Proceedings of the Federated Conference on Computer Science and Information Systems pp. 1549–1552
DOI: 10.15439/2016F360
ACSIS, Vol. 8. ISSN 2300-5963

AgileSafe – a method of introducing agile practices into safety-critical software development processes

Katarzyna Łukasiewicz
Gdańsk University of Technology ul. Narutowicza 11/12, 80-233 Gdańsk, Poland
Email: katarzyna.lukasiewicz@pg.gda.pl

Janusz Górski
Gdańsk University of Technology ul. Narutowicza 11/12, 80-233 Gdańsk, Poland
Email: jango@pg.gda.pl

Abstract—This article introduces AgileSafe, a new method of incorporating agile practices into critical software development while still maintaining compliance with the software assurance requirements imposed by the application domain. We present the description of the method covering the process of its application and the input and output artefacts.

I. Introduction

AGILE software development methods have been introduced in response to particular concerns emerging from the changing needs of the market. In practice, volatile requirements, demanding clients from diverse backgrounds and a growing need for shortening time-to-market made a growing number of software companies seek alternatives to their own traditional approaches. A similar tendency can be presently observed in many domains, including the public sector [1] and further in what seemed to be a leading plan-driven environment - the safety-critical software domain. In this case, however, strictly agile methods will not be the answer as they are insufficient on the safety assurance and certification side. The question is if and how to enrich the agile practices with safety and risk management practices without sacrificing agility and still providing the necessary assurance level.

While process optimization is vital to the business and economical aspect of a software development project, in the safety-critical software domain its profits will not be sufficient unless a company is able to conform to standards and guidelines, which regulate a particular industry. Clients demand their products to be of high quality, on time and within a reasonable budget but at the same time the software need to be certified by an appropriate authority in order to be licenced for use in its destined environment. For this reason, in safety-critical software domain it is not enough to improve the software development process to provide financial profits to the company. It is also necessary that the safety requirements are adequately identified and assured throughout the process. Changing the process will likely result in changes in the safety evidence collected during the development, which may affect the scope and structure of the certification process. Therefore, each change to the process should be carefully analysed from the viewpoint of future audits and potential consequences to the outcome of these audits. This makes a change in safety-critical software development process a complicated and potentially costly operation depending on how the company is introducing new elements to the process. This may be a problem for SMEs where budget constrains can be a significant barrier in introducing the change.

Attempts to provide a hybrid, disciplined-agile, approaches bringing together best of the two worlds are already in effect for several years. A growing body of evidence, including industrial reports, shows that obtaining the right balance is doable and profitable especially when the companies decide to employ competent experts to develop a custom made approach. Examples of such reports can be found in [2], [3], [4], [5] and were surveyed in [6]. What is more, in 2012 FDA (Food and Drug Administration) recognized the AAMI TIR45:2012 - Guidance on the use of AGILE practices in the development of medical device software [7]. It concludes that agile practices can be successfully used in safety-critical software development and that such practices can be compliant with IEC 62304 [8] standard. It also provides a mapping between agile methods and IEC 62304 activities.

However encouraging these reports are, 'tailoring' a software development method can be a costly and complicated process. If hybrid approaches are to be applied in a larger scale, more available and ready to use solutions are needed. An example can be SafeScrum [9], which concentrates on adapting Scrum into safety-critical software development. The method has been already applied in a number of real life projects and most of them ended in success as well as required standard certification [10]. Another approach is AV-Model [11] combining the traditional V-Model with Scrum and focusing on medical device software development and the IEC 62304 standard.

In this article we propose a new method, called AgileSafe, of incorporating agile practices into critical software development while still maintaining compliance with the software assurance requirements imposed by the application domain, which is complementary to the existing methods. AgileSafe is addressed mainly to SMEs developing safety-critical software, to support them in the process of introducing new practices into their software development environment. AgileSafe provides a user with tools enabling her/him to create, with a help of guidelines and questionnaires, a hybrid agile approach customized for the project. It also provides a tool

978-83-60810-90-3/$25.00

for handling conformance with standards and norms while introducing this new approach.

II. Overview of the AgileSafe Method

To provide and maintain control over the safety requirements and over the scope and level of their assurance, AgileSafe employs evidence-based arguments which are explicitly maintained during the software development process. These arguments follow the ISO/IEC 15026 [12] recommendations on *assurance cases*. The main idea is to provide assurance cases both for the software development process and for the end product itself. While the latter is the essence of demonstrating product conformance with the stated safety objectives, the former is complementary to it and allows to demonstrate adequacy of the chosen software development practices and in particular their conformity with safety requirements imposed by relevant standards. AgileSafe focuses on an explicit development process assurance case demonstrating that the selected range of software development practices is conformant with the requirements of the relevant safety related standards.

Although, for demonstration reasons we concentrate on the medical domain, the method is generic and can be adapted to different safety-critical domains. In order to address a broad range of products resulting from development, the process assurance arguments are based on the standards that are relevant for a particular application domain.

The prerequisite for applying AgileSafe to a particular (planned) safety-related software development project is that we have identified a set of relevant standards we want to be compliant with. Then, the main results of applying AgileSafe to this project are:

- *Project Practices Set* (*PPS*)– a custom prepared hybrid approach composed of plan-based and agile software development practices;
- *Assurance arguments*, for each selected standard.

Figure 1 presents a BPMN model of the AgileSafe.

Based on the *Project characteristics,* prepared during the AS.P.1 *Analyse the project process*, a user is guided through the set *AgileSafe Practices Knowledge Base,* which contains descriptions of software development practices. The method suggestions are based on the good practices for software development as well as the results of experiments conducted in the course of our research.

The customized *Project Practices Set*, prepared in the AS.P.2 *Select practices* process, should be later implemented in the software development process (AS.P.7 *Apply Practices*). For each given *Standard* a *Practices Compliance Argument* need to be *developed/updated* (AS.P.3). These *Practices Compliance Arguments* are then *adapted* (AS.P.4), depending on the *PPS*, into *Project Practices Compliance Arguments*. Based on them, the *Project Compliance Arguments* are *prepared* (AS.P.5) and they are the end products of the method allowing the user to AS.P.6 *Assert conformance*, using the *Evidence* prepared during the AS.P.7 *Apply practices* process.

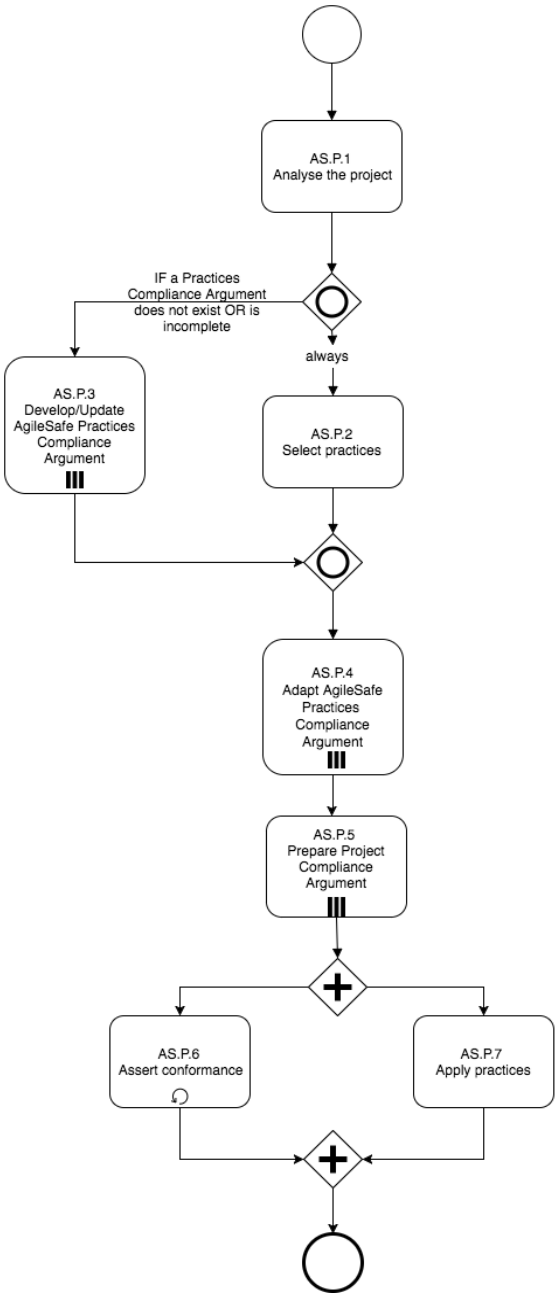

Fig. 1 Diagram of AgileSafe method

In further sections we explain components of the method in more detail.

III. Assurance Arguments in AgileSafe

An assurance argument is a structure of claims, which is based on explicitly provided evidence and demonstrates that a product or a system satisfies a detailed set of requirements. Recommendation on the structure of assurance arguments (called *assurance cases)* can be found in ISO/IEC 15026 standard [12].

Since 2005 the idea of safety assurance arguments has been analysed in depth by both FDA and SEI (Software Engineering Institute) [13]. This partnership resulted in series of documents presenting potential uses of assurance arguments in FDA certification process [13],[14]. With FDA

currently recommending the use of assurance arguments in the process of qualification of medical devices in order to present compliance with safety requirements, explicit use of assurance cases is gaining increasing recognition.

In AgileSafe there are two types of assurance arguments: for process assurance and for product assurance, as shown in Figure 2.

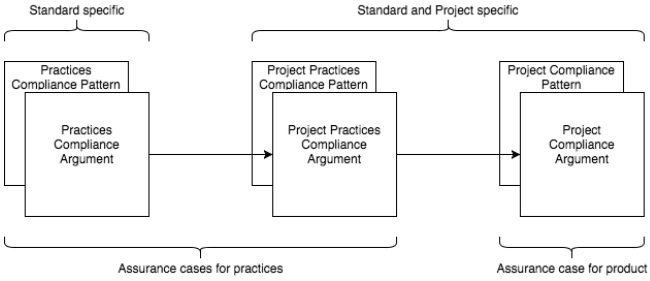

Fig. 2 Assurance cases in AgileSafe

The patterns for assurance arguments hold the information about the argument's structure and are used as a template for creating specific arguments.

In AgileSafe, assurance arguments are developed separately for each applicable standard in order to support certification on the standard by standard basis. A more detailed description of the assurance cases employed in AgileSafe (see Figure 2) is given below.

A. Practices Compliance Argument

Practices Compliance Argument is a template which is developed separately for each relevant standard. Its structure is based on the standard requirements. To make such templates uniform, each template is following the *Practices Compliance Pattern*. This pattern is generic and focuses on the conformance of practices from *AgileSafe Practices Knowledge Base* with particular requirements of the considered standard. For each such requirement it proposes an argumentation strategy and the range of software engineering practices used for collecting evidence demonstrating the compliance. It also contains explicit justification that the argumentation strategy is adequate on the condition that the evidence is collected and integrated with the argument. A list of claims concerning different types of practices, which may contribute to satisfying the standard demand, is presented, each claim postulating the potential of a given practice to generate the evidence needed to demonstrate compliance.

If the *Practices Knowledge Base* is complete, the *Practices Compliance Argument* once prepared for a specific standard remains unchanged and can be used in multiple projects which have to comply with the standard. Nevertheless, in the 'learning period' of the AgileSafe method the *Practices Knowledge Base* is expected to grow acquiring new practices and therefore the *Practices Compliance Arguments* will change and evolve in time.

B. Project Practices Compliance Argument

Project Practices Compliance Argument is a *Practices Compliance Argument* adapted to a specific project and is characterized by the *Project Practices Set* specific to this project. The *Project Practices Compliance Argument* refers only to the practices used in the project along with the description of *Evidence* they are providing. Its structure is defined in the *Project Practices Compliance Pattern*.

C. Project Compliance Argument

Project Compliance Argument is an assurance argument in its traditional form. It is structured around a particular standard and is used to collect the required product related evidence to demonstrate conformance with given standard. The *Evidence* should be collected with accordance to the *Project Compliance Pattern* and be an effect of AS.P.7 *Apply practices* process.

D. Tool support for assurance cases

To handle assurance cases AgileSafe follows the TRUST-IT methodology [15] and uses the Argevide NOR-STA [16] services. NOR-STA provides means for developing, maintaining and assessing assurance cases and integrating them with the supporting evidence. All the assurance cases presented in Section VI were developed using this tool.

In NOR-STA, argument conclusion is represented by a *claim* node. A node of type *argumentation strategy* links the claim with the corresponding premises and uses a *rationale* node to explain and justify the inference leading from the premises to the claim. A premise is a sort of assertion and can be in particular another claim to be further justified by its own premises or a *fact* represented by an assertion to be demonstrated by the supporting evidence. The evidence is integrated by nodes of type *reference* which point to external resources (files of any type, web pages, etc.). In addition, *information* nodes can be used in any place to provide explanatory information.

IV. Practices Selection Process

In order to support users while selecting the software engineering practices from the *AgileSafe Practices Knowledge Base,* AgileSafe offers the guidance, which is based on *Project characteristics.* The practices maintained in the *AgileSafe Practices Knowledge Base* include both plan-driven and agile methods documented in the literature, and may also include custom developed hybrid practices. Upon the selection of the practices an assurance argument is composed along with assurance arguments for selected standards.

Users are able to introduce their own practices into the *Knowledge Base* by following the AS.P.2.1 *Introduce new practice p*rocess.

V. Assessing Conformance

The process of assessing conformance with given standards is based on the set of assurance arguments, mainly the *Project Compliance Arguments*.

The *Project Compliance Arguments* are being developed in parallel to the software development process (AS.P.7 *Apply practices*). The *Evidence* prepared in the course of AS.P.7 *Applying practices* from the *Project Practices Set*

should be placed in the accurate nodes as indicated in the *Project Practices Compliance Arguments*. Upon the certification process for a particular *Standard* User should be able to prove conformance by presenting the applicable *Project Compliance Argument* along with *Project Practices Compliance Argument*, which contains additional reasoning behind the choice of practices (*PPS*) and *Evidence* used in the process.

VI. Conclusion

In this article we presented an overview of AgileSafe, a method of agile software development with simultaneous controlling conformity with selected safety related standards.

The ultimate objective is to help SMEs involved in safety-critical software development to introduce agile practices in the most profitable way while meeting the requirements imposed by safety standards and certification bodies.

Recently we conducted a case study which goal was to use AgileSafe to incorporate selected risk management practices into an agile project with safety requirements. We have followed all of the steps of the AgileSafe method algorithm and prepared the artefacts as specified in the method, collecting all the metrics that we planned to collect. A complete set of AgileSafe assurance cases was prepared for ISO 14971 standard. The basic result of the case study was that it was possible to conduct an agile project while still controlling its conformity to the ISO 14971 requirements.

In the nearest future we plan to interview experts in the field of safety certification as well as practitioners who were involved in adapting agile practices to safety-critical system development in order to obtain their feedback on AgileSafe.

Acknowledgment

This work was partially supported by the Statutory Grant by the Polish Ministry of Higher Education for the Faculty of Electronics, Telecommunications and Informatics of Gdansk University of Technology.

References

[1] A. Kaczorowska, "Traditional And Agile Project Management In Public Sector And ICT," in *Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS)*, Łódź, Poland, September 2015. DOI: 10.15439/2015F279

[2] G. B Alleman., M Henderson., C. H. M. Hill, R.Seggelke, "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*, June 2003. DOI: 10.1109/ADC.2003.1231460

[3] R. Paige, R. Charalambous, X. Ge, P. Brooke, "Towards Agile Engineering of High- Integrity Systems," in *Proceedings of 27th International Conference on Computer Safety, Reliability and Security (SAFECOMP)*, September 2008. DOI: 10.1007/978-3-540-87698-4_6

[4] K. Petersen, C. Wohlin, "The effect of moving from a plan-driven to an incre- mental software development approach with agile practices," in *Empirical Software Engineering*, 15(6):654–693, 2010. DOI: 10.1007/s10664-010-9136-6

[5] R. Rasmussen, T. Hughes, J. R. Jenks, J. Skach, "Adopting Agile in an FDA Regulated Environment," in *Proceedings of Agile Conference*, Chicago, USA, August 2009. DOI: 10.1109/AGILE.2009.50

[6] J. Górski, K. Łukasiewicz. „Assessment of Risks Introduce to Safety Critical Software by Agile Practices – a Software Engineer's Perspective" in. *Computer Science*, 13(4), AGH University of Science and Technology Press 2012. DOI: http://dx.doi.org/10.7494/csci.2012.13.4.165

[7] AAMI TIR45: 2012 Technical Information Report Guidance on the use of AGILE practices in the development of medical device software

[8] ISO/IEC 62304 Medical device software — Software life cycle processes, standard

[9] SafeScrum, http://www.sintef.no/safescrum

[10] T. Stålhane, T. Myklebust, G. K. Hanssen, "Safety standards and Scrum – A synopsis of three standards", http://www.sintef.no/globalassets/safety-standards-and-scrum_may2013.pdf

[11] M. Mc Hugh, F. Mc Caffery, G. Coady "An Agile Implementation within a Medical Device Software Organisation", in Proceedings of The 14th International SPICE Conference Process Improvement and Capability dEtermination 2014. DOI: 10.1007/978-3-319-13036-1_17

[12] ISO/IEC 15026 Systems and software engineering -- Systems and software assurance

[13] C. B. Weinstock, J. B. Goodenough. "Towards an Assurance Case Practice for Medical Devices". TECHNICAL NOTE Software Engineering Institute October 2009. http://www.sei.cmu.edu/reports/09tn018.pdf

[14] FDA: Guidance – Total Product Life Cycle: Infusion Pump-Premarket Notification Submissions [510(k)], 2010

[15] J. Górski, "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, pp. 125-142 (2005). DOI: 10.1007/1-4020-3381-8_7

[16] Argevide NOR-STA, https://www.argevide.com/en