The 4+1 Principles of Software Safety Assurance and Their Implications for Scrum

Osama Doss(✉) and Tim Kelly

High Integrity Systems Engineering Research Group, Department of Computer Science, University of York, York YO10 5DD, UK
{osad500,tim.kelly}@york.ac.uk

Abstract. As part of our research concerning the integration of assurance case development with Scrum, we are planning to conduct semi-structured interviews with participants to gain feedback on a proposed approach. We will be interviewing individuals who have been involved with safety-critical systems development and Agile methods. Participants will be presented with an overview of the challenges associated with applying the 4+1 software safety assurance principles to Scrum. Initial recommendations concerning how the principles can be accommodated within a Scrum development will also be presented. Participants will be led through a series of questions to gain feedback on the feasibility of the approach, and for an assessment as to whether the 4+1 principles can be addressed without compromising agility. The motivation behind this research is to gain a deeper insight into the difficulties experienced when integrating assurance case in to Scrum process.

Keywords: Scrum · Safety · Assurance · Certification · Assurance case · Software safety

1 Research Aim

This study is part of the research under the High Integrity System Engineering Group, Computer Science Department, of the University of York. This paper introduces the 4+1 Principles of Software Safety Assurance [1] and their implications for Scrum [2], specifically, the impact on the processes, roles and artefacts associated with Scrum development.

Historically, there has been a reluctance to adopt agile methods within safety-critical systems development. However, feedback from our initial research in this area suggests that there are benefits to be gained from the application of agile methods to safety critical systems [3, 8]. Following this feedback we have done further work to assess how the 4+1 principles of software safety assurance can be integrated with Scrum, and have developed an initial proposal for how Scrum could be modified to better address the principles. The aim of the proposed study at XP2016 is gain practitioner feedback on these proposals. The feedback we receive will ultimately be used to help refine the proposal before further empirical evaluation.

H. Sharp and T. Hall (Eds.): XP 2016, LNBIP 251, pp. 286–290, 2016.
DOI: 10.1007/978-3-319-33515-5_27

2 Research Questions and Their Motivations

Our research, as a whole, is focused on answering the following questions:

- 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 Can the integration of incremental assurance case development and evaluation within the existing "Scrum" methodology alleviate the concerns identified in answer to RQ1?
- RQ3 What changes can Scrum Process has to undertake in order to be compliant with the safety standard?

This study, specifically furthers our investigation into RQ2 and 3. We now how have initial proposals for changes and a description of assurance case development integrated with Scrum. However, what we lack is a substantial and varied practitioner base to help assess the credibility, feasibility, and efficacy of our proposals.

3 Importance of Research

Despite progress in the use of agile development methods in safety critical systems development (e.g. [4]), there are still those with doubts about the potential for successful integration. There are also reported experiences [5] that highlight the complementary nature of the iterative and incremental approach underlying many agile methods, and recognised best practice in risk management in safety critical systems development. Rather than start with a theoretical evaluation of the compatibility of the principles of agile development with software safety assurance, we decided to draw out these experiences, opinions (and possibly preconceptions) by means of a practitioner's semi-structured interview. In particular, our first round of semi-structured interviews drew out specific responses relating to (possible) incremental and iterative nature of safety requirements development, hazard analysis and safety (assurance) case developments. The responses we received showed both the potential for benefits from agile development of safety-critical software, together with residual concerns about the ability to provide software safety assurance in a manner compatible with current software safety assurance standards. Rather than focusing on a single safety assurance standard (as some have done, e.g. [4]) we have used the framework of the 4+1 software safety assurance principles to tackle the common and broad issues of software safety assurance that exist across multiple industry domains and safety standards. These principles have been developed to highlight the commonality of purpose of multiple existing safety standards, and are being adopted by industry (e.g. in Defence Standard 00-55) as a framework against which software safety assurance can be judged.

The importance of this research is that it represents one step along the path of pushing beyond simplistic and over-generalised preconceptions of the compatibility of agile and safety-critical systems development, and potentially unlocking the benefits of agility within the safety domain.

4 Data Collection Methods to Be Used, Including

- ***Who the participants should be***

It is not easy task to find practitioners with both experience in the field of Agile and Safety. However, XP2016 will involve various categories of experts (software engineers, industry and academia etc.) as well there being a significant opportunity to link this study with the XP2016 workshop Agile Development of Safety-critical Software (ASCS).

We would like to interview individuals who have been involved with Safety Critical-Systems, Agile methods, or both, during XP2016 in order to use their experience and insight to gain feedback on our proposed approach.

- ***What methods will be used and why these have been selected.***

The study will be conducted as a qualitative survey using "semi-structured interviews" for data collection [6]. Shull et al. [7] illustrate the advantage and disadvantage of conducting 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 responses received from XP2016 will also be compared with 1-to-1 semi-structured interviews conducted with some of the respondents from our initial survey [3]; the purpose of this interview study is to investigate the success of the proposed integration of 4+1 principles and assurance safety case development with Scrum.

Interviews will be conducted face-to-face at the XP2016 location. Further interviews (with further participants) may be conducted over phone. Interviews will be recorded and transcribed to facilitate subsequent analysis.

- ***What will happen during data collection activity?***

Participants would take part in an approximately 40 min interview to explore perceptions around the 4+1 Principles of Software Safety Assurance and their implications for Scrum.

The interview will start by introducing the research aims and the topics to be discussed. Then the 4+1 principles will be explained, together with an outline of the proposal for integrating these principles within a Scrum development. Questions will then be asked relating to the proposal – picking out specific features one-by-one (e.g. our recommendations for team composition). The questions will tackle both aspects of (a) whether the proposed approach challenges agility and (b) whether the proposed approach challenges safety assurance.

Documents that will be prepared for the interview:

1. Interview guide - main pointers to guide the interview
2. Information sheet - to be provided to the interviewee to provide the context of the interview
3. Consent form - for interviewee to sign.

5 Data Analysis Methods to Be Subsequently Used

Transcripts will be analysed using thematic analysis. One researcher will read all of the interview transcripts, and will code the transcripts using first-cycle coding (Open Coding or Initial Coding), supported by the NVivo 11 software package. Main categories (or topics) will be identified through clustering of codes in project review meetings. Codes and categories will be constantly compared with the data and revised or refined as appropriate.

The results of the thematic analysis will then be written up in a form suitable for sharing with participants and subsequent publication.

6 How the Results Will Be Used

Initially, we will present the key findings within the High Integrity System Engineering Group at York. The findings will also potentially form part of the final thesis of the ongoing PhD research on Assurance Case Integration with An Agile Development Method. Our findings will be submitted, in the future, for publication in peer-reviewed journals.

Ultimately, the findings will be used to refine our proposed approach before proceeding to empirical case studies.

References

1. Kelly, T.: Software certification: where is confidence won and lost? addressing systems safety challenges. In: Anderson, T., Dale, C. (eds.) Safety Critical Systems Club (2014)
2. Rubin, K.S.: Essential Scrum: A Practical Guide to the Most Popular Agile Process, 1st edn. Addison-Wesley Professional, Upper Saddle River (2012)
3. Doss, O., Kelly, T.: Challenges and opportunities in agile development in safety critical systems – a survey. In: Agile methods applied to development and certification of safety-critical software Workshop, XP 2015, Helsinki, Finland (2015)
4. Stålhane, T., Myklebust, T., Hanssen, G.K.: The application of safe scrum to IEC 61508 certifiable software. ESREL (2012)

5. Bedoll, R.: A Tail of Two Projects: How ‘Agile’ Methods Succeeded after ‘Traditional’ Methods Had Failed in a Critical System-Development Project, Extreme Programming and Agile Methods-XP/Agile Universe 2003, pp. 25–34. Springer, Heidelberg (2003)
6. Flink, A.: The Survey Handbook, 2nd edn. Sage Publications, Thousand Oaks (2003)
7. Shull, F., Singer, J., Sjberg, D.I.K.: Guide to Advanced Empirical Software Engineering, 1st edn. Springer, London (2010)
8. Doss, O., Kelly, T.: Assurance case integration with an agile development method. XP 2015, LNBIP, vol. 212, pp. 347–349 (2015)