The Sprint #2 implementation has happened on the RT-ACMIS project, as hardware embedded software integration between STs.

ST01 and ST05 (from the Civil Defense Segment) were able:

- To make available the basic information about accidents and/or crises occurrence registrations in the RT-ACMIS project database from data obtained at SCADE, in a safety and reliable way.

ST02 and ST06 (from the Health Segment) were able:

- To perform scripts integration created from Sprint#1 with the web portal and the RT-ACMIS project database;
- To start the embedding software development in hardware;
- To acquire victims' vital signals from ambulances; and also
- To receive and maintain historical data from victims' vital signs to support the software development.

ST03 and ST07 (from the Firefighter Segment) were able:

- To define and integrate their data into a single screen;
- To process RFID bracelet data for victims' classification;
- To develop a mobile application, using a Simple Screening And Rapid Treatment Plan (START) method [25, 26];
- To register victims in the project database, by using a Raspberry Pi device connected to an RFID reader;
- To define a firefighter architecture;
- To develop an Application Program Interface (API) in Node.js; and also
- To create a CDS, for displaying signals obtained from noise sensors used to rescue victims from earthquakes, landslides, and/or debris.

ST04 and ST08 (from the Police Segment) were able:

- To implement an infrastructure for collecting sensor's data (like gases, fuel, smoke, temperature, and/or humidity);
- To develop a CDS for sensor data visualization;
- To control the Drone flight with a claw;
- To develop the Drone user-control interface connected on real-time to a mobile device; and
- To operate the Drone claw for delivering small packages, medicines, and/or supplies in inhospitable areas.

35.3.3.4 Sprint #3

On this Sprint #3, both ST01 and ST05 (from the Civil Defense Segment) were able:

- To finish the prototype implementation of a web portal for real-time crises management;
- To include on this accident and/or crises portal occurrences or resources (sensors, ambulances, operation bases, hospitals, among others); and
- To control a web map, involving a real-time monitoring and processing of historical data from alerts and/or crises in progress.

ST02 and ST06, (from the Health Segment) were able to develop and delivery the following working functionalities for the RT-ACMIS project web portal:

- A registration for recording historical data of occurrences in the project database;
- A management equipment module, for listing operational ambulances and its integrated embedded software functionalities;
- An integrated process for ambulances to receive, on real-time, the geographical position of accidents or crises, involving victim's vital signs (like temperature and heart beating);
- The total number of available beds and victims received by each hospital and also death notifications and released beds for receiving accident and/or crises new victims; and
- For the RT-ACMIS project architecture, the following devices and functionalities: two sensors (for temperature and heart beats) connected to an Arduino device capturing sensors' data, using C code; data sent, via network, to a Raspberry Pi microcomputer, running a server with Flask framework, using Python, to program data received from the cloud used by ST06—Hospitals, and to program data sent to the ambulances integrated screen, displaying on real-time, the embedded data, by using SCADE.

ST03 and ST07 (from the Firefighter Segment) were able:

- To implement a mobile application for the START method: by integrating ST03 with ST02, providing victims' data; by integrating ST03 with ST04, providing sensor data (light, gas, ethanol, and smoke) to aftermath functionalities; and also by integrating data on the RT-ACMIS project web portal; and
- To demonstrate the Raspberry Pi, as a PoC, for reading signals from noise sensors: by integrating with ST02 for reading vital signals and assigning victims with RFID bracelets.

ST04 and ST08 (from the Police Segment) were able:

- To implement the CDS and perform integration with hardware created from Sprint#2 for the persistent data collected from database and available to the firefighter segment; and also
- To implement a police report for crises monitoring, through the RT-ACMIS project web portal, integrating with all other project segments, the images taken by the Drone.

35.4 Conclusion

The goal of this paper was to report the agile conceptualization, design, build, and demonstration of a computerized system, containing correct-by-construction software, for safely managing critical information, during alerts or crises situations.

The implemented scenario has allowed students from 8 (eight) different Scrum Teams (STs) assigned to the 4 (four) segments of CIVIL DEFENSE, HEALTH CARE, FIRE DEPARTMENT, and POLICE DEPARTMENT of the Real-Time Accident and Crises Management Integrated System (RT-ACMIS) project to prove that: it is possible to use the two approaches, Agile Development and Formal Methods; and also it is possible to state that both approaches are able to be effectively applied to mitigate risks and/or increase community resilience.

The following challenges and requirements were successfully tackled on this research: formal specifications, aerospatial-level reliability, agile development, embedded systems, controlled testability, and product assessment.

An Interdisciplinary Problem-Based Learning (IPBL), involving a Scrum of Scrums Agile Framework was adapted for managing the cohesive, productive, and collaborative development team of around 100 undergrad and graduate students remotely working.

In addition, the following hardware technologies, for supporting the software development were used: environment sensors, Radio Frequency Identification (RFID), and Unmanned Aerial Vehicles (UAVs) as a Drone, together with other software technologies like cloud-based web-responsive platforms and mobile applications to geographically manage resources on real-time.

Finally, the ANSYS® SCADE (Safety-Critical Application Development Environment) was employed to support the embedded and correct-by-construction module of this system, according to Model-Driven Architecture (MDA) and Model-Driven Development (MDD).

The authors recommend that those implemented elements associated to different public agencies efforts be used to improve and speed up service quality on attendance to accidents and crises services in Brazil, thereby optimizing existing resources and contributing for saving lives.

35.5 Future Works

As a natural continuation of this research and due to its importance on the global context, the authors of this paper suggest the following works for further research, involving the expansion of the RT-ACMIS project:

- Its use in more complex scenarios of accident and crises management, incorporating new technologies;
- The use of new and emergent technologies to identify and predict accidents, disasters and/or crises, right after occurrences applying appropriate integrated actions like some of those already implemented; and
- Finally, the use the ANSYS® SCADE together with Model-Driven Architecture (MDA) and Model-Driven Development (MDD) to be expanded and applied to support the embedded and correct-by-construction modules of systems also in other knowledge domains.

Acknowledgment The authors would like to thank: the Brazilian Aeronautics Institute of Technology (*Instituto Tecnologico de Aeronautica*—ITA); the Casimiro Montenegro Filho Foundation (*Fundacao Casimiro Montenegro Filho*—FCMF); the Ecossistema Digital Business Ltd.; and the 2RP Net Ltd, for their infrastructure and financial support to the development of this research project, allowing its PoC in an academic and simulated real environment.

References

1. United Nations Office for Disaster Risk Reduction (UNISDR) Terminology. https://www.unisdr.org/we/inform/terminology. Accessed 18 Aug 2017
2. Swiss Re Institute, Natural catastrophes and man-made disasters in 2016: a year of widespread damages. http://www.preventionweb.net/publications/view/52534. Accessed 26 Aug 2017
3. UNISDR, International Strategy for Disaster Reduction. https://www.unisdr.org/who-we-are/international-strategy-for-disaster-reduction. Accessed 12 Nov 2016
4. UNISDR, Technology: the future of disaster risk reduction?. https://www.unisdr.org/archive/51043. Accessed 10 Jan 2017
5. Federal Communications Commission, April 2014 Multistate 911 outage: cause and impact. https://apps.fcc.gov/edocs_public/attachmatch/DOC-330012A1.pdf. Accessed 10 Aug 2017
6. W. Sunne, L. Hovmarken, Scrum goes formal: agile methods for safety-critical systems, in *Proceedings of the First International Workshop on Formal Methods in Software Engineering: Rigorous and Agile Approaches* (FormSERA, Zurich, Switzerland, 2012)
7. K.S. Rubin, *Essential SCRUM: A Practical Guide to the Most Popular Agile Process* (Addison-Wesley, New York, 2013)
8. J. Sutherland, K. Schwaber, The Definitive Guide to Scrum: The Rules of the Game. http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf. Accessed 18 Mar 2016
9. RTCA DO-178C, *Software Considerations in Airborne Systems and Equipment Certification* (Radio Technical Commission for Aeronautics (RTCA), Washington, DC, 2011)
10. RTCA, *DO-278A. Software Integrity Assurance Considerations for Communication, Navigation, Surveillance and Air Traffic Management (CNS/ATM) Systems* (RTCA, Washington, DC, 2011)
11. Esterel Technologies Automatic Code Generation. http://www.ansys.com/products/embedded-software/ansys-scade-suite/scade-suite-capabilities#cap6. Accessed 20 Dec 2017
12. L. Rierson, *Developing Safety-Critical Software: A Practical Guide for Aviation Software and DO-178C Compliance* (CRC Press, New York, 2013)
13. T. Stober, U. Hansmann, *Agile Software Development Best Practices for Large Software Development Projects* (Springer, Heidelberg, 2010)
14. R.S. Pressman, *Software Engineering: A Practitioners Approach* (McGraw-Hill, New York, 1997)
15. L. Copeland, *A Practitioner's Guide to Software Test Design* (Artech House Publishers, Norwood, 2007)
16. L. Crispin, J. Gregory, *More Agile Testing* (Addison-Wesley, New York, 2015)
17. P. Jorgensen, C. Software, *Testing—A Craftsman's Approach* (CRC Press, Boca Raton, 2014)
18. G. Goncalves, et al., An agile developed interdisciplinary approach for safety-critical embedded system, in *14th International Conference on Information Technology: New Generations, vol 2017* (ITNG, Las Vegas, 2017)
19. D. Astels, *Test-Driven Development: A Pratical Guide* (Prentice Hall, Upper Saddle River, 2003)
20. K. Beck, *Test-Driven Development by Example* (Addison-Wesley, New York, 2002)
21. I. Sommerville, *Software Engineering*, 9th edn. (Addison-Wesley, Harlow, 2010)
22. J. Martins, et al., Agile testing quadrants on problem-based learning involving agile development, big data anda cloud computing, in *14th International Conference on Information Technology: New Generations (ITNG 2017)*, (Las Vegas, NV, 2017)
23. G. Berry, The foundations of Esterel, in *Proof, Language and Interaction: Essays in Honour of Robin Milner, Foundations of Computing Series*, ed. By G. Plotkin, C. Stirling, M. Tofte, (MIT Press, Cambridge, 2000)
24. Esterel Technologies. http://www.esterel-technologies.com/products/scade-arinc-661/. Accessed 26 Mar 2016
25. Esterel Technologies "SCADE Suite". http://www.esterel-technologies.com/products/scade-suite/. Accessed 22 Mar 2017
26. G. Super, S. Groth, R. Hook, et al., *START: Simple Triage and Rapid Treatment Plan* (Hoag Memorial Presbyterian Hospital, Newport Beach, 1994)