

# A New Agile Paradigm for Mission-Critical Software Development

**Angelo Messina, Defence & Security Software Engineering Association - DSSEA**

**Franco Fiore, NATO Communication and Information Agency, Directorate Application Services**

**Mario Ruggiero, Defence & Security Software Engineering Association - DSSEA**

**Paolo Ciancarini, Department of Computer Science & Engineering - DISI, University of Bologna & CINI**

**Daniel Russo, Department of Computer Science & Engineering - DISI, University of Bologna & CINI**

**Abstract.** The Agile paradigm, as intended in the 2001 “Agile Manifesto,” brought a disruptive software development methodology. However, with regard to mission- and security-critical organizations, traditional Agile methodologies are quite ineffective because they do not clearly address issues of (1) quality and (2) security. Within the Italian Army General Staff Logistic Department, a new Agile methodology was introduced to tackle quality and security issues in a classified and mission-critical context.

## I. Introduction

Continual evolution of the operational environment generates instability of the command and control (C2) systems requirements, obliging developers to work with unstable and unconsolidated mission needs. NATO’s new resolute support (RS) mission in Afghanistan is focusing on the training and advising of the Afghan National Defense and Security Forces (ANDSF), introducing a new dimension that transcends the canonical range of military operations. In 2013, the Italian Army General Staff Logistic Department decided to overcome the problem of the “volatile requirement,” transitioning to a completely different software development methodology derived from the commercial sector but almost completely new to mission-critical software applications: the so-called “Agile” methodology.

The introduction of Agile in the development of high-reliability software was not easy and required the generation of a brand-new Agile methodology called “Italian Army Agile” or “ITA2.” Setting up the LC2Evo (the evolution software of the land C2) required the solution of many problems and the construction of a solid structure based on four principles: user community governance, specific Agile training, new Agile CASE tools and custom Agile development doctrine.

This paper gives two major contributions to the community. First, it is an experience report of the Italian Army C2 system in military scenarios. Second, we outline the new Agile methodology introduced within the Italian Army: iAgile. This paper is structured as follows: Section II gives a major insight about the operational scenario and requirements management. Section III shows briefly the Italian LC2Evo model. In Section IV the new iAgile model is presented with some cost reduction evaluations. Finally, in Section V, we conclude our paper with the presentation of further works.

## II. Asymmetric Operations and Volatile Requirements

To run an analysis on the evolution of the C2 requirements, reference is taken to the operation ISAF (Afghanistan 2003 to 2014), in which Italy took part almost since the beginning, and its natural evolution, the operation “resolute support” (RS) (Afghanistan from Jan. 1, 2015 to present). As part of the operation, Italian forces have contributed to the NATO force in Afghanistan and to the Provincial Reconstruction Team.

The initial mission was relatively limited and included providing security for Kabul and its surrounding areas. But as the conflict in Afghanistan continued, more nations, international organizations and non-governmental organizations (NGOs) began various assistance efforts in Afghanistan, and ISAF’s mission scope was expanded (UNSCR 1510). In 2003, when Italy joined the operation, ISAF’s mandate had been expanded to the entirety

of Afghanistan. Many of the assignments of the force on the ground were different from traditional military operations, and so were the connected mission needs.

The first relevant transition in the C2 architecture was caused by NATO's assumption of leadership. Ideally, the coalition network was supposed to transition to a full NATO network, which was almost non-existent at the time. The second relevant change was the geographic dispersion of the forces. At the end of 2006, ISAF expanded its bases of operation throughout Afghanistan, covering the whole territory. Increasing the geographical footprint and adding more members and more non-military organizations made operations radically more complex and unconventional.

The number and type of potential customers of the C2 systems significantly changed as well. Initially, ISAF consisted of roughly 5,000 troops concentrated near Kabul. By 2010, ISAF consisted of well over 100,000 troops from 48 different countries, including NATO, NGOs and Afghan partner institutions. The third relevant change was the "advise and assist" (AA) function, exerted at the regional and central levels, starting at the beginning of 2015. In particular, the AA support at the security institutions level was a new challenge in terms of C2 support. A new specific tool following the Mission Thread approach is in development. Functional military areas to cope with are mainly humanitarian assistance, stability operations, counterinsurgency operations and combat operations.

It is now clear that mission needs are largely unpredictable while the mission support system (hardware and networks) are, for the most part, substantially the same throughout the mission. The volatility of the mission requirements has to be mitigated by the flexibility of an innovative software development process capable of continuity of change. The original concept of network-centric warfare (NCW), ready to deploy in all situations, has been integrated by the "mission-oriented approach" based on mission threads.

### III. The Functional Area Service Approach and LC2EVO

The doctrinal reference point to start building an evolutionary C2 system was the mission thread-based approach adopted by NATO with regard to quality and security issues.

#### A. Scrum is Not Enough

In 2014, the Italian Army General Staff Logistic Department began the development of the Land Command and Control EVOLution system. The product major item is software, while most of the hardware systems are the same supporting standard systems with the addition of some COTS. The main reason for this engagement was the need to support the evolution of the land C2, keeping high customer satisfaction in a volatile requirement situation in a mission-critical context. Another major issue to be solved was the need to substantially reduce the budget necessary for this software development and subsequent maintenance.

The Army software engineers experimented with the principal Agile software development methodology available at the time, "Scrum Agile." This methodology is very successful in commercial environments, where it is the method of choice for a majority of software applications producers, especially Android- and Linux-based products. LC2Evo started by using Scrum Agile with production cycles of three weeks and one experimental "Scrum team" with programmers, subject matter experts, security specialists, a Scrum master and a product owner. The team was composed of both developers from the industry and military people and was based at the Army staff major facility. [1] The initial phase of production was extremely successful, and even the very first sprint (production cycle of three weeks), that was supposed to be only a trial, actually delivered the planned product. While the product became more and more complex and the stakeholders' expectations grew, it became clear that

## CALL FOR ARTICLES

If your experience or research has produced information that could be useful to others, **CROSSTALK** can get the word out. We are specifically looking for articles on software-related topics to supplement upcoming theme issues. Below is the submittal schedule for the areas of emphasis we are looking for:

**Operations & Maintenance**  
*May/June 2017 Issue*  
Submission Deadline: Dec 10, 2016

**Model Based Testing**  
*July/August 2017 Issue*  
Submission Deadline: Feb 10, 2017

**Software Release Management**  
*September/October 2017 Issue*  
Submission Deadline: Apr 10, 2017

Please follow the Author Guidelines for **CROSSTALK**, available on the Internet at [www.crosstalkonline.org/submission-guidelines](http://www.crosstalkonline.org/submission-guidelines). We accept article submissions on software-related topics at any time, along with Letters to the Editor and BackTalk. To see a list of themes for upcoming issues or to learn more about the types of articles we're looking for visit [www.crosstalkonline.org/theme-calendar](http://www.crosstalkonline.org/theme-calendar).

commercial scrum methodology was not capable of handling the peculiarity of a high-reliability software production with an articulated user community as the one in charge of the land C2 operational requirement. [2]

The Army designed a particular Agile software development process called "Italian Army Agile" (ITA)<sup>2</sup> and tested it in the LC2Evo production. This methodology is currently shared by people with a broad community of interests, including those in defence industries, people from universities, and software engineers taking part in the DSSEA (Defence & Security Software Engineers Association).

### B. LC2Evo's FAS: A Direct Implementation of iAgile

LC2Evo is based on "core services" and "functional area services" (FAS). Unlike systems in the past, the core services are web-based. FAS are derived by the "mission threads" definition of the ISAF concept of operations and have been adopted by NATO. Both the core services and the individual FAS software components can be separately changed to accommodate particular mission needs defined by the user. At the same time, all the FAS can share the data and the artifacts developed individually, maximizing code reuse.

The LC2Evo has been tested in a NATO exercise for the first time at CWIX 2015 ([www.act.nato.int/cwix](http://www.act.nato.int/cwix)) with very favorable results. The FAS are under continuous development processes. Some of the most significant are:

**Battle Space Management.** Originally designed to provide C2 support for homeland security operations such as "Strade Sicure," battle space management is capable of tracking friendly secure mobile units on various types of cartography. It implements voice and messaging capability and has an integrated NATO JOCWHATCH feature. Most of the code and functions realized for this FAS have been reused for the LC2Evo-Infrastructure FAS, [3] which provides an extensive and detailed set of functionalities needed to manage Army real estate.

**Joint ISR FAS.** This provides management and analysis functions for the intelligence preparation of the battlefield.

**Joint Fires and Targeting FAS.** This supports all the coordination and planning activities related to fire power support, including all available effectors.

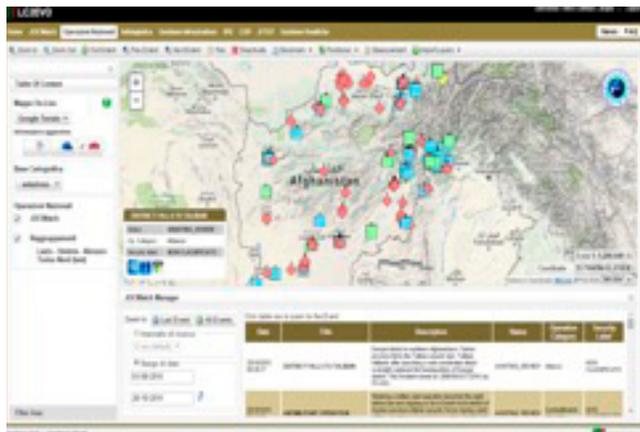


Figure 1. The Battle Space Management: The Events, Simulated Screenshot from Afghanistan

**Military Engineer and Counter IED FAS.** This was initially designed to support the collection and management of data about unexploded ordnances (UnExO) from World War II found on the national territory, but was soon implemented to provide support to the Counter IED operations (attack the net and force protection).

To comply with SIP's (Software Intensive Programs - DoD 5000.02) performance principles of NATO, two major Agile principles have to be followed:

- Deliver early and often. This principle is aimed at changing the culture from one that is focused typically on a single delivery at the end of the development phase to a new model with multiple deliveries during development, leading to an ultimate version that supports the full set of requirements supported by the DevOps approach. [4]
- Need for incremental and iterative development and testing. This principle embraces the concept that incremental and iterative development and testing, including the use of prototyping, yields better outcomes than those resulting from trying to deploy large, complex IT network systems in one "big bang."

These two principles tell us that the old-fashioned waterfall approach in which the customer, after months of software development, is given a release he or she may not be happy with, needs to be replaced by more modern and innovative software engineering techniques and methodologies.

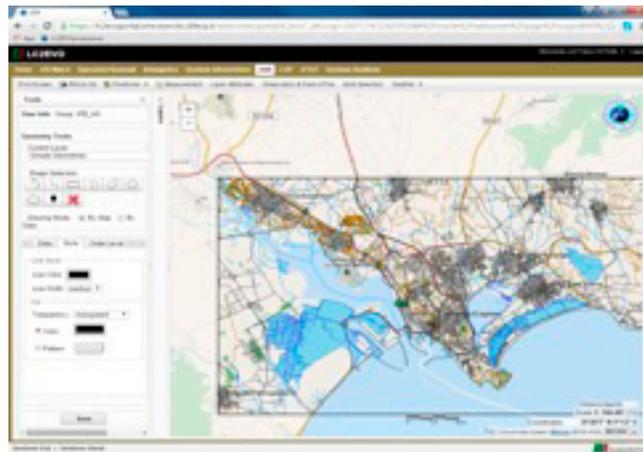


Figure 2. Joint ISR: A Simulated Screenshot of the Intelligence Preparation of the Battlefield

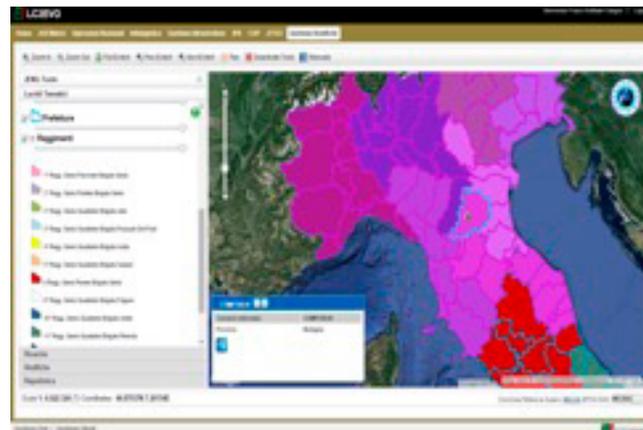


Figure 3. Military Engineer and Counter IED: Simulate Screenshot of the UnExO Areas of Interest.

After two years in the process, the Italian Army General Staff has clearly demonstrated the effectiveness of the new software development methodology, realizing the LC2Evo Command and Control software. The product is a continual development effort that produces new segments every five weeks. The first FAS of the product, the LC2Evo-infrastructure, which was published online in June 2015, serves more than 1,000 users daily and has registered customer satisfaction levels close to 100 percent.

### IV. Toward iAgile

The transition to Agile was not only needed to accommodate quicker adaptation to dynamic mission needs changes and quality and security needs but was also mandated by a drastic reduction in the defense budgets experienced in many NATO countries, particularly Italy.

#### A. Building the Four Pillars

Most of the effort to generate an adequate production structure for the LC2Evo has been devolved to the creation of an innovative cultural and technical environment. Most of the difficulties found during this innovation process were human-based, essentially due to cultural resistance based on consolidated practices. An entire brand-new environment had to be built. The four pillars of this innovative software engineering paradigm are:

- User community governance.
- Innovative Agile training.
- Innovative CASE tools.
- High-reliability Agile doctrine.

#### User Community Governance Pillar

This is of paramount importance and can be considered a prerequisite for the entire development process. In the area of land command and control, the number and articulation of the reference stakeholders and users is huge. Functions such as the "third dimension control" may have multiple stakeholders and users at the same time — for example, artillery might be using the 3D space to plan its firepower delivery while the same space is used by the Army Light Aviation and the Air Force in joint operations. This situation makes it necessary to rationalize the requirements management. The Army general staff has dedicated a huge effort to creating the coordination lines and the permanent structure to allow an orderly fashion collection of user needs and to provide the availability of subject matter experts to be placed on the development teams. Ad hoc social networks have been designed for this scope.

As stated before, the Agile training easily available from the market was not able not provide the particular skills needed to work in the mixed military and industry multidisciplinary teams, and the traditional roles described by the Scrum doctrine, such as the product owner and scrum master, had to be modified to be able to perform in the Italian Army Agile methodology. Within NATO partners, DSSEA is carrying out new training courses to match such specific needs.

#### Innovative CASE Tools

Replacing the traditional and CASE tools poses a difficult challenge: keeping the momentum of the Agile innovation while implementing new concepts for designing the high-reliability-related software development environments. The core of the Agile methods is the human element, which is positioned at the center of the development process again, using the brain's non-linear capability to overcome the difficulties related to user requirement incompleteness, volatility and redundancy.

Agile methods, properly implemented, can take care of a significant part of this problem by capturing user needs in lists of short user stories and then giving the user the working segments of the product after a few weeks or even days. This way, part of the non-linearity of the requirement conceptual design is overcome by the interaction between humans — the software developer is directly assisted by the user, and they essentially design the application together. In the process, the two different complex 3D representations of the application (run time) imagined by the mind of the user and the one detailed by the mind of the software developer tend to converge. This method also reduces the number of translations needed to convert the requirements into coding tasks, significantly decreasing the loss of relevant information [5].

#### High-Reliability Agile Doctrine

The last pillar regards the definition of a doctrine as a set of rules and procedures encompassing all the needed practices and artifacts to be used in the implementation of a new Agile methodology. DoD Instruction 5000.02 (Dec. 2013) heavily emphasizes tailoring program structures and acquisition processes to the program characteristics. Agile development can achieve these objectives through:

- Focusing on small, frequent capability releases.
- Valuing working software over comprehensive documentation.
- Responding rapidly to changes in operations, technology and budgets.
- Actively involving users throughout development to ensure high operational value.

These indications are a clear encouragement to use Agile practices to integrate planning, design, development and testing into an iterative life cycle to deliver software at frequent intervals.

Moreover, assuring high code quality for mission-critical applications is a core mission of iAgile. There are two great and opposite tensions for delivering software in such a volatile operational scenario: reliability and velocity. Both are crucial and, apparently, diametrically opposite. The characteristic of iAgile is that the most effort and focus is put on the development. Short and focused sprints are able to provide both velocity, due to time boxing, and reliability, since developers put the most attention into developing high-quality code using reliable and known libraries. Documentation and maintenance efforts are minimal. Development itself is boosted by redundancy by people with different professional expertises. Oriented pair programming in such contexts is useful when one of the components of the pair is a software security expert or a mission-specific application expert. Testing in iAgile has to be performed continually with a Test Driven Development (TDD).

The apparent redundancy of resources does not impact the production effectiveness because of the dramatic reduction of the code rework activity due to errors.

Security is enhanced by both methodology and redundancy. Since the code developed is supervised by a security expert through pair programming, leaks and bugs are fixed before testing. However, penetration testing is used extensively as TDD methodology for security to reveal leaks not fixed in the development phase.

From a quality and security point of view, there is a paradigm shift from “deliver and maintain” to “continuous development.”

### B. Waterfall vs. Agile: Cost Reduction

The Agile Manifesto in 2001 changed the focus of software engineering, putting the programmers at the center of the production line and creating a straight communication line between the customer (in charge of the requirement) and the developers, minimizing the need for formal documentation. [7] Within the iAgile methodology, every single step of the procedure is monitored, tracked and evaluated by the developers themselves.

An initial internal assessment of LC2Evo product cost per line of code equivalent with respect to other comparable internally produced software showed a cost reduction of 50 percent. To consider those costs, we computed comparable software by dimension (LOC) and functional area (command and control). We considered all relative costs of personnel, documentation and maintenance costs, and fixed costs for office utilities. The assessment after two years showed more significant cost reduction.

Generally speaking, we know from literature that, on average, cost per ELOC in military domains is about \$145 and, with regard to ground operation, the cost is about \$90. [8] This study in particular was carried out in a “waterfall” or procedural context. Based on Reiter’s study, we carried out our evaluation regarding iAgile’s cost. It was quite surprising to realize that the measured software LC2Evo had an average cost per ELOC of \$10. This was possible because of the decreased maintenance and documentation costs, which represent the most relevant parts of software development cost. [9]

### V. Conclusions

The Italian Army experience in developing the command and control software LC2Evo confirms that the major problem in dealing with complex scenarios and rapidly evolving user needs is the management and evolution of the user requirements with both high quality and security standards. The linear development cycles focus on the production process once the requirement is consolidated, but even in the defense and security software applications areas, this is no longer possible. Similar to applications in the commercial sector, military applications experience a quickly changing operational environment. The high reliability, security and quality of the product cannot be pursued with procedures and agents working outside the development cycle, but they have to be embedded in the production cycle.

Standard Agile methodologies seem not to comply with a mission-critical context with high quality and security requirements. Within the Italian Army, a new Agile paradigm that addresses these issues has been developing.

Future work will focus both on the development of a theoretical model and on validation in related domains (e.g., banking sector) with the development of dedicated tools.

### Acknowledgements

The authors wish to thank the Italian Army General Staff Logistic Department for the availability of data and the courtesy of some pictures. We also thank the Consorzio Interuniversitario Nazionale per l’informatica (CINI) for the financial support. Special thanks go to Major General Francesco Figliuolo and Col. Franco Cotugno for their support.

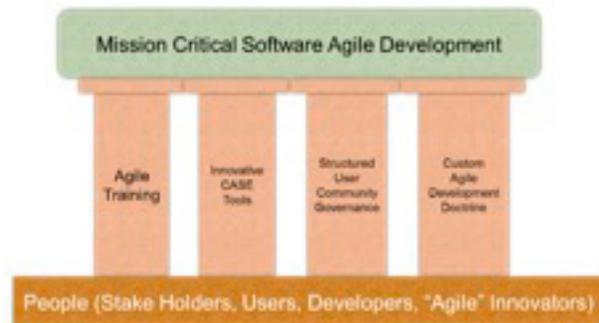


Figure 4. The Four Pillars of Innovation.

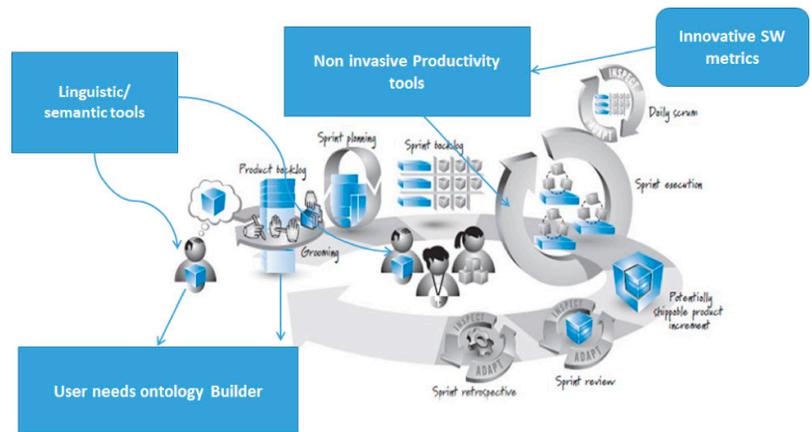


Figure 5. iAgile’s Graphic Representation, inspired by [6].

## ABOUT THE AUTHORS



**Angelo Messina** B. General (ret) Ita- Army received a MD Electronic Engineering degree at the Politecnico of Turin and a doctorate degree in non-com, electronics and missile systems. He also attended the Army General Staff Course and Joint General Staff Course. He is secretary of the Defence and Security Software Engineers Association (DSSEA) and an independent consultant in Agile Software Engineering for NATO NCIA. He also serves as senior scientist at CASE Research Ltd. He was previously appointed as Deputy Chief 4th Dep. Army General Staff and head of the 1st Department of the Land Armaments Directorate. In his career, he also served as the European Defence Agency R&T deputy director.

**Via A. Bertoloni, 1/E, 00197 – Rome**  
**angelo.messina@esercito.difesa.it**



**Franco Fiore** has been serving as the NATO Communication and Information Systems Agency Service Support and Business Application Service Line since March 2016. As former Italian Army Corps Engineer (EW and Communications), Dr. Fiore spent four years in the USA serving at the NATO Medium Extended Air Defense System Management Agency (NAMEADSMA) as sensor simulation engineer, and joined NC3A (now NCI Agency) in January 2005. He holds master's degrees in computer engineering and complex electronic systems and a Ph.D in telecommunications and electronics.

**The Hague, Netherlands**  
**franco.fiore@ncia.nato.int**



**Mario Ruggiero** MG (ret) Ita- Army received a degree in applied military strategic sciences from the University of Turin. After that, he attended the Army General Staff Course, Joint General Staff Course and French Joint Staff Course. He is an active member of the Defence and Security Software Engineers Association (DSSEA). He served as Deputy Chief of Staff Support– Resolute Support Mission (AFGH). He was also chief of the 4th Dep. Army General Staff and the director of the Italian Center for Defense Innovation. In his career, he was also appointed as Chief General Planning Branch at Defense General Staff.

**Via A. Bertoloni, 1/E, 00197 – Rome**  
**mario.ruggiero@esercito.difesa.it**



**Paolo Ciancarini** is full professor of computer science at the University of Bologna. He is currently the president of the Italian Association of University Professors in Computer Science.

He is also the vice-director of CINI (National Inter-University Consortium for Informatics). His research focus is on software engineering. He is the author of more than 120 scientific published scientific articles.

**Mura Anteo Zamboni, 7, 40126 – Bologna**  
**paolo.ciancarini@unibo.it**



**Daniel Russo** is a research fellow at the department of computer science and engineering at the University of Bologna and a research associate at the Consorzio Interuniversitario per l'Informatica (CINI) in Rome, currently pursuing a Ph.D. in computer science and engineering.

His main research interests include Agile software development methodologies in mission-critical environments and quality and security assurance in mission-critical software.

**Mura Anteo Zamboni, 7, 40126 – Bologna**  
**daniel.russo@unibo.it**  
**(Corresponding Author)**

*The Defence & Security Software Engineers Association is a nonprofit organization aimed at the development of a new software engineering paradigm. The association includes members from the defense and security areas as well as from universities and industry. ([www.dssea.eu](http://www.dssea.eu)).*

## REFERENCES

1. Messina, A. & Cotugno, F. (2014, May.) Adapting SCRUM to the Italian Army: Methods and (Open) Tools. The 10th International Conference on Open Source Systems. San Jose, Costa Rica.
2. Cotugno, F. & Messina, A. (2014, September.) Implementing SCRUM in the Army General Staff Environment. The 3rd International Conference on Software Engineering for Defence Applications – SEDA. Rome, Italy.
3. Ventrelli, C.; Trenta, D.; Dettori, D.; Sanzari, V. & Salomoni, S. (2015, May.) ITA Army Agile Software implementation of the LC2EVO Army Infrastructure strategic management tool. The 4th International Conference in Software Engineering for Defense Applications SEDA. Rome, Italy.
4. Russo, D. (2015, May.) Benefits of Open Source Software in Defense Environments. Advances in Intelligent Systems and Computing, Springer. The 4th International Conference in Software Engineering for Defense Applications SEDA. Rome, Italy.
5. Ciancarini, P., Sillitti, A., Succi, G., & Messina, A. (Eds.). (2016). Proceedings of 4th International Conference in Software Engineering for Defence Applications: SEDA 2015 (Vol. 422). Springer.
6. Rubin, K. S. (2012). Essential Scrum: A Practical Guide to the Most Popular Agile Process. Addison-Wesley.
7. Schwaber, K. (2004.) Agile Project Management with SCRUM. Microsoft Press. ISBN 9780735619937.
8. Reifer, D. J. (2004.) Industry Software Cost, Quality and Productivity Benchmarks. The DoD Software Tech, 7 (2), 3-8.
9. Pressman, R. S. (2009.) Software Engineering: A Practitioner's Approach, 7th Ed. McGraw Hill Int. Press.