A Process Model based on STAMP for Collecting and Management of Safety Evidence

Luiz Eduardo Galvão Martins, Tony Gorschek

A Process Model based on STAMP for Collecting and Management of Safety Evidence

Luiz Eduardo Galvão Martins
Institute of Science and Technology
Federal University of São Paulo, UNIFESP
São José dos Campos, Brazil
legmartins@unifesp.br

Tony Gorschek
Software Engineering Research Lab
Blekinge Institute of Technology, BTH
Karlskrona, Sweden
tony.gorschek@bth.se

***Abstract*— Safety evidence collection is an essential activity for companies that develop safety-critical systems (SCS), software intensive product (SIP) and service development (SD). The collected evidence of safety along the SCS development are used during the certification process to show the auditors that the systems, software, services and products developed are safe. In this article we discuss the importance for the companies to have a robust safety evidence collection process. We discuss the use of STAMP as a basis to build a comprehensive safety evidence collection and management process that also correspond and do not conflict with agile methodologies. We believe that a proper integration between product development and evidence gathering process can be achieved by adopting STAMP as the basis for a full safety evidence collection and management process.**

Keywords*— *Safety-Critical Systems, Software Intensive Product, Service, Safety Evidence, Process Model, SCS, STAMP.

I. INTRODUCTION

Safety-Critical Systems (SCS) are becoming increasingly present in the daily lives of modern societies, increasing people's dependence on them. Current SCS are strongly based on computational technology; possible failures in the operation of these systems can lead to accidents and endanger human life, as well as to damage the environment and property [1][2]. SCS are present in many areas such as avionics, automotive systems, industrial plants (chemical, oil & gas, and nuclear), medical devices, railroad control, defense and aerospace systems, among others [2][3]. Companies that develop SCS and SIP must present evidence of their safety before society, in order they can obtain certification and authorization to market their systems and software.

Safety is an essential attribute in SCS. To ensure that SCS are safe and reliable they must go through a certification process, where a thorough checking is made taking into account features and functionalities of the system and software, based on evidence provided by developers [5][12]. SCS certification processes are driven by safety standards, which are defined and approved by national and international organizations, such as ISO, IEC, IEEE, ABNT among others [10].

In recent years, an increasingly adopted way of presenting safety evidence of SCS and SIP is through the specification of safety cases [3][5][7]. A safety case must present a set of evidence accompanied by convincing arguments that a SCS is safe enough to operate in a given environment [6]. A safety case consists of three parts: objectives, arguments, and evidence [4][6]. Demonstrating the satisfaction of safety case objectives involves obtaining safety evidence during the SCS development process, as well as building arguments that relate evidence to safety case objectives [3]. The safety evidence gathering process is essential for building convincing safety cases [8]. However, while the safety case argumentation aspects have been extensively studied, there are few works focusing on the accurate characterization of evidence that should support the safety arguments in SCS and SIP. Moreover, there is still a lack of appropriate guidance on what evidence should be collected and managed during the SCS development process [3][9]. This is also relevant from the perspective of not over specification and collecting data not necessary or used.

STAMP is a System-Theoretic Accident Model and Process proposed by Leveson [2][11]. This model offers a solid groundwork which can be used as a basis to build a safety evidence collection process. We present more details about STAMP in section 3.

We want to develop a process model called the safety evidence collection process (SECP). It is based on STAMP for collecting and management of safety evidence. SECP is intended to be integrated with the SCS development lifecycle. SECP will be supported by a software tool, which will help the developers to trace the safety evidence obtained along the SCS and SIP development. Moreover, SECP is intended to ensure safety as a property along the SCS development lifecycle. SECP contributes with a good-enough approach to avoid under or over specifying as well as appropriate data collection. SECP can be especially relevant for companies new to the SCS domain, or in combination with the use of more agile methods to make sure that mandatory certification is not lost in the normally document averse informal nature of many agile realizations where human interaction often replaces documentation that could also been part of the data collection/documentation.

The remainder of this paper is organized as follows: in Section 2 we present some related work; in Section 3 we present our position in relation to the need of a robust safety evidence collection process, and why we intend to use STAMP as a basis for the proposed process (SECP); in Section 4, we discuss how to use the concepts of the STAMP to support the proposed process (SECP); in Section 5 we present the final remarks.

II. RELATED WORK

Nair et al. [5] carried out a systematic literature review on provision of evidence for safety certification. The main goal of their work was to synthesize the existing knowledge in the literature about safety evidence, concentrating on three facets: the information that constitutes evidence; structuring of evidence; and evidence assessment. The main results from this work were the following: (a) a general taxonomy of

safety evidence types under safety analysis results, requirements specification and design specification; (b) a classification of existing techniques for structuring evidence information into three categories (argumentation-induced evidence structure, model-based evidence specification, and textual templates); and a classification of existing techniques for evidence assessment into four categories (qualitative assessment, checklists, quantitative assessment, and logic-based assessment).

Huan et al. [6] proposed a systematic model-based approach for collecting safety evidence. This approach is organized into four phases: evidence requirement analysis, traceability management, evidence collection preparation, and evidence collection. In the evidence requirement analysis phase the evidence requirements are extracted from each safety goal and presented as UML class models. During the traceability management phase the traceability between the extracted concepts from the evidence requirements and the original safety goals are maintained in the form of a generated traceability table. In the evidence collection preparation phase all safety goal conceptual models are combined and the redundant information is removed. The combination of the safety goal conceptual models generates the safety case conceptual model (SCCM). Finally, in the evidence collection phase the SCCM is used as a guideline for evidence collection. Three evidence collection activities are defined: class-based evidence collection, attribute-based evidence collection, and association-based evidence collection.

In the SCS domain, in order to get a system certification it is necessary to demonstrate compliance with safety standards. The demonstration of compliance with standards and regulations requires collecting safety evidence which indicates that the safety goals have been met. In real SCS contexts, the safety evidence is collected from thousands of sources, including requirements document, hazard analysis document, design artifacts, test logs etc. As discussed by Huan et al. [6], manual safety evidence collection tends to be time-consuming, tedious and error-prone. Moreover, without a clear process, there is a risk of invalid information being collected as safety evidence, decreasing the confidence on the evidence. This is also relevant for any organization not wanting to expend resources for collection unless central to the needs and actual use. We intend to create a new safety evidence collection process, SECP, based on STAMP, which will address the challenges of improving the confidence on the evidence, supporting safety argumentation, and reducing the certification costs [6].

III. Process Model for Collecting and Management of Safety Evidence

Practitioners who develop SCS and SIP in different domains, whether in the automotive, aeronautics, defence, health, or industrial plants, face similar challenges when confronted with the task of demonstrating to society that the systems and software they develop are safe. The challenge of convincingly demonstrating that systems and software are safe has been increasing as they are becoming increasingly complex and dependent on computer technology (software, hardware, communication, and their integration).

In the last years, approaches that have been gaining strength for safety demonstration are those based on Safety Cases [4-7]. The construction of effective safety cases requires the use of robust evidence [4]. A systematic process is essential for the collection of correct evidence, which should occur at the proper time during the development of SCS. Little attention has been given to the safety evidence collection process [5][6], leading to rework and delay during the construction of safety cases, with negative impacts during the SCS certification process [5]. Few studies addressing the problem of safety evidence collection are found in the literature [6].

The evidence collection process proposed by Huan et al. [6] (presented in section 2) is a model-based approach that uses UML class diagrams to represent conceptual models of safety goals and safety cases. However, this approach lacks effective integration with the SCS development process. Effective integration between the evidence gathering and SCS development process is essential for the collection of appropriate evidence at the proper time, otherwise there is a risk that the evidence collection will occur only at the end of the product development. Such situation leads to loss of safety evidence generated throughout the product development process, decreasing the quality of the safety cases and causing delay in the certification process.

In the last decade, traditional models for safety and accident analysis have been questioned for effectiveness [2]. Such traditional models are based on the analysis of failure event chains, where each failure can result in a subsequent failure within a chain, leading to a "domino effect" which may end up to accidents. Leveson [2] argues that traditional models can no longer keep up with the complexity of the SCS currently developed. In this context, Leveson proposed a new model for safety and accident analysis. This model is called STAMP: System-Theoretic Accident Model and Process.

STAMP is based on three main concepts: safety constraint, hierarchical control structure, and process models. The safety constraint is the basic concept in STAMP, which can take the form of a design, implementation, or operation constraint of the SCS. The safety constraint is a goal to be achieved by a controller (human or machine). The hierarchical control structure maps the relationships between the actors/controllers involved at the different levels of control, design, development, and operation of the SC. Process models are abstract representations of the real process that should be controlled. These models can represent the mental model of a human operator as well as the control logic of an automated controller [2].

The conceptual framework of STAMP, coupled with its holistic view of a SCS, provides a solid foundation that may underpin a process model for collecting and management of safety evidence. Typically, the safety evidence gathering process consists of four steps: requirements analysis, evidence gathering preparation, evidence gathering execution, and evidence validation [6]. In the requirements analysis step, the main objective is to analyze safety goals and identify the necessary evidence resulting in evidence requirements. At the evidence preparation step, evidence requirements are refined to identify relevant project artifacts (such as data sources), resulting in an evidence collection guide. In the execution step of the evidence collection, data items are extracted to provide information to build evidence associated with the project artifacts. In the last step, a validation of the evidence is performed to ensure completeness, correctness, and consistency of the collected evidence [6].

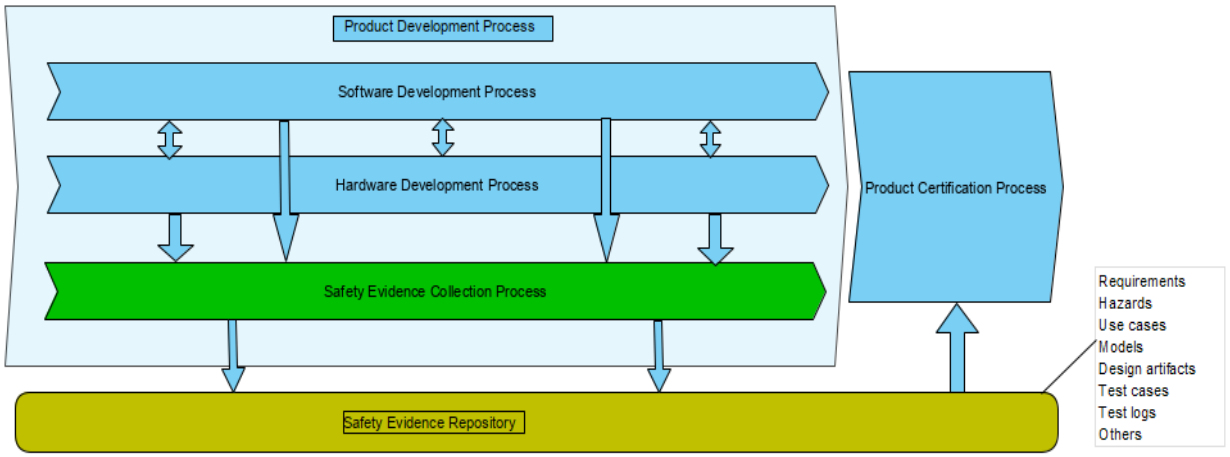

Figure 1. Relationship between product development, safety evidence collection and certification process.

We believe that a proper integration between product development and evidence gathering process can be achieved by adopting STAMP [2] as the basis for a full safety evidence collection and management process. Fig. 1 presents our view of the essential relationships among three main processes carried out during SCS development: product development process (1) and safety evidence collection process (2), which should be tightly integrated; safety evidence collection process (2) and certification process (3). The safety evidence collection process should be responsible to feed a safety evidence repository, which will supply the necessary safety evidence during the certification process.

IV. Using STAMP as a Basis for SECP

During the SCS and SIP development the safety evidence related to the systems and software produced may rise from different sources at different time, assuming a variety of artefact types. In order to have compliance with safety standards commonly used in the certification process, the safety evidence produced along the system and software development must be collected, stored, and linked to the final product.

We believe that STAMP offers a solid groundwork to support an innovative safety evidence collection process (SECP). As explained in Section 3, STAMP is based on three concepts: safety constraint, hierarchical safety control structure, and process models. In SECP we visualize the product development process as a hierarchical control structure, where each phase (or level) of the development process enforce a set of safety constraints to the phase/level bellow (or next). According to Leveson [2], between the hierarchical levels of the control structure, communication channels are necessary (downward and upward) in order to provide information to impose safety constraints on the level bellow and feedback to the level above informing whether the safety constraints are being satisfied.

The separation of concerns allows different groups of experts to focus on their expert area, and allow for a realistic decision making in relation to how to best collect good-enough evidence on their respective level. In addition, the formal, but even more importantly, informal communication channels of agile methodologies can be re-used for the purpose of SECP. By coordination and separation of concerns you can delegate work and responsibility to the teams, rather to an "audit" or "certification" group. Fig. 2 illustrates the model of communication channels between the levels proposed by Leveson.

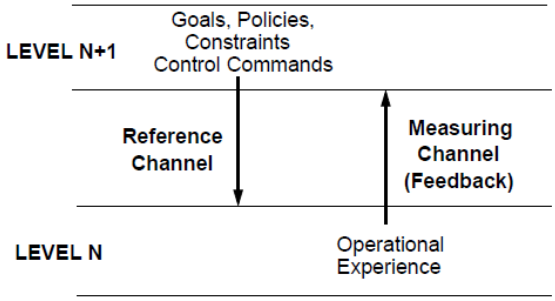

Figure 2. A model of communication channels between the hierarchical levels proposed by Leveson [2].

In Fig. 3 we instantiate the model of communication channels presented in Fig. 2, in order to show how this model can be applied into our conception of SECP. In this example, we stress the hierarchical safety control between software requirements and design phases. The same idea should be performed to all system development phases.

According to the hierarchical safety control showed in Fig. 3 the software requirements level imposes safety constraints to the software design level. In this example, such constraints are the requirements specs, safety requirements, formal specs, uses cases, hazards/risks and others, which must be satisfied in the software design level. Additionally, each satisfied constraint will generate safety evidence, which should be collected, stored and linked to the product to be certified by the authorities. The software design level must return a feedback to the software requirements level showing that the constraints are satisfied (specs accommodation).

Metrics should be defined to measure in what extent the constraints are satisfied. Therefore, SECP should be able to handle constraints in the form of safety evidence that can be assessed by proper metrics inside of a hierarchical safety control structure. Communication channels between the hierarchical levels are essential to guarantee the change of information between the levels. SECP, process and tool, will help SCS practitioners to address the challenges of improving the confidence on the evidence, supporting safety argumentation, and reducing the certification costs.

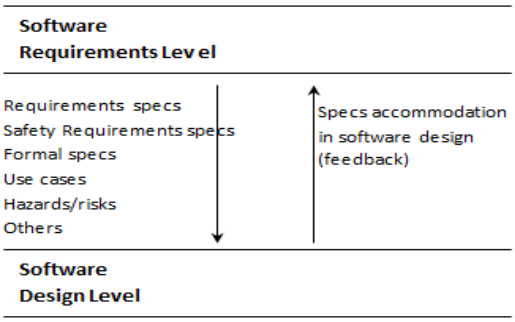

Figure 3. Hierarchical safety control between software requirements and design phases.

V. Conclusion

The adoption of safety cases as an approach to demonstrating system and product safety has gradually increased in the SCS industry. However, few studies are found in the literature addressing the definition of systematic processes that assist in the management and collection of safety evidence. We believe that STAMP provides a solid conceptual framework for creating a safety evidence collection process (SECP). According to STAMP safety is not a feature present only in technological components of the systems and products developed, but rather it is a feature that should permeate the entire development cycle, covering all levels of development and decision making of companies that produce safety-critical systems and software intensive products.

Thus, the evidence/data collection and documentation need to be distributed as is the work and separation of concerns – with strong coordination mechanisms. Our intent going forward is to develop SECP to be in-line with agile ways of working from two perspectives. One, organizations that have long SCS tradition can decentralize evidence/data collection and documentation and move responsibility to teams and introduce coordination rather than control as its main tool. Companies that have not traditionally worked with SCS can through SECP and the realization of STAMP get support in what parts need to be considered the least common denominator for collection and documentation.

Acknowledgment

This work was funded by São Paulo Research Foundation (FAPESP) under the grant no. 2019/09396-0.

References

[1] Hatcliff, J., Wassyng, A., Kelly, T., Comar, C., and Jones, P. (2014). Certifiably safe software-dependent systems: challenges and directions. In Proceedings of the on Future of Software Engineering - FOSE, (pp. 182–200).

[2] Leveson, N. G. (2011). Engineering a Safer World: Systems Thinking Applied to Safety. The MIT Press.

[3] Nair, S., de la Vara, J. L., Sabetzadeh, M., and Falessi, D. (2015). Evidence management for compliance of critical systems with safety standards: A survey on the state of practice. Information and Software Technology, 60, (pp. 1–15).

[4] Kelly, T. P. (1998). Arguing Safety - A Systematic Approach to Managing Safety Cases, PhD Thesis, University of York.

[5] Nair, S., de la Vara, J. L., Sabetzadeh, M., and Briand, L. (2014). An extended systematic literature review on provision of evidence for safety certification. Information and Software Technology, 56, (pp. 689–717).

[6] Huan, L., Ji, W., Chunchun, Y., Yaping, L., van den Brand, M., and Engelen, L. (2015). A Systematic Approach for Safety Evidence Collection in the Safety-Critical Domain. In Proceedings of the Annual IEEE Systems Conference (SysCon), (pp. 194-199).

[7] De la Vara, J. L., Génova, G., Álvarez-Rodrigues, J. M., and Llorens, J. (2017). An analysis of safety evidence management with the structured assurance case metamodel. Computers Standards & Interfaces, v. 50, (pp. 179-198).

[8] Gannous, A., Andrews, A, and Gallina, B. (2018). Toward a Systematic and Safety Evidence Productive Verification Approach for Safety-Critical Systems. In Proceedings of the IEEE International Symposium on Software Reliability Engineering Workshops, (pp. 329-336).

[9] De la Vara, J. L., Borg, M., Wnuk, K., and Moonen, L. (2016). An Industrial Survey of Safety Evidence Change Impact Analysis Practice. IEEE Transactions on Software Engineering, v. 42, issue 12, (pp. 1095-1117).

[10] Martins, L. E. G. and Gorschek, T. (2017). Requirements Engineering for Safety-Critical Systems: Overview and Challenges. IEEE Software, v. 34, (pp. 49-57).

[11] Abdulkhaleq, A., Wagner, S., and Leveson, N. (2015). A comprehensive safety engineering approach for software-intensive systems based on STPA. Procedia Engineering, 128, (pp. 2–11). http://doi.org/10.1016/j.proeng.2015.11.498

[12] Nair, S., de la Vara, J. L., Sabetzadeh, M., and Briand, L. (2013). Classification, Structuring, and Assessment of Evidence for Safety. In IEEE Sixth International Conference on Software Testing, Verification and Validation, (pp. 94–103).