Combining STPA and BDD for Safety Analysis and Verification in Agile Development

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

ABSTRACT

Agile development is in widespread use, even in safety-critical domains. However, there is a lack of an appropriate safety analysis and verification method in agile development. In this poster, we propose the use of Behavior Driven Development for safety verification with System-Theoretic Process Analysis for safety analysis in agile development. It shows a good capability on communication effectiveness through a preliminary controlled experiment.

CCS CONCEPTS

• **Software and its engineering → Agile software development**; *Software verification*; *Software safety*;

KEYWORDS

Agile Development; Safety Verification; Safety-Critical Systems

ACM Reference Format:
Yang Wang, Stefan Wagner. 2018. Combining STPA and BDD for Safety Analysis and Verification in Agile Development. In *Proceedings of 40th International Conference on Software Engineering Companion (ICSE '18 Companion).* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3183440.3194973

1 INTRODUCTION

Agile practices have been widely used in software industries to develop systems on time and within budget with improved software quality and customer satisfaction [7]. The success of agile development has led to a proposed expansion to include safety-critical systems [3]. However, to develop safety-critical systems in an agile way, a significant challenge exists in the execution of safety analysis and verification [4]. The traditional safety analysis and verification techniques, such as failure mode effect analysis (FMEA), fault tree analysis (FTA), hazard and operability study (HAZOP) are difficult to apply within agile development. They need a detailed and stable architecture design [1].

In 2016, we proposed to use System-Theoretic Process Analysis (STPA) [5] in agile development for safety-critical systems [11]. First, STPA can start without a detailed and stable architecture. It can guide the design. Second, Leveson developed STPA based on the systems theoretic accident modeling and processes (STAMP) causality model, which considers safety problems based on system theory rather than reliability theory. In today's complex cyber-physical systems, accidents are rarely caused by single component or function failures but rather by component interactions, cognitively complex human decision-making errors and social, organizational, and management factors [5]. System theory can address this.

The safety requirements derived from STPA need verification. However, there is no congruent safety verification in agile development. Most agile practitioners mix unit test, integration test, field test and user acceptance testing (UAT) to verify safety requirements [3]. In 2016, we proposed using model checking combined with STPA in a Scrum development process [11]. However, using model checking, a suitable model is necessary but usually not available in agile development. In addition, the formal specification increases the difficulties to communicate. BDD, as an agile technique, is an evolution of test driven development (TDD) and acceptance test driven development (ATDD). It relies on testing system behavior by implementing a template to describe scenarios and generate test cases with three main steps: Given[Context], When[Event], Then[Outcome] [2]. The context describes pre-conditions or system states, the event describes a trigger event, and the outcome is an expected or unexpected system behavior. Yet, it has not been used to verify safety requirements. In this article, we propose STPA-BDD. We believe that BDD might be suitable for safety verification with STPA for safety analysis in agile development.

2 RELATED WORK

Modern agile development processes for developing safety-critical systems advocate a hybrid mode through alighment with standards like IEC 61508. There are many successes [9]. However, a lack of integrated safety analysis and verification to face the changing architecture through each short iteration is a challenge for using such standards. In 2016, we proposed to use STPA in a Scrum development process. It showed a good capability to ensure agility and safety in a student project. However, we verified the safety requirements only at the end of each sprint by executing UAT together with TDD in development. A lack of integrated safety verification causes some challenges. The previous research regarding safety verification in agile development suggested using formal methods. However, they need models and make intuitive communication harder. In addition, they have not considered specific safety analysis techniques. BDD is specifically for concentrating on behavior testing. It allows automated testing against multiple artifacts throughout the iterative development process. Moreover, it bridges the gap between natural language-based business rules and code language. Okubo et al. [8] mentioned the possibilities of using BDD for security and privacy acceptance criteria, while Lai, Leu, and Chu [6] have combined BDD with iterative and incremental development for security requirements evaluation.

ICSE '18 Companion, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5663-3/18/05.
https://doi.org/10.1145/3183440.3194973

3 STPA-INTEGRATED BDD FOR SAFETY ANALYSIS AND VERIFICATION (STPA-BDD)

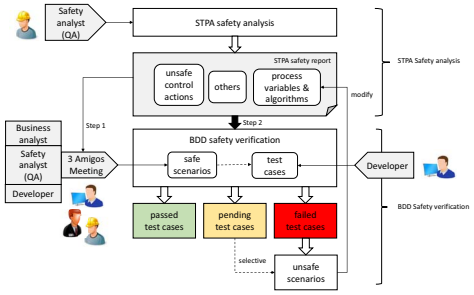
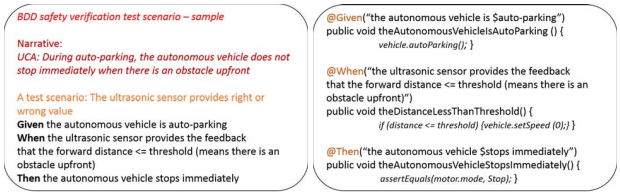


Figure 1: STPA-BDD Concept

In this article, we propose STPA-BDD. We mainly focus on safety verification. As we can see in Figure 1, we have two main parts: STPA safety analysis and BDD safety verification. A safety analyst[1] (QA) starts performing STPA safety analysis with a sufficient amount of code[2]. STPA is executed by firstly identifying potentially hazardous control actions, and secondly determining how unsafe control actions (UCAs) could occur. STPA derives the safety requirements, which constraint the UCAs, as well as system behaviors. Additionally, it explores the causal factors in scenarios for each UCA. The output from the safety analyst (QA) is an STPA safety report with system description, control structure, accidents, hazards, UCAs, corresponding safety requirements, process variables[3] and algorithms[4].

In BDD safety verification, to generate and test scenarios, the UCAs (in STPA step 1), process variables and algorithms (in STPA step 2) from the STPA safety report are needed. We write other data into "others". BDD safety verification has two steps: In step 1, the business analyst, the safety analyst (QA) and the developer establish a "3 Amigos Meeting" to generate test scenarios. In a BDD test scenario[5], we write the possible trigger event for the UCA in **When [Event]**. The other process variables and algorithms are arranged in **Given [Context]**. **Then [Outcome]** presents the expected behavior - a safe control action. In Figure 2 (a), we present an example. The safety analyst (QA) has provided a UCA as *During auto-parking, the autonomous vehicle does not stop immediately when there is an obstacle upfront.* One of the process variables with relevant algorithms detects the forward distance by using an ultrasonic sensor. The developer considers a possible trigger as the ultrasonic sensor provides the wrong feedback. Thus, a BDD test scenario should test if *the ultrasonic sensor provides the feedback that the forward distance <= threshold (means there is an obstacle upfront)* and whether the vehicle stops. They write this after **When**. The context could be *the autonomous vehicle is auto-parking*. We write them after **Given**. **Then** constraints the safe control action as *the autonomous vehicle stops immediately.* More possible triggers are expected to be generated after **When** to test them. In step 2, after the three amigos discuss and determine the test scenarios, the developer starts generating them into test cases, as shown in Figure 2 (b). BDD test cases use annotations such as **@Given**, **@When**, and **@Then** to connect the aforementioned test scenarios with real code. The developer produces code to fulfill each annotation. We can identify unsafe scenarios when the test cases fail. We correct the trigger event to pass the test cases to satisfy the safety requirement.



(a) Test scenario example **(b) Test case example**

Figure 2: BDD safety verification example

4 CONCLUSION

We propose a possible way to use BDD for safety verification with STPA for safety analysis in agile development. The preliminary controlled experiment shows that STPA-BDD has a good capability on communication effectiveness between developers and business analysts. Further research is needed to validate it.

REFERENCES

[1] Fleming CH. 2015. *Safety-driven early concept analysis and development.* Ph.D. Dissertation. Massachusetts Institute of Technology.
[2] North D. 2012. *JBehave: A framework for behaviour driven development.*
[3] Cleland-Huang J and Rahimi M. 2017. A case study: Injecting safety-critical thinking into graduate software engineering projects. In *Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track.* IEEE Press.
[4] Arthur JD and Dabney JB. 2017. Applying standard independent verification and validation (IV&V) techniques within an agile framework: Is there a compatibility issue?. In *2017 Annual IEEE International Systems Conference.* IEEE.
[5] Leveson N. 2011. *Engineering a safer world: Systems thinking applied to safety.* MIT Press.
[6] Lai ST, Leu FY, and Chu WCC. 2014. Combining IID with BDD to enhance the critical quality of security functional requirements. In *Proceedings of the 9th International Conference on Broadband and Wireless Computing, Communication and Applications.* IEEE.
[7] Dybå T and Dingsøyr T. 2008. Empirical studies of agile software development: A systematic review. *Information and Software Technology* (2008).
[8] Okubo T, Kakizaki Y, Kobashi T, Washizaki H, Ogata S, Kaiya H, and Yoshioka N. 2014. Security and privacy behavior definition for behavior driven development. In *Proceedings of the 15th International Conference on Product-Focused Software Process Improvement.* Springer.
[9] Stålhane T, Myklebust T, and Hanssen GK. 2012. The application of Safe Scrum to IEC 61508 certifiable software. In *11th International Conference on Probabilistic Safety Assessment and Management Conference and the Annual European Safety and Reliability.*
[10] Wang Y, Ramadani J, and Wagner S. 2017. An exploratory study of applying a scrum development process for safety-critical systems. In *Proceedings of the 18th Product-Focused Software Process Improvement Conference.* Springer.
[11] Wang Y and Wagner S. 2016. Towards applying a safety analysis and verification method based on STPA to agile software development. In *IEEE/ACM International Workshop on Continuous Software Evolution and Delivery.* IEEE.

[1] Since we focus on safety in our research, we assign a safety analyst as the QA role in our context.
[2] More descriptions of STPA for safety analysis are given in [11] concerning an example of using STPA in an airbag system and [10] concerning the use of STPA in a Scrum development process.
[3] The process variables exist in the process model for each component at the system functional design [5].
[4] The algorithm is responsible for processing inputs and feedback, initializing and updating the process model, and using the process model plus other knowledge and inputs to produce control outputs [5].
[5] We illustrate a BDD test scenario using only three basic steps "Given" "When" "Then". More annotations, such as "And", can also be added.