Compliance of Agilized (Software) Development Processes with Safety Standards: a Vision

Barbara Gallina, Faiz Ul Muram and Julieth Patricia Castellanos Ardila
School of Innovation, Design and Engineering, Mälardalen University, Västerås, Sweden
{barbara.gallina,faiz.ul.muram,julieth.castellanos}@mdh.se

ABSTRACT

Hybrid software development, meant as a combination of traditional and agile methods/practices, has become a reality in safety-critical systems engineering. The spreading of hybrid software development stems from the impossibility to face the manyfold challenges via the definition of a process by the book. In this context, compliance management becomes challenging and the role of existing means for compliance should be clarified/rethought. In this position paper, we discuss the challenges and we propose our compliance management vision, which is being implemented in the context of the EU ECSEL AMASS project.

CCS CONCEPTS

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

KEYWORDS

Process compliance checking, Safety standards, Agile development, Compliance management

ACM Reference Format:
Barbara Gallina, Faiz Ul Muram and Julieth Patricia Castellanos Ardila. 2018. Compliance of Agilized (Software) Development Processes with Safety Standards: a Vision. In *Proceedings of 4th international workshop on agile development of safety-critical software (ASCS' 18).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Hybrid software development, meant as a combination of traditional and agile methods/practices, has become a reality. This statement is the result of an online survey, recently conducted within the HELENA (Hybrid DEveLopmENt Approaches in software systems development) project [21, 22]. In the survey, authors evaluate the usage of agile development methods/practices in the context of regulatory domains (e.g., medical, automotive, railway, avionics, space, etc.). The survey revealed that agile is mainly used in early development phases, specifically in non-critical systems (with the exception of automotive and aviation domains). Furthermore, the size of company or external standards has no impact on the development and use of hybrid approaches. The spreading of hybrid software development is a consequence of the "agilizing" trend, which in turn stems from the impossibility to face the manyfold challenges via the definition of a process by the book (i.e., compliant with prescriptive normative documents without any tailoring in place). Even the aviation and automotive domains contribute to make hybrid software development a reality. In this context, compliance management becomes challenging and the role of existing means for compliance should be clarified/rethought. In this position paper, we discuss the challenges and we propose our compliance management vision embracing the most common and/or promising means of compliance, which are being implemented in the context of the EU ECSEL AMASS project [3].

The rest of this paper is organised as follows. In Section 2, we provide background information. In Section 3, we formulate the challenges related to compliance management for hybrid development methods/practices. In Section 4, we propose our vision to face such challenges. Finally, in Section 5, we provide concluding remarks.

2 BACKGROUND

In this section, we present the background information related to the problem space. In particular, in Section 2.1, we recall agile principles and methods. In Section 2.2, we recall essential information on safety standards. In Section 2.3, we recall agilized methods. Finally, in Section 2.4, we recall basic principle of compliance management.

2.1 Agile Manifesto

As summarised in a previous work [14], the agile manifesto, published in 2001, consists of twelve principles [24]. These principles describe a way of organising the development process that can be summarised by the following slogans: emphasising frequent communication throughout the whole development life cycle, working software rather than comprehensive documentation, change responsiveness over following a plan, continuous delivery of working software and short iterations.

2.2 Safety Standards

Safety (de-facto) standards consist of a set of norms (requirements). In the case of prescriptive standards, part of these requirements is focused on the prescription of process reference models at system (e.g., ISO 26262 [19], part 3-4) as well as at subsystem level (e.g., ISO 26262, part 5 and part 6). In case of objective-based standards (e.g., DO-178C [31]), instead, part of these requirements is focused on the definition of the objectives and on the description of process elements (activities, work products, methods), which could be adopted to achieve the objectives. In what follows, essential information regarding DO-178C and ISO 26262 is briefly recalled.

DO-178C provides guidance for the development of software for airborne systems and equipment. Its purpose is to guarantee

ASCS' 18, 21 May, 2018, Porto, Portugal

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

a level of confidence in the correct functioning of the software developed in compliance with airworthiness requirements. To do that, it provides a series of processes characterised by a set of objectives, activities and expected deliverables. Process planning is one of these processes. Among its expected deliverables, we have: software development plan (SDP). DO-178C also defines the so-called certification liaison process, where the interactions that are expected to take place between the applicant and the certification body are defined. These interactions are expected to take place at specific development stages (Stages of Involvement-SOI). The first interaction (SOI#1) is expected to take place after the planning phase to ensure plan's approval.

ISO 26262 proposes a tripple V-model: a V-model to be followed at system as well as subsystem (software/hardware) level. A planning phase is also mandated. Since in the automotive domain, a formal certification process is not in place, ISO 26262 does not explicitly specify a liaison process. However, a process similar to the one defined in DO-178C could be adopted for performing self-assessment.

In this paper, we limit the attention to these two (de-facto) standards. It should be noted, however, that these standards represent significant samples of the normative space. ISO 26262, automotive sector specialisation of the so-called meta standard IEC 61508 [2], exhibits important similarities [15] with the parent and siblings standards (other sector-specific specialisations of IEC 61508 [2], e.g., EN 50126 [1] in the rail sector).

2.3 Hybrid software development

During the past decade, we have witnessed the so-called agilizing trend: traditional development methods and practices have been progressively influenced and transformed. As a result a plethora of hybrid development approaches is now available and as recently surveyed these methods/practices are widely used in industry [21, 22]. Hybrid approaches originate from the combination of the agile and/or traditional approaches to adapt and customize to organizational context needs (e.g., application domain, culture, processes, project, organizational structure, techniques, technologies, etc.). The research in Kuhrmann et al. [21] lists the hybrid approaches that are widely used in practice. In this subsection, we only recall those, which have been used for critical application development or which have been taken as starting point to develop ad-hoc hybrid methods. In this respect, it is worth to mention the combination of the well-known Waterfall model with the agile principle: Scrum, and the combination of V-Model Derivate(s) with Lean Development or Agile Portfolio Management. In particular, we have selected three hybrid approaches, which concentrate on adapting Scrum into safety-critical software development. These hybrid approaches have been already applied in the real life projects and most of them required standard certification.

In the remaining part of this subsection, the discussion is focused on Scrum. Scrum[1] is a widely-used agile method, which consists of three phases: pre-game (planning), development and post-game (review and retrospective + release). During the planning, user stories (requirements) as well as tasks are defined, and sprint (fixed-length iteration) backlogs (comprehensive planning work product) are produced. Scrum defines three main roles: (i) product owner is responsible for product backlog, (ii) scrum master interacts with other roles and is responsible for ensuring that Scrum process is followed, (iii) the team members need to deliver a product at the end of each sprint.

The "hybridity" of Scrum is not sufficient in the context of safety-critical development, where, as recalled in Section 2.2, planning represents a crucial and rather demanding phase, which requires the achievement of a complete view. This phase in the agile world only requires a partial view. Agile planning follows an *empirical logic* (in a plan-do-check-act cycle), which contrasts with the *defined logic* desired in regulated environments [11]. R-Scrum [11], an augmented Scrum implementation for regulated environments, proposes to have: an initial planning (authorized by the product council), containing the overall objectives, key phases, strategic decisions, personnel and resources required for the project management; and risk mitigation for quality (including safety and security), facilitated by the so-called *continuous compliance phenomenon* via assurance check points at the end of each sprint.

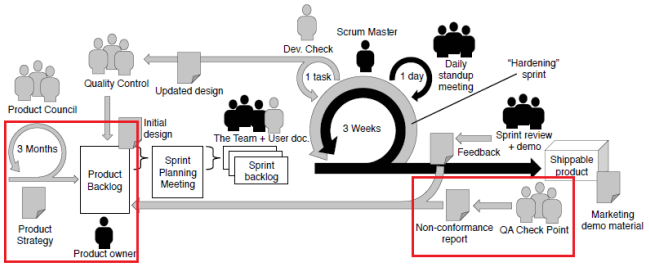

Figure 1: R-Scrum [11]

The *Agile approach for certifiable avionics systems* [25] (see Figure 2) also builds on top of Scrum. This approach proposes a *"way of working"* section in the software plans, which includes definition of small functionalities and quality gates supported with the evidence provided by the *"micro-reviews"* (done in every process transition). This section promotes the *"planning to re-plan"* mindset. SOI#1 is carried out after the first sprint, since confidence in the plans may be increased. In the planning, user-stories (requirements) should include descriptions of how the customer will use the function from which test cases/acceptance criteria can be derived, highly-automated continuous integration and verification should be foreseen to minimise repetition of manual work and to detect negative impacts, and quality assurance records should be generated in micro-reviews.

Safe Scrum [33] is an adapted version of Scrum, applicable to IEC 61508 certifiable software development (see Figure 3). In this approach, the IEC 61508 steps, needed for developing the environment description (Phases 1-4), results in the initial requirements of the system. These initial requirements are documented as *product (functional and safety) backlogs*, prioritised by the customer. Each backlog item also indicates the estimated amount of resources needed to complete the item. The software process is considered during the initial risk analysis and all later analysis on per iteration. If appropriate, the independent safety validator may take part in

[1] See https://www.scrum.org/

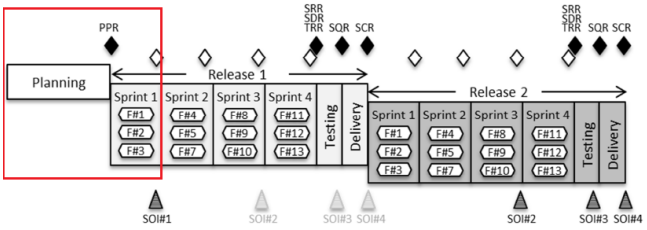

Figure 2: Agile Process for Avionics [25]

the RAMS (Reliability, Availability, Maintainability and Safety) validation for each sprint. The process enables *Re-planning*, based on the most recent understanding of the requirements and the system under development.

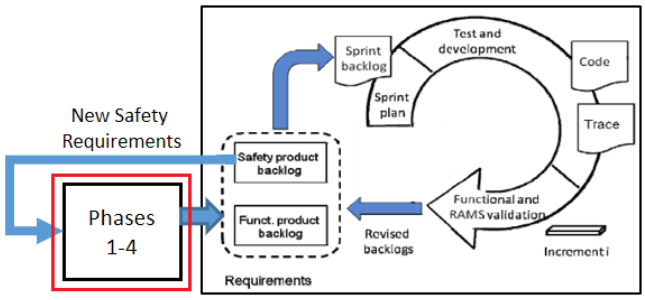

Figure 3: Safe Scrum [33]

2.4 Process Compliance Management

Compliance management deals with company's adherence to the standard's requirements. In this paper, we focus our attention on process compliance management, thus on the adherence to those requirements imposed on the system engineering process. The adherence to the standard's requirements can be shown via the provision of a justification regarding the fulfilment of these requirements. Such justification is expected to be scrutinised by an auditor. In what follows, we present four methods for compliance management. These methods are complementary and vary in terms of the type of justification used to show compliance. As Figure 4 depicts, all these four methods are all expected to confront the normative space with the process space. The AMASS platform [10], which represents the first de-facto European tool platform for assurance and certification, implements all these methods and offers scenarios to demonstrate their usefulness and applicability. The justification can be provided through a contextualized structured argument that links evidence to claims; it is known as a safety case [32]. The standards ISO 26262 and DO-178C require the use of safety case to show the compliance with standard's requirements. A safety case consists of two types of arguments: *product-based argumentation*, which shows that the product satisfies the safety requirements derived from hazards analysis and *process-based argumentation*, which shows that a safety-critical system has been developed in compliance with the development process defined in the standards. In this paper, we focus on the process-based argumentation (see Section 2.4.2).

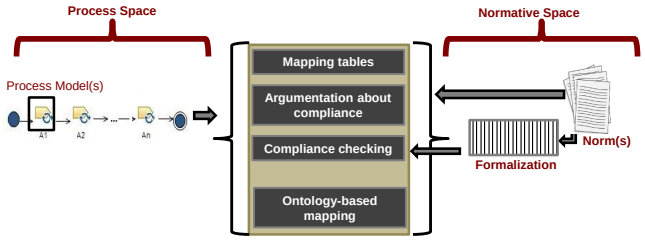

Figure 4: Compliance Management Vision

2.4.1 Compliance via mapping tables (generation). This approach consists of the provision of a table (or matrix) containing the mapping between process (elements) to standard requirements. A compliance table summarises the status of the process compliance. "A compliance matrix is often used where each requirement of a standard is explicitly recorded and demonstration of evidence is identified or linked to the particular clause in question" [23]. Typically, compliance tables are created via excel files. As observed in [9], in the automotive domain, for instance, compliance tables may contain the tailoring of the methods' related tables recommended by ISO 26262. Process authoring tools could also be used. In [27], for instance, an approach for mapping a process to a standard is described. This approach consists of the capturing of the standard's requirements as elements in Rational Method Composer (RCM) [2], a process management framework based on Software Process Engineering Meta-model (SPEM) 2.0 [29]. To confirm that the standard's requirements have been addressed by the process model, every requirement must be fully satisfied by assigning single process elements, i.e., tasks, roles, work products or guidelines to the requirements. The requirements are listed in a tree view, which makes visually easy to understand which requirements are fulfilled, which are not fulfilled, and which ones are only partially fulfilled. From this tree view, the compliance table can be easily generated.

2.4.2 (Generation of) argumentation about compliance. Process-based argumentation consists of arguments showing that a required development process has been planned (and/or executed) according to the normative space (e.g., safety standards). In particular, it plays a central role in justifying that the available evidence, in the form of staff competencies, tool qualifications, guidelines, and work products has achieved a set of safety requirements, and in turn an acceptable level of safety has been achieved. Process-based argumentation can be generated semi-automatically. Model-Driven Safety Certification (MDSafeCer) [12, 13] is a method that enables the semi-automatic generation of arguments from process models using model-driven engineering principles. In particular, process models compliant with SPEM 2.0 are transformed into process-based argumentation models compliant with the Structured Assurance Case Meta-model (SACM) 1.0 [30]. An implementation of MDSafeCer is being implemented in the AMASS platform, based on the most recent, and still under development, version of SACM. However, the process-based argumentations cannot ensure that the evidences are sufficient to support the claim. If these argumentations contain insufficient information (i.e., fallacious) they

[2] https://www.ibm.com/us-en/marketplace/rational-method-composer

may result in a loss of confidence on system's safety. In another research work [28], we have presented a method that validates whether the process models contain sufficient information, in order to prevent the occurrence of fallacy (i.e., *omission of key evidence*) in process-based argumentations.

2.4.3 Automatic compliance checking. This approach consists of the checking whether a (software) process model adheres to the normative (safety) requirements. The checking can be automated by using a compliance checker e.g., Regorous [18]. In this paper, we limit our attention to the Regorous-based compliance checking. Regorous only requires the information regarding the process enriched with compliance effect annotations (the cumulative interactions between process tasks that are adhered to the standard requirements influence [20]) and the normative safety requirements (formalized in a rule-base approach). This information allows Regorous to define a finite state model of the process, where normative safety rules provide the permissible states of the process elements.

Compliance checking with Regorous produces a *compliance report*, in which rule violations and uninvoked rules (which can be the caused of hidden uncompliant situations) are highlighted. Explorations of automatic compliance checking can be seen in our previous work [7, 8]. The rule base is modelled with Formal Contract Logic (FCL) [17], a language based on defeasible Logics [4] and deontic logic of violations [17], which allows reasoning with incomplete and inconsistent information. In [6], we have drawn a general definition of safety compliance pattern to facilitate the formalization of standard's requirements, in particular with ISO 26262.

2.4.4 Ontology-based mapping. This approach, originally called OPER (Ontology-based Process Elements Reuse), proposed by Gallina et al. [16], consists of the identification and merging of commonalities and variabilities among processes. This identification and merging is conducted via the exploitation of inference rules based on ontological representations of the processes. Once the commonality among the processes is identified, compliance can be shown by reusing previously used evidence.

3 CHALLENGES IN THE HYBRID CONTEXT

As mentioned, the hybrid context is represented by those approaches that combine agile practices and traditional methods/practices. Previous work in the medical domain [26] and in the avionics domain [25] has identified several challenges to adopting agile methods for engineering safety-critical systems. The main challenges regarding compliance management can be summarised as follows:

- The process plan has to be validated to support manufacturers in achieving certification of their products by satisfying the planning requirements. Accordingly, the challenge is how much information and definitions should be specified in the user story to enable compliance management.
- Continuous improvement implies that teams may change the software during the development. Changes may entail conflicts with the previously negotiated/approved plans. Thus compliance justifications can be threatened.

4 COMPLIANCE MANAGEMENT VISION

As recalled in Section 2.4, process compliance requires the provision of a justification regarding fulfilment of the requirements. Such justification may take different forms: a mapping table (alias checklist, indicating which process elements act as evidence for the satisfaction of the requirements coming from the standards), an argument (an argument explaining why certain evidence is linked to certain requirements), a proof (e.g. a verification report, proving that a certain process trace satisfies a certain set of formalized requirements). An inferred ontological equivalence enabling the linking of standard-related concepts with process-related concepts via the exploitation of previously used evidence could also be used. All these existing and complementary methods have the potential to play a crucial role in the hybrid world.

In Section 2.3, we have recalled three approaches for hybrid development processes, which solve the problem of planning by proposing an initial software development process plan with a minimal but enough set of process elements. For R-Scrum (see Figure 1), information for planning is obtained from a previous general planing strategy, which should be applied to all projects, and particularities of the project. In Safe-Scrum (see Figure 3) information is taken from the product safety backlog. Finally, in the agile approach for avionics (see Figure 2) information is obtained from the user stories.

In our compliance vision for hybrid/agilized processes, these *initial plans* can benefit from the compliance means (see Section 2.4), which are being implemented within the AMASS platform, as follows:

4.1 Hybrid Processes Compliance Tables

As required by ISO 26262 and recommended by Tüv [5], a safety manager shall be appointed. This means that to the Scrum's roles, an additional role should be added (or its competence should be guaranteed by the product manager). This role should offer guidance with respect to standards' interpretation, which could be used to enrich the plans by adding specific information to enable compliance matrix generation. Even where a safety manager is not explicitly recommended (e.g., DO-178C), the need of interpreting the standard at planning phase is sound to be able to guide the development towards production of certifiable evidence. In [25], the planning phase is not explicitly discussed but only assumed and expected to benefit from the delayed first SOI interaction. In order to provide a better understanding of our hybrid process compliance vision via mapping tables, we have created an activity diagram in EPF (Eclipse Process Framework) Composer[3], as shown in Figure 5.

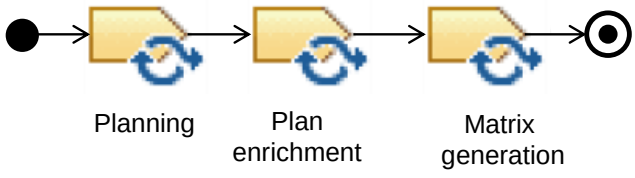

Figure 5: Hybrid Process Compliance Tables

[3]See https://www.eclipse.org/epf/

4.2 Hybrid Process-based Argumentation

Given the interpretation proposed by the safety manager, an argument could be formulated and eventually generated. In order to exploit MDSafeCer's principle for generating process-based argumentation (as mentioned in Section 2.3) user stories or product backlog can be used as initial or high-level plan as shown in Figure 6. By adopting a hybrid approach, the process-based argumentation shall continually provide the evidence for compliance with the plan at every sprint. Safety manager should validate the process plan to detect whether it contains all the essential information for supporting the key evidence(s). In case deviations are detected due to missing/wrong information in the process plan when writing the user story or product backlog, the feedback is provided regarding detected fallacies. The plan will be modified accordingly. Detecting fallacies and generating process-based argumentation are iterative and incremental tasks, in particular, the development project can be continuously re-planned based on found deviations. Then the process-based argumentation is generated from the modified process plan, consequently, it ensures that the argument is valid. Between the iterations (sprints), it is the duty of the customer or product owner to use the most recent evidence to re-prioritize the product backlogs. Therefore, a plan would benefit from the feedback from the certification body at each sprint. By so doing, safety awareness, understanding and confidence would increase.

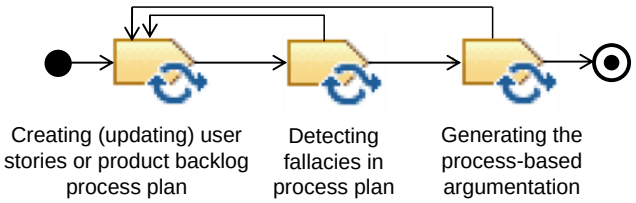

Figure 6: Hybrid Process-based Argumentation

4.3 Hybrid Processes Compliance Checking

The *initial plans* developed in the hybrid context can be automatically checked for compliance, with the benefit that a compliance report can help the manager in charge to understand how far (or close) the process plans are from the desirable state required by the standards requirements regarding processes. *R-Scrum* and *Safe Scrum* have explicitly defined feedback for re-planning the software process during its execution, after better understanding of the process is acquired. In addition, *R-Scrum* explicitly defines *Check points* carried out at the end of the Sprint. Check points can produce process enhancements and therefore re-planing. An static version of the *"new process"* can be automatically checked for compliance in the same way that the initial plan. We consider that automatic compliance checking could benefit the application of hybrid process by providing a general understanding of current software process state in term of compliance and providing future process improvements, supported by the information regarding non-compliant situations (violations of the requirements). Figure 7 summarises the hybrid process compliance checking vision. Specifically, the compliance checker takes the initial plans as inputs and produces a compliance report. The report can be used directly to make the adjustments required in the process plans and/or execution of interaction for the process. However, process executions and evaluations may also bring feedback for the process plans, that should be used to update the initial plans, to be checked iteratively.

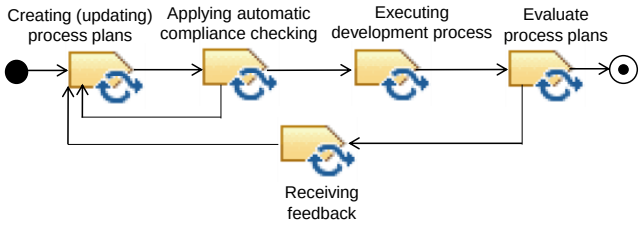

Figure 7: Hybrid Process Compliance Checking

4.4 Hybrid Processes Ontology-based Mapping

The *initial plans*, developed in the hybrid context, could be ontologically compared with previous plans in order to exploit potential reuse opportunities of compliance justifications. Figure 8 shows ontology-based mapping approach within the context of hybrid processes.

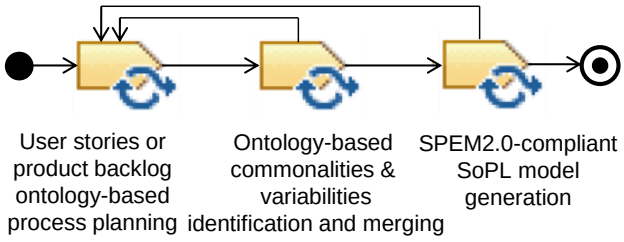

Figure 8: Hybrid Processes Ontology-based Mapping

4.5 Research Agenda

Our vision is expected to be further developed in the context of the AMASS project. In particular, the plan is to benefit from the different use cases stemming from various domains, capture state of practice way of working and use the compliance means, which are being implemented, to ease the communication with auditors. We are also interested in applying our implemented means to see if we can justify the hybrid development methods recalled in the background, at least those that, based on what was stated by the authors, have already been applied to produce certifiable evidence for limited portions of the development process.

5 CONCLUSION

In this paper, we have taken as starting point the outcome of a recent on-line survey, which has highlighted that hybrid development processes/practices have become a reality. Based on that outcome and based on the well-known challenges, posed by agile principles in the context of safety critical systems engineering, we have formulated the challenges related to compliance management of agilized

(software) development processes with safety standards. Then, we have presented our vision regarding the potential role of existing and complementary compliance means in the hybrid/agilized context.

Acknowledgments. This work is supported by EU and VINNOVA via the ECSEL Joint Undertaking under grant agreement No. 692474, AMASS project [3].

REFERENCES

[1] BS EN 50126. 1999. Railway Applications: The Specification and Demonstration of Reliability. Availability, Maintainability and Safety (RAMS). (1999).

[2] IEC 61508. 2010. Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems. (2010).

[3] AMASS. 2016. Architecture-driven, Multi-concern and Seamless Assurance and Certification of Cyber-Physical Systems. http://www.amass-ecsel.eu/. (2016).

[4] G. Antoniou, D. Billington, G. Governatori, and M. J. Maher. 2001. Representation Results for Defeasible Logic. *ACM Transactions on Computational Logic* 2, 2 (2001), 255–287. https://doi.org/10.1145/371316.371517

[5] A. Bärwald. 2014. ISO 26262 compliance-Addressing the compliance complexity of safety-relevant E/E systems. White Paper, Tüv Süd. (2014). Last accessed: 2018-06-02.

[6] J. Castellanos Ardila and B. Gallina. 2017. Formal Contract Logic Based Patterns for Facilitating Compliance Checking against ISO 26262. In *1st Workshop on Technologies for Regulatory Compliance, CEUR Workshop Proceedings, Luxembourg, December 13, 2017.* 65–72.

[7] J. Castellanos Ardila and B. Gallina. 2017. Towards Efficiently Checking Compliance Against Automotive Security and Safety Standards. In *7th IEEE International Workshop on Software Certification (WoSoCer) satellite event of ISSRE, Toulouse, France, October 23-26, 2017.* IEEE Computer Society, 317–324. https://doi.org/10.1109/ISSREW.2017.33

[8] J. Castellanos Ardila and B. Gallina. 2017. Towards Increased Efficiency and Confidence in Process Compliance. In *24th European Conference on Systems, Software and Services Process Improvement (EuroSPI), Ostrava, Czech Republic, September 6-8, 2017.* Springer, 162–174. https://doi.org/10.1007/978-3-319-64218-5_13

[9] M. Conrad. 2012. Artifact-Centric Compliance Demonstration for ISO 26262 Projects Using Model-Based Design. In *GI-Jahrestagung*, Vol. 208. GI, 807–816.

[10] ECSEL - AMASS. 2016. D2.2 AMASS reference architecture (a). (2016). Last accessed: 2018-03-09.

[11] B. Fitzgerald, K. J. Stol, R. O'Sullivan, and D. O'Brien. 2013. Scaling Agile Methods to Regulated Environments: An Industry Case Study. In *35th International Conference on Software Engineering (ICSE), San Francisco, CA, USA, May 18-26, 2013.* IEEE Computer Society, 863–872. https://doi.org/10.1109/ICSE.2013.6606635

[12] B. Gallina. 2014. A Model-driven Safety Certification Method for Process Compliance. In *2nd International Workshop on Assurance Cases for Software-intensive Systems, joint event of ISSRE, Naples, Italy, November 3-6, 2014.* IEEE, 204–209. https://doi.org/10.1109/ISSREW.2014.30

[13] B. Gallina, E. Gómez-Martínez, and C. Benac Earle. 2016. Deriving Safety Case Fragments for Assessing MBASafe's Compliance with EN 50128. In *16th International Conference on Software Process Improvement and Capability Determination (SPICE), Dublin, Ireland, June 9-10, 2016 (Communications in Computer and Information Science)*, Vol. 609. Springer, 3–16.

[14] B. Gallina and M. Nyberg. 2015. Reconciling the ISO 26262-compliant and the Agile Documentation Management in the Swedish Context. In *third Workshop on Critical Automotive applications: Robustness & Safety (CARS), Paris, France, September 7-11, 2015*, M. Roy (Ed.). open-access eternal archive HAL. https://hal.archives-ouvertes.fr/hal-01192981

[15] B. Gallina, I. Sljivo, and O. Jaradat. 2012. Towards a Safety-Oriented Process Line for Enabling Reuse in Safety Critical Systems Development and Certification. In *35th Annual IEEE Software Engineering Workshop (SEW), Heraclion, Crete, Greece, October 12-13, 2012.* IEEE Computer Society, 148–157. https://doi.org/10.1109/SEW.2012.22

[16] B. Gallina and Z. Szatmári. 2015. Ontology-based Identification of Commonalities and Variabilities among Safety Processes. In *16th International Conference on Product-Focused Software Process Improvement (PROFES), Bolzano, Italy, December 2-4, 2015.* Springer, 182–189. https://doi.org/10.1007/978-3-319-26844-6_13

[17] G. Governatori. 2005. Representing Business Contracts in RuleML. *International Journal of Cooperative Information Systems.* 14, 2-3 (2005), 181–216. https://doi.org/10.1142/S0218843005001092

[18] G. Governatori. 2015. The Regorous Approach to Process Compliance. In *IEEE 19th International Enterprise Distributed Object Computing Workshop (EDOCW), Adelaide, Australia, September 21-25, 2015.* IEEE, 33–40. https://doi.org/10.1109/EDOCW.2015.28

[19] ISO26262. 2011. Road vehicles Ð Functional safety. International Standard, November. (2011).

[20] G. Koliadis and A. Ghose. 2007. Verifying Semantic Business Process Models in Verifying Semantic Business Process Models in Inter-operation. In *IEEE International Conference on Service-Oriented Computing (SCC), Salt Lake City, Utah, USA, July 9-13, 2007.* IEEE Computer Society, 731–738. https://doi.org/10.1109/SCC.2007.128

[21] M. Kuhrmann, P. Diebold, J. Münch, P. Tell, V. Garousi, M. Felderer, K. Trektere, F. McCaffery, O. Linssen, E. Hanser, and C. R. Prause. 2017. Hybrid Software and System Development in Practice: Waterfall, Scrum, and Beyond. In *International Conference on Software and System Process (ICSSP), Paris, France, July 5-7, 2017.* ACM, New York, NY, USA, 30–39. https://doi.org/10.1145/3084100.3084104

[22] M. Kuhrmann, P. Diebold, J. Münch, P. Tell, K. Trektere, F. Mc Caffery, V. Garousi, M. Felderer, O. Linssen, E. Hanser, and C. R. Prause. 2018. Hybrid Software Development Approaches in Practice: A European Perspective. *IEEE Software* (2018).

[23] R. Maguire. 2006. *Safety Cases and Safety Reports: Meaning, Motivation and Management.* CRC Press.

[24] The Agile Manifesto. 2001. Manifesto for Agile Software Development. http://agilemanifesto.org/. (2001). Last accessed: 2018-03-01.

[25] J. Marsden, A. Windisch, R. Mayo, J. Grossi, J. Villermin, L. Fabre, and C. Aventini. 2018. ED-12C/DO-178C vs. Agile Manifesto: A Solution to Agile Development of Certifiable Avionics. In *9th European Congress Embedded Real Time Software and Systems (ERTS), Toulouse, France, January 30-February 2, 2018.*

[26] F. McCaffery, M. Lepmets, K. Trektere, Ö. Özcan-Top, and M. Pikkarainen. 2016. Agile Medical Device Software Development. *International Journal on Advances in Life Sciences* (2016), 181–216.

[27] B. McIsaac. 2015. *IBM Rational Method Composer: Standards Mapping.* Technical Report. IBM Developer Works. 1–19 pages.

[28] F. UL Muram, B. Gallina, and L. Gomez Rodriguez. 2018. Preventing Omission of Key Evidence Fallacy in Process-based Argumentations. In *11th International Conference on the Quality of Information and Communications Technology (QUATIC), Coimbra, Portugal, September 4-7, 2018.* (in press).

[29] Object Management Group (OMG). 2008. Software & Systems Process Engineering Metamodel Specification (SPEM), Version 2.0. http://www.omg.org/spec/SPEM/2.0/. (2008). Last accessed: 2018-03-01.

[30] Object Management Group (OMG). 2013. Structured Assurance Case Metamodel Specification, Version 1.0. https://www.omg.org/spec/SACM/1.0/About-SACM/. (2013). Last accessed: 2018-03-01.

[31] RTCA Inc. 2013. *Software Considerations in Airborne Systems and Equipment Certification, RTCA DO-178C (EUROCAE ED-12C).* Washington DC.

[32] J. M. Rushby. 2007. Just-in-Time Certification. In *12th International Conference on Engineering of Complex Computer Systems (ICECCS), Auckland, New Zealand, July 10-14 2007.* IEEE Computer Society, 15–24. https://doi.org/10.1109/ICECCS.2007.26

[33] T. Stålhane, T. Myklebust, and G. Hanssen. 2012. The application of safe scrum to IEC 61508 certifiable software. In *11th International Probabilistic Safety Assessment and Management Conference and the Annual European Safety and Reliability Conference (PSAM ESREL), Helsinki, Finland, June 25-29, 2012*, Vol. 8. 6052–6061.