Journal of Universal Computer Science, vol. 27, no. 8 (2021), 868-893
submitted: 8/4/2021, accepted: 2/5/2021, appeared: 28/8/2021 CC BY-ND 4.0

Lean integration of IT security and data privacy governance aspects into product development in agile organizations

Alexander Poth
(Volkswagen AG, Berliner Ring 2, D-38436 Wolfsburg, Germany
https://orcid.org/0000-0002-2868-5633, alexander.poth@volkswagen.de)

Mario Kottke
(Volkswagen AG, Berliner Ring 2, D-38436 Wolfsburg, Germany
mario.kottke@volkswagen.de)

Kerstin Middelhauve
(Audi AG, D-85045 Ingolstadt, Germany
kerstin.middelhauve@audi.de)

Torsten Mahr
(Volkswagen Financial Services AG, Gifhorner Str. 57, D-38122 Braunschweig, Germany
torsten.mahr@vwfs.com)

Andreas Riel
(Université Grenoble Alpes, CNRS, G-SCOP, F-38000 Grenoble, France
https://orcid.org/0000-0001-9859-019X, andreas.riel@grenoble-inp.fr)

Abstract: This article deals with the design of a product development-specific framework to support lean and adequate governance. This framework is based on layers of product-specific standards and regulations. The layers can be merged into a specific set to address the demands of a product to fit the state-of-the-art requirements of its domain. For the product domain, specific layers are presented with examples from IT security and data privacy for the software development phase. The approach is generic and can be extended to other domains like finance services or embedded products and their life-cycle phases.

Keywords: Lean Software Development, Agile Software Development, IT Governance, IT Compliance
Categories: D.2
DOI: 10.3897/jucs.71770

1 Introduction

Many business domains have established ways to address regulations like the General Data Protection Regulation (GDPR) [GDPR, 21] for privacy in the European Union, and standards like the ISO 27000 series for IT Security Management Systems (ISMS) [ISO27000, 18]. IT products developed for these legal areas and business domains have to be compliant with the state-of-the-art regulations and standards. Organizations have

to ensure the product compliance with controls and checks of their products. Depending on the structure and culture established in a company, the organization can choose different approaches to comply with regulations and standards. This is possible because applicable regulations and standards mostly impose requirements with focus on what and not on how. The main driver for the selected instantiation approach is the type of accountability that is used within the organization. Many concepts like [Seal, 06], experiences like [Herbert, 12] and examples like [Abdel-Kader, 08] or [Karhapää, 21] exist to show how to instantiate compliance in established classical hierarchical organizations of companies and large enterprises. Agile organizations structure this accountability differently, i.e., in autonomous product teams. To support this way of working, the governance has to be aligned with these organizations' types of accountability and responsibilities. In agile environments, a shared responsibility approach is common [McHugh, 11]. The expectation of the product teams is that shared responsibility [Scott, 05] approach will be supported by the governance, too. This leads to the expectation that a lean governance is established, and procedures for compliance are aligned with the agile mindset and procedures of product development and delivery working.

With this expectation, a set of questions around accountability and responsibility arises: Who is accountable and responsible for the specific governance instantiation in terms of

1. the *selection* of all relevant (regulation) requirements?
2. the *implementation* of (regulation) requirements in the product and its organizational setup?
3. the *check* of the compliant application of the (regulation) requirements by the teams?

A risk management of the shared responsibility approach is needed to make the current state of the instantiation and application of the specific governance actions transparent for an active handling of the identified risks on organizational level. In [Poth, 20a], a systematic check of the decentralized instantiations is proposed and the autonomy grows with team maturity [Poth, 20b].

Typical objective of an agile organization is to adapt specific product team demands by

- designing an approach which fosters agility of product teams by keeping compliant;
- fostering lean governance by reference/base to the source requirements.

Various different more or less suitable solution approaches to implementing compliance governance exist:

- *A central governance* unit for security, privacy, Free/Open Source Software (FOSS) compliance etc. These organizations tend to establish one big governance framework to address all (edge) cases. This may lead to a one-size-fits-all approach, bureaucracy, and frustration in agile teams.
- *Local Governance* units for domain-specific instances of governance like ISMS. This kind of organization tends to multiply efforts for compliance

in all phases of the life cycle with plan, build, run the local governance instantiation.
- *Meta models* are mapping all regulation and governance relevant aspects of a business domain into one model. One generic, however, probably complex, point of truth as base for specific derivations. All derivations are based on the indirection of the meta model and focused/reduced by context-specific filtering. The filter is the key for the outcome completeness and leanness.

We are building our approach based on the assumption that industrial organizations have an increasing demand for lean approaches that are adaptable to different organizations. Therefore, the solution approach has to address the following requirements:

a) Define the scope of the product-specific compliance setup to avoid unnecessary efforts.
b) Build a transparent base of implemented regulation and standard requirements to make transparent for everybody what is handled and what is not (base for sharing responsibility) and to make it easy to identify non-necessary aspects for the specific product context.
c) Have the possibility to combine different regulations and standards to make the approach generic and applicable in different product domains.
d) Foster a lean and agile mindset by design to get acceptance in modern organizations.

Section 2 presents related work. Section 3 describes the methodology. Section 4 presents applications in the security and privacy domains. Section 5 evaluates the insights and results obtained during the iterative development process. Section 6 concludes with a discussion, while section 7 gives an outlook.

2 Related work

The accountability and responsibility are a basic concept to established a shared-responsibility approach [Lindkvist, 03]. However, the approach has to be designed for a lean governance environment to ensure leanness across its entire life cycle by design. To ensure adaption to different organizations and business domains, process tailoring approaches are relevant, too.

2.1 Accountability and responsibility

Accountability and responsibility are a widely discussed topic in different contexts like e.g. Cooperate Social Responsibility [Ribstein, 05]. Various types of accountability exist [Erkkilä 07], see table 1.

Type of accountability	Features	Mechanisms of Accountability	Context (Structure)
Political accountability	Democratic, external	Democratic elections, chain of accountability	Democratic state
Bureaucratic accountability	Hierarchic, legal	Rules, regulations, supervision	Bureaucracy
Personal accountability	Internal, normative, moral	Culture, values, ethics	Collective
Professional accountability	Complex, 'deferent to expertise', peer-oriented	Expert scrutiny, peer review, professional role	Expert organisation
Performance	Output or client-oriented	Competition, self-regulation	Market
Deliberation	Interactive, deliberative, open, public	Public debate, deliberation, transparency, access to information	Public sphere

Table 1: Types of accountability according to [Erkkilä 07]

Shared responsibility is established in agile approaches in the safety domain with SafeScrum® [Hanssen, 18] or R-Scrum [Fitzgerald, 13]. Both approaches use a shared responsibility approach to ensure that all relevant safety related artefacts are built and maintained.

2.2 Lean governance

Scaling agile software development through lean governance is described in [Ambler, 09]. There, the bridge from traditional IT governance to agile value-driven work is proposed.

Ensuring regulations compliance is a big topic in governance. In [Mussmann, 20], mappings of security standards like ISO 27001, ISO 27002, GDPR, COBIT [COBIT, 21] and BSI C5 [BSI, 21] are analyzed, as well as how they can be mapped to each other directly or via an ontology. This shows that generic IT standards like the ISO 27000 series (short ISO 27000 in this article) need alignment with technology domain-specific standards like BIS C5 for cloud computing. In [Di Giulio, 17], this technology domain-specific comparison is made in more depth. This leads to the challenge that depending on the specific product, business and technology domain-specific mappings are needed.

Lean governance frameworks have been developed over years like [Pinheiro, 14]. To combine them with agile [Ambler, 09] is not new, either. However, the scope on teams has been established only later. In [Horlach, 18], lean governance aspects of frameworks are compared down to the teams by offering practices for different governance aspects. The safety domain agile approaches R-Scrum and SafeScrum® use team external assessors or auditors to ensure compliant instantiations over the product life-cycle. These are steps for continuous compliance. However, a systematic and critical self-reflection e.g. with retrospectives [Przybyłek, 17] to keep focus and stay lean can be a useful approach, too.

2.3 Process Tailoring

Tailoring approaches to agile processes are investigated and analyzed in [Akbar, 19]. Different types of process tailoring operations are identified like add, delete, modify, split, merge, shrink and wrap up. The RegTech approach comes from the finance domain [Butler, 19], however, is not limited to it [Johansson, 19]. It works with meta-models [Feltus, 17] to describe the different regulation requirements in a generic and holistic model. Then the model is implemented into different IT workflows to automate parts of them. Combination of standards like security and privacy are created in [Lopes, 19a] and [Lopes, 19b] or to the ISO 9000 [Tzolov, 18] to realize holistic compliance approaches. An approach or framework has to be designed openly and foster business agility [Triaa, 16].

3 Methodology

The development of the proposed approach is based on a design science research approach according to [Hevner, 07] with the three cycles for relevance, design and rigor. In a first step, the relevant concepts like shared responsibility are analyzed and then combined and integrated to the proposed approach. Then the evaluation starts and iterates as long as the approach needs refinement to get acceptance by the practitioners of the evaluation context.

3.1 Shared responsibility for regulations compliance and standard requirements

To ensure that the responsibility and accountability for the compliance is established adequately, the organization has to identify the appropriate type of accountability. To map the table 1 to enterprises and organizations, the established main type of accountability has to be identified to enable an adequate instantiation of a lean governance approach. In classical enterprises, accountability is mainly allocated per hierarchy and legal competences. Often less pronounced are personal accountability by culture, value and ethics as a second type of accountability. Other types of accountability are included as a kind of "supporting" accountability to the main type.

In agile mindsets and methods, the professional accountability with peer reviews and professional roles with expert scrutiny are dominant. Furthermore, deliberation, transparency and information access are part of the accountability, too. This can lead to holacracy [Holacracy, 19] based on democratic election and chains of accountability. Having these different types of accountability in one organization makes it difficult to

work in a hybrid organization because they all have to be strong enough to run the business. Hybrid organizations can exist with the classic enterprise setup driven by hierarchy. At one point, the "agile silo" is "integrated" by one accountable person who builds the connector to the "agile silo". Without this connector, it is difficult for the established classical accountability system to work with the agile organization. A hybrid organization coming from the agile type of accountability can add the classical accountability elements gradually without too much pain during the transformation.

To move from accountability to responsibility, the relevant aspect is that accountability is the ultimate instance (not shareable) after something happens or not. Responsibility is the owner of (future) tasks and can be shared between different owners. The shared responsibility approach is part of the autonomy in the agile mindset. The product teams have the autonomy to decide by having the responsibility for their decisions. In the case of compliance aspects, the paradigm leads to the teams getting support for structuring and ensuring their compliance. The governance supports the teams to decide how to instantiate the regulation and standard requirements. Furthermore, the ways of measuring their compliance to the requirements needs to be addressed.

For the proposed approach, governance experts are enabler by providing support through Self-Service Kits (SSK) [Poth, 20c] for specific regulation requirements sets, including examples showing their instantiation. This gives guidance to the product teams by keeping up their autonomy. This builds a shared responsibility in which the governance is responsible to select the relevant regulation and standard requirements for specific product domains. The product teams are responsible for the adequate instantiation. In terms of accountability, the governance has to ensure that the complete set of currently applicable regulation and standard versions provide the base for the SSKs. The product teams are accountable for the instantiation, compliant application and delivery of evidence about compliance status, since organizations depending on their regulations environment need to be able to know about their overall compliance status. Using Scrum terminology, the Scrum master is responsible for the rituals and procedures. Compliance is part of the Scrum master's duties as described in the Scrum guide [ScrumGuide, 20]. E.g. "The Scrum Master is accountable for the Scrum Team's effectiveness" by "Helping the Scrum Team focus on creating high-value Increments that meet the Definition of Done", "Causing the removal of impediments to the Scrum Team's progress" or "Removing barriers between stakeholders and Scrum Teams". The Scrum master is the ultimate instance in the team regarding the topic, and therefore accountable for compliant outcomes of the team as part of an effective team. However, the Scrum master can delegate tasks which are conducted by all team members. In other agile approaches like the Spotify model, the squad lead, who is responsible for delivery, can be the compliance accountable person. In all cases, the teams work in the companies' governance frameworks. They have the freedom to act in alignment within their area of autonomy in the governance setup of their organization, which is typically authorized and managed by the senior management.

Figure 1 shows the schematic concept behind the shared responsibility of compliance and accountability. Each team has an exposed person who is accountable for the team. However, everyone in the team can take over tasks as a responsible person. The shared responsibility between the governance and the product teams is realized by the instantiation of the relevant LoD (Level of Done, according to the concept presented in [Poth, 20a]) layers for the specific product setting. Independent checks about

compliance can be conducted by regulation experts of the company's internal governance teams or by company external experts depending on the regulation requirements. Furthermore, it is possible to establish cyclic checks conducted by other product teams to get compliance feedbacks and practical tips about instantiation alternatives from other practitioners. The overall accountability for compliance is assigned to a dedicated person like the Chief Governance Officer. The governance can conduct audits to check for adequate instantiation of LoD layers in product teams.

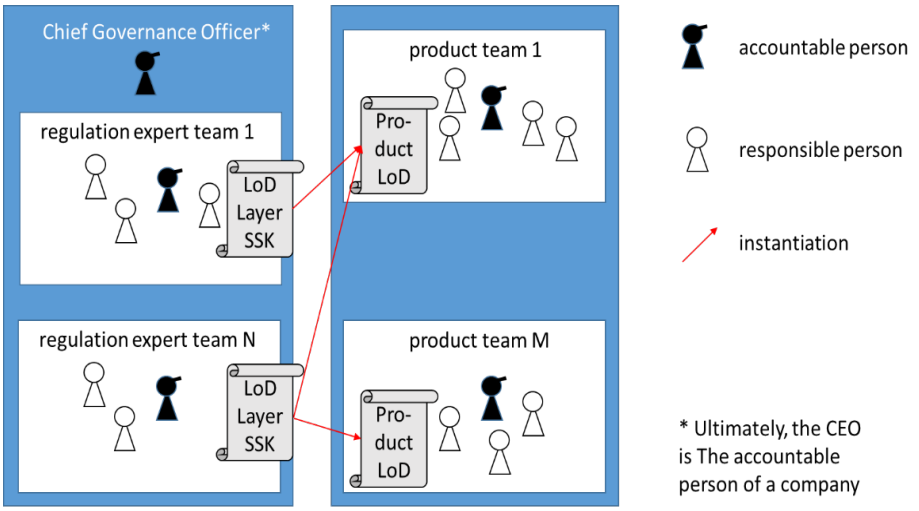

Figure 1: Schematic view of accountability and shared responsibility for compliance

3.2 LoD layer concept

The authors' LoD approach [Poth, 20a] provides the fundamental basis for the LoD layer concept presented here. First, the organization's handover points are identified. Each handover leads to an additional level of the LoD. At the core of the LoD concept are the experts for governance who identify all relevant requirements to be compliant for a specific organization which operates in a dedicated business domain. The selected requirements are mapped to the levels. Where possible, the requirements are de-duplicated if required (i.e., if e.g. they overlap). Then, the LoD for an organization of a specific business domain is ready for instantiation by their product teams. As the LoD only focuses on regulation and standard compliance, each product team has to identify product-specific risks to mitigate them adequately with associated actions. These actions are added to the LoD to build a product-specific LoD. By that, the LoD has been refined specifically for the business domain and is therefore ready to be used.

3.3 Lean governance

As long as the LoD is streamlined to the regulation requirements, there is not much room for tailoring by reducing aspects. However, the adaptation to the specific product team context is possible and needed for the integration into the "DNA" of the product

teams' workflows and deliverables for adequate responsibility for LoD compliance. The adaptation is based on integrating the specific activities and tasks into the product teams' delivery procedures. Furthermore, product-specific changes of the LoD have to be validated (with the LoD provider) and safeguarded with at least an additional compliance control to ensure that this modification is transparent for external compliance checks. The product team is responsible for the compliant implementation and continuous application of the LoD. However, the governance will still conduct external compliance checks or demands regular evidence about compliance status, which is preferable, because it gives the team autonomy how to achieve evidence and reduces the external compliance check scope and needs.

3.4 Shared responsibility for regulations compliance and standard requirements

For the presented large-scale industrial context, the lean governance charge in the shared responsibility approach is:

- Stay up-to-date about external regulations and standards.
- Offer up-to-date LoD layer.
- Offer a Self-Service Kit (SSK) and additional training and learning material to facilitate the LoD layer instantiation.
- Pre-validate the LoD layer with typical conflict areas that offer also LoD layers; resolve conflicts or give clear recommendations for operational ways of handling these conflicts.
- Offer support to the product teams for instantiation issues and questions (serving governance).
- Cyclically check compliance of the LoD layer application of the instantiations in product teams, including learning what should be improved in the provided LoD layer for further increasing the compliance level.

In case of growing complexity in the decentralized domain-specific instantiation, a domain accountability for compliance can be useful. To establish decentralized accountability, local stakeholders can be made accountable at some or each of the company's sites. This helps making the accountability transparent and staying lean, because the local domain knowledge can be used to perform adequate instantiation, controls and checks. Figure 2 presents an approach to establishing decentralized accountability. The central Chief Governance Officer is accountable for the bullet points above. The Local Governance Officer is accountable for the business domain-specific instantiation and application within the product teams. This can reduce the compliance check efforts of the Chief Governance Officer significantly by giving autonomy to the local organization and thereby foster shared responsibility. Mastery about compliance is needed to ensure that the local responsibility will be instantiated adequately. To establish a lean governance, the layers should be kept as simple as possible to facilitate their acceptance and practical application in product teams.

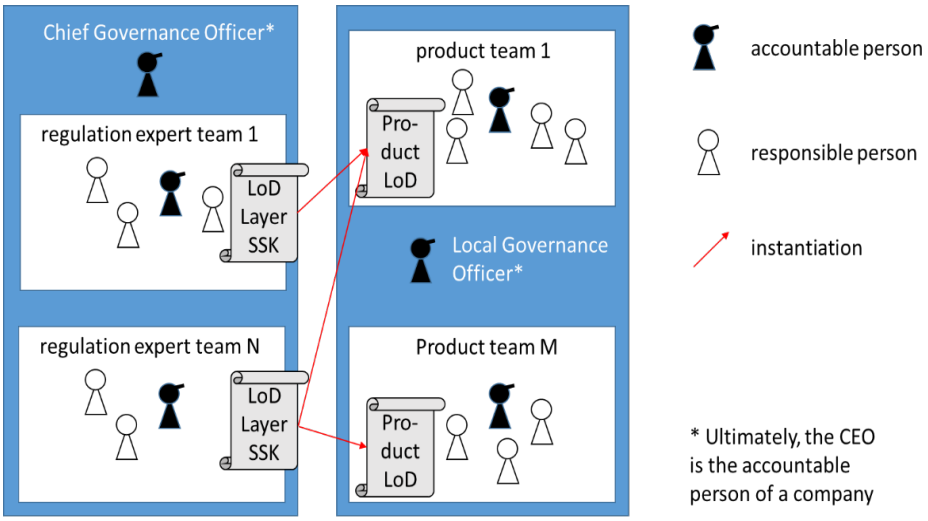

Figure 2: Schematic view of accountability and shared responsibility about local compliance

An example for the proposed setup can be the ISO 27000 with the ISMS, which typically the organization's Chief Information Security Officer (CISO) is accountable for. However, it is possible to have Local Information Security Officers (LISO) in the specific business areas or domains. The LISO is accountable for an adequate establishment of the ISMS in the organization for their specific business domain. This includes selection and refinement of relevant aspects of ISMS.

3.5 LoD layers

The LoD approach as described in [Poth, 20d] combines all relevant regulation and governance requirements in one single instance of LoD. This makes it difficult to see in the LoD, which source each part of the LoD comes from. This limits the method's flexibility of reusing parts of the LoD in another product context, which has similar but not equal conditions. To address this, the introduction of LoD layers enables an enterprise to have specific topic-related LoDs maintained by experts. LoD layers are dedicated for regulations like the GDPR or standards like the ISO 27000. Each LoD layer is self-contained and can be build and maintained by its independent experts. Different LoD layers are combined to build a complete LoD for a specific product. The layers can address domain-specific regulations and organizational compliance aspects. This makes it possible to stack layer by layer to a fitting domain-specific LoD. For example, the base layer could be the domain-specific regulation layer, the organization layer can be stacked on top, and finally the product-specific regulations layer could be added. Figure 3 visually presents the merging of two LoD layers. The two layers are for example offered by the independent regulation or standard experts and are merged by the product team. The merge process will have to handle the levels (columns of the Kanban board in Figure 3) of the LoDs that are combined. It also has to ensure that the dependencies of levels are maintained. This is the main work at the merge of different

LoDs to identify the level dependencies - respectively their tasks - between the different LoDs.

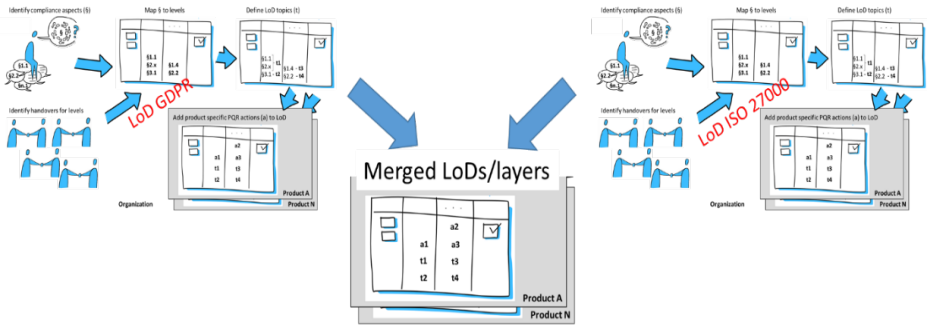

Figure 3: Schematic view of the merge of different standards and regulations into one LoD

The merge of more than two layers is done iteratively by merging the LoD layers with the highest amount of levels. For this, edge cases concerning the best and worst case of increasing the amount of layers need to be considered. In the best case, the amount of levels of the LoD with the higher number of level maintained by sorting the level of the other LoD's into the existing levels. In the worst case, all tasks of the LoD with the lower amount of levels have sequence dependencies with LoD containing the highest amount of levels. This leads to the point that each dependency requires a new level. However, this case is a theoretical edge case with a very low probability of occurrence in practice. Hence, the following relationship holds for the amount of levels in layered LoD after merge of LoDs:

$$\#LevelmoreLevelLoD <= \#LevellayerLoD <= (\#TaskslessLevelLoD * 2) + 1 + (\#LevelmoreLevelLoD - 1)$$

with "#" meaning "amount of", and "<=" meaning "smaller or equal to".

Edge case: the semantic content of layers are contradicting. In this case, the formal merge can be conducted, but the semantic conflict cannot be solved. While the presented LoD layer approach helps identifying this issue, its resolution is beyond the scope.

4 Generic instantiation examples

This section elaborates on a minimal example case for the merging of two specific layers related to regulations and standards concerning data privacy and cybersecurity. The assumed organizational context is given by the following preconditions:

- The focus of the organization is on software development.
- The software is developed on customer demand.
- The software is developed using agile and lean methods and practices.
- The software is developed with in-house resources (software engineers and development infrastructure).
- The product is for the European Union market.

- The software development organization can rely on enterprise services like facility management for physical access and human resource departments caring about on/off-boarding – so these service providers are responsible for their process implementations.

The example focuses on LoD layers for the standardized security management aligned with an ISO 27000 based ISMS and privacy regulations compliance with the GDPR.

4.1 IEC/ISO 27000/1/2 layer for IT product development

Given by the context assumptions, the organizational (like employee on/off-boarding, physical access or teleworking) and operational (like monitoring and logging) aspects are not in scope. The LoD layer is motivated by the ISO 27000 chapter 4.6 and mentioned in the standard as critical success factor for implementing an ISMS with "information security policy, objectives, and activities aligned with objectives" and "an approach and framework for designing, implementing, monitoring, maintaining, and improving information security consistent with the organizational culture". Given by the context assumptions, the ISO 27000 [ISO27000, 18], ISO 27001 [ISO27001, 13] and 27002 [ISO27002, 13] are relevant. Especially domain specific refinements like the ISO 27009 and the 2701x are not in scope.

4.1.1 Identified requirements and relevant aspects for the LoD layer

The ISO 27001 controls dedicated to IT systems development are defined in annex 14. In addition to these explicit controls, there are generic aspects relevant to ensure the sensitiveness of the developers for security, and to ensure that the controls are up-to-date, applied and realized during the daily development business. The compliance has to be ensured by the (development) organization aligned with ISO/IEC 27002:2013 section Compliance. The source links to the ISO standard requirements are marked for traceability with the related number in parenthesis from which the text is extracted or derived. Especially the technical compliance reviews (18.2.3) for the software artefacts have to be checked. It is recommend to have this performed by an experienced system engineer with tool support for analysis and report generation. Moreover, security testing (like pen-testing or vulnerability assessments) has to be performed in a repeatable way and documented. The accountable and qualified person for this task is authorized to perform/supervise the reviews. The reviews are conducted by independent persons in intervals (18.2.1) and aligned with the security policies and standards (18.2.2).

The developers using cryptographic libraries need authorizations that are aligned with the product usage taking into account e.g. encryption restrictions in China or export restrictions from the European Union e.g. to North Korea (18.1.5). These controls have to be instantiated by the software development organization and teams.

The Intellectual Property (IP) rights (18.1.2) have to be ensured with appropriate procedures. Free/Open Source Software (FOSS) and proprietary licenses have to be fulfilled. An appropriate asset registry has to be maintained, in particular a Software Bill of Material (SBOM), which includes a listing of the reputable sources. The assets registry/lists have to be maintained and reviewed.

The privacy aspects (18.1.4) are refined by the relevant legislation and regulation like GDPR and have to be based on an organization's data policy for privacy and protection of personally identifiable information.

The developers have to care about (bug) tickets with focus on security incidents (16) and availability (17) (like business continuity and recovery) topics to support the operations. Especially the responsiveness to security incidents (16.1.5) has to be ensured. Furthermore, the developers have to establish a knowledge base from the analysis and resolving learnings of security incidents (16.1.6).

An appropriately protected secure development environment (14.2.6) like a dedicated dev-environment is required.

Development of software and systems is based on established rules of the organization (14.2.1) by e.g. secure coding guidelines, security in the development methodology, security requirements in the design phase, security checks within milestones (like sprints, release trains), required application security knowledge and developers' capability of avoiding, finding and fixing vulnerabilities. Security is established for repositories and the version control.

The systems have to be acceptance tested (14.2.9), security tested (14.2.8) and the test data have to be protected (14.3). Changes to software packages have to be limited to necessary and strictly controlled (14.2.4). System changes shall be controlled by formal change control procedures (14.2.2) which include review and approval by authorized users. The changes shall be documented and have an audit trail.

Secure system engineering principles (14.2.5) have to established, documented, maintained and applied.

Aspects like data classification (8) and labeling, access control (9), cryptography (10) and logging and monitoring (12.4) capabilities are requirements to the software under development too, but are refined and documented about their specific implementation with the relevant stakeholders and reviewed by all stakeholders (14.1.1). Furthermore, procedures like technical vulnerability management (12.6) have to be supported by the developers.

The developers have to establish awareness about security and be trained, educated, and updated about organizational policies and procedures, as well as their job function regularly (7.2.2).

Table 2 is an extract of the ISO 27001 with the scope-related audit controls. This is helpful to cross-check the LoD layer for completeness of the extracted requirements of the ISO 27000.

A.14 System acquisition, development and maintenance		
A.14.1 Security requirements of information systems		
Objective: To ensure that information security is an integral part of information systems across the entire lifecycle. This also includes the requirements for information systems which provide services over public networks.		
A.14.1.1	Information security requirements analysis and specification	*Control* The information security related requirements shall be included in the requirements for new information systems or enhancements to existing information systems.
A.14.1.2	Securing application services on public networks	*Control* Information involved in application services passing over public networks shall be protected from fraudulent activity, contract dispute and unauthorized disclosure and modification.
A.14.1.3	Protecting application services transactions	*Control* Information involved in application service transactions shall be protected to prevent incomplete transmission, mis-routing, unauthorized message alteration, unauthorized disclosure, unauthorized message duplication or replay.
A.14.2 Security in development and support processes		
Objective: To ensure that information security is designed and implemented within the development lifecycle of information systems.		
A.14.2.1	Secure development policy	*Control* Rules for the development of software and systems shall be established and applied to developments within the organization.
A.14.2.2	System change control procedures	*Control* Changes to systems within the development lifecycle shall be controlled by the use of formal change control procedures.
A.14.2.3	Technical review of applications after operating platform changes	*Control* When operating platforms are changed, business critical applications shall be reviewed and tested to ensure there is no adverse impact on organizational operations or security.
A.14.2.4	Restrictions on changes to software packages	*Control* Modifications to software packages shall be discouraged, limited to necessary changes and all changes shall be strictly controlled.
A.14.2.5	Secure system engineering principles	*Control* Principles for engineering secure systems shall be established, documented, maintained and applied to any information system implementation efforts.

A.14.2.6	Secure development environment	*Control* Organizations shall establish and appropriately protect secure development environments for system development and integration efforts that cover the entire system development lifecycle.
A.14.2.7	Outsourced development	*Control* The organization shall supervise and monitor the activity of outsourced system development.
A.14.2.8	System security testing	*Control* Testing of security functionality shall be carried out during development.
A.14.2.9	System acceptance testing	*Control* Acceptance testing programs and related criteria shall be established for new information systems, upgrades and new versions.
A.14.3 Test data		
Objective: To ensure the protection of data used for testing.		
A.14.3.1	Protection of test data	*Control* Test data shall be selected carefully, protected and controlled.

Table 2: The controls of the ISO/IEC 27001:2013 for IT systems development

4.1.2 Derivation of the ISO 27000 LoD layer

The ISO 27000 standard requires three layers for our example with the customer- (user-)driven development organization. The review of the security requirements with all stakeholders (14.1.1) leads to a handover of the refined and reviewed requirements to the development organization. Authorized users accept changes before operation (14.2.2). All other external/independent reviews could be modeled as handover, however, not all outcomes have to be reviewed in this case. Therefore, the explicit formal modeling of a level is not useful. This makes it possible to have the option to realize all ISO 27000 aspects in three LoD levels. The requirements 14.1.1 and the related refinements of 8, 9, 10 and 12 are assigned to the first level for the handover. As the final act of the development, the formal authorization of the change (release/version) is made (14.2.2) in the third level. All other identified aspects are mapped to the second level for the development. Table 3 presents an LoD layer for the ISO 27000 that has been established according to the logic explained above. Each line of the table addresses a topic. Some pillars have to handle more topics than others.

Customer level	**Development level**	**Approval level**
Perform an information security requirements analysis and specification (14.1.1)	Principles. Development of software and systems is based on established rules of the organization (14.2.1) by e.g. secure coding guidelines, security in the development methodology, security requirements in the design phase, security checks within milestones (like	Obtaining formal approval and acceptance of the change (release) by authorized users or customers to

including aspects like data classification and labeling (8), access control (9), cryptography (10) and logging (12).	sprints, release trains), required application security knowledge and developers' capability of avoiding, finding and fixing vulnerabilities. Secure system engineering principles (14.2.5) have to be established, documented, maintained and applied.	ensure (rigor) implementation of security requirements (14.2.2).
Info: technical operating aspects should be refined with the future operator team (key stakeholder) to fit the expectations for an effective ops.	Devstack. Establish an appropriately protected secure development environment which also includes segregation between different development environment and access control. (14.2.6) Security is established for repositories and version control. The changes have an audit trail. (14.2.2) Changes to software packages limited to necessary and strictly controlled (14.2.4).	
	Networking and data transfer. Securing application services on public networks includes authentication, authorization and protection of confidential information. Avoid the loss or duplication of transaction information. (14.1.2) Transactions should be protected to prevent incomplete transmission, mis-routing, unauthorized message alteration, unauthorized disclosure and unauthorized message duplication or replay. (14.1.3)	
	Privacy. The privacy aspects (18.1.4) are refined by the relevant legislation and regulation like GDPR and have to be based on an organization's data policy for privacy and protection of personally identifiable information. (Info: use the LoD layer GDPR)	
	Reviews. Especially the technical compliance reviews (18.2.3) for the software artifacts have to be checked. It is recommended to perform this with tool support for analysis which generates reports and by an experienced system engineer. Additionally security testing (like pen-testing or vulnerability	

	assessments) have to be performed within a repeatable way and documented. The accountable and qualified person for this task is authorized to perform/supervise the reviews. The reviews are conducted by an independent person in intervals (18.2.1) and aligned with the security policies and standards (18.2.2).	
	Testing. Upon platform changes, the business critical applications should be reviewed and tested to avoid impact on the organizational operations or security. Ensure that changes are made to the business continuous plans and notify operating appropriate in time. (14.2.3) Security function testing is established (14.2.8) Acceptance testing is established (14.2.9) Test data is selected carefully, protected and controlled (14.3.1)	
	Bug-Handling. The developers have to care about (bug) tickets with focus on security incidents (16) and availability (17) (like business continuity and recovery) topics to support the operations. Especially the response to security incidents (16.1.5) is ensured. Furthermore, the developers establish a knowledge base from the analysis and resolving learnings of security incidents (16.1.6).	
	Vulnerabilities. Vulnerability management is established to obtain technical vulnerabilities, act timely and risk-appropriate (12.6)	
	Legal. The developers using cryptographic libraries need authorizations that are aligned with the product usage taking into account e.g. encryption restrictions in China or export restrictions from the European Union e.g. to North Korea (18.1.5). The Intellectual Property (IP) rights (18.1.2) have to be ensured with appropriate procedures. Free/Open Source Software (FOSS) and proprietary licenses have to be fulfilled. An appropriate asset registry has to	

	be maintained, in particular a Software Bill of Material (SBOM), which includes a listing of the reputable sources. The assets registry/lists have to be maintained and reviewed.	

Table 3: LoD layer for ISO 27000 for in-house software development in an agile organization

4.2 GDPR

In this section, we present a schematic example of how to instantiate a GDPR LoD layer which may differ from enterprise to enterprise, as well as within enterprises between the interpretations of the different legal departments of the independent legal entities. The software aspects related to the GDPR have to be developed with the business owner to ensure effectiveness and completeness related to the use case and regulation compliance. However, developers should have sensitiveness to some aspects closely related to the software architecture, design and implementation to fulfill their part of the shared responsibility. The scope of the developers is to ensure that privacy by design is the default architecture for software and IT systems. Furthermore, they have to ensure that the records of processing activities are documented in a compliant way. Aspects like data processing outside the EU, i.e., in third countries, or with external processing partners - the processor - (like cloud providers or partner companies) are not considered here, based on the given frame conditions of an in-house development scenario.

4.2.1 Identified requirements and relevant aspects for the LoD layer

Below we provide some sample GDPR clauses [GDPR, 21] to show that the approach to designing a GDPR LoD layer is analogous to the one shown in the ISO 27000 series:

Art. 5 §1 Personal data shall be:

a) processed lawfully, fairly and in a transparent manner in relation to the data subject (‘lawfulness, fairness and transparency’);
b) collected for specified, explicit and legitimate purposes and not further processed in a manner that is incompatible with those purposes; … (‘purpose limitation’);
c) adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed (‘data minimization’);
d) accurate and, where necessary, kept up-to-date; every reasonable step must be taken to ensure that personal data that are inaccurate, having regard to the purposes for which they are processed, are erased or rectified without delay (‘accuracy’);
e) kept in a form which permits identification of data subjects for no longer than is necessary for the purposes for which the personal data are processed; … (‘storage limitation’);
f) processed in a manner that ensures appropriate security of the personal data, including protection against unauthorized or unlawful processing and against accidental loss, destruction or damage, using appropriate technical or organizational measures (‘integrity and confidentiality’).

Art. 5 §2 The controller shall be responsible for, and be able to demonstrate compliance with, paragraph 1 ('accountability').

Art. 6 §1 Processing shall be lawful only if and to the extent that at least one of the following applies:

a) the data subject has given consent to the processing of his or her personal data for one or more specific purposes;
b) …

Art. 6 §4 Where the processing for a purpose other than that for which the personal data have been collected is not based on the data subject's consent … the controller shall, in order to ascertain whether processing for another purpose is compatible with the purpose for which the personal data are initially collected, take into account, inter alia:

a) any link between the purposes for which the personal data have been collected and the purposes of the intended further processing;
b) the context in which the personal data have been collected, in particular regarding the relationship between data subjects and the controller;
c) the possible consequences of the intended further processing for data subjects;
d) the existence of appropriate safeguards, which may include encryption or pseudonymization.

Based on this selection, we provide some generic examples from the GDPR for inclusion in the LoD layer: Developers support the records of processing activities of Art. 30 with their implementation knowledge and data protection impact assessment of Art. 35 with their technology knowledge. Furthermore, developers act compliant to the code of conduct of Art. 40. The joint controllers' approach of Art. 26 is not suitable for the proposed shared responsibility approach because it impacts the external communication of the organization with its reference to Art. 13 and 14.

4.2.2 Derivation of the GDPR LoD layer

One level is demanded by the standard, because no handover points are required. Independent reviews could be modeled as handover, but in this case, not all outcomes have to be reviewed. Therefore, the explicit formal modeling of a level is not useful. The objective is to reduce the amount of handover points were possible to reduce organizational and process complexity to instantiate the handovers – keep it lean were possible.

This makes it possible to have the option to realize all GDPR aspects in one LoD level. However, for a customer driven development organization, as of our initial assumptions for this case study, a two-level LoD is the only practical option. It is needed to make transparent which aspects of the GDPR are managed by the development team, and which have to be handled outside of it. This leads to the mapping of Art. 25 and Art. 15 to the stakeholder/customer handover level, because the final accountability for data protection is a business topic. All other identified GDPR aspects are assigned to the development level. Table 4 presents an LoD layer for the GDPR. Art. 5 is used to structure the LoD layer GDPR. The GDPR's Articles are assigned to the structure like Art. 6 to purpose limitation. Table 4 does not limit the responsibility of the data owner.

Typically the data owner is located in the business unit which demands the data processing by IT systems or services.

Customer level	**Development level**
Data protection by design takes into account the state of the art, the cost of implementation and the nature, scope, context and purposes of processing as well as the risks of varying likelihood and severity for rights and freedoms of natural persons posed by the processing and necessary safeguards into the processing (Art. 25).	Lawfulness, fairness and transparency. Personal data shall be processed lawfully, fairly and in a transparent manner in relation to the data subject (user or other person) (Art. 5) Offer interfaces for structured, commonly used and machine-readable format and have the option to transmit those data to another controller/system (Art. 20). Be able to collect fast all data of an individual person on demand (Art. 16). Implement an easily accessible information for users/persons processing about the data subject in a concise, transparent, intelligible and easily accessible form, using clear and plain language – Art. 13 and Art. 14 offer additional information about typical content (Art. 12). The data subject shall have the right to withdraw his or her consent at any time. It shall be as easy to withdraw as to give consent – established by a rigor UX. The withdrawal of consent shall not affect the lawfulness of processing based on consent before its withdrawal. Prior to giving consent, the data subject shall be informed thereof (Art. 7). Be able for restriction, like temporal disabling, of data processing for individual personal data (Art. 18).
The controller shall implement appropriate technical and organisational measures for ensuring that, by default, only personal data which are necessary for each specific purpose of the processing are processed. That obligation applies to the amount of personal data collected, the extent of their processing, the	Purpose limitation. Personal data shall be collected for specified, explicit and legitimate purposes and not further processed in a manner that is incompatible with those purposes (Art. 5) - keep in mind to have consent for data usage in the context of testing, bug-reproduction and training of Machine Learning (ML) algorithms. The request for data subject's (user/person) consent shall be presented in a manner which is clearly distinguishable from the other matters, in an intelligible and easily accessible form, using clear and plain language. Any part of such a declaration which constitutes an infringement of this Regulation shall not be binding. Be able to demonstrate the consent of a person to legitimate the data processing (Art. 7). Where the processing for a purpose other than that for which the personal data have been collected is not based on the data subject's consent (like for ML training data or test data) the controller (developer/tester in the testing context) shall, in order to ascertain whether processing for another purpose is compatible with the purpose for which the personal data are initially collected, take into account, inter alia:

period of their storage and their accessibility (Art. 25)	a) any link between the purposes for which the personal data have been collected and the purposes of the intended further processing; b) the context in which the personal data have been collected, in particular regarding the relationship between data subjects and the controller; d) the possible consequences of the intended further processing for data subjects; e) the existence of appropriate safeguards, which may include encryption or pseudonymization (Art. 6). Processing of personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership, and the processing of genetic data, biometric data for the purpose of uniquely identifying a natural person, data concerning health or data concerning a natural person's sex life or sexual orientation shall be prohibited – safeguard the demand it in case of an implementation request (Art. 9).
Define what and how the data of an individual person are processed (Art. 15).	Data minimization. Personal data shall be adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed (Art. 5). Be able to delete data on an individual person on request or automatically if purpose for processing is not given anymore (Art. 17). If processing of personal data do not or do no longer require the identification of a data subject the data can be deleted (Art. 11).
	Accuracy. Personal data shall be accurate and, where necessary, kept up-to-date; every reasonable step must be taken to ensure that personal data that are inaccurate, having regard to the purposes for which they are processed, are erased or rectified without delay (Art. 5).
	Storage limitation. Personal data shall be kept in a form which permits identification of data subjects for no longer than is necessary for the purposes for which the personal data are processed (Art. 5). Enable policy or life-cycle driven delete of expired data (Art. 15)
	Integrity and confidentiality. Personal data shall be processed in a manner that ensures appropriate security of the personal data, including protection against unauthorized or unlawful processing and against accidental loss, destruction or damage, using appropriate technical or organizational measures (Art. 5).

	Appropriate security which includes encryption, pseudonymization of data and the confidentiality, integrity, availability and resilience of the processing systems and services. This includes regularly testing, assessing and evaluation of the effectiveness of the technical security (Art. 32). (Info: use the LoD layer ISO 27000)
	Accountability. Be responsible and be able to demonstrate compliance (Art. 5). Be accountable for adequate data-protection by design and by default contributions to the overall software and its derived component implementation of data protection by design and by default (Art. 25). Be accountable for acting compliant to the privacy code of conduct of the organization (Art. 40). Be accountable for correct and updated information contributed into the records of processing (Art. 30). Be accountable for technical knowledge contributed to the data protection impact assessment (Art. 35).

Table 4: LoD layer for GDPR for in-house software development in an agile organization

4.3 Merge of the ISO 27000 layer and the GDPR layer

In this example, the merge is trivial because no conflicts or dependencies between a sequential work-order between the ISO 27000 and GDPR are identified. In the ISO 27000 18.1.4 the privacy aspects are "delegated" to the GDPR for refinement. In this case, both layers can be stacked into the same level of an LoD. This is beneficial, because it allows the developer team to map the merged level without required interfaces. This enables a dev-team to work autonomously in the context of ISO 27000 and GDPR as long as mastery for autonomy is given.

5 Evaluation

The presented evaluation setting is the Volkswagen Group IT. The evaluation took place in the Volkswagen AG, Audi AG and Volkswagen Financial Services AG.

5.1 Build of the approach

The focused layers for security and privacy are derived in a Community of Practice (CoP) initiated by Volkswagen's Agile Center of Excellence (ACE). The CoP was built with experts of agile methods, quality management and assurance, software engineers and governance standard specialists.

5.2 Offer to the product teams

The LoD approach itself is provided as SSK, as well as each layer. The basic layer contains the basics for IT software development aligned with the ISO 9000. The

security layer addresses the IT software development related aspects of the ISO 27000. The privacy layer addresses the GDPR aspects for IT software development. The combination of the layers is "pre-verified" by the VW Group-IT Agile Center of Excellence (ACE) for fast and easy instantiation in the product teams.

Some additional information concerning the layer derivation that is not presented in this article: The ISO 9000:2015 requires in clause 8.2 the determination of requirements with (potential) customers and clause 8.6 authorization (like sign-off) of the product or service by the customer where possible or another authorizer after verification and validation of the product or service. This leads to at least a two level LoD. However, in a typical customer-driven development a third level is needed to fulfill clause 8.3.1 to hand over the product or service requirements from? the customer or interested parties to the design and development process.

This "pattern" of customer requirements, development, customer acceptance maps with the presented levels of the LoD layer for the ISO 27000 and GDPR.

5.3 Usage and application

The product teams used the LoD layer SSKs to decide about the relevance of each potential layer for their product. Then they merged the relevant layers and added specific instantiations where they considered useful like for the GDPR Art. 5 §1 a) practices on how the data flows shall be documented. Once the building of the LoD starts and the teams commit to the content, the teams start to take over responsibility. The tour of mastery for more governance autonomy starts. During the evaluation, some teams – depending on factors like the current product and its life-cycle state – also maintained the established governance tasks and procedures for safeguarding the evaluation experiment on compliance.

5.4 Learnings of the product teams

The minimal amount of levels for a customer-driven development organization aligned with the ISO 9000 is a three level LoD. The handover from the customer to the development team and the handover for the authorization of the release by the customer leads to the three levels. In case of an additional operation, the fourth level is needed for the operating/serving. This has an implication for devops-teams, which have to establish a level for the release authorization by the customer between dev and ops. The development organization can establish a pre-merged LoD with the ISO 9000 layer and ISO 27000 layer for all products and for products with privacy aspects the GDPR layer as an additional layer.

6 Discussion and Conclusion

The general contribution of this paper is an approach to ensure a systematic shared responsibility in large agile organizations between the governance and product teams. The approach presents examples of accountability in large organizations and the possibilities to delegate responsibility to autonomous teams via the LoD layers. The LoD layers are built and maintained by the governance and compliance experts of the entire organization and the instantiation is made by the autonomous product teams with

their knowledge about the delivered services and products. This leads to a lean governance.

6.1 Contribution for practitioners

Practitioners are focused on observations and practices applicable/transferable to their context:

- The presented LoD layer approach can be used as a common base for product development in inter-legal entities to establish a governance environment without adding more complexity by merging all existing governance frameworks of the legal entities.
- The presented LoD layer approach can reduce a complex set of internal rules and regulations of enterprises based on relevant external regulation and standards without additional interpretations and extensions.
- Depending on the enterprise's accountability type, the change to a lean governance approach will be more or less difficult.
- The shared responsibility of adequate compliance instantiation in agile organizations leads to a shift from classical centralized governance to product teams; autonomy comes with the mastery of compliance.
- The central governance provides to agile product teams the domain-specific regulations and standards requirements. Furthermore, governance makes independent checks of the adequate instantiation and application within the organization.
- This is a chance for classical enterprises to stop growing or start reducing centralized governance functions by developing this knowledge and awareness in the affected product teams.
- The moving responsibility is a chance for classical organization to reduce hierarchy and its management functions by developing more qualified job profiles in the product teams.

6.2 Contribution for researchers

Researchers are primarily interested in new insights and potential investigation areas:

- Not all parts of an enterprise need an agile organization. A connector is needed for transforming the instantiated types of accountability between the established and the agile part of the enterprise beyond the separation into independent legal entities or install the "hero-interface-manager" who is "silo-accountable".
- The accountability type changes over time have to be investigated more to build clear transition paths – especially in the direction from classical to agile organizations.
- The current initiative is initiated and driven bottom-up. The initiative has identified central governance functions as potential entities which are slowing down delivery performance. There is no work on the possibilities and limitations of bottom-up transitions in governance to evaluate the chances of success of a bottom-up initiative.

- The support with technology to instantiate regulations and standards transparently has to be pushed and needs investigation to ensure a trustful usage in product teams and help to scale efficiently.
- Establishing automated checks of compliances controls will be the next big step. Efficient scaling methods for compliance checks and compliance derivation risk evaluation and mitigation within large enterprises will have to be investigated.

6.3 Limitations

The evaluation is a limited study of selected cases and not a systematic application within a large organization. It demonstrates opportunities, but does not provide evidences on a large company level – the design science research rigor cycle is limited at this point. The approach works better in purely agile organizations because their types of accountability are more compatible with the decentralized handling of compliance in a lean governance approach. Furthermore, the amount of investigated LoD layers is limited. We also did not demonstrate edge cases like a lot of layers in one product team, or conflicting handling of contrary requirements in LoD layers.

7 Future Work

Upcoming steps will show if the evaluation of selected field studies becomes a transition. The change support from all involved governance parties will be crucial. Additionally, over time we expect that the amount of LoD layers will grow, driven by the company's obligation of addressing the different product and business domains of their increasingly heterogeneous product portfolio.

References

[Akbar, 19] Akbar, R., 2019. Tailoring agile-based software development processes. IEEE Access, 7, pp.139852-139869.

[Ambler, 09] Ambler, S. W., 2009. Scaling agile software development through lean governance, 2009 ICSE Workshop on Software Development Governance, Vancouver, BC, Canada, 2009, pp. 1-2, doi: 10.1109/SDG.2009.5071328.

[Abdel-Kader, 08] Abdel-Kader, M. and Luther, R., 2008. The impact of firm characteristics on management accounting practices: A UK-based empirical analysis. The British Accounting Review, 40(1), pp.2-27.

[BSI, 21], C5 (Cloud Computing Compliance Criteria Catalogue), 2021, https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/Empfehlungen-nach-Angriffszielen/Cloud-Computing/Kriterienkatalog-C5/C5_AktuelleVersion/C5_AktuelleVersion_node.html;jsessionid=642822DF7E4EF5CB0B44C5842A17455E.internet082

[Butler, 19] Butler, T. and O'Brien, L., 2019. Understanding RegTech for digital regulatory compliance. In Disrupting Finance (pp. 85-102). Palgrave Pivot, Cham.

[COBIT, 21] Effective IT Governance at Your Fingertips, 2021 https://www.isaca.org/resources/cobit

[Di Giulio, 17] Di Giulio, C., Sprabery, R., Kamhoua, C., Kwiat, K., Campbell, R.H. and Bashir, M.N., 2017, June. Cloud standards in comparison: Are new security frameworks improving cloud security?. In 2017 IEEE 10th International Conference on Cloud Computing (CLOUD) (pp. 50-57). IEEE.

[Erkkilä, 07] Erkkilä, T., 2007. Governance and accountability-A shift in conceptualisation. Public Administration Quarterly, pp.1-38.

[Feltus, 17] Feltus, C., Grandry, E. and Fontaine, F.X., 2017. Capability-driven design of business service ecosystem to support risk governance in regulatory ecosystems. Complex Systems Informatics and Modeling Quarterly, (10), pp.75-99.

[Fitzgerald, 13] Fitzgerald, B., Stol, K.J., O'Sullivan, R. and O'Brien, D., 2013, May. Scaling agile methods to regulated environments: An industry case study. In 2013 35th International Conference on Software Engineering (ICSE) (pp. 863-872). IEEE.

[GDPR, 21] Complete guide to GDPR compliance, 2021, https://gdpr.eu/

[Hanssen, 18] Hanssen, G.K., Stålhane, T. and Myklebust, T., 2018. SafeScrum®-Agile Development of Safety-Critical Software. Springer International Publishing.

[Herbert, 12] Herbert, I.P. and Seal, W.B., 2012. Shared services as a new organisational form: Some implications for management accounting. The British Accounting Review, 44(2), pp.83-97.

[Hevner, 07] Hevner, A.R., 2007. A three cycle view of design science research. Scandinavian journal of information systems, 19(2), p.4.

[Holacracy, 19] HolacracyOne, 2019. EVOLVE YOUR ORGANIZATION. https://www.holacracy.org

[Horlach, 18] Horlach, B., Böhmann, T., Schirmer, I. and Drews, P., 2018. IT governance in scaling agile frameworks. Proceedings of the Multikonferenz Wirtschaftsinformatik, Lüneburg, pp.1789-1800.

[ISO27000, 18] Information technology — Security techniques — Information security management systems — Overview and vocabulary - https://www.iso.org/standard/73906.html

[ISO27001, 13] Information technology — Security techniques — Information security management systems — Requirements, 2013, https://www.iso.org/obp/ui/#iso:std:iso-iec:27001:ed-2:v1:en, Last request: 09.06.2021

[ISO27002, 13] Information technology — Security techniques — Code of practice for information security controls, 2013, https://www.iso.org/standard/54533.html, Last request: 09.06.2021

[Johansson, 19] Johansson, E.., Sutinen, K.O.N.S.T.A., Lassila, J., Lang, V., Martikainen, M.. and Lehner, O.M., 2019. Regtech - a necessary tool to keep up with compliance and regulatory changes. ACRN Journal of Finance and Risk Perspectives, Special Issue Digital Accounting, 8, pp.71-85.

[Karhapää, 21] Karhapää, P., Behutiye, W., Rodríguez, P. et al. Strategies to manage quality requirements in agile software development: a multiple case study. Empir Software Eng 26, 28 (2021). https://doi.org/10.1007/s10664-020-09903-x

[Lindkvist, 03] Lindkvist, L. and Llewellyn, S., 2003. Accountability, responsibility and organization. Scandinavian Journal of Management, 19(2), pp.251-273.

[McHugh, 11] McHugh, O., Conboy, K. and Lang, M., 2011. Agile practices: The impact on trust in software project teams. Ieee Software, 29(3), pp.71-76.

[Lopes, 19a] Lopes, I. M., Guarda, T. and Oliveira, P.,2019. Implementation of ISO 27001 Standards as GDPR Compliance Facilitator. Journal of Information Systems Engineering & Management, 4(2), em0089. https://doi.org/10.29333/jisem/5888

[Lopes, 19b] Lopes I. M. , T. Guarda and P. Oliveira, "How ISO 27001 Can Help Achieve GDPR Compliance," 2019 14th Iberian Conference on Information Systems and Technologies (CISTI), Coimbra, Portugal, 2019, pp. 1-6, doi: 10.23919/CISTI.2019.8760937.

[Mussmann, 20] Mussmann, A., Brunner, M. and Breu, R., 2020, Mapping the State of Security Standards Mappings. https://library.gito.de/open-access-pdf/L4_Mussmann-Mapping_the_State_of_Security_Standards_Mappings-305_c.pdf

[Pinheiro, 14] Pinheiro, M.G. and Misaghi, M., 2014, December. Proposal of a framework of lean governance and management of enterprise IT. In Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services (pp. 554-558).

[Poth, 20a] Poth, A., Jacobsen, J. and Riel, A., 2020, June. A systematic approach to agile development in highly regulated environments. In International Conference on Agile Software Development (pp. 111-119). Springer, Cham.

[Poth, 20b] Poth, A., Kottke, M. and Riel, A., 2020, June. Evaluation of agile team work quality. In International Conference on Agile Software Development (pp. 101-110). Springer, Cham.

[Poth, 20c] Poth, A., Kottke, M. and Riel, A., 2020, September. Scaling agile on large enterprise level with self-service kits to support autonomous teams. In 2020 15th Conference on Computer Science and Information Systems (FedCSIS) (pp. 731-737). IEEE.

[Poth, 20d] Poth A., Jacobsen J., Riel A., 2020, September. Systematic Agile Development in Regulated Environments. In: Yilmaz M., Niemann J., Clarke P., Messnarz R. (eds) Systems, Software and Services Process Improvement. EuroSPI 2020. Communications in Computer and Information Science, vol 1251. Springer, Cham. https://doi.org/10.1007/978-3-030-56441-4_14

[Przybyłek, 17], A. Przybyłek and D. Kotecka, "Making agile retrospectives more awesome," 2017 Federated Conference on Computer Science and Information Systems (FedCSIS), 2017, pp. 1211-1216,

[Ribstein, 05] Ribstein, L.E., 2005. Accountability and responsibility in corporate governance. Notre Dame L. Rev., 81, p.1431.

[Scott, 05] Scott, L. and CARESS, A.L., 2005. Shared governance and shared leadership: meeting the challenges of implementation. Journal of nursing management, 13(1), pp.4-12.

[ScrumGuide, 20] The Scrum Guide, 2020, https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf#zoom=100, Last request: 09.06.2021

[Seal, 06] Seal, W., 2006. Management accounting and corporate governance: An institutional interpretation of the agency problem. Management Accounting Research, 17(4), pp.389-408.

[Triaa, 16] Triaa, W., Gzara, L., & Verjus, H., 08/2016, Organizational agility key factors for dynamic business process management. In 2016 IEEE 18th Conference on Business Informatics (CBI) (Vol. 1, pp. 64-73). IEEE.

[Tzolov, 18] Tzolov, T., 2018. One Model For Implementation GDPR Based On ISO Standards. International Conference on Information Technologies (InfoTech), Varna, 2018, pp. 1-3, doi: 10.1109/InfoTech.2018.8510716.