Agile Development of Critical Software – Can It be Justified?

Janusz Górski and Katarzyna Łukasiewicz
Department of Software Engineering, Faculty of Electronics, Telecommunications and Informatics,
Gdańsk University of Technology, Narutowicza 11/12, 80-233 Gdańsk, Poland

Keywords: Agile Practices, Safety-critical Software, Medical Devices, Experimental Assessment.

Abstract: The paper introduces the problem of application of agile practices in critical software development projects. It summarizes the present state of research and identifies the need for having a ready-to-use model of being agile while still meeting the required assurance levels which could be particularly useful for small and medium sized safety-critical software companies. Then the objective and scope of a research aiming at delivering such a model is presented together with a case study description which is a step of this research project. The case study will investigate how software engineers perceive risks associated with introduction of agile practices and collect their ideas on how these risks could be mitigated.

1 INTRODUCTION

Following the introduction of the Agile Manifesto in 2001 (Agile Manifesto, 2001) the agile methodologies have been increasingly attracting developers, offering more flexible and 'human' approach towards the software development. However, it is often argued that they are suitable only for small, non-critical projects as a less disciplined alternative to plan-driven methodologies. For many years it was a common misconception that agile methodologies are at odds with the guidelines of formal certification programs and maturity models (Glazer, et al., 2008). Nevertheless this attitude does not give enough credit to agile methodologies. In their essence, they present an approach towards volatile requirements environment, address needs for shorter delivery times, better customer satisfaction and cost reduction. These concerns can as well apply to bigger projects which may be subject to certifications, standards and maturity models at the same time. It also applies to safety-critical domains such as medical devices, where there is a growing competition between companies making the software part very often a critical success factor. Consequently, providing products in shorter time and for less in comparison to competitors, while satisfying the clients, becomes crucial (Petersen and Wohlin, 2010). The plan-driven methodologies with their heavy-weight processes are more likely to loose contact with the stakeholders needs and restricted flexibility can be a risk for the project success (Boehm and Turner, 2003). Still, there is the safety part with all the standards and certifications associated with it. Plan-driven development addresses the safety assurance needs with well developed risk management techniques, appropriate documentation and traceability. Moreover, many companies have already years of experience in executing safety-critical projects using such methodologies and therefore the acquired know-how. For these reasons the plan-driven methodologies have been the methodologies of choice in the safety-critical domain for a long time (Ge, Paige and McDermid, 2010).

2 INTRODUCING AGILITY TO SAFETY-CRITICAL DOMAIN

The idea of tailoring agile practices in order to make them more compliant with various standards and maturity models has been increasingly developed in the last few years.

2.1 Balancing Agility and Discipline

Models which would balance the agile and more heavy-weight practices have been introduced as early as in 2003 (Boehm and Turner, 2003) bringing the subject to public attention. The need for

Górski J. and Łukasiewicz K..
Agile Development of Critical Software – Can It be Justified?.
DOI: 10.5220/0003984101340139
In *Proceedings of the 7th International Conference on Evaluation of Novel Approaches to Software Engineering* (ENASE-2012), pages 134-139
ISBN: 978-989-8565-13-6

combining the best of the two worlds together kept breaking the ice and resulted in models adapting agile practices to maturity models such as CMMI (Fritzsche and Keil, 2007; Marçal, et al., 2008; Diaz, Garbajosa and Calvo-Manzano, 2009; Bulska, 2010), bringing new possibilities to the larger software companies as well. Some well documented applications of such balanced approaches have been reported since, mentioning the benefits obtained from introducing agile practices into software development process (Lindvall, et al., 2004; Poppendieck M. and T., 2003; Babuscio, 2009; Glazer, et al., 2008; Potter and Sakry, 2009; Pikkarainen and Mantyniemi 2006).

2.2 Models for Safety-critical Systems

With growing body of evidence for potential improvements offered by incorporating agile practices into projects, the question arose if it is also possible for companies involved in safety-critical software domains to benefit from becoming more agile and, as a matter of fact, if this is feasible at all. In fact, as early as in 2003, Alleman et al. (2003) reported a successful implementation of agile practices in a government contracted project, proving it doable and profitable in their case. They proposed an approach combining eXtreme Programming (eXtreme Programming, 2009) practices with Earned Value Management and described their experiences along with benefits they had managed to achieve. Nevertheless little was mentioned about the exact approach used in the project and which features of the product and its certificates influenced the choice of practices.

Subsequent reports gave more promising descriptions of implementations of agile practices. Rasmussen et al. (2009) concentrated on describing their experiences with adopting a tailor made agile approach Agile+ in an FDA regulated project in Abbott company. The company was interested in investing in new methodology due to the rapid growth of the market which put extra pressure on responding to the changing requirements as well as the need to reduce cost of producing the software. Interestingly, they managed to address these needs entirely by introducing the Agile+, concluding that "this experience has convinced us that an agile approach is the approach best suited to development of FDA-regulated medical devices." (Rasmussen et al., 2009). Still, the methodology was prepared by AgileTek (AgileTek, 2012), a software engineering firm, which narrowed down the scope of the methodology's description in the article, thus despite the encouraging success of Abbott, other companies would not be able to recreate it without external help.

Another interesting investigation along with case study was reported by Petersen and Wohlin (2010). They studied a successful migration from plan-driven methodology to a more agile approach which had taken place in Ericsson AB which is certified with ISO 9001:2000.

With increasing number of reports suggesting that by adapting agile practices to suit safety-critical processes, companies can achieve calculable profits, a need for a model of such adaptation arose.

Attempts to create such models include FDA-compliant practice for medical software (Weiguo and Xiaomin, 2009), "evenly weighted" eXtreme Programming for high-integrity systems (Paige, Charalambous, Ge and Brooke, 2008), incorporating risk management (Ge, Paige and McDermid, 2010) and safety modelling (Stephenson, McDermid and Ward, 2006) techniques into agile practices.
Main focus in these models was set on

- Incremental approach towards safety argument which would mean delivering a safety argument for each iteration and adding subsequent ones with the progress of the project, resulting in a complete argument;
- Adding necessary risk management techniques into agile project lifecycle along with safety assurance activities. This means a need for i.e. preparing a description of architecture of the system and a prioritized list of requirements;
- The idea of incremental certification adjusted to the incremental development presented by the agile approach;

While these models provide a valuable source of knowledge, there is a pressing lack of a more ready-to-use model which could be used by small and medium sized safety-critical software companies.

3 PROPOSED RESEARCH

The goal of the proposed research project is to develop a comprehensive set of guidelines supported by tools, to help software developers in combining the agile and more disciplined practices in order to improve the effectiveness and efficiency of their critical software development process while keeping the sufficient levels of assurance.

3.1 Assurance Argument Patterns

In our research we will use *argument patterns* to guide the software developers in building explicit and incremental safety arguments growing in parallel with their software development projects. The patterns will be derived from the relevant standards, regulations and guidelines. They will follow the TRUST-IT approach taken while applying argument structures to support application of standards (NOR-STA, 2012; Cyra, Górski, 2011a), in particular for the medical domain (Górski, Jarzębowicz, Miler, 2012).

TRUST-IT (Górski, 2005; Górski et al., 2005; Górski, 2007) is an approach to promoting trust by presenting in the cyberspace 'live' arguments integrated with the supporting evidence and providing means for assessing and visualizing the compelling power of the arguments. Evidence is a document in any form: text, graphics, image, web page, video, audio etc. which is used to demonstrate the facts referred to in the argument. Integrating an argument with supporting evidence helps to make it more convincing. TRUST-IT introduces a model of an argument, a graphical language for expressing arguments and a technique for integrating arguments with the evidence. It also offers a general purpose argument appraisal mechanism based on Dempster-Shafer belief functions (Sentez K., Ferson S., 2002) and the corresponding mechanism of visualisation of the argument compelling power (Cyra, Górski, 2011b). TRUST-IT arguments were already applied to analyze safety, privacy and security issues of personalized health and lifestyle oriented services (Górski, Jarzębowicz, Miler, 2008), monitoring of environmental risks (ERM, 2009) and support of standards conformance (Cyra, Górski, 2011a; Górski, Jarzębowicz, Miler, 2012). TRUST-IT is offered to its users by means of software services deployed in accordance with the SaaS (Software-as-a-Service) cloud computing model. The approach is generic and can be applied in any context were evidence based argumentation brings added value to decision making processes and disputes.

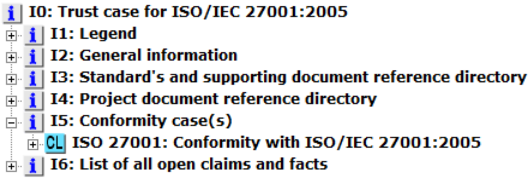
i I0: Trust case for ISO/IEC 27001:2005
⊞ i I1: Legend
⊞ i I2: General information
⊞ i I3: Standard's and supporting document reference directory
⊞ i I4: Project document reference directory
⊟ i I5: Conformity case(s)
⊞ CL ISO 27001: Conformity with ISO/IEC 27001:2005
⊞ i I6: List of all open claims and facts

Figure 1: Example argument pattern.

An example argument pattern expressed in accordance with TRUST-IT and demonstrating conformance with a standard is given in figure 1. The key part of it is the row labelled 'I5' and the claim below it (denoted 'CL') which postulates conformity with ISO/IEC 27001. The claim is further decomposed into more specific claims (which is not shown in the figure). The decomposition ends at *facts* which are assertions about the state of the world. Validity of a fact is justified by the evidence linked to it. Figure 2 presents how a higher level claim is decomposed into five more specific claims.

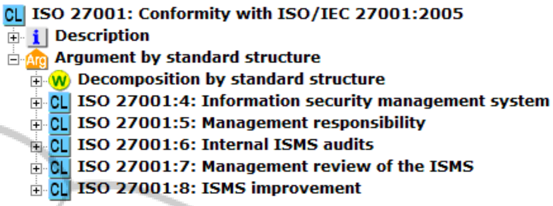
CL ISO 27001: Conformity with ISO/IEC 27001:2005
⊞ i Description
⊟ Arg Argument by standard structure
⊞ W Decomposition by standard structure
⊞ CL ISO 27001:4: Information security management system
⊞ CL ISO 27001:5: Management responsibility
⊞ CL ISO 27001:6: Internal ISMS audits
⊞ CL ISO 27001:7: Management review of the ISMS
⊞ CL ISO 27001:8: ISMS improvement

Figure 2: Claim decomposition.

3.2 Justifying the Patterns

While developing the argument patterns we will refer to already identified best practices recommended by relevant guidelines, standards and publications. In addition to this, we plan for a series of experiments during which we will acquire additional knowledge on how to incorporate agile practices into critical software development processes in a way which do not hinder the possibility for being conformant with the relevant assurance requirements. For each argument pattern to be used to justify agility in critical software development, we plan to develop a separate meta-argument justifying this argument pattern. In our research we are planning for the experiments aiming at collecting evidence supporting these meta-arguments. An example of such assurance arguments can be found in (Ge, Paige, McDermid, 2010).

4 CASE STUDY

Medical safety-critical software domain has developed in recent years at a very fast pace. It has moved from supplying only hospitals and providing solutions for doctors to e-health technology and personal medical equipment. A growing competition among companies and willingness to satisfy a bigger and more diverse customer group can provoke such companies to look for new solutions for their software development processes. For this reason we chose this domain as a subject for the case study.

4.1 Description of the Case Study

The objective of this case study will be to collect evidence about how software engineers perceive risks associated with selected agile practices while applied to critical software. Such evidence will be needed to support meta-arguments justifying the assurance argument patterns mentioned in the previous section.

The case study will be carried out from March to the end of May 2012 in a group of 36 students of our university, all at the last year of their master course, specializing in software engineering. Most of the students are already part-time employees of software companies. All have already attended courses on plan-driven and agile methodologies and risk assessment methods.

They will work in groups of 3, forming 12 project groups. All of the groups will be given the same product specification of an insulin infusion pump and a short description of a fictional company called MediSoft which produces software for such pumps. The company's management have been observing a growing popularity of agile methodologies and became interested in potential benefits obtained by introducing such methodologies. To investigate the potential of agile approaches, MediSoft decided to carry out a pilot project concerning software for insulin infusion pump. The project will employ eXtreme Programming (eXtreme Programming, 2009) and Scrum (Schwaber and Beedle, 2001) methodologies.

An insulin pump is a device for patients with diabetes who need to control their blood sugar level by administrating insulin. The pump is attached to the patient's body along with a small container filled with insulin. At the correct times, small and precisely calculated amounts of insulin are released from the container into the patient's bloodstream.

The insulin pump description used in the project is based on a real pump available on market, the Animas OneTouch Ping (Animas One Touch Ping, 2012) characterized by the following features:

- Calculator for carbohydrates, blood glucose corrections and insulin;
- Insulin bolus very precise, should allow dosing even the lowest amounts of insulin in order to respond to every glucose deviation;
- Reminders for when to perform blood glucose checks;
- Easily available insulin dose corrections;
- Measuring the level of active insulin in the body;
- Wireless communication, the pump can be controlled with a wireless remote;
- Wireless bolus calculation and delivery;
- Uploading data from the pump to a computer using dedicated software;
- Information about the state of the body showed on the screen;
- Waterproof;

Students will work on the project in three phases:

A. Preparing a List of Hazards associated with the functioning of the insulin pump. Here they will be supported by some hazard identification techniques, like Preliminary Hazard Analysis and HAZOP adapted for this case;

B. Conducting a Risk Analysis for the introduction of agile methodologies (6 groups for eXtreme Programming and 6 groups for Scrum) to a project with safety-critical requirements. The students will use the Designsafe tool (Designsafe, 2012) to support their risk analysis. In first step of the analysis they will be given a list of practices from either eXtreme Programming or Scrum grouped into sub-processes (Planning Game, Sprint Planning etc.). They will be encouraged to complete the list if they should feel the need to specify given practices in more details. In next step they will narrow down the list of potential hazards to the ones which can be connected with the software and human errors, They will also reflect on possible causes. They will be encouraged to employ FTA analysis at this stage. Having prepared the list, students will assign the hazards to the practices which may have influence on materializing given hazard. After completing these tasks, they will be able to carry out another step of the project – assessing the risk using a selected risk scoring system. This will result in risk levels assigned to the identified hazards.

C. Composing a List of Additional Practices which could extend the methodology they were working with (either eXtreme Programming or Scrum) in order to reduce potential risks in areas they perceive as important.

4.2 Research based on the Case Study

The main goals of the case study are:

- Obtaining suggested lists of hazards rooted in the chosen software development process and then analyzing and merging these lists in order to result in a checklist of hazards related to application of agile practices in safety-critical software development;
- Obtaining the estimates of risk (risk levels) related to introduction of agile practices to a project with safety-critical requirements;
- Obtaining suggestions of additional practices (for risk mitigation) which would scale down the risk of introducing agile methodologies to

a project with safety-critical requirements;

We expect that through this research we will be closer to having answers to the following questions:

- How is the risk related to the agile practices used?
- To what extent do the agile practices influence the risk?
- How and to which extent the risk can be mitigated by additional practices introduced together with the agile ones?

The metrics we will collect during the case study:

- List of hazards for an insulin pump applied in its target environment;
- Complete list of agile practices;
- The list of hazard scenarios explaining how agile practices applied contribute to software hazards
- Risk assessment (risk levels) associated with each agile practice used;
- Agile practices which carry the highest risk;
- List of risk mitigation recommendations;

Obviously students will prepare their analysis based on limited knowledge and experience. Neither their risk analysis will be equal to the expert professional ones nor the proposed set of additional practices will be sufficient to implement the methodologies in safety-critical environment straight away and these are not the goals of the case study. Based on the results we will be able to detect which areas are perceived as problematic, which bring the most fear when it comes to implementing agile methodologies in safety-critical projects and which practices can ease this fear by delivering safety-assurance qualities. These more or less intuitive evaluations will provide a valuable source of knowledge which can be a great point of reference for building relationships with the companies - potential customers in future as well as for preparing the model of adapting agile practices into safety-critical projects.

5 CONCLUSIONS

The main goal of our research project is to provide companies which produce safety-critical software with a model and a supporting tool of choosing the right balance between agile and more disciplined practices. In this paper we explained the project objectives and scope and explained what is its expected result. The result will be a knowledge base containing guidelines on how to incorporate agile practices into critical software development projects together with patterns of arguments to be developed to demonstrate that introduction of the agile practices do not hinder the assurance levels required by the corresponding standards and regulations. The patterns themselves will be supported by meta-arguments justifying their validity. To construct such meta-arguments we need more evidence on how the risks associated with agile practices can be effectively assessed and managed.

The case study described in this paper is a step on the way of collecting such evidence. The results of the case study should be known in mid-June and if the paper is accepted will be presented during ENASE 2012 conference.

REFERENCES

Agile Manifesto, 2001. *Agile Manifesto.* [online] Available at: <http://agilemanifesto.org/> [accessed January 2012]

AgileTek, 2012. *AgileTek*. [online] Available at http:// www.agiletek.com/ [accessed January 2012]

Alleman, G. B., Henderson, M., Hill, C. H. M., & Seggelke, R., 2003. Making Agile Development Work in a Government Contracting Environment Measuring velocity with Earned Value. In *Proceedings of the Agile Development Conference 2003,* Salt Lake City, Utah, 25-28 June 2003, *IEEE Computer Society*, pp. 114-120.

Animas One Touch Ping, 2012. Insulin pump. [online] Available at <http://www.animas.com/animas-insulin-pumps/onetouch-ping> [accessed February 2012]

Babuscio, J., 2009. How the FBI Learned to Catch Bad Guys One Iteration at a Time. In *2009 Agile Conference Proceedings*,Chicago, USA, 24-28 August 2009. *IEEE Computer Society*, Los Alamitos 2009, pp. 96-100

Boehm, B. & R. Turner, 2003. Balancing Agility and Discipline: A Guide for the Perplexed. *Addison Wesley*, 2003.

Bulska, K., 2010. Integration of the agile software development methodologies with maturity models – good practices assistant, MSc thesis, *Gdansk University of Technology*, Gdansk, 2010. (in Polish)

Cyra L., Górski J., 2011. SCF - a Framework Supporting Achieving and Assessing Conformity with Standards. In *Computer Standards & Interfaces*, Elsevier, 33, 2011, pp. 80-95

Cyra L., Górski J., 2011. Support for Argument Structures Review and Assessment. In *Reliability Engineering and System Safety*, Elsevier, 96, 2011, pp.26-37

Designsafe, 2012. *Designsafe tool*. [online] Available at <http://www.designsafe.com/> [accessed Jan. 2012]

Diaz J., Garbajosa J., Calvo-Manzano J. A., 2009. Mapping CMMI Level 2 to Scrum Practices: An Experience Report. In *EuroSPI 2009 Proceedings*. Alcala, Spain, 2-4 September 2009. Springer, Heidelberg, pp. 93-104

ERM - Workshop on Selected Problems in Environmental Risk Management and Emerging Threats, 2009. *Proc. of the Workshop on Selected Problems in Environmental Risk Management and Emerging Threats,* June 2009, Gdansk, Poland [online] Available at <http://kio.pg.gda.pl/ERM2009/> [accessed February 2012]

eXtreme Programming, 2009. *Extreme Programming: A gentle introduction*. [online] Available at <http://www.extremeprogramming.org/> [accessed January 2012]

Fritzsche, M., Keil, P., 2007. Agile methods and cmmi: Compatibility or conflict? In *e-Informatica Software Engineering Journal*, 1(1), pp. 9-26

Ge, X., Paige, R. F., McDermid, J., 2010. An Iterative Approach for Development of Safety-Critical Software and Safety Arguments. In *2010 Agile Conference Proceedings*, Orlando, USA, 9-13 August 2010. *IEEE Computer Society*, Los Alamitos 2009, pp. 35-43

Glazer, H., Anderson, D., Anderson, D. J., Konrad, M., & Shrum, S., 2008. CMMI or Agile : Why Not Embrace Both! In *Software Engineering Process Management – Technical Note* for Software Engineering Institute, Carnegie Mellon University.

Górski J, Jarzębowicz A, Leszczyna R, Miler J, Olszewski M., 2005. Trust Case: Justifying Trust. In *IT Solution. Elsevier, Reliability Engineering and System Safety*, 2005, 89, p. 33-47.

Górski J., 2005. Trust Case – a case for trustworthiness of IT infrastructures. In *Cyberspace Security and Defense: Research Issues*, NATO Science Series II: Mathematics, Physics and Chemistry, 196 (). Springer-Verlag, 2005, pp. 125-142.

Górski J., 2007. Trust-IT – a framework for trust cases, Workshop on Assurance Cases for Security - The Metrics Challenge. In *Proceedings of DSN 2007*, June 25-28, Edinburgh, UK, 2007, pp. 204-209.

Górski J., Jarzębowicz A., Miler J., 2008. Arguing trustworthiness of e-health services with the Trust-IT framework. In *Proceedings of 25th Anniversary Healthcare Computing: Invitation to the Future: Conference & Exhibition* (HC 2008), Harrogate 21-23 April, 2008.

Górski J., Jarzębowicz A., Miler J., 2012 (in press). Validation of services supporting healthcare standards conformance. (accepted for publication in Metrology and Measurement Systems, 2012)

Lindvall M., Muthig D., Dagnino A., Wallin C., Stupperich M., Kiefer D., May J. & Kähkönen T., 2004. Agile Software Development in Large Organizations. In *Computer,* 37(12), pp. 26-34.

Marçal, A. C., de Freitas B. C., Furtado Soares F. S., Furtado M. S., Maciel T. M., Belchior A. D., 2008. Blending Scrum practices and CMMI project management process areas. In *Innovations in Systems and Software Engineering*, 4(1), pp. 17-29

NOR-STA, 2012. *NOR-STA project Portal*. [online] Available at <www.nor-sta.eu> [accessed February 2012]

Paige R., Charalambous R., Ge X., Brooke P., 2008. Towards Agile Engineering of High-Integrity Systems. In *Proceedings of 27th International Conference on Computer Safety, Reliability and Security (SAFECOMP),* 22-25 September 2008, Newcastle upon Tyne, UK.

Petersen, K., & Wohlin, C., 2010. The effect of moving from a plan-driven to an incremental software development approach with agile practices. In *Empirical Software Engineering*, 15(6), pp. 654-693.

Pikkarainen M., Mantyniemi, A., 2006. An Approach For Using CMMI in Agile Software Development Assessments: Experiences From Three Case Studies. In *Proceedings of SPICE Conference*, Luxembourg, 3-5 May 2006.

Poppendieck M. and T., 2003. *Lean software development: an agile toolkit*, Addison-Wesley, 2003.

Potter, N., Sakry M., 2009. Implementing Scrum (Agile) and CMMI together. [online] Process Group Post Newsletter, 16(2). Available at: <http://www.itmpi.org/assets/base/images/itmpi/Potter-ScrumCMMI.pdf> [accessed January 2012]

Rasmussen, R., Hughes, T., Jenks, J. R., & Skach, J., 2009. Adopting Agile in an FDA Regulated Environment. In *2009 Agile Conference Proceedings*, Chicago, USA, 24-28 August 2009. IEEE Computer Society, Los Alamitos 2009, pp. 151-155.

Sentez K., Ferson S., 2002. *Combination of evidence in Dempster-Shafer theory*. SANDIA National Laboratories.

Schwaber, K., Beedle, M., 2001. *Agile Software Development with Scrum*. Prentice Hall, 2001.

Stephenson Z., McDermid J., Ward A., 2006. Health Modelling for Agility in Safety-Critical Systems Development. In *Proceedings of the First IET International Conference on System Safety Engineering*, 6-8 June 2006, London, UK.

Weiguo L., Xiaomin F., 2009. Software Development Practice for FDA-Compliant Medical Devices. In *Proceedings of the 2009 International Joint Conference on Computational Sciences and Optimization*, 24-26 April, 2009, Sanya, China.