Trends in Agile Development of Safety-Critical Software: A Summary of the 3d International Workshop on Agile Development of Safety-Critical Software (ASCS 2017)

Geir K. Hanssen
Sintef Digital
Norway
ghanssen@sintef.no

Thor Myklebust
Sintef Digital
Norway
Thor.Myklebust@sintef.no

Stig Ole Johnsen
NTNU
Norway
Stig.O.Johnsen@sintef.no

Osama Doss
Leeds Beckett University
osamadoss@gmail.com

ABSTRACT

Agile development of safety-critical software has evolved from an early conceptual idea to, presently, an approach that is gaining uptake in the industry. As we now get more and more experience we also discover new challenges and related ideas that needs further investigation. The third international workshop on agile development of safety-critical software (ASCS) gathered some of the leading researchers and practitioners in the field to discuss recent ideas and developments. This paper presents an overview of the motivation and background for the workshop and the talks that were given.

CCS CONCEPTS

• Software and its engineering → Agile software development • Software and its engineering → Software safety

KEYWORDS

Agile software development, safety-critical software.

1 INTRODUCTION

Development, certification and maintenance of safety-critical software systems is complex and costly. In particular, having a high safety integrity system certified according to mandatory standards such as IEC61508 (process), DO178C (avionics) or EN50128 (railway) is fundamental to keep a competitive advantage but also one of the most severe cost drivers. An estimated 25-50% of total costs may be related to documentation of proof of compliance to standards and the assessment by external certification bodies.

The trend of implementing larger parts of safety systems in software has led to a growing interest in agile software development methods and techniques to improve performance with respect to development efficiency, system quality and safety integrity, as well as resource optimization and effective assessment and certification [3]. This raises a series of challenges, for example how to adapt agile principles to large and complex projects, how to implement changes in a conservative and plan-driven practice, how to involve external certification and notified bodies, and how to enable efficient and cost effective traceability and documentation management.

The third international workshop on agile development of safety-critical software (ASCS) gathered a mix of practitioners and researchers to address industrial and scientific challenges related to the adoption and exploitation of agile methods and techniques to improve development and certification of safety-critical and high-integrity systems.

The ½-day workshop opened with a keynote: "Experiences with the STAMP/STPA method for

ASCS 2017, 3rd International Workshop on Agile Development of Safety-Critical Software, 26th May, Cologne, Germany.

hazard analysis and its application to security and privacy” by professor Stefan Wagner from the University of Stuttgart. Further, the workshop consisted of two paper presentations: “The Dynamics of Agile Practices for Safety-Critical Software Development” by Peter Axel Nielsen from Aalborg University, and “A Study of Safety Documentation in a Scrum Development Process” by Yang Wang from the University of Stuttgart. The workshop also included an invited talk “The Agile Safety Case” by Thor Myklebust from SINTEF Digital.

2 KEY ASPECTS FROM PRESENTATIONS AND DISCUSSIONS

The keynote by Stefan Wagner challenged some of the common assumptions of safety systems and considered the use of Nancy Levesons System-Theoretic Process Analysis (STPA) approach [1]. Wagner also looked into how STPA may be applied to dealing with security issues in safety systems, which is an increasingly important challenge. The keynote was motivated by the realization that high system reliability is not sufficient for safety, we also need to consider e.g. operator behaviour as a product of the environment and that risk and safety may be best understood and communicated in ways other than probabilistic risk analysis. Software is reliable but unsafe when:

- The software correctly implements the requirements, but the specified behaviour is unsafe from a system perspective.
- The software requirements do not specify some particular behaviour required for system safety (that is, they are incomplete).
- The software has unintended (and unsafe) behaviour beyond what is specified in the requirements.

Wagners conclusion was that software systems fit well to system-theoretic analysis and that STPA may be applied to strengthen analysis of security and privacy issues in safety systems, and that STPA may very well be integrated in an agile development process [5].

Peter Axel Nielsen presentation was based on a review of 54 articles on safety-critical software and agility (in collaboration with Lise Tordrup Heeager). Based on the review he proposed a framework showing dynamic relations between four common areas of concern in agile methods: flexible requirements in user stories, light documentation, iterative & incremental lifecycle, and test-first strategy [2]. He explained five relationships between these concerns and claims that they are mutually dependent in the sense that altering one concern will affect the others. The framework was exemplified through a medical device case, showing that it may provide a clearer understanding of what happens when introducing agile process in safety-critical development.

Yang Wang presented a study of safety documentation in a Scrum development process (in collaboration with Ivan Bogicevic and Stefan Wagner). She considered how the format of safety-related documentation can be improved to enable better communication in a development project. Through a case study [4] based on participant observations, reviews of Scrum artefacts and documentation and a combination of questionnaires and interviews, she concluded that a safety story pattern and a safety epic pattern are strongly suggested to be used for reducing communication problems in Scrum for safety-critical systems. She also found that the agile safety plan had little positive effect on communication.

Thor Myklebust explained the concept of agile safety cases. Safety cases are (like in court) structured arguments supported by evidence, intended to justify that a product or system is acceptably safe for a specific application in a specific operating environment. Safety cases are used in some domains, such as rail, where EN 50129:2003 include requirements for safety cases. An agile safety case should be managed adaptively and be flexible in order to insert information when it becomes available; throughout an agile development process. Experience shows that it takes a lot of experience to master safety cases and that there is little training material available so far. Myklebust however concludes that safety cases fit well with agile processes and that they can be constructed incrementally.

Based on the talks and the following discussions we see that the trend of applying agile processes to the development and certification of safety-critical software is maturing and gaining uptake. This, however, opens a lot of new questions and challenges that needs to be resolved to achieve the full potential.

3 ACKNOWLEDGEMENTS

The work on this summary and the ASCS workshop was partly funded by the Research Council of Norway under grant #228431 (the SUSS project). Thanks to the workshop participants and to the program committee.

REFERENCES

1. Abdulkhaleq, A., Wagner, S., and Leveson, N., *A comprehensive safety engineering approach for software-intensive systems based on STPA.* Procedia Engineering, 2015. **128**: p. 2-11.
2. Heeager, L.T., Nielsen, P.A., *Agility in Development of Safety-Critical Software: A Conceptual Model*. 2017.
3. Stålhane, T., Myklebust, T., and Hanssen, G.K. The application of Scrum IEC 61508 certifiable software. In proceedings of ESREL. 2012. Helsinki, Finland.
4. Wang, Y., Ramadani, J., and Wagner, S., *An Exploratory Study of Applying a Scrum Development Process for Safety-Critical Systems.* arXiv preprint arXiv:1703.05375, 2017.
5. Wang, Y. and Wagner, S. *Toward Integrating a System Theoretic Safety Analysis in an Agile Development Process*. in *Software Engineering (Workshops)*. 2016.