Remote and agile improvement of industrial control and safety systems processes

Thor Myklebust

SESS, SINTEF Digital, Norway. E-mail: thor.myklebust@sintef.no

Mary Ann Lundteigen

EC, NTNU, Norway. E-mail: mary.a.lundteigen@ntnu.no

Lars Bodsberg

SESS, SINTEF Digital, Norway. E-mail: lars.bodsberg@sintef.no

Geir K. Hanssen

SESS, SINTEF Digital, Norway. E-mail: Geir.K.Hanssen@sintef.no

Digitalization and remote operations introduce new possibilities for continuous and agile improvements of products in operation by exploiting inherent possibilities in software which is easily changeable and deployable. This approach is driven by data analysis, customer expectations and the possibility of frequent deployment over the air of improved software. Adding functionality into software, combined with connectivity to products, opens possibilities for manufacturers and operators, enabling new features and new operational models.

This has also become relevant for regulated environments like industrial control and safety systems used in critical infrastructures. Adapted agile processes like SafeScrum and DevOps may be used to achieve continuous improvement. They enable speed and a continuum between development, maintenance and operation. For instance, experience and data from operation on new cybersecurity threats, must be fed back to the maintenance process to be resolved fast. Hence, the DevOps concept, which is imperative in non-safety domains, is now highly relevant in regulated environments as well. The speed of this process is vital where in particular cybersecurity threats must be resolved fast to avoid safety threats.

The Agile Safety Case is an enabler of ensuring structured proof of compliance of safety performance for the involved stakeholders. This paper proposes a solution for a safety case which may be applied for continuous product improvements during operation considering safety as well as security. The solution involves the relevant stakeholders and results in a shift in responsibilities.

Keywords: Agile, DevOps, SafeScrum, safety case, safety, security.

1. Introduction

With the mindset that a non-secure system is a non-safe system, it is necessary to bridge the safety and cybersecurity domains of industrial automation and control systems (IACS). The industry has for a long-time developed management systems for functional safety to prevent and mitigate safety consequences in line with the IEC standards 61508/61511. This has led to significant developments in the field of safety. However, these standards provide limited requirements and guidance on protection of the new industrial threat from cyber-attacks. It is expected that these safety standards mainly will refer to relevant security standards to take care of security challenges. The IEC 62443 series of standards, technical reports, and related information are becoming the main standards with design guidance and requirements for implementing and managing secure IACS. Some of the standards are still drafts and there is limited practical experience in using the standards.

In this paper we suggest improving the processes including updates, upgrades, patching and related safety evidence.

The rise of the safety–security interactions in systems have stimulated academic research and standardization communities, but concrete achievements are very few. The challenge of mastering the interdependencies between safety and cybersecurity is still significant. Some clarifications are included in IEC TR 63069:2018. This is further elaborated by Lundteigen and Gran (2019). The industry needs more practical guidelines and standardized approaches for integrating the safety and security domain to ensure that cybersecurity measures does not interfere with the proper safe operation of the production process.

According to IEC, "*Functional safety is the part of the overall safety that depends on a system*

Proceedings of the 30th European Safety and Reliability Conference and
the 15th Probabilistic Safety Assessment and Management Conference
Edited by Piero Baraldi, Francesco Di Maio and Enrico Zio
Copyright © ESREL2020-PSAM15 Organizers.*Published by* Research Publishing, Singapore.
ISBN: 978-981-14-8593-0; doi:10.3850/978-981-14-8593-0

or equipment operating correctly in response to its inputs. Functional safety is the detection of a potentially dangerous condition resulting in the activation of a protective or corrective device or mechanism to prevent hazardous events arising or providing mitigation to reduce the consequence of the hazardous event." In this paper, safety is related to accidental threats whereas security is related to malicious threats.

Assessing a security threat is radically different from assessing a safety hazard (see Kriaa, 2015). Unlike safety hazards, all sources of security threats are usually not known by the analyst and cover an extremely broad range of possible scenarios. The characteristics of the safety hazards are well-known, and the required number of scenarios to consider is limited. The use of probabilities is widely adopted in assessing safety hazards, whereas cyber attacks are less predictable and depend on many factors (e.g., attacker profile, skills, motivation, etc.).

Safety case is an efficient method for helping system suppliers and operators to focus on the simple but important question "How do you know that your system is safe enough? Compared to the traditional safety case approach, there are several benefits using an agile safety case approach when presenting proof of compliance:

- Improves communications between stakeholders and the progress of the project
- Strengthens communication in all phases of a project
- Easily navigate the status of the safety case
- Less time used on the development of the safety case
- Less information and documentation needed
- More emphasis on the reuse of safety case structure information and other generic documents, e.g. documentation of employees' competence and experience
- Improved management of changes and corrections during development and after the first release (operation phase)
- Shorter time from the last code is written to the finalization of the safety case
- More effective procedures for updates of the software due to security threats

1.1 *Delimitation and definition of SIS*

This paper focuses on safety instrumented systems (SIS) used to implement one or more safety instrumented functions (SIF) (see IEC 61511: 2016). It should be noted that human action may be part of a SIF. A SIS is composed of any combination of sensor(s), logic solver(s), and final elements(s). It also includes communication and ancillary equipment (e.g., cables, tubing, power supply, impulse lines, heat tracing). As modern SIS normally includes software, we have assumed that the SIS discussed in this paper includes software.

2. Background

2.1 *Process control vs safety*

IACS is a collection of personnel, hardware, and software that can affect or influence the safe, secure, and reliable operation of an industrial process (see IEC 61511-1:2016, IEC 62443-1-1:2009). These systems include industrial control systems (e.g. basic process control system (BPCS), distributed control systems (DCSs), and safety instrumented systems (SIS), programmable logic controllers (PLCs), remote terminal units (RTUs), intelligent electronic devices, supervisory control and data acquisition (SCADA), and networked electronic sensing and control, and monitoring and diagnostic systems. This paper addresses the relationship between BPCS and SIS.

Suppliers and operators of process control and safety systems must have an organisation that is able to handle the technical requirements as well as organisational and operational requirements for safety and security given in the IEC 61511 and IEC 62443 series, respectively.

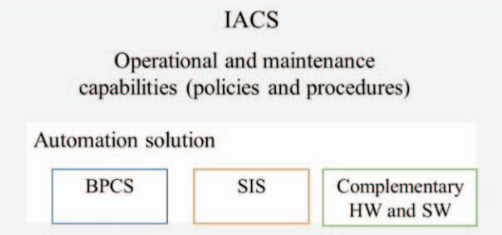
IACS
Operational and maintenance capabilities (policies and procedures)
Automation solution
BPCS
SIS
Complementary HW and SW

Fig. 1. Relationship between IACS, BPCS and SIS. Adapted from IEC 62443-2-4:2015

It is important to realize and understand the fundamental difference between process control and safety control.

BPCS is active, or dynamic, and responds continuously to changes in process states so that the operation follows pre-set targets and boundaries. It normally has analogue inputs and outputs, performs mathematics and have feedback loops. The control algorithms of the BPCS is made to obtain best performance of the process, which in some cases may result in operation close to the boundaries of set by safety considerations. For example, the optimal throughput of a gas compression system could potentially be with a downstream pressure close to the setpoint set by the SIS to trigger a process shutdown. A BPCS must be flexible enough to allow frequent changes. Process parameters e.g. set points require frequent changes as called by a plant wide control strategy, Jahanshahi et al. (2020).

Depending on the state of facility, parts of the BPCS may also be placed in bypass, and the process may be controlled manually. The BPCS is primarily built for optimal control, and safety is allocated to other independent systems (SISs) following a defense-in-depth approach.

Unlike the BPCS which is executing its functions continuously, the SIS is normally dormant, or passive, during normal operation. Only when provoked, i.e. when the process exceeds the preset setpoints for process safety, the SIS should be activated and transfer the process to a safe state. For instance, a process shutdown system (PSD) would close inlet valves to a pressurized vessel if the pressure exceeds the high-trip setpoint. A SIS is usually complemented with dormant mechanical barriers, such as pressure relief valves. These valves will open if the pressure continuous to increase and route flow to the flare system to depressurize the vessel. A facility will also have additional SISs, which act upon global events such as fire and gas releases and manual activation. Such SIS systems, often named as fire and gas detection systems, fire extinguishing systems, and emergency shutdown systems, will carry out mitigative actions to reduce the consequences of a situation where a failure of the PSD system and potentially also the pressure relief valves have led to a release of pressurized and flammable fluids and gases.

Many process plants are moving towards an increased use of remote operation. Remote operation means to locate the central control room in a remote location, which can be at another facility or (if we consider the offshore oil and gas industry) onshore in a typical office environment. Plants may also keep central control rooms at the facility, but allocate specific support tasks (e.g. in relation to condition-monitoring of critical equipment and software upgrades) to a remote support centre. The allocation of monitoring, support, and control tasks to a remote location make BPCS and SIS potentially more vulnerable to cyber attacks.

Compared to safety hazards, cyber attacks may require immediate software modifications. Since security threats changes with time, it is important to have systems and procedures in place to ensure that the most optimal decision on when to make the safe patching of BPCS and SIS controllers, when security issues are revealed. Countermeasures for the period when awaiting the patching must also be identified. On a less frequent basis, the BPCS and SIS may be subject to upgrades of hardware and software (application program and firmware) that are initiated from operational needs and considerations. These may have direct and indirect implications for the SIS and BPCS ability to resist and detect cyber attacks. In general, it is a need to be able to manage the specification and implementation of new and modified software requirements in an efficient and trustworthy way, where technical safety, process safety, and cybersecurity disciplines are collaborating actively to solve and resolve conflicting issues in relation to safety and cybersecurity.

2.2 Agile and *SafeScrum*

Agile software development is a way of organizing the development process, emphasizing direct and frequent communication, frequent deliveries of working software increments, short iterations, active customer engagement throughout the whole development life cycle, and change responsiveness rather than change avoidance. This is in contrast to waterfall and V-like models, which emphasize thorough and detailed planning, and design upfront and consecutive plan conformance. Several agile methods are in use. Scrum, Schwaber et al. (2001) and XP, Beck et al. (2004) are the most commonly used.

SafeScrum (see Hanssen et al., 2018), is based on the Scrum process framework for incremental and iterative development, which has become the standard process model for most industrial software engineering projects. In order to be compatible with requirements found in safety standards, in particular the generic IEC 61508:2010, SafeScrum proposes additional activities and roles.

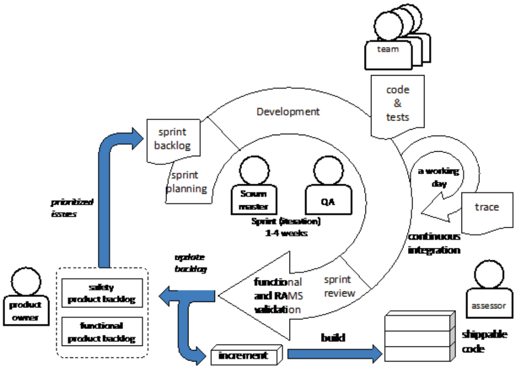

Figure 2 The SafeScrum process model

Requirements are kept in the product backlog in the form of user stories, and in SafeScrum, functional (not safety related) and safety-related requirements are kept separately. Simply put, functional user stories come from the users and safety-stories comes from preliminary safety analyses. If a user story is presumed to be related to a safety story (see Myklebust et al., 2016) or hazard story (see Stålhane et al. 2018) a link should be established. Development is done in sprints, which are short and repeated work iterations. Each sprint starts with a sprint planning meeting where stories from the product backlog are prioritized, selected, and broken down into solution ideas and added to the sprint backlog. Development is done by a fixed team which has a short status meeting (known as the scrum) regularly, maybe every day, to share progress, plans and discuss any problems. Development of software should be done according to the principles of test-driven development, which also assures high test coverage and documentation of testing. SafeScrum defines an additional QA-role that ensures that all process steps and documentation is done according to the relevant safety standard. Each sprint is supposed to produce an increment which is a small part of the solution. Normally this is software that can be integrated into the solution, but it may also be architecture descriptions, documentation or other needed artefacts - basically work items that are needed to build the final solution. Each sprint ends with a sprint review which may encompass a RAMS validation to check that the increment fulfils any safety requirements and complies with the relevant safety standard. If a resulting work item, e.g. a software module, is approved, it is kept in the solution which may be built for further testing. However, if it is not approved, the story that initiated the development is improved (rewrited) including the new knowledge and moved back to the product backlog for later improvement or completion. As a final note, it should be added that this process needs to be supported by tools for requirements (backlog) management, unit testing, and workflow management. Normally, standard agile process tools can be used.

An agile process has several properties that are beneficial for development of safety critical software:

- It provides testable results (increments) frequently
- It builds tests and code in parallel
- It enables traceability, both of the development itself, but also the quality assurance
- It enables a learning process where (functional) requirements can be improved over time
- It enables transparency, which may be beneficial in the dialogue with an assessor. Especially the agile safety case approach and sprint review (see Myklebust et al., 2018).
- It enables building the agile safety case in parallel with development (not leaving it as a final activity)

2.3 *DevOps*

DevOps is all about communication & organization and is not intended to be a development process. The Gartner Glossary calls it a culture: "*DevOps represents a change in IT culture, focusing on rapid IT service delivery through the adoption of agile, lean practices in the context of a system-oriented approach. DevOps emphasizes people (and culture) and seeks to improve collaboration between operations and development teams.*"

DevOps is in many ways just what developers and operators always have done but in a more efficient and coordinated way. The operator experiences some problem or a need to change or extend the systems functionality, the responsible person write a report and send it to the company that has developed the system. This can also be partly automated, both the monitoring of the system and the feedback. Sooner or later, the operator will receive a new version of the system where the reported needs have been covered. What is new with DevOps is that DevOps extend the development team by "including" site operations in the development process. In this sense, a system is under constant development where experience from the field and monitoring of the system is feed into the improvement of the system. This will benefit both developers and operators. Operators will be able to bring their problems to the attention of the developers quicker and thus get the problems solved earlier. Developers will get a better understanding of the operations problems and the consequences of delivering systems containing errors or not fully meeting the needs of its users.

2.4 *Safety case*

The idea of a safety case is to argue that a system is safe in the same way as one would do in a court of law - thus the name safety case. One should start planning for the agile safety case (see Myklebust et al., 2018) at one of the first phases of the development of the SIS. We need to know which evidence we need and when and then plan how we will produce them during the development process. However, before we develop a safety case we need to define the system – what is part of the system and what is not, the context – where are our arguments valid? – and which assumptions we have made in order to construct our arguments. A safety case has four components:

- Claims – e.g., "The system is safe." Risk mitigated e.g. ALARP (As Low As

Reasonable Practical) approach and satisfy relevant legislation safety standards and guidelines like NOG 070 (2018) that present relevant SIL for the different products and systems.
- Description of planned use and environment – also named the context
- Arguments (also called justification or strategy) – supporting the claims.
- Proofs (also called evidence) – supporting the arguments. Evidence without argument is unexplained

As soon as the claim is written and the context is defined, we need to start on the arguments. First and foremost – keep it simple. Long, complicated arguments will be difficult to read and understand and will create the impression that you are trying to hide something. It is important to decide what is a valid argument. Consider the following: "We have found all errors because we used test method X". We might think that this is OK if method X is a method that is specified in the relevant standard. However, there are several ways to use a method and for a safety case, we must show that the method or technique applied is suitable for its intended purpose, is used in the right way and by competent persons who have sufficient resources (time, tools, etc.) allocated. The arguments must be supported by evidences.

Safety case are used in many industries. In the Northern sea, the UK have a directive for safety case (2015) the Offshore Installations (Offshore Safety Directive) (Safety Case etc.) Regulations 2015. The relevant safety standards for the Oil&gas industry, IEC 61508:2010 and IEC 61511-01:2016 does not require a safety case. A typical proof of compliance (PoC) documentation consists of:
- 50-200 documents. Several of them are named in the relevant safety standards
- E.g. 82 documents are mentioned in the IEC 61508:2010 series, 101 documents in EN 5012X series and 106 work products (documents) in ISO 26262:2018 series. IEC 61511-1:2016 is not as concrete when it comes to named documents, but our estimation is that it has approximately 70 documents depending how the document/information management are implemented
- The referenced documents in the safety case also include references, typically 1 – 20 references
- Total number of pages developed by the manufacturer: 2000 – 10000 pages

A safety case is of great help to have the complete picture as the safety case sums up and refer to the important documents. In addition, the safety case is a starting point when stakeholders shall look for relevant topics and documents to be reviewed and updated.

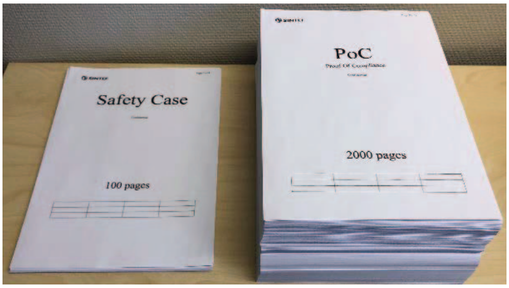

Fig. 3. A safety case and the PoC (the referenced documents in the safety case)

An agile safety case improves the process by inserting information when available, includes all relevant agile practices (see Myklebust et al. 2018) relevant for the project and is aligned with an agile and DevOps approach.

4. Suggestion for a Safety case approach for continuous product improvements during operation

To satisfy the need for frequent updates and upgrades after the SIS have been installed and taken into operation, we have to combine the best from the safety and security domains, agile community and the DevOps approach. There have to be a proactive reaction to unknowns that will inevitably manifest during operation of IACS including SIS. In addition, there may be security issues. On top there will be planned improvements to move towards a more digitalized system. To satisfy safety and security requirements both when developing and operating IACS including SIL, we should implement the agile safety case approach to ensure continuous update of the safety and security evidence, Myklebust et al (2018), as illustrated in the figure below.

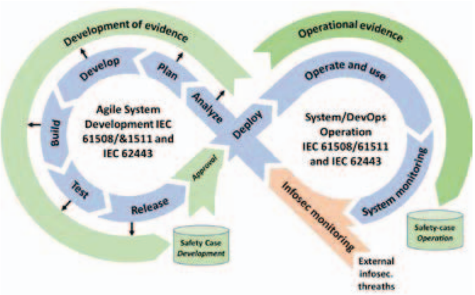

Fig. 4. The SafeScrum process model including both an Agile safety case and a DevOps approach.

It is important to have a DevOps process in place to ensure quick but safe patching to ensure a short-term response. Remote monitoring and control become more and more relevant both due to cost issues and to improve the operations. For a remote operator to gain adequate situational awareness, sufficient information must be transferred from the sensors to a remote control centre in a timely manner. This results in requirements on the type, volume and latency of information transmitted and the way it is presented to the remote operator(s).

When performing upgrades or e.g. safe patching this may have an effect on the operational safety case or the development safety case.

In the future, the safety requirements will not be as stable as they have been since safety and security standards have to be reissued more often due to the rapid development and implementation of new technology. The safety standards, including guidelines, are foreseen to be more wide-ranging as there are a few missing topics like e.g. AI (Artificial Intelligence), ML (Machine Learning), deployment and OTA (Over The Air) issues. As a result, the safety case has also to be updated to adapt to these changes in the safety and security standards.

5. Conclusion

Remote and continuous improvement of industrial control and safety systems processes including continuous improvements like updates and upgrades of IACS and SIS requires agile approaches including DevOps and an agile documentation approach. A safety case is of great help to have the complete picture as the safety case sums up and refer to the important documents. In addition, the safety case is a starting point when stakeholders shall look for relevant topics and documents to be reviewed and updated.

References

Kriaa, S., L. Pietre-Cambacedes, M. Bouissou, and Y. Halgand (2015). A survey of approaches combining safety and security for industrial control systems. Reliability Engineering and System Safety, 139, 156–178.

Piggin, R.S.H. (2013), Development of industrial cyber security standards: IEC 62443 for SCADA and Industrial Control System security. In Control and Automation 2013: Uniting Problems and Solutions.

Kanamaru, Hiroo (2017). Bridging Functional Safety and Cyber Security of SIS/SCS, In Proceedings of the SICE Annual Conference 2017, Kanazawa, Japan

DNV-GL-RP-G108 (2017), Cyber security in the oil and gas industry based on IEC 62443, www.dnvgl.com/oilgas/download/dnvgl-rp-g108-cyber-security-in-the-oil-and-gas-industry-based-on-IEC-62443.html

NOG 070 070 – NORWEGIAN OIL AND GAS APPLICATION OF IEC 61508 AND IEC 61511 IN THE NORWEGIAN PETROLEUM INDUSTRY (Recommended SIL requirements). Rev.03, June 2018

UK legislation directive No 295 seen 2020-02-07: www.legislation.gov.uk/uksi/2015/398/contents

IEC Basecamp, https://basecamp.iec.ch/download/flyer-iecee-international-cyber-security-certification-en/

T. Myklebust and T. Stålhane. Safety stories – A New Concept in Agile Development. SafeComp 2016-09, Trondheim.

K. Beck and C. Andres:Extreme programming explained: embrace change, 2nd Edition. 2004, Boston: Addison-Wesley Professional.

T. Myklebust, N. Lyngby and T.Stålhane. Agile practices when developing safety systems. PSAM14 Los Angeles September 2018

T. Stålhane and T. Myklebust. Hazard stories, HazId and safety stories in SafeScrum. XP 2018 Porto

Schwaber, K., Beedle, M.: Agile Software Development with Scrum. 2001, New Jersey: Prentice Hall.Author, N. (Yearb). Title. In N. Editor and N. Editor (Eds.), *Book Title*, pp. Pagestart–Pageend. Publisher Name.

M. A. Lundteigen and B. A. Gran. The need of improved methods to handle functional safety and cybersecurity in industrial control and safety systems. OECD Halden reaktorprosjektet 2019.E.

Jahanshahi, D. Krishnamoorthy, A. Codas, B. Foss and S. Skogestad. Plantwide control of an oil production network. ELSEVIER, Computer&Chemical Engineering, February 2020