Toward Integrating a System Theoretic Safety Analysis in an Agile Development Process

Yang Wang, Stefan Wagner
Institute of Software Technology
University of Stuttgart, Germany
{yang.wang, stefan.wagner}@informatik.uni-stuttgart.de

Abstract: Agile development methodologies are becoming a tendency in today's changing software development. However, due to a lack of safety assurance activities, especially safety analysis, agile methods are criticized for being inadequate for the development of safe software. In this paper, we introduce an agile "Safe Scrum" by mapping a novel systematic safety analysis method, called STPA (System-Theoretic Process Analysis) into an existing agile development process "Safe Scrum" for safety-critical systems. This work is done by (1) performing safety-guided design inside each sprint, and (2) replacing the traditional RAMS (Reliability, Availability, Maintenance, and Safety) validation. We aim to extend Safe Scrum by integrating STPA, to find a balance point between Safe Scrum and basic Scrum.

1 Introduction

To apply agile methodologies into safety-critical systems, most research prefers combining agile methods with traditional development processes relying on safety standards. However, little emphysis is put on the nature of agile techniques, which prevents the proceeding of traditional safety assurance activitie−a continuously changing architecture design. **Problem Statement:** (1) the lack of safety assurance activities, especially safety analysis in agile methodologies prevents the application of agile methods in safety-critical systems. (2) Current safety analysis technologies are inadequate for agile methods due to a lack of a stable architecture design. **Research Objective:** The main objective of this article is to integrate a novel systematic safety analysis technology STPA [Lev11] into an existing agile development process "Safe Scrum" [SMH12] guiding safe architecture design. **Contribution:** (1) STPA drives the safe architecture design in Safe Scrum. (2) Applying STPA to the final product of each sprint instead of the traditional complicated RAMS (Reliability, Availability, Maintenance and Safety) validation, which makes Safe Scrum more agile. (3) we introduce the first concept of mapping STPA into agile software development processes. This article is a position paper, where the extension of Safe Scrum is still in the concept stage. A student project is planned in the future to further explore and investigate the development process.

2 Related Work

There has been little experience published on the utilization of safety analysis technologies in agile development methodologies. Most of the research is from the viewpoint of a hybrid

development process model to accept agile methods in safety-critical systems. Safe Scrum, proposes by Stålhane, Myklebust and Hanssen [SMH12], is motivated by the need to make it possible to use methods that are flexible with respect to planning, documentation and specification while still being acceptable to IEC 61508 [IEC04], as well as making Scrum a practically useful approach for developing safety-critical systems.

Undoubtedly, Safe Scrum is a considerable success for its innovative combination. However, too much adherance to the safety standard IEC 61508 makes the process lacking agility. First, all the additional safety assurance activities are kept outside Scrum. Little focus is put on the incremental architectures inside each sprint. Second, each sprint in Scrum should be swarming [Rub12] rather than sequential as mini waterfall or mini V-model [SMH12]. We believe that safety-guided design is strongly needed in agile methods instead of the purely "add-on" safety assurance.

In addition, Ge et al. [GPM+10] published an iterative approach to develop safety-critical software. Vuori [Vuo11] proposed a hybrid model like Safe Scrum. However, both of them suggest an up-front design, which is not recommended in Scrum [Coh10].

In comparison to the research above, we try to find out a suitable safety analysis technique for agile methods, to abandon this heavy weight architecture ahead and still keep safety in agile methods.

3 Background

We suggest STPA to confront the aforementioned problems. It is a new hazard analysis technique based on systems thinking and a new model of accident causation based on systems theory rather than reliability theory. It consists of two main steps: (1) Identify the potential for inadequate control of the system that could lead to a hazardous state. (2) Determine how each potentially hazardous control action identified in step 1 could occur [Lev11]. We recommend this novel technique for two reasons: (1) The current safety analysis techniques, such as FMEA (Failure Mode and Effects Analysis) or FTA (Fault Tree Analysis), assume that accidents are caused by component failures, which is not true for software. The primary advantage of STPA is that it emphasises causal factors from the system view, such as component interaction accidents or cognitively complex human decision-making errors. (2) Current safety analysis techniques start from a complete design, which is not consistent to agile development methodologies. STPA, however, provides the necessary information to guide the design process.

4 Concept

In this section, we integrate STPA in Safe Scrum. To clarify our approach, we use the airbag system as an example.

We extend Safe Scrum in three aspects: (1) During each sprint we integrate STPA as safety-guided design. (2) At the end of each sprint, we use STPA on the product instead of a RAMS validation. (3) We replace the final RAMS validation with STPA. The other parts which are still kept consistent to Safe Scrum are: (1) The environment description and the SSRS phases 1-4. (2) Test Driven Development. (3) Safety product backlog. (4) A safety

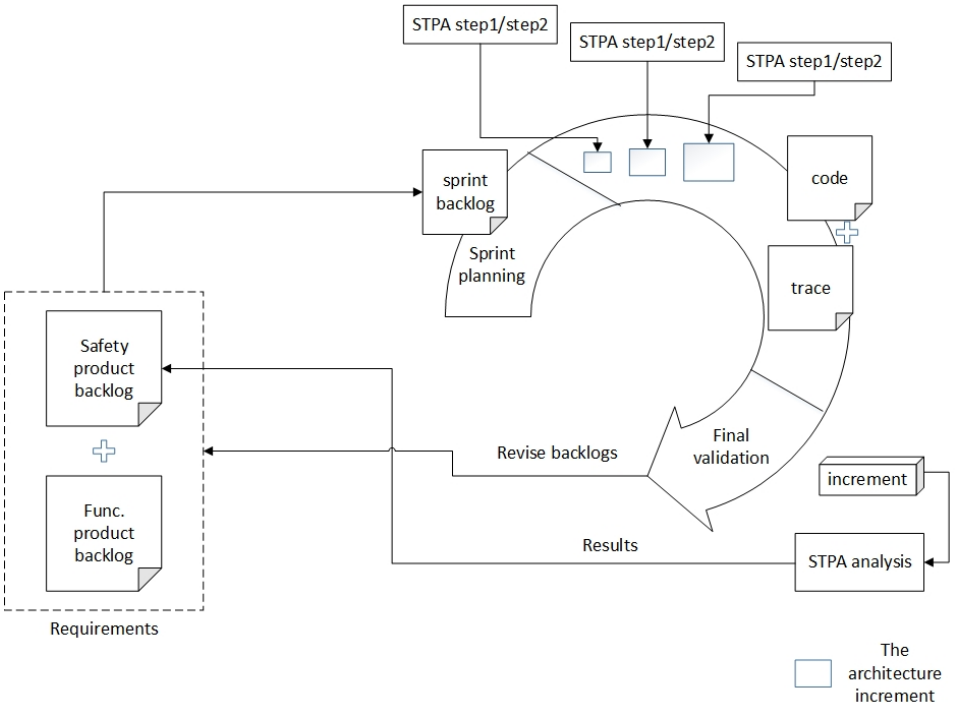

Figure1: Mapping STPA in Safe Scrum

expert. See Figure 1.

In STPA, the accidents are regarded as resulting from inadequate control. Thus, the control structure (architecture) is considered as the cross point between STPA and Safe Scrum.

We start from a general description of the environment and initial systems through SSRS phases 1-4 (concept, overall scope definitions, hazard and risk analysis and overall safety requirements) of IEC61508. An approximate safety target or safety-related targets are determined in up-front plannings and story boarding in agile [Mad10]. For example, in airbag system, we formulate the safety analysis results as:

Accident: The human beings in the target vehicle are injured, when a traffic accident occurred.

Hazard: The airbag is not ignited even though a critical crash occurred.

System requirement: The airbag shall be ignited, when a critical crash occurred.

During each sprint, we arrange STPA in development when there is a sufficient amount of new architecture design. The safety analysis results could be a driving force for the next architecture design. By applying STPA step 1 in each sprint, we fomulate:

Unsafe control action: A sensor delivers the wrong amplitude of the target vehicle.

Safety constraint: The data of the sensor must be checked to be right.

This constraint is translated as a safety requirement for the following architecture design. By applying STPA step 2, the factors that could lead to violate the safety constraints are determined. More detailed safety requirements are to be elicited depending on the stepwise system design.

After each sprint, a product is created. We finally apply STPA to it for the following reasons: (1) Getting a final safety assessment. (2) Combining with safety verification in the system level. (3) Driving the next sprint development.

All the safety analysis activities aforementioned are executed by a safety expert and the results are documented in the safety product backlog.

5 Conclusion and Future work

In this paper, we propose an agile "Safe Scrum" by integrating STPA to perform safety-guided design instead of adding traditional safety-related assurance methodologies on it. Traditionally successful safety analysis technologies are based on traditional development process. It′s not certain if they can solve the safety-related problems caused by the nature of agile development methodologies. Rather than a hybrid combination between traditional safety standard and agile development process, it would be a good direction from the standpoint of the nature in agile methods and try to find out more safety assurance technologies, which are commit to and get use of the agile principles. For future work, we focus on the verification between safety requirements from STPA and the code under development.

Literatur

[Coh10] Mike Cohn. *Succeeding with agile: software development using Scrum*. Pearson Education, 2010.

[GPM+10] Xiaocheng Ge, Richard F Paige, John McDermid et al. An iterative approach for development of safety-critical software and safety arguments. In *Agile Conference (AGILE), 2010*, Seiten 35–43. IEEE, 2010.

[IEC04] IEC. *IEC61508, Functional safety of electrical/electronic/programmable electronic safety-related systems.* International Electrotechnical Commission, 2010-04.

[Lev11] Nancy Leveson. *Engineering a safer world: Systems thinking applied to safety*. Mit Press, 2011.

[Mad10] James Madison. Agile architecture interactions. *Software, IEEE*, 27(2):41–48, 2010.

[Rub12] Kenneth S Rubin. *Essential Scrum: A practical guide to the most popular Agile process*. Addison-Wesley, 2012.

[SMH12] T Stålhane, T Myklebust und GK Hanssen. The application of Safe Scrum to IEC 61508 certifiable software. *Haettu*, 1:2014, 2012.

[Vuo11] Matti Vuori. Agile development of safety-critical software. *Tampere University of Technology. Department of Software Systems; 14*, 2011.