Integration of the 4+1 Software Safety Assurance principles with Scrum

Osama Doss[1], Tim Kelly[2], Tor Stålhane[3], Børge Haugset [4], and Mark Dixon[1]

[1] School of Computing, Creative Technologies, Leeds Beckett University UK
o.doss7788@student.leedsbeckett.ac.uk m.dixon@leedsbeckett.ac.uk
[2] Department of Computer Science, University of York, UK
tim.kelly@york.ac.uk
[3] Norwegian University of Science & Technology, Trondheim, Norway
tor.stalhane@idi.ntnu.no
[4] SINTEF ICT, Trondheim Norway
borge.haugset@sintef.no

Abstract. Some researchers have attempted to tailor agile methods to comply with specific standards (e.g. SafeScrum and IEC61508). However, this risks over-configuring the agile method in such a way as to make it difficult to apply it to another safety standard.

Our approach sought to look at the problems of addressing the more fundamental principles of safety assurance by adopting the 4+1 safety principles and investigating how a Scrum process challenges, and can be adapted to give strong indication that the practitioners felt that there is a significant potential for successful integration of the 4+1 principles within Scrum. There were some issues where practitioners were concerned to focus only on one safety standard, and neither the agile practitioners nor the safety practitioners had a clear understanding of the outlook and work of the other group. However, we used these issues to inform a further set of questions. We conducted semi-structured interviews with participants to explore the general feasibility of the approach, and to provide an assessment as to whether the 4+1 principles can be addressed without compromising agility.

Keywords. Safety-Critical Systems, Agile Methods, Assurance Case, Scrum

1 Introduction

Evidence and experience concerning the integration of Agile development in the field of safety-critical software development is limited. However, there are some published case studies and research on successes or failures in that field, [1,2]). Agile methods are known for being fast, efficient and adaptive, as well as fostering discipline and good

adfa, p. 1, 2011.

practices in engineers. The use of agile methods can support both quality and team productivity [3]. We need to investigate whether it is possible to use agile methods that are flexible with respect to planning, documentation and specification while still being acceptable by standards [9]. In particular, we need to consider how a structured argument providing assurance of the safety of the system can be incorporated with a typical agile development method.

2 Research problem and questions

We are concerned with the research problem of how assurance case development (including the incremental development of structured arguments) can best be integrated with a typical agile development method.

We propose the following two research questions (RQ) concerning the relationship between Agile Methods and Safety-Critical systems:

- *RQ1 What are the current concerns and opportunities voiced by safety-critical systems professionals regarding the use of agile development methods for safety-critical systems development?*
- *RQ2 What changes are necessary to the Scrum Process in order to address the 4+1 Software Safety Assurance Principles?*

Our research will focus on investigating best practice evolution of GSN arguments as a means for safety case development as part of a Scrum process. We expect to develop guidance that will support the development of a goal structure as an integral part of this process.

3 The 4+1 - Fundamental Principles of Software Safety Assurance

There are many standards that either directly or indirectly addresses software safety assurance (e.g. IEC 61508, ISO 26262, EN 50128, parts of UK Defense Standard 00-56, and DO-178B). Although there are differences between these standards, there are a number of common principles that can be observed [4]

- Principle 1: Software safety requirements shall be defined to address the software contribution to system hazards
- Principle 2: The intent of the software safety requirements shall be maintained throughout requirements decomposition
- Principle 3: Software safety requirements shall be satisfied
- Principle 4: Hazardous behavior of the software shall be identified and mitigated
- Principle 4+1 The confidence established in addressing the software safety principles shall be commensurate to the contribution of the software to system risk [5].

The focus of our work is to identify and address challenges associated with the integration of Agile methodologies into safety-critical systems development. In order to gain a deeper insight into the challenges identified for the integration of safety assurance into Scrum, and into the practicality of the recommendations [16], we conducted semi-structured interviews with safety engineers and Agile developers.

4 Related Work

Previous work discussed about how to integrate agile methods into a regulator environment [6]. Jonsson [7] describes a mapping between requirements in a regulated environment (EN 50128) and agile practices. They define a modular approach for building safety arguments incrementally using the Goal Structuring Notation (GSN) [8]. The process they propose "...precisely captures the notion of sufficient up-front design..." They also illustrate how to use safety patterns and introduce the notion of a modular safety argument to enable the iterative development of a safety argument. However, their conclusion is that agile practices may not change the nature of the entire safety-critical development procedure model, but might improve the agility of the development. Furthermore, it is important to note that this paper is intended as a conceptual proposal, and is far from industrial practice. Fitzgerald et al. [6] presented a case study to demonstrate that agile methods could be scaled to regulated environments. Their paper described a number of issues - quality assurance, safety and security, effectiveness, traceability, and verification and validation - and illustrated how they react with the model. The work also focused on how to implement an integrated model to achieve compatibility in terms of agility, safety/security, certification, and Quality Assurance. Stålhane et al. [9] proposed the "SafeScrum" approach. This was motivated by the need to make it possible to use methods that are flexible with respect to planning, documentation and specification while still adhering to the safety standard IEC 61508. Stålhane illustrates two kinds of Product Backlog within SafeScrum: the Functional Product Backlog and Safety Product Backlog. Other researchers have examined how conventional methods and techniques used for security assurance suit agile methodologies. Beznosov and Kruchten [10] evaluated how well security assurance practices match the typical practices of agile methods. For example, they identify that the informal review practices of agile methods match well, but that many agile approaches lack the external review and formal validation required for security. In addition, they identify that many security practices are independent of the adoption of agile practices – e.g. the use of security design principles. Similarly, Lotfi [11] found that some security assurance practices can be easily integrated.

However, our work focuses on the satisfaction of 4+1 principles within an agile framework "Scrum". Specially, the focus of the work is on assurance rather than being seen to comply with a specific safety standard.

5 Data collection and methods

The study was conducted as a qualitative survey using "semi-structured interviews" for data collection [12]. Shull et al. [13] illustrate the advantage and disadvantage of conducting a semi-structured interview. The interview will include some simple (e.g. Likert scale-based) question, as well as more open-ended questions that allow for greater depth of response. The purpose of these interviews was to investigate the proposed integration of the 4+1 principles and assurance safety case development with Scrum. The interviews lasted for at least 40 minutes, and explored perceptions around the 4+1 Principles of Software Safety Assurance and their implications for Scrum. Interviews were voice recorded with consent from the participants.

The interviews started by introducing the research goals and the topics to be discussed. Then the 4+1 principles were explained, together with an outline of the proposed integration of these principles in a Scrum development setting. Questions were then asked relating to the proposal – selecting specific features one-by-one (e.g. our recommendations for team composition). The questions touched both aspects of a) whether the proposed approach challenges agility and b) whether the proposed approach challenges safety assurance. [14]

Transcripts were analyzed using thematic analysis. A researcher examined all of the interview transcripts, and coded the transcripts using first-cycle coding (Open Coding or Initial Coding), supported by the NVivo 11 software package. Main categories (or topics) were identified through clustering of codes. Codes and categories were constantly compared with the data and revised or refined as appropriate [14]

5.1 Participants and Interviews

We interviewed 12 participants at the XP 2016 Conference in Edinburgh, from the academic and industrial domains in order to use their experience and insight to gain feedback on our proposed approach. The participants had experience with safety critical-systems, agile methods, or both. The majority of our participants were based in the UK, Sweden, Germany, the USA, and Norway. The table below shows participant qualifications:

PARTICIPANT DESCRIPTION

Participant #	INTERVIEWEE JOB TITLES
1	Development Engineer
2	Professor of Software Engineering
3	Owner, evolution and Information Technology and Services Consultant
4	Director and Consultant
5	Professor Vice-Director, Chair at…
6	Professor of Software Engineering
7	Independent programmer and consultant
8	Emeritus Professor of Computing Science
9	Founder and Director

10	Distinguished Consultant
11	VP Engineering
12	Professor, Vice-Director

5.2 Interview Findings

This section discusses the responses from the feedback sessions that have been conducted during the XP2016 conference and Agile Development of Safety-critical Software Workshop. We used the following categories in the final coding:

1. Safety requirements and functional requirements must be considered together
2. It is important to keep safety requirements and functional requirements separated
3. Safety expert needed to keep safety in focus and to make the right decisions
4. Needs to build a safety culture, not to rely on safety experts
5. QA-experts are needed
6. Difficult to get customer feedback
7. Safety-expert may be a bottleneck
8. Involve all competent stakeholders
9. Involve assessor
10. Tools needed
11. Daily stand-ups are not (should not be treated as just) status meetings
12. Analysis of architecture and safety up front
13. All requirements need a criticality score
14. Status / review meetings take too much time

Not all interviewees touched upon all categories. We have summed up who brought up which category in the following graph.

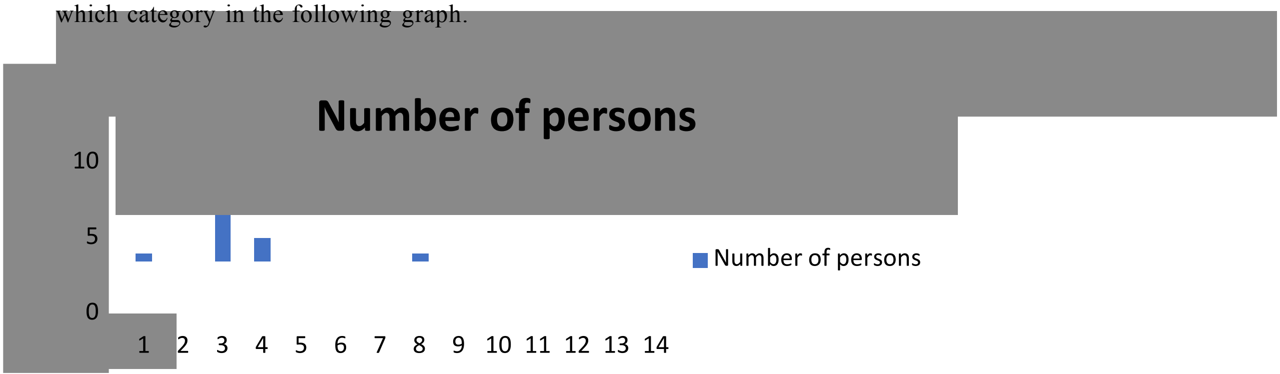

Figure 1- topics discussed by which interviewee

A. **Second product backlog – a safety backlog**

There are major differences of opinion among our respondents concerning the proposal in SafeScrum to introduce a second Product Backlog, the Safety Backlog, to track safety-critical concerns through the Scrum-based process. As will be seen from the extracts quoted below, the practitioners were split on this issue – four interviewees held the opinion that safety requirements and functional requirements should be handled together while two said that they should be separated. Note that "handled together" does not necessarily mean "in the same backlog". Nor does having separate backlogs mean a strict split – they may be kept in the same backlog but separated by tagging. However, some of the literature promotes the importance of a Security Backlog [15], to help to deal with the security "safety" issues in Scrum. The following extracts from our interviews indicate the practitioners' differing views:

The first feature to discuss is this need or otherwise to have an explicit safety backlog, the concept of safety backlog was agreed by the **[Interviewee2]** through experience and knowledge of 2 product backlogs

"*The safety requirements separate from the others because they are ... the first few requirements and the safety requirements and the fact that requirements somehow give rise to safety requirements and the other way around. It is nice to have them separated because they have to be treated differently. A lot of the thing also is that the safety requirements are usually at least more stable. The functional requirements would change much more often due to changed needs or so on. The problem is, of course, that the functional requirements will sometimes change and introduce new risks. But it opens up a new safety concern, then they would go back and they'll update the safety requirements and make sure it's linked to the original functional requirements.*" **[Interviewee1]** points out a different way of dealing with safety issues:

"*We do not have 2 backlogs, we use only one backlog, so we mark every issue with criticality, complexity and risk, as well we have three levels low medium and high, we discussed if we could mark issues to safety issues and non-safety issues, we need to distinguish between issues and requirements.* **[Interviewee8]** suggested to build safety features into the evaluation of everything, validating that every change still fulfills the safety need, through automating those checks: "*If you want another way, it's closer to a checklist that you do all the time. Yes. Basically, this checklist applies to any feature you implement. Whatever it is, you go through the activities to decide. Keep in mind; these are the safety implications for this feature because when you discuss the functionality, you may flush them out. We have to go against the checklist just to make sure that all the important things are done and none is forgotten. Some of them apply always. Some of them maybe apply only in some circumstances*" **[Interviewee6]** supported the need for safety backlog, "*It seems to me quite reasonable but of course it's starting to be very much like some requirements, documents, safety requirements, so I'm all for it*".

The general view is that addition of the Safety Backlog may be a good idea, but that we need to conduct more investigation to see if this is true. The interviews show that there is a question about the granularity of what should go into the backlog. One of the challenges is to keep track of the safety requirements explicitly since they require a different process that might not fit in with agility. This needs to be addressed by tailoring the agile process.

B. Safety team member

Almost all of the interviewees felt that our proposed addition of a team member with specific responsibility for safety issues was a sensible idea. We present extracts from the interviews below:

[Interviewee8] "*Yes, could be. Somebody in the team that is an expert about safety and so while the things are discussed and designed, he's there to ask the questions, so to make sure safety is taken into consideration because being the expert, he's the one that can ask the right questions to bring the problems to the surface*". **[Interviewee1]** mentioned to a new quality assurance (QA) role. We would, however, never instantly put QA and safety together because it's two different things, but we're saying there should be somebody who is the keeper of the flame of the safety issues that understands what's in the backlog. "*We have been trying to do quality assurance role in Scrum, the product owner is problem in Scrum, because he or she does not exist as a person, in our situation the product owner is a set of persons, we have a product owner but he is the marketing person, he does not know the system. We will start to define QA, the problem if you have too many, and I think the QA can be like a safety team member."*

The implementation of the safety team member can help to spread the safety culture into development team. **[Interviewee2]** *"There should be somebody who say, have you, well, they should be doing safety analysis because you have taken this user story, broken it down into 3 functions and somebody has to say, "Has he really done some safety analysis there?" Mostly, I think if you have a real safety culture the little person will mostly be able to do that, but somebody has to think, "Has it been done?"*

The idea behind the safety team member is to spread the safety culture across team to ensure safety. The safety team member will also improve communication between the development team and the independent assessor. This ensures that the safety requirements, safety criticality, and safety case will meet the customer requirements. In addition, it will help to ensure that the development team has a good understanding of the safety requirements, and helps spreading the safety culture.

C. Independent stakeholders are required as part of sprint review

The idea is having an identifiable, focal person within the team that is quote, "*The safety person*". The reason for that being having somebody who can be keeper of the flame. **[Interviewee11]** complained that independent stakeholders do not know enough about the features, as well, he mentioned to some concerns:

"*We tried it in each sprint review. But they never argue about what we are doing, what I think is to find someone who disagrees with you and has experience. So, you need to find a person whose only job is to disagree with you. And it is difficult to get a company to hire someone*." Contrariwise **[Interviewee6]** had good experience on dealing with independent stakeholders: "*I think it is a good point, if you have an independent stakeholder, like in our case was TÜV regulatory and standards services compliance", so I probably ask that person, not in each sprint but when you realize something."* **[Interviewee3]** explained his interest with some concerns: "*I would fear that such a person would become a bottleneck, in the sense that I need something, but I'm fifth in line. What I would probably hope for, would be that that person, or people* *would be over-resourced, in the sense that it would be desirable if they were a developer, that did work with the other developers, but also had this special role, rather than they only had this special role."* **[Interviewee5]** introduced an important link between the independent stakeholders and the daily Scrum meetings: "*Having someone responsible on the customer side almost directly implies there's also going to be someone on the development side as well, these Scrum meetings, we're saying that we think it's possible, even if only in a very light touch way that safety can be reviewed or highlighted as part of daily Scrum meetings."*

However, the response to the questions related to stakeholders varies quite a lot. There seems to be two important messages: (1) it is difficult to get good feedback and (2) it is important to involve all stakeholders – e.g. safety experts and assessors. In addition, there are some worries, e.g. safety experts might be a bottleneck in the process.

D. Daily Scrum meetings

The daily scrum meetings are important both to give informal feedback on problems and solutions. In addition, it enables the safety expert to influence solutions that directly or indirectly influence safety. This will, in the long term, help to create a safety culture in the team. There are two concerns with a negative impact: (1) the daily stand-ups must not degenerate to a status meeting and (2) a strong focus on safety can make the availability of the safety expert a bottleneck in the process.

Almost all the interviewees found that the daily scrum meeting useful, was also agreed by interviewees 1, 5 and 12 that the Daily scrum meetings can help the safety issues, especially safety be on the top of daily scrum meeting agenda:

"*This is a good idea, as a Scrum-master I like the idea of asking [a] question every day: Have you checked safety-related issues?"*

"I mean, that's important because you have what they're going to do, what you have done, you having problems. You should also get a question, "Have you done anything unrelated, that aren't on the safety requirements down here?" That's important just to keep safety on top of the agenda all the time."

"I completely agree, it is very important to have the safety requirements in the daily Scrum meeting and the safety person should always be at the daily Scrum meeting because of the early feedback. [This] can be used to provide an early opportunity to identify unintended side-effects emerging from a chosen implementation approach."

Daily scrum meetings can provide informal feedback of progress and satisfaction of safety requirements.

E. Sprint planning

Before you start the Scrum development, the chosen modeling, implementation and development assessment approach and the tools to be used should be documented and assessed against the systems' criticality. The split between those who want to keep safety requirements and functional requirements together versus those who want to keep them apart is visible also here. We also register the need for an architectural analysis in the sprint planning process.

[Interviewee10] has a deeper view of planning safety requirements and it is sensible idea. "*Yes, but there's always the question. ...When [do] you start the sprint planning? You'll take functions out of functional backlog, but they're only used as stories and you have to detail them, turn them into functions. Doing that, you might introduce new safety problems, meaning that you'll have to maybe look at, "OK, this requirement, this link to this safety requirement, maybe that should be changed also." This is somewhat traceability issue."* In Scrum, every iteration begins with a sprint-planning meeting. Another view was introduced on organising the sprint planning **[Interviewee 7]** "*safety requirements need to be checked, at least, they have addressed all the safety questions. What will happen if ...]"* **[Interviewee11]** *"The criticality of chosen requirements for the sprint (or requirements related to those chosen) should be highlighted at the beginning of the sprint planning. Because the team should know how critical that particular requirement is for the sprint. Yes, hazard analysis should be conducted to on the chosen languages, processes and tools to identify the potential for the introduction of implementation errors."*

F. Sprint review

How much we can get done in sprint reviews? Ideally, we should check whether somebody has fully satisfied a safety requirement and review the verification evidence. The main concern for a sprint review is that it is time-consuming, therefore we are recommending that hazard analysis should be an activity that takes place within a sprint.

[Interviewee7] *"I agree with you, but: If you should go through everything, which is dumb, people lose interest. For example a person, like QA person's look at the general work of the team, see how long has this been done, has everything been done properly? Have you looked at all the failing models, for instance? Sometimes you need somebody to do that."* **[Interviewee5]** has a different approach on dealing with sprint reviews: "*We don't use Sprint reviews as in "Scrum", In our project we use retrospective to improve safety, it takes half a day, so that is the time were you really look at the whole process, so we invite all the stakeholders, development team, safety person, product owner and the manager that is the big retrospective. As well we do small retrospective each week from one to two hours, and hazard analysis should be during the sprint not only in the review.*

Important issues to consider are participation of a safety team member and that the team has a safety culture. The presence of a safety-culture will, in the long term, remove the need for a separate safety expert, which might be a bottleneck in the review process.

5.3 Conclusion and Issues Arising

We have reported on the results of 12 semi-structured interviews which were conducted in order to gain actual practitioners' reactions to our approach to integrating the 4+1 safety assurance principles within Scrum, and our initial assessment of the challenges associated with the approach. Although the research sample is limited, the study has benefitted from the introduction of a more pragmatic perspective, which complements our rather theoretical viewpoint.

We summarise the results in forms of answers to the two research questions we posed at the beginning of the study:

RQ1 *What are the current concerns and opportunities voiced by safety-critical systems professionals regarding the use of agile development methods for safety-critical systems development?*

We encountered some difficulties during the interviews. For example, on aspect that we briefly touched on was misunderstanding and lack of knowledge or awareness from both the Agile and the safety practitioners concerning each other's outlooks and work. Nonetheless, the semi-structured interviews have motivated us to propose further work involving interviews on a much larger scale in order to achieve better results. We need to move from the basic questions that we asked during the current research – i.e. "is it feasible to integrate safety into Agile methods?" – towards more specific questions, such as the following:

- Is it permitted to break a safety requirement which has been satisfied in one Sprint?
- Safety backlog - Yes or No?
- Safety team member - Yes or No?
- Hybrid agile approach - Yes or No?
- Sprint duration: 1 to 2 weeks? Or longer?
- Independence is particularly important for the developers of the verification procedures (testing, formal methods, etc.) Surely they should participate in the review as well?
- How will the software safety requirements be verified?
- Would it be desirable/possible to implement a "feature" in a particular Sprint, but NOT to implement its related safety requirements in the same Sprint?
- Is Scrum a good way to produce a prototype (as recommended in 26262, 61508)?

RQ2 *What changes are necessary to the Scrum Process in order to address the 4+1 Software Safety Assurance Principles*?

The findings indicate clear support for the recommendations that we propose to integrate the 4+1 safety assurance principles into the Scrum process [16] in order to help demonstrate compliance with safety standards, and with our initial survey of the challenges presented by such an approach. Our recommendations stem from the use of the 4+1 principles to build on the strengths of the Scrum process to improve management of safety issues in system development.

Ultimately, this research will develop and evaluate a process model of an adapted version of Scrum that clearly integrates the activities of software safety requirements evolution, software hazard analysis and software safety (assurance) case development. To support the assurance case development aspect of this process, the results from the survey and semi-structured interview have provided a clear direction in terms of the importance of incremental hazard analysis, safety requirements development, and assurance case development (i.e. they indicate clearly that these activities must be performed within an incremental, rather than simply being up-front or end-of-development activities).

5.4 Limitations

Our study suffered from some limitations which should be addressed in future work:
a) We had considerable difficulty finding practitioners who were sufficiently expert in both Agile and safety – in the end, we were able only to interview those with an interest in the integration of agile and safety;
b) The limited research sample meant that not enough data was collected from the practitioners;
c) We needed to establish specific criteria in order to avoid deviation from the interview script.

5.5 Future Work

It would be desirable for future researchers to conduct a pilot project. This should be formulated in such a way as to address the particular themes that emerge from our survey: for example, the pilot project could evaluate how difficult it was to establish safety requirements at the outset, and how much they change during the project.

Our initial survey in this area highlighted some areas of interest in the role of the safety case. Further work is required to explore how GSN safety cases could be linked to a notion of safety Product backlog within Scrum. The research indicates that existing assurance case activities need to be adjusted.

Software safety argument patterns provide a way of capturing good practice in software safety arguments. Future research could develop a pattern-based approach to integrating software safety cases, Scrum’s Safety Product Backlog, risk-based planning, and requirements-based evaluation. Software safety argument patterns describe the nature of the argument and safety claims that would be expected for any software safety case.

It would be useful for future researchers to engage in a larger-scale interview-based evaluation of an approach for safety case development within Scrum. In particular, research should address the development of pragmatic techniques to ensure that evidence to validate the safety case is developed and collected in all incremental (Sprint) processes.

A realistic case study should be developed, to investigate where there are opportunities to build up a safety case as a part of an Agile development, to determine the risks and conflicts associated with this approach and how these risks could be mitigated.

Reference:

1. Bowers, J., May, J., Melander, E., Baarman, M., Ayoob, A.: Tailoring XP for Large System Mission Critical Software Development. In: Wells, D., Williams, L., (Eds.): XP/Agile Universe, LNCS 2418, pp. 100–111. © Springer-Verlag Berlin Heidelberg (2002).
2. Bedoll. R. A Tail of Two Projects: How 'Agile' Methods Succeeded after 'Traditional' Methods Had Failed in a Critical System-Development Project. In: F. Maurer and D. Wells (Eds.): XP/Agile Universe, LNCS 2753, pp. 25–34. Springer- Verlag Berlin Heidelberg (2003).
3. Bruce,D. 19 February 2013 (Agile analysis practices for safety-critical http://www.ibm.com/developerworks/rational/library/agile-analysis-practicessafety-critical-development/
4. Kelly, Tim. "Software Certification: Where is Confidence Won and Lost?" Addressing Systems Safety Challenges, T. Anderson, C. Dale (Eds), Safety Critical Systems Club (2014)
5. O. Doss and T. P. Kelly. 2016. Challenges and Opportunities in Agile Development in Safety Critical Systems: A Survey. SIGSOFT Software Engineering. Notes 41, 2 (May 2016), 30-31.
6. B. Fitzgerald, K. J. Stol, R. O, Sullivan and D. O. Brien, "Scaling agile methods to regulated environments: An industry case study," 2013 35th International Conference on Software Engineering (ICSE), San Francisco, CA, 2013, pp. 863-872.
7. Jonsson, S. Larsson and S. Punnekkat, "Agile Practices in Regulated Railway Software Development," Software Reliability Engineering Workshops (ISSREW), 2012 IEEE 23rd International Symposium on, Dallas, TX, 2012, pp. 355-360.
8. Kelly, T.; Weaver, R.: The Goal Structuring Notation – A Safety Argument Notation. In: Proceedings of the International Conference on Dependable Systems and Networks – Workshops on Assurance Cases, Florence, Italy, (2004)
9. Stålhane, T., Myklebust, T., Hanssen, G. The application of Scrum IEC-61508 certifiable software, Unpublished, (2011).
10. Konstantin Beznosov and Philippe Kruchten. 2004. Towards agile security assurance. In Proceedings of the 2004 workshop on New security paradigms (NSPW '04). ACM, New York, NY, USA, 47-54.
11. L. b. Othmane, P. Angin and B. Bhargava, "Using Assurance Cases to Develop Iteratively Security Features Using Scrum," Availability, Reliability and Security (ARES), 2014 Ninth International Conference on, Fribourg, 2014, pp. 490-497
12. Flink, A.: The Survey Handbook, 2nd edn. Sage Publications, Thousand Oaks (2003)
13. Shull, F., Singer, J., Sjøberg, D.I.K.: Guide to Advanced Empirical Software Engineering 1st edn. Springer, London (2010)
14. Osama Doss, Tim Kelly "The 4+1 Principles of Software Safety Assurance and Their Implications for Scrum" (Eds.): XP 2016, LNBIP 251, pp. 1–5, 2016
15. Z. Azham, I. Ghani and N. Ithnin, 'Security backlog in Scrum security practices' Software Engineering (MySEC), 5th Malaysian Conference in, Johor Bahru, 2011, pp. 414-417
16. Osama Doss and Tim Kelly. 2016. Addressing the 4+1 Software Safety Assurance Principles within Scrum. In Proceedings of the Scientific Workshop Proceedings of XP2016 (XP '16 Workshops). ACM, New York, NY, USA, Article, 17, 5 pages. DOI: 10.1145/2962695.2962712