Adapting Agile practices in globally distributed large scale software development

Mario Ivček
Research and Development Centre
Ericsson Nikola Tesla
Krapinska 45, 10 000 Zagreb, Croatia
Telefon: +38513654619 Fax: +38513653548 E-mail:
mario.ivcek@ericsson.com

Tihana Galinac
Faculty of Engineering
University of Rijeka.
Vukovarska 58, 51 000 Rijeka, Croatia
Telefon: +385 51 651 583 Fax: +385 51 675 818 E-mail:
tihana.galinac@riteh.hr

Abstract – Nowadays, in time of rapid software development and continuous change, the software development organizations have increasingly been adapting the Agile practices within their software development methodologies. This paper discusses experiences and observations of applying a set of selected Agile practices within globally distributed organization responsible for development of large scale software systems for evolving telecommunication products. The paper provides valuable material for practitioners who plan to adopt agile principles in similar environment conditions. Also we give possible direction for future research.

I. INTRODUCTION

Rapid environmental changes bring uncertainty into software development, introduce frequent changes, shorter time to market and frequent deliverables of new and enhanced functionalities of software product. To cope with these new conditions a group is formed called 'Agile Alliance' with revolutionary different development approach from those product and process oriented. The group has specified the main ideas for the forthcoming new era of software development and unified it under the common agile development method as follows, [1]:

- individuals and interactions over processes and tools,
- working software over comprehensive documentation,
- customer collaboration over contact negotiation,
- responding to change over following a plan.

Based on these overall ideas many different software development methods have been categorized as Agile such as eXtreme Programming (XP) [2], Crystal [3], Scrum [4] and others, all reviewed in [5]. Agile development methods were proven as good for rapid development environments with high change rate because they reduce risk and time to market, and still secure required quality levels. However, comparison of the resulting quality levels by use of Waterfall and Agile methods is difficult because their initial development conditions as stated in [6].

Even though the Agile methods were primarily designed for co-located small and medium size teams, there is increased research effort in the field of how to adopt them in large globally distributed environment (GSD). The study performed in [7] has identified numerous possible benefits of applying Agile methods and principles on GSD teams. Reported benefits are listed as follows:

- requirements prioritization is focused on efficiency and productivity, however, efficiently produced features that nobody needs is still waste according to [8],
- continuous integration involves more communication and coordination effort,
- short iterations and frequent build bring transparency into work progress to all partners involved into software development,
- focus on high quality work builds trust and respect,
- early involvement of customer allow frequent changes, early corrections of directions to desired point, better understanding of requirements, etc.

There are already many experience reports of implementing Agile practices into GSD as [9], [10], [11], [12], [13], [14]. The Agile practices that have been applied by most of these organizations are following: daily build, iterations, increments, frequent integration, system metaphor, scheduling according to feature priorities, feedback from expert users, pair development, self organized teams, etc. With increasing deployment of agile principles within GSD organizations there is increasing need for evidence about their applicability. Due to that reason a group of large companies have shared their experiences that are generalized as common experiences in [11].

Additionally to the experience reports on the applicability of Agile practices there are also reported experiences in improving the existing processes and implementing Agile practices. Experience described in [15] reports failure in implementing Agile practices and suggests team training, prepared infrastructure prior to actual feature development and team involvement into decision to adapt agile practices. On the other hand, [11] identifies process maturity as prerequisite for successful Agile implementation in GSD.

In this paper we report experiences of implementing selected Agile principles within an GSD industrial context. The rest of the paper is organized as follows: in next section we describe industrial context were the study was performed, as well as business drivers for introducing Agile practices. Selection of the best practices and improvement project that applied these practices is elaborated in section three. Then, in fourth section we discuss experiences in introducing Agile practices, and finally fifth section concludes the paper.

II. INDUSTRIAL CONTEXT AND BUSINESS DRIVERS

In this section we describe industrial context were the study was performed as well as business drivers for introducing Agile practices.

A. Industrial context

The study was performed within the Ericsson organization. The organizational unit is globally distributed having long tradition of working in global environment. The development department and integration & verification (I&V) department belong to separated organizational subunits and both are divided into several dislocated development sites such as Sweden, Italy, Croatia, Germany, China, Canada and even some external consultants. The organizational unit is responsible for large scale software (LSS) product used in number of telecommunication network solutions that consist of distributed network nodes, which have to harmonically serve billions of end users. The main goal of organization is to effectively and efficiently develop parts of LSS that should consistently behave as a whole within one network node and moreover, coordinated with other network nodes in particular network scenario.

The LSS product that the target organization is responsible for is characterized by more then thousand software components organized in modules with number of connections and interdependencies. The product has been evolutionary developed over the last 40 years. Currently, several hundreds of developers distributed across all over the world modify the LSS in distributed teams and several hundreds of integrators across the world integrate the LSS into number of different network scenarios.

Such distributed global development requires huge communication effort. Furthermore, development work gets even more complicated with high quality and performance requirements, mainly caused by the fact that the system has to work in real time conditions, simultaneously serving millions of subscribers.

The software development process that was traditionally used for development of this LSS product is an in house developed, mature sequential process that is very well established and documented. It is based on extensive usage of collaboration tools and well known by organizational personnel. Important is to mention that the target organization has a long experience of successful development of this target LSS product in GSD environment and has already implemented many of the improved methods, tools and techniques that enable effective and efficient software development.

B. Business drivers

The target organization has been faced with continuously growing market demands for faster, better, more efficient and more complex products delivery that continuously builds a rapid development environment. In this condition the organizations experience numerous problems. Development projects usually overlap each other on the same software base as is described in [16], thus implying increasing number of overlapped software product life cycles, and product versions. As a consequence the maintenance cost has been increased due to maintenance of several different versions of the same products, increased planning problems that have to avoid overlapping project issues and even more complicating communication, increased need for coordination, increased fault removal cost because every fault found in one project has to be mapped to all overlapping projects, increased product management complexity, increased product complexity, increased complexity of keeping right architecture modeling, increased need for special attention on dimensioning issues, backward compatibility issues, resource usage, uncertainty during whole project, frequent changes, customer trust, product quality issues, etc.

Besides already mentioned problems, the challenges of working in GSD organization [17] are communication and coordination. Additional challenges due to development of LSS product are explained in [18]: how to coordinate design capabilities, how to keep groups responsible for development, maintenance and evolution enough agile to respond effectively to changes, how to adopt the process to handle these changes, how to minimize effort needed to integrate components built independently, how to organize processes so they converge on high quality designs, how to control activities within the process to make elements working in harmony and to ensure fulfillment of objectives, how to maintain overall quality of service while enabling flexibility in providing different levels of service to different groups, etc.

Solution to these problems is seen through usage of evolutionary Agile ideas. The main opportunities for improvements applying Agile practices are following: incremental and iterative development allows parallel changes to many different systems that exist at the same time, much more testing is done 'in the field' and process of learning about effective solutions occurs through feedback from environment, the focus of engineering efforts is on change to small parts of the system rather than on change to the system as a whole, multiple small teams are involved in design and implementation of these changes.

III. AGILE PRACTICES AND IMPROVEMENT PROJECT

The selection of Agile practices offered by different research and industry studies that could add on business value has to be carefully planned. This is especially important in case of large GSD organization developing complex LSS, as in this study. The main goal before investing into improvement initiative is to address and carefully select required changes to the existing company standard process and to plan their incremental implementation into target software development project.

A. Agile practices

The key Agile practices are selected based on Ericsson development needs explained above, best

practices published by other researchers, Ericsson internal best practices, and Ericsson best practices published in [19] and [21].

The concept that has evolved over time based on experience from a number of organizations is founded on incremental system growth as oppose to big-bang integration, as illustrated in Fig 1.

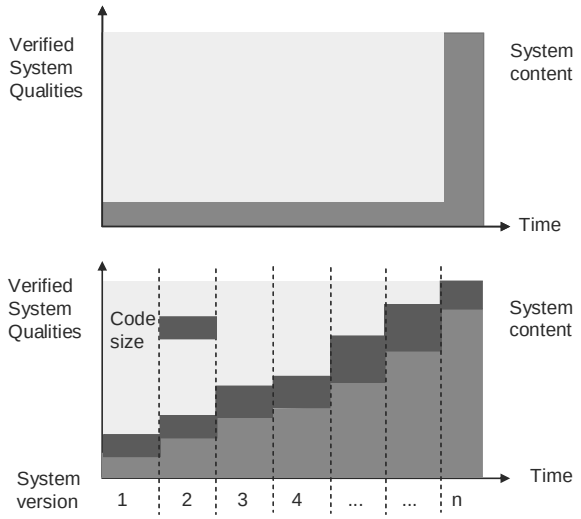

Figure 1. "Big-bang" vs. incremental system growth

Incremental system growth implies frequent small deliveries, continuous verification of each delivery before integration with rest of the system, and planning and development from an integration perspective.

1) Requirements: The requirements are prioritized prior of being inserted into the projects. Also, the project activities are committed according to organizational capability such as budget and available resources. In [20] it is stated that Agile development is more focused on developer's knowledge and due to that the requirements do not have to be so in detail. In case of GSD of LSS products the requirements are the main communicating tool, not only among GSD teams within the development project but also among GSD teams working on different product versions. The requirement details are needed not only for maintenance personnel but also for developers of future development iterations.

To achieve efficient management and control of requirements though whole LSS lifecycle the requirement handling tool is introduced. The tool ensures the global visibility of changes, tracking of requirements through product life cycle and incremental integration plan based on shared milestone concept to integrate and coordinate GSD development.

2) Project Planning and Monitoring: The first task of project planning is to identify pieces of system enhancements (called deltas „Δ") from a set of high level features and system requirements. The pieces are chosen with focus on system architecture rather than organizational aspects. Each piece is fully implemented and verified separately.

To find an optimal way of implementing new system changes, a project anatomy (also called delta anatomy) is used. The project anatomy is a center of planning cycle that shows all currently planned system changes and how they depend on each other (Fig 2). In that sense, the project planning is continuous planning of integration activities according to delta anatomy.

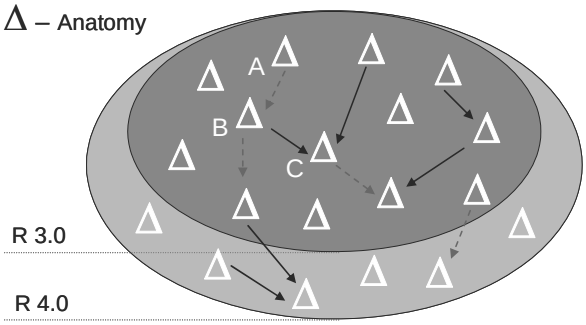

Figure 2. Project anatomy

The dependencies constrain the order in which changes can be done. Changes without dependencies, for example, can be developed in parallel with each other. The project has to plan small software deliveries and frequent inspections of project progress based on working software.

3) Software Design: The design is divided into verifiable system changes and behaves like a software factory that frequently delivers a new component tested system version, also called the Latest System Version (LSV). The system is developed in small steps according to Fig3.

In between the steps there are intermediate versions of the design base used for daily development work. The idea is to get early feedback by having the system subject for testing as quickly and often as possible. A new LSV can ultimately be available every three weeks.

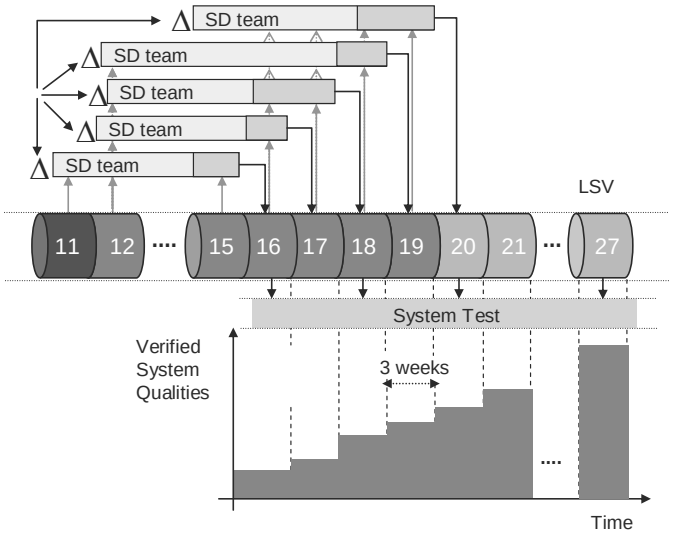

Figure 3. Agile development strategy

4) Testing and verification: It is proven that large changes that are verified late in the project, or not at all, leads to disaster. One of the strategies to attain good quality and delivery precision is to verify the system or product as early as possible, and before the system is integrated.

To obtain that, test activities are done in parallel so the regression test of existing functionality at verification level is run in parallel with function test activities.

The first quality check is done by design teams doing basic test on component level. Focus is on automatization of all new test cases enabling also automatic regression test in future. The second level of tests is then done by the integration and regression test team.

5) Software integration: Every week software delivery is integrated and regression tested on system level in the target environment. Regression test ensures that functionality that worked in the previous LSV is not broken. Weekly build and regression test of components is highly automated.

There is a difference in incremental and iterative development. The implementation in this study implies both but on different levels. Due to system complexity, the system is integrated according to incremental development taking several features into one increment. Also, the features that are not ready for that system release are delivered in parts so the backward compatibility and their influence on the existing already working system could be checked.

Unfortunately, the frequency of delivery is still not daily because of the problems already explained by [19]. The usage of weekly build instead of daily build technique is reported as more appropriate for similar industrial context in [9].

6) Teams: The Agile method does not require any special project or team organization. Teams can be organized as a cross-functional team or a component design team depending on the design environment, product complexity and product organization.

To avoid parallel work in the same software a component design team may be the best choice, where each component (e.g. block) in the system is owned by a designer who has the full responsibility for its construction and qualities. This is the most common way to team up in Agile project. It is also possible to have different sub-systems organized in different ways. The important thing is to have synchronization points at the weekly build for all teams.

B. Improvement project

In order to implement above explained practices within development organization, the improvement project has been defined. The main task of the project was to integrate Agile practices into existing company standard processes and guide GSD teams in their implementation. The improvement project consisted of improvement workgroups, each covering one process area. Following process areas were identified: requirement handling, integration strategy, project monitoring and control, reviews and inspections, software quality measurement. The improvement workgroups were organized as cross-organizational teams formed of local experts in their process areas. Each workgroup has defined its own leader and group of workgroup leaders were forming cross-process group that was responsible for alignment of activities within different process areas. The improvement time plan and budget were defined per each workgroup, monitored and controlled at regular steering meetings involving the key stakeholders, such are unit manager, software development project manager and improvement project manager.

IV. DISCUSSION AND EXPERIENCES

In this Section we elaborate experiences from applying selected Agile practices and discuss findings along with reviewing the existing literature.

According to [20], an organization should manage its processes in accordance to management process, establish the infrastructure under the processes according to the infrastructure process, provide training to its personnel according to the training process, and improve the processes following the improvement process. These processes are discussed in the following subsections.

A. Management process

In our experience, the organizational setup traditionally used for GSD development of LSS product has to change to accommodate needs of rapid development environment and selected improvement practices. According to our knowledge there is not so much said about the appropriate GSD organizational setup and division of work responsibility for implementing agile practices.

For gaining maximal efficiency and effectiveness in implementing improvement activities described above, the GSD organizational setup should be adequately adopted in order to secure maximal possible independence among development sites. This means not only division of work following the LSS product architecture but also definition of all involved stakeholders and definition of clear communication interfaces with level of information transferred. The influence of organizational issues (especially local site issues) on GSD should be minimized.

The most important aspect of organization is handling with its capability. The work prioritization is strongly dependent on resource availability, so when distributed teams develop the same LSS the most effective mechanisms for agreement should be secured. For the purpose of global resource management the proposal is to use collaboration tool. Also enhancements to project steering and control model where introduced compared to one presented [19] with additional more frequent check points named Go decisions included between so called Tollgates, which enable better and more efficient control.

B. Infrastructure process

There are many research studies on the topic of evaluating organizational readiness to implement agile principles based on the established infrastructure, as for example in [19]. The paper provides list of activities that an organization has to secure to implement daily build

practice. Additionally to the list we define the following: capability of configuration management process (especially handling with system versions), level of automate tests, human competence to develop backward compatible smaller system improvements (functional or nonfunctional) and processes supported by collaboration tools.

C. Training process

In the highly mature organizations that have evolved through decades dealing always with the same business it is natural to have human resistance to change. According to our experience the right setup of opinion makers across organizational units is essential. Managers and their in time training with strong involvement into improvement programs by responsibility and authority proved to be successful. Also, frequent discussions with all involved suspicious and worried personnel are of outmost importance. This includes finding out the real root cause of their worries, strong change management leadership by finding the quick and efficient solutions to problems and then presenting solutions to everybody or making them visible on company walls.

Agile practices require shift in the thinking of all involved within software life cycle such as customer, developer and manager. It is important to stress that the main changes are made by developers and these changes have many implications to their stakeholder. Also, having them as a part of prioritization team while organization has to balance its capabilities is not an easy task especially in large scale GSD development environment, when there are numerous customers of the same product.

D. Improvement process

The improvement project in this study followed the company defined improvement process with addition of the Agile main ideas specified in the introduction. This means following: constant integration of improvement work done in each process area and workgroup, frequent reviews by software developers thus securing frequent feedback and alignment to their needs, working in cross organizational and cross process teams, prototyping solutions whenever possible, introduction of simple and small changes to the existing standard company process. Main benefit of such improvement process is that enables continuous iterative improvement by constant collaboration of organizational and project levels.

Transition time while the changes are introduced is risky for the all ongoing regular development activities. Organization has to be aware that except all regular activities that are ongoing within one organization all work related to improvement has to be done and much energy must be spent on change management and regular training. Also reorganization according to new development philosophy in GSD organization requires significant time and energy. In our case one year of preparation and deployment is still not enough for fully applied selected agile practices. We believe that fully applying of Agile practices is the matter of practice gained through number of iterations applying them.

E. Methods and tools

Methods and processes that are new to already existing ways of working are best to be secured in advance, communicated prior to deployment, and training provided to personnel. Aligned with introducing the proposed methods, the needed tool support was secured. The main identified best practice is proactive collaboration of experts in the field (process practitioners) and tool development organization. Thus, all needs are communicated in the most efficient way for both tool and process developers.

During the deployment of new tool, the tool champions were identified per each development location. Since the organization where the Agile was introduced was already mature for GSD conditions there was a number of instruments that support such environment already in place.

Usage of resource management tool improves planning and managing organizational capability and was used within organizational management processes for requirement prioritization process. Anatomy planning tool was used for integration planning purposes that improves management of changes over project integration plan. Rational collaboration tools such as Clear case for code and product document storage and CDM for project documentation storage are company best practice already implemented that have just accelerated the improvement deployment. There are also collaboration tools that help in providing frequent feedback from all distributed personnel such as tools for fault reporting based on IBM Rational Clear Quest and MHWeb applications.

There are many tools that belong to group of tools for common development environment that are available to distributed teams. For implementing Agile practices especially important for suppressing the chaos within the GSD of LSS are following tools: tool for signal coordination (interfaces), tool for handling of system data that are commonly used by all software components, tool for handling of system commands for system operation and maintenance. Additionally we have introduced the tool for common configuration management that is in our case IBM product MARS which proves efficient in GSD environment. Thanks to these tools, introducing Agile does not grow into chaotic behavior as stated in [12].

Another important issue that we have to solve is to adopt the testing tools for its limited usage to parts of the system whose complexity is appropriate to the tool in use. We experienced the problems with in house developed tools and their adaptation. The organizational parts that have used globally available commercial tools had better capability to adapt to the agile principles.

V. CONCLUSION

The current environment conditions force software development organizations to continuous change without exception. The way of organizing and managing software development activities is evolving to more efficient and effective models. The common opinion is

that some software characteristics such as application, complexity, programming language, etc are the reasons for not evolving towards modern agile principles specified by manifesto. We believe that these principles are cornerstones for future software development and prerequisite for further evolution. We also believe that all software developments, no matter of its specifics, has to adapt to these principles. The implementation itself and methodology used for their implementation may however differ from application to application.

In this study we presented application of agile principles to telecommunication large scale software with very large installed design base, that is network-centric product with evolution history of more than 30 years and very large mature global software development organization, which continuously adapts its product following very well documented and defined sequential processes. All these specifics do not prevent implementation of agile principles, but of course their interpretation and implementation is specific.

The success of implementation of principles specified in [19] and [21] has been already proved within the Ericsson company and is applying across whole Ericsson organization as best practice. Even more, the implementation of these principles have pushed further improvements that are connected to tool supported usage of these principles and for improving collaborated agile work in GSD organization. So, the main contribution we identified with this paper is presenting Ericsson best practice in adopting Agile principles and the experiences gained.

REFERENCES

[1] K. Beck, M. Beedle, V.A. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, et. al., *Manifesto for Agile development*, www.AgileManifesto.org

[2] K. Beck, *Extreme Programming Explained: Embrace Change*, Addison-Wesley, Boston, 1999.

[3] A. Cockburn, *Agile software Development*, Addison-Wesley, Boston, 2001.

[4] K. Schwaber, M. Beedle, *Agile Software Development with Scrum*, Prentice Hall PTR, New York, 2002.

[5] P. Abrahamsson, O. Solo, J. Ronkainen, J. Warstra, *Agile software development methods: Review and Analysis*, VTT Publications 478, Oulu, 2002.

[6] M. Huo, J. Verner, L. Zhu, M. A. Babar, *Software Quality and Agile Methods*, Proc. of the 28th Annual International Computer Software and Applications Conference (COMPSAC'04).

[7] M. Paasivaara, C. Lassenius, *Could Global Software Development Benefit from Agile Methods?*, Proc of the International Conference on Global Software Engineering, 2006. ICGSE '06. pp. 109-113.

[8] C. Rand, B. Eckfeldt, *Aligning Strategic Planning with Agile Development: Extending Agile Thinking to Business Improvement*, Proc. of the Agile Development Conference (ADC'04), pp. 78-82, IEEE Computer Society, Washington, DC, USA.

[9] M. Leszak, M. Meier, *Successful Global Development of a Large-scale Embedded Telecommunications Product*, Proceedings of the International Conference on Global Software Engineering (ICGSE 2007), pp. 23-32, IEEE Computer Society, Washington, DC, USA.

[10] C. Sepulveda, *Agile Development and Remote Teams: Learning to Love the Phone*, Proc. of the Conference on Agile Development, pp. 140, IEEE Computer Society, Washington, DC, USA.

[11] M. Lindvall, D. Muthig, A. Dagnino, C. Wallin, M. Stupperich, D. Kiefer, J. May, T. Kahkonen, *Agile Software Development in Large Organizations*, Computer, pp. 26-34, IEEE Computer Society Press, Los Alamitos, CA, USA.

[12] T. Kähkönen, *Agile Methods for Large Organizations - Building Communities of Practice*, Computer, Proc. of the Agile Development Conference (ADC'04), pp. 2-11, IEEE Computer Society, Los Alamitos, CA, USA.

[13] P. Manhart, K. Schneider, *Breaking the Ice for Agile Development of Embedded Software: An Industry Experience Report*, Proc. of the 26th Int. Conference on Software Engineering (ICSE'04), pp. 378-386, IEEE Computer Society, Washington DC, USA.

[14] B. S. Boelsterli, *Iteration Advocate/Iteration Transition Meeting: Small Sampling of New agile Techniques Used at a Major Telecommunications Firm*, Proc. of the Conference on Agile Development, pp. 109, IEEE Computer Society, Washington, DC, USA.

[15] P. Hodgetts, *Refactoring the Development Process: Experiences with the Incremental Adoption of Agile Practices*, Proc. of the Conference on Agile Development (ADC'04), pp. 106-113, IEEE Computer Society, Washington, DC, USA.

[16] T. Galinac and S. Golubić, *Project Overlapping and Its Influence on the Product Quality*, Proc. of the 8th International Conference on Telecommunications, pp. 655-662, IEEE.

[17] J. D. Herbsleb, "*Global Software Engineering: The Future of Socio-technical Coordination*", Future of Software Engineering, International Conference on Software Engineering, IEEE Computer Society, Washington, DC, p. 188-198.

[18] L. Northrop, P. Feiler, R. P. Gabriel, J. Goodenough, R. Linger, T. Longstaff, R. Kazman, M. Klein, D. Schmidt, K. Sullivan, and K.Wallnau, *Ultra-Large-Scale Systems: The Software Challenge of the Future*, Software Engineering Institute, Carnegie Mellon University, 2006.

[19] E.A. Karlsson, L. G. Andersson, P. Leion, *Daily build and feature development in large distributed projects*, Proc. of the 22nd international conference on Software engineering (ICSE'00), pp. 649—658, ACM, New York, NY, USA.

[20] F. Paetsch, A. Eberlein, F. Maurer, *Requirements Engineering and Agile Software Development*, Proc. of the Twelfth International Workshop on Enabling Technologies (WETICE '03), pp. 308, IEEE Computer Society, Washington, DC, USA.

[21] T. Galinac, Ž. Car, *Software Verification Process Improvement Proposal Using Six Sigma*, Lecture Notes in Computer Science Vol. 4589 (2007), pp. 51-64.