Integrating agile practices into critical software development

Katarzyna Łukasiewicz, Janusz Górski

Integrating agile practices into critical software development

Katarzyna Łukasiewicz, Janusz Górski
Gdańsk University of Technology
Gdańsk, Poland
{katarzyna.lukasiewicz, jango}@eti.pg.gda.pl

***Abstract*— Development of safety-critical software is constrained by the requirements of numerous standards and recommendations. In consequence, the development costs and time are considerably higher. In order to deliver high quality products faster and at lower cost safety-critical software developers may look for more efficient approaches and in particular the agile development practices are considered as a promising alternative. In this text we describe our research towards introducing agile practices into critical software development processes**

Keywords— safety-critical software; agile practices; software development; process improvement; safety assurance

I. PROBLEM STATEMENT

The need to deliver high quality systems, faster and at lower cost in comparison to competitors encouraged companies to look for more efficient solutions [1], [2]. Agile methodologies are known to successfully address these issues for non-critical projects. Presumably agile practices can reduce both cost and time to market when applied to safety-critical projects as well. While benefits can be significant, the main concern are quality and safety assurance. Plan-driven methodologies adequately address these objectives and have been integrated into the safety lifecycle for a long time. A growing body of evidence demonstrates that agile practices with their flexibility and the potential for shortening the development time and lowering the development costs could be complemented by more disciplined approach and therefore bring the best of the two worlds together [3], [4], [5], [6], [7], [8]. In particular, the challenge is to find an effective way of introducing agile practices into the critical software development process while ensuring that the level of assurance required by the corresponding domain specific standards is sufficient from the regulatory and system certification viewpoints. A mechanism for maintaining such control over the critical (agile) software development process could result in an explicit assurance (safety) case which integrates the assurance requirements with the arguments and evidence demonstrating that the requirements are satisfactorily implemented. Providing a methodological framework and tools for building, maintaining and assessing such assurance cases for the software development processes could help safety-critical software developers (in particular SMEs) to streamline their processes with agile practices and to maintain conformance with safety standards and certification requirements.

II. OBJECTIVES

The main goal of the research is to develop a comprehensive solution that would help safety-critical software developers to incorporate agile practices in the most profitable way while meeting the safety requirements imposed by standards and certification bodies. In particular, we are interested in a solution that provides for incremental development of an assurances case in paralel to the progress of the software development process which would make the assurances case a sort of 'side effect' of the software development.

Although we want the solution to be as generic as possible, we decided to focus on medical software domain. Medical devices are becoming increasingly software intensive and the market of suppliers as well as clients is growing rapidly. Health related electronic devices find their use in hospitals, homes, pharmaceutical companies and in many contexts has safety relevance. Most of such products need to be compliant with appropriate the related standards (e.g. GAMP [9], ISO 13485 [10], IEC 62304 [11]) and explicit assurance cases for medical devices are expected to be required by the relevant regulatory bodies.

III. METHODOLOGY

First, a literature review has been performed together with the review of the recommended, currently applied practices of critical software development and the review of currently used agile software development practices.

To better understand the risks related to application of the agile practices in critical software development processes we plan to perform a series of case studies with the involvement of software engineers. During these case studies the participants are requested to analyze and assess risks related to the agile practices and to propose the ways of controlling these risks. The case studies are bound by target system and its environment (the common insulin pump example has been selected for that purpose) and by the process context of software development (here, we concentrate on Scrum and Extreme Programming).

To represent the constraints imposed by the relevant standards and recommendations and to incrementally construct a related

assurance case we will use the TRUST-IT methodology [12] and in particular its application scenario related to standards conformance [13], [14] and the related platform of services [15]. Referring to standards and the present good practices we will develop a set of argumentation patterns that will also reflect the knowledge on risk mitigation related to the agile development practices acquired during the case studies. In order to justify each pattern we will prepare complementary meta-arguments. We described these ideas in more detail in [16], [17].

Upon completion of the planned case studies we will analyze the results and determine which of the agile practices raise most doubt when applied in safety critical projects and thus require extended evidence when building assurance arguments.

Alongside the case studies we will prepare templates for meta-arguments as well selected assurance argument patterns, for software development processes which incorporate agile practices, both presented in NOR-STA tool [15].

The complete method is planned to be a subject of validation with active participation of stakeholders and experts. Our aim is to establish cooperation with medical software developing companies to provide a satisfactory validation context.

IV. Expected results

The results of this research will be delivered as a set of argument patterns justified by associated meta-arguments supporting introduction of agile practices into critical software development and the related models of business processes explaining how the argument patterns are to be used to incrementally develop an assurance case for the resulting software. These results will be packaged on the top of the NOR-STA platform of generic services supporting application of evidence based argumentation [15].

V. Present state

The research described in this paper is an ongoing project. To date we carried out two case studies in 2012 (CS1) and 2013 (CS2) with the goal to investigate how junior software engineers identify and assess risks associated with applying selected agile practices to critical software development and what are their suggestions concerning risk mitigation. The results of the CS1 can be found in [16] and [17]. We plan to carry out workshops for more advanced practitioners in the nearest future as a continuation of the case studies.

References

[1] K. Petersen, C. Wohlin, "The effect of moving from a plan-driven to an incremental software development approach with agile practices," Empirical Software Engineering, vol. 15(6), pp. 654–693, 2010.

[2] M. McHugh, F. Mc Caffery, V. Casey, M. Pikkarainen, "Integrating Agile Practices with a Medical Device Software Development Lifecycle," Proceedings of European Systems and Software Process Improvement and Innovation Conference (EuroSPI), Vienna, Austria, 25-27 June, 2012

[3] M. Lindvall, D. Muthig, A. Dagnino, C. Wallin, M. Stupperich, D. Kiefer., J. May. and T. Kähkönen, "Agile Software Development in Large Organizations," Computer, vol. 37(12), pp. 26-34, 2004

[4] H. Glazer, D. Anderson, M. Konrad and S. Shrum, "CMMI or Agile : Why Not Embrace Both!" Software Engineering Process Management – Technical Note for Software Engineering Institute, Carnegie Mellon University, 2008

[5] M. Poppendieck, T. Poppendieck, "Lean software development: an agile toolkit," Addison-Wesley, 2003

[6] J. Babuscio, "How the FBI Learned to Catch Bad Guys One Iteration at a Time," 2009 Agile Conference Proceedings,Chicago, USA, 24-28 August 2009, pp. 96-100

[7] N. Potter, M. Sakry, "Implementing Scrum (Agile) and CMMI together," Process Group Post Newsletter, 16(2), http://www.itmpi.org/assets/base/images/itmpi/Potter-ScrumCMMI.pdf, 2009

[8] M. Pikkarainen, A. Mantyniemi, "An Approach For Using CMMI in Agile Software De-velopment Assessments: Experiences From Three Case Studies," Proceedings of SPICE Conference, Luxembourg, 3-5 May 2006

[9] ISPE GAMP 5 Publications, http://www.ispe.org/index.php/ci_id/11614/la_id/1.htm

[10] ISO 13485:2003 , http://www.iso.org/iso/catalogue_detail?csnumber=36786

[11] IEC 62304, http://webstore.iec.ch/preview/info_iec62304%7Bed1.0%7Den_d.pdf

[12] J. Górski, "Trust-IT – a framework for trust cases", Workshop on Assurance Cases for Security - The Metrics Challenge. Proc. of DSN 2007, Edinburgh, UK, 2007, pp. 204-209

[13] J. Górski, L. Cyra, J. Górski, "SCF - a Framework Supporting Achieving and Assessing Conformity with Standards," Computer Standards & Interfaces, Elsevier, vol. 33, 2011, pp. 80-95

[14] J. Górski, A. Jarzębowicz, J. Miler, "Validation Of Services Supporting Healthcare Standards Conformance," Journal on Metrology and Measurement Systems, vol. XIX, No. 2, 2012, pp. 269-282

[15] NOR-STA project Portal, http://www.nor-sta.eu

[16] J. Górski, K. Łukasiewicz, "Agile development of critical software, can it be justified?" 7th International Conference on Evaluation of Novel Approaches to Software Engineering, Wrocław, Springer, 2012

[17] J. Górski, K. Łukasiewicz, "Assessment of risks introduced to safety critical software by agile practices - a software engineer's perspective," Computer Science, vol. 13(4), 2012