Success Factors in Secure Software Development of Cloud Applications in Germany: A Qualitative-explorative Expert Study

Marc Aurel Schubert
University of
Applied Sciences, Mainz
marc.schubert@hs-mainz.de

Sven Pagel
University of
Applied Sciences, Mainz
sven.pagel@hs-mainz.de

Harald F. O. von Korflesch
University of Koblenz
harald.vonkorflesch
@uni-koblenz.de

Abstract

The use of cloud computing continues to increase in Germany, according to Germany's digital association. However, threats to information security stand in the way of a comprehensive acceptance and penetration of cloud computing. Secure software development is described in the standard ISO/IEC 27001, security control A.14.2 "Security in development and support processes" and in the cloud related code of practice standard ISO/IEC 27017 in chapter 14.2 "Security in development and support processes". Secure software development has the potential to reduce vulnerabilities and thus increase the security level of applications. When implementing a secure software development process of cloud applications, the question for organizations is which factors have a positive influence on success, where success is defined as an increased security level of cloud applications. This paper contributes to answering the questions of (RQ1) what potential success factors exist in secure software development of cloud applications in Germany and (RQ2) what role does strategic and operational aspects play.

Keywords: Secure Software Development, Cloud Application, Success Factor Research.

1. Introduction

According to the Cloud Monitor 2021, a study by the Germany's digital association (*Bundesverband Informationswirtschaft, Telekommunikation und neue Medien – bitkom e.V.*) 82% of companies in Germany with 20 employees or more use cloud computing (Heidkamp, Vogel, and Gentemann 2021). In the studies of the same name from 2020 and 2019, it was only 76% of companies (Heidkamp, Vogel, and Pols 2020). Correspondingly, the use of cloud computing will continue to increase over the next several years. According to the Cloud Monitor 2021, this is countered by security concerns in the use of public cloud computing in particular, which are still 'persistent' in 2021. 75% of companies fear unauthorized access to data and 60% fear the loss of data. The use of cloud computing is thus impaired by threats to information security.

Cloud computing enables enterprises and public institutions (hereafter referred to as organizations) to obtain resources (networks, servers, storage, applications and services) on demand over high-speed networks with minimal management overhead (Chen and Zhao 2012; Mell and Grance 2011; Suryateja 2018). A cloud application supports business processes of organizations and is a composition of cloud computing services operated in the service models SaaS, PaaS or IaaS. According to the Cloud Application Maturity Model, cloud applications can be differentiated into four maturity levels (Kratzke 2018; Kratzke and Quint 2017): cloud ready, cloud friendly, cloud resilient and cloud native. The use of cloud computing is impaired by threats to information security in cloud computing (Suryateja 2018). "A threat is a class of potential events [...]" (Freiling et al. 2014:15). If there is a corresponding vulnerability in addition to a threat, it results in a hazard. (Freiling et al. 2014; Suryateja 2018). Damage can result from a hazard. In the ISO/IEC 27001:2013 standard, secure software development is described in security control A.14.2 "Security in development and support processes" (ISO 2013). Secure software development has the potential to reduce vulnerabilities and thus increase the security level of applications (Assal and Chiasson 2018). A process for secure software development integrates security practices (such as training, security requirement analyses, code reviews and the use of security tools) in all phases of a software development project (Dodson, Souppaya, and Scarfone 2020; Waidner 2013). A process for secure software development specifically for cloud applications compared to a process for secure software development of traditional applications (multitier architecture style) may differ because cloud applications have a different architecture. For example, distributed loosely coupled services (each publicly accessible) in a cloud

URI: https://hdl.handle.net/10125/103439
978-0-9981331-6-4

application must fulfil different security requirements than a traditional multitier architecture operated in the classic perimeter-protected data center with a single-entry point for public access. When implementing a secure software development process of cloud applications, the question for organizations is what factors have a positive influence on success, where success is defined as an increased security level of cloud applications. Knowing and considering success factors in secure software development of cloud applications is important for organizations in Germany, as it can increase effectiveness and efficiency when designing their own approach to secure software development and can result in cloud applications with an increased level of security.

2. Related Work

2.1 Success and success factor research

Research into success factors is based on the assumption that, despite the multidimensionality of success and the multicausality of potential success factors, there are a few success factors that have a decisive influence on success (Baumgarth, Eisend, and Evanschitzky 2009). These potential success factors are then evaluated in the explication according to success indicators. The evaluation according to subjective and objective success indicators is intended to determine whether there is a connection to success for the identified potential success factors and whether these can be considered success factors.

2.2 Standards and best practices for secure software development

Secure software development describes an approach consisting of phases such as design, creation and quality assurance of software, and integrates security practices in all phases (Dodson et al. 2020; Waidner 2013). For organizations there is a wide range of standards and best practices available: Assal and Chiasson (2018) have derived twelve best practices (AB) based on the Microsoft Security Development Lifecycle (MS-SDL), the Building Security in Maturity Model (BSIMM) and the Open Web Application Security Project (OWASP) Software Assurance Maturity Model (SAMM): (1) Identify security requirements, (2) Design for security, (3) Perform threat modelling, (4) Perform secure implementation, (5) Use approved tools and analyze third-party tools' security, (6) Include Security Testing, (7) Perform code analysis, (8) Perform code review for security (9) Perform post-development testing, (10) Apply defense in depth, (11) Recognize that defense is a shared responsibility and (12) Apply security to all applications. Dodson et al. (2020) have consolidated practices from 18 sources[1] into a high-level Secure Software Development Framework (SSDF). The Software Assurance Forum for Excellence in Code (SAFECode) is a global nonprofit organization that brings business leaders and technical experts together to establish software security programs. They outline the following Key Principles (SKP): (1) Organizational Commitment, (2) Clear Business Requirements, (3) a Security Development Lifecycle (SDL) and (4) Training. The SAFECode also stated, that the following aspects should be considered by an organization (SSA): (1) culture, (2) SDL, (3) consideration of existing culture in planning, (4) creating a new security culture, (5) the use of champions to challenge culture and (6) strengthening security awareness. Those standards and best practices will be mapped (by the abbreviations defined here) to the identified potential success factors afterwards (see Table 2).

2.3 German standards for secure software development

The Federal Office for Information Security in Germany (*Bundesamt für Sicherheit in der Informationstechnik – BSI)* is the publisher of the German standards the IT-Grundschutz[2] and the C5 (Cloud Computing Compliance Criteria Catalogue)[3]. The IT-Grundschutz is a methodology for setting up a security management system and for securing information networks via standard security measures. Requirements for software development were formulated in module CON.8. The C5 specifies minimum requirements for secure cloud computing and is primarily aimed at professional cloud providers, auditors and customers. The C5 chapter Procurement, Development and Modification of Information Systems (DEV) aims to ensure information security in the development cycle.

[1] Referenced standards/best practices: BSIMM, BSA, IDASOAR, ISO/IEC 27034, MS-SDL, NISTCSF, OWASPASVS, OWASPTEST, PCISSLRAP, SAMM15, SCAGILE, SCFPSSD, SCSIC, SCTPC, SCTTM, SP80053, SP800160 SP800181. For further explanation see Dodson, Souppaya, and Scarfone (2020).

[2] Referred as BSI-Standard 200-2

[3] Referred as BSI-Standard C5:2020

2.4 Cloud computing and cloud application

Mell and Grance (2011) define cloud computing as a model with five essential characteristics (Fehling et al. 2014; Mell and Grance 2011; Suryateja 2018): (1) On-demand self-service, (2) Broad network access, (3) Resource pooling, (4) Rapid elasticity, (5) Measured service. Within the model, a distinction is made between three service models: Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS). A SaaS provides an application, such as an Enterprise Resource Planning (ERP) system. A PaaS provides services like portal or application servers, which can be used by higher-level applications. An IaaS provides components such as virtual servers, storage or networks, on which applications can be based. A cloud application is a composition of cloud computing services (SaaS, PaaS or IaaS) and differs from a multitier architecture in that it is not divided into tiers, but rather into services that take into account the structural principles of a cloud application architecture. According to the Cloud Application Maturity Model a cloud application can be differentiated into four maturity levels (Kratzke 2018; Kratzke and Quint 2017): A cloud application in the cloud ready maturity level can operate a traditional application in the multitier architecture style (presentation tier, logic tier and data tier) on a virtual infrastructure in a cloud. The potentials of cloud computing are almost disregarded here. In the cloud friendly maturity level, loosely coupled services based on cloud design patterns are used. Cloud applications in the next maturity level, cloud resilient, are characterized by the fact that their state is only isolated in a few services and they are independent of failures of other services. The full maturity level is referred to as cloud native. Cloud applications on this maturity level are capable of exploiting the full potential of cloud computing. They are able to switch infrastructure providers at runtime without service interruptions and automatically scale the required resources according to demand. If cloud applications are implemented in the cloud friendly maturity level or higher the structural principles of a cloud application architecture must be considered.

2.5 Success factors in secure software development of cloud applications

No study could be identified that provides a precise definition of success in secure software development of cloud applications, nor one that defines the factors which positively influence the success of secure software development specifically of cloud applications. The studies are primarily concerned with finding out why companies do not introduce secure software development processes in general instead of optimizing their success. Chow and Cao (2008) have investigated factors influencing the success of agile software development projects in their study, based on agile software development. Dhir et al. (2019) have identified failure and success factors in agile software development by a case study. Geer (2010) refers in his publication to a survey, which presented the results on the adoption of secure software development at the RSA conference, stating that the majority of the companies surveyed find a secure software development process too time-consuming or are not aware of secure software development. Alghamdi (2020) has investigated what company characteristics have a positive influence on the adoption of a secure software development process and its subdivided security practices. What factors in the application of secure software development have an influence on the success of creating secure software has not been investigated.

3. Research needs and design

This paper aims to identify success factors in secure software development of cloud applications in Germany within the framework of a qualitative-explorative expert study. However, a definition of success is left to the perception of the experts. An expert of the strategic perspective stated a secure software development for cloud applications is successful when no vulnerabilities are exposed. An expert of the operative perspective stated that the customer's perception of security is crucial. Nevertheless, all experts agreed that secure software development for cloud applications is successful when the (increased) security level of cloud applications meet required security needs. The following two research questions were formulated, which the present work contributes to answering: *(RQ1) What are the potential success factors in secure software development of cloud applications in Germany? (RQ2) What role does strategic and operational aspects play in secure software development of cloud applications?*

3.1 Preparation

An expert is a construct of research interest (Meuser and Nagel 2009) and not a personal characteristic or ability (Bogner, Littig, and Menz 2014). The following organizational criteria were used to identify an expert: *(1) The organization is based in Germany, (2) The organization must be active in software development, (3) The developed software is*

designed for use in cloud computing, (4) The software development must follow principles of secure software development or relate to software for processing information with increased protection needs. Criterion (3) is crucial to take into account the specific characteristics of a cloud application (especially in the maturity levels cloud friendly and above) and thus enable a differentiation from secure software development of traditional applications. According to the ISO/IEC 27001:2013 standard (ISO 2013; Kissel 2019), the strategic perspective is taken by the Chief Information Security Officer (CISO). Organizational structures for information security change depending on size, industry and corporate cultures (Williams 2007). For the identification of an expert from a strategic perspective, the following additional personal criteria are required: *(1) The candidate is a CISO or responsible for secure software development, (2) The candidate has worked in the aforementioned role for at least 5 years, (3) The candidate has demonstrable knowledge in secure software development or in software development of applications with increased protection needs.* Software developers are responsible for implementing software for their own organization or for use by a customer. Through security trainings software developers are empowered to identify security vulnerabilities, eliminate them and produce more secure software (Assal and Chiasson 2018; Waidner 2013). The following personal criteria were used for identifying an expert from an operational perspective. *(1) The candidate is a software developer for software with increased protection needs. (2) The candidate has worked in the aforementioned role for at least 5 years. (3) The candidate has demonstrable knowledge of expertise in secure software development or in software development for applications with increased protection needs.* Additionally, a mixed-form perspective was constructed, that stands in between of strategic and operational perspective (e.g. auditors or software development leads). The expert indication here of the strategic perspective has been applied, knowing that there are some blurbs.

3.2 Data collection

An expert interview is considered a survey instrument that is used for exploratory purposes (Meuser and Nagel 2009). A semi-structured expert interview based on a guideline is a systematizing expert interview whose aim is to gather as much expert knowledge as possible (Bogner et al. 2014). It allows experts to talk in some depth, choosing their own . The results of the study can be analyzed further by theory-driven hypotheses using quantitative research methods. The expert is the source of specialized knowledge about the knowledge to be researched. Using a structured guideline, an attempt is made to close gaps in the researcher's level of information (Bogner et al. 2014). Based on the aforementioned criteria for identifying experts from a strategic perspective and an operational perspective, two guides were created. 15 expert interviews (see Table 1) were conducted. The expert interviews are intended to explore the coverage of all empirical variants and manifestations of a given phenomenon. The number of expert interviews was based on 'theoretical saturation'. According to Bogner et al. (2014), saturation is reached when no more additional data can be found, which may change the already existing findings. To reduce the influence of the researcher, the acquisition of subjects was primarily done through associations.

Table 1. Expert interviews conducted

No.	Role	Perspective	Industry
1	CISO	Strategic	Health
2	CISO	Strategic	Finance & Insur.
3	CISO	Strategic	Logistics
4	CEO	Strategic	Inform. & comm.
5	Head of Software Development	Mixed	Inform. & comm.
6	Cloud Sales Engineer and Security Architect	Mixed	Inform. & comm.
7	Security Managing Consultant	Mixed	Inform. & comm.
8	Auditor Information Security	Mixed	Inform. & comm.
9	Head of Solution Architect	Mixed	Professional services
10	Cloud Administrator	Operational	Inform. & comm.
11	Senior Cloud Eng.	Operational	Inform. & comm.
12	Senior Software Eng.	Operational	Transport
13	Microsoft 365 Arch.	Operational	Inform. & comm.
14	Senior Software Eng.	Operational	Logistics
15	Senior Software Eng.	Operational	Inform. & comm.

3.3 Data evaluation

The data were analyzed with the qualitative content analysis according to Mayring (2014). Content analysis dates back to the first half of the 20th century. It was developed in order to be able to quantitatively analyse aspects of content (e.g. topics and headings) in mass media (e.g. newspaper or radio) (Mayring 2014). By means of qualitative content analysis, the text is analysed qualitatively and interpretatively. It elicits and statistically analyses category frequencies and consists of two steps. In the first step, categories are formed inductively or in a theory-guided, deductive manner on the basis of the available material. Individual text passages are assigned to the categories. In a second step, it is checked whether several text passages can be assigned to the categories. (Mayring 2014).

4. Results

All the experts surveyed develop software using the Scrum or DevOps approach and are thus subject to the agile manifesto. The studies on factors in agile software development projects divide their identified factors into the dimensions of organization, process, technology, human and project. For the classification of the 38 identified potential success factors, the dimensions according to Chow and Cao (2008) have been used as clusters, as they allow a more multifaceted classification (see Table 2).

Cluster: Organization (aggregates leadership behavior and measures and cultural factors). Before a software development project begins, (advanced) security training for software developers *(ORG-1)* must already have been completed, according to the experts. Experts from all perspectives agree that this is a success factor. Security training can be provided in various depths to suit the target group. According to OWASP (2021), security knowledge is still not sufficiently available.

In addition to security training, for experts from the operational perspective alignment with standards/best practices *(ORG-2)* plays a decisive role in terms of a success factor for secure software development of cloud applications. Among other things, it was pointed out that especially in the implementation of security-critical software functions (e.g. for encryption and signature verification), priority should not be given to proprietary solutions, but rather to standard implementations that have proven themselves in practice.

From a strategic perspective in Germany, dependencies on the respective states and governments must be considered when using cloud providers. An analysis and measures for protection against foreign state activities/sovereign acts *(ORG-3)* were therefore highlighted as a success factor. Many companies from the banking and insurance sectors invest in the public cloud solutions of US hyperscaler's Microsoft, Amazon or Google. After the European Court of Justice ruling on the EU-US Privacy Shield (Schrems-II), legal uncertainty exists when companies choose a US cloud provider (Heidkamp et al. 2021). An alternative is offered by the GAIA-X project, which is a European alliance of the political, business and science sectors, is working on to create a secure infrastructure for cloud and data. According to the Cloud Monitor 2021, 38% of companies have looked into GAIA-X. Of these, 65% of companies see GAIA-X as a must-have (Heidkamp et al. 2021). *"State regulations can throw a spanner in the works of the cloud approach. [...] Data locality is one of the important issues. Which cloud provider do I trust with my data?" (Interview NY1C, 46:35)*

Establishing a security culture (ORG-4) is crucial from strategic and operational perspective and is therefore a success factor for secure software development of cloud applications. There must be a common understanding of security in the organization and an environment must be created in which security concerns can be freely expressed.

From the mixed-form perspective, expectation management on the topic of technical debts *(ORG-5)* with the customer and were singled out as success factors. In customer contact, it is the task of the project management team to explain the topic of secure software development to the customer and to explain the interaction between the quality, costs and time of a software product. The project could be completed faster with the required functions, but at certain technical debts. *"Technical debts are incurred when speed is required and then the building blocks of the secure construction kit are not used. These are then debts that have to be explained to the customer and then in due course they have to be reduced." (Interview BN7W, 38:10).*

According to the experts, software developers are mainly measured by two criteria 'in-time' and 'in-budget'. Software developers have little time left for deeper examination of security or secure software development. From a strategic and an operational perspective, it was emphasized as a success factor that free space/time *(ORG-6)* must be made available.

Experts from strategic, mixed-form and operational perspectives agree that knowledge exchange/lessons learned *(ORG-7)* contribute significantly to the development of secure software and thus represent a success factor. Experiences must be made transparent and shared – at least within one's own organization. Special events, such as the Red/Blue team approach, where different teams try to penetrate systems in an explorative and playful way, are of crucial importance.

From a strategic perspective, management attention *(ORG-8)* to secure software is a success factor for secure software development. Only if the management is aware of the topic can guidelines be made or acted upon from a strategic perspective. From an operational perspective, management attention was not mentioned.

Management commitment *(ORG-9)* from a strategic perspective is crucial for secure software, and thus a success factor.

According to experts of the strategic and mixed-form perspective, it is not enough to use cloud computing as a standard solution in the company. Cloud must be securely consumable. Cloud computing

must be conceptualized in individual building blocks that are already securely configured in the standard. The provision of a platform strategy for cloud computing *(ORG-10)* in the sense of an ecosystem is a success factor.*"A pure cloud strategy is too short-sighted. A platform strategy is necessary to make things accessible." (Interview BN7W, 38:34)*

From a strategic, mixed-form and an operational perspective, experts see it as a duty to deal with the topic of secure software development and to build up knowledge in this area: security as a duty/license to operate *(ORG-11).* Software developers who reject the topic and do not cooperate are not tenable from the middle management's point of view. *"The first message in security is 'deal with it'. Security is a topic that every developer has to deal with up to a certain level." (Interview QL9F, 56:45).*

In combination with the success factors of security training and secure construction kit, the introduction of security champions/experts *(ORG-12)* is a success factor from the mixed-form and operational perspective. A security champion/expert has specialized knowledge on the topic of secure software development and serves as a contact person for the other software developers for further questions on the topic or on the secure construction kit.*"I strongly support the use of security champions. Here, operational units are empowered to recognize attacks and to think about them when software is being developed." (Interview KL3D)*

According to an auditor from the mixed-form perspective, the transparency towards the employees *(ORG-13)* by the organization's management is a success factor. This functions on the one hand as a confidence-building measure and on the other hand as a basis for assessing the importance of secure software development.

Cluster: Process (aggregates security practices aligned to a secure software development process). Automated scanning was considered particularly important from a strategic perspective. Automated code, platform and container scanning *(PRC-1)* was highlighted as a success factor. The specifics of cloud computing technology allow for increased use of automation. Container scanning makes it possible to identify executed malicious code in a dedicated cell and react to it automatically.*"We also need to work in testing: with source code scanning, with software composition analysis, with container scanning." (Interview G3YN, 25:30)*

A stringent development procedure with rules *(PRC-2)* was named as a success factor by experts from strategic and operational perspective. The experts explained that different rules apply to different components. For example, stricter security standards are applied to technical basic components than to components that are in turn based on the basic components. This ensures that few software developers need in-depth security knowledge and that other software developers benefit from these secure components.

Experts from all perspectives agree that, in addition to specifications, tools and assistance, there is still a need for control. An internal inspection process/quality gate/security audits/code reviews *(PRC-3)* are an elementary component of secure software development in general and for cloud applications as well. "*However, in addition to self-reliance, time and resources, processes and technologies, and knowledge transfer among software developers, there must be an additional controlling security function that defines quality gates where security colleagues have veto power over the go-live of applications." (Interview G3YN, 43:00)*

According to experts from all perspectives, one way of testing the security level of the cloud application is measurement by means of penetration testing *(PRC-4)*. In a penetration test, malicious attack behavior of a perpetrator will be simulated, trying to find out what vulnerabilities can be exploited and what damage can be caused. *"Having a good test framework is important – ideally with 100% code coverage. Results of a penetration test are a very good indicator." (Interview DR4E)*

The secure architecture and design of a cloud application *(PRC-5)* is of crucial importance and a success factor according to the mixed-form perspective and the operational perspective. Lipner (2004) describes this requirement as 'security by design'. If security aspects are already considered in the design, they do not have to be dealt with separately in further development. *"We do security by design. We don't run after security." (Interview BN7W, 38:15)*

The result of a platform strategy for cloud computing is the provision of secure construction kit/development platform *(PRC-6)* that contain consumable elements released for the organization and, according to the experts, is a success factor from all perspectives. It should function in such a way that security is already taken into account here. However, a security champion should still be available.

Every employee contributes to the organization's information security. Awareness of the threats, methods and countermeasures in the area of cybersecurity is therefore crucial. The task of raising awareness among employees is summarized under the term security awareness *(PRC-7)*. In addition to the aforementioned security training for software developers, security awareness is also a success factor

in general and also for cloud applications from a strategic and operational perspective.

In order to integrate security practices into a development process the introduction of security guidelines for the development *(PRC-8)* is a success factor from an expert of the strategic perspective.

Threat modelling *(PRC-9)* is a success factor from a strategic and operational perspective, according to the experts. Threat modelling is a process by which potential threats, such as structural vulnerabilities or the lack of appropriate safeguards, can be identified and enumerated, and mitigation measures prioritized. Cloud applications are due to their architecture and nature more open accessible, and almost every service serve a potential attack surface. Therefore, threat modelling for cloud applications is more important in comparison to traditional multi-tier architectures, an expert stated. *"In the design phase, we already receive vulnerabilities from threat modelling. […] Threat modelling is an absolute basic building block of secure software development. [...] It saves money and solves the problems before they are implemented." (Interview G3YN, 18:30)*

Unsurprisingly, vulnerability identification and remediation *(PRC-10)* contributes decisively to the increased security level of cloud applications. This was named by the experts of all perspectives as a success factor. The process of vulnerability identification and remediation is far less trivial in its concrete design. An expert stated that cloud applications are based on up to 80% open-source software, which can include vulnerabilities. As the log4j vulnerability (CVE-2021-44228) in December 2021 made clear, vulnerability identification and remediation are crucial to application security. The use of vulnerability scanning tools is therefore indispensable. *"If we are very fast, then we have to control the speed. We therefore need sensors along the entire supply chain to report weak points, which then have to be dealt with in the weak point management." (Interview G3YN, 25:20).*

Cluster: Technology (aggregates techniques and tools towards secure software development). From the experts' point of view, the use of an automated CI/CD pipeline *(TEC-1)* represents a success factor from all perspectives. Continuous Integration (CI) and Continuous Delivery (CD) are processes from software engineering to increase the efficiency of projects (Rangnau et al. 2020). Routine tasks can be carried out reliably through automated integration and deployment. In addition, no intermediate paths can be taken. The manual exchange of individual files that are being corrected is not possible. This also ensures traceability as to which person exchanged or provided which artefacts. It is also possible to integrate tests into the CI/CD pipeline. One of these tests is the Dynamic Application Test (DAST), which can be divided (Rangnau et al. 2020) into Web Application Security Testing (WAST), Security API Scanning (SAS) and Behavior-Driven Security Testing (BDST). Implementing CI/CD for cloud application development is an elementary aspect. *"CI/CD is not only used in the software development process, but also in everything downstream such as testing. It forces us to look at everything with automation glasses." (Interview BN7W, 39:30).*

When using cloud computing, the choice of the service model to be used is crucial. According to an expert, the use of IaaS is not purposeful, as it does not take into account the level of automation and standardization. The use of cloud native technology *(TEC-2)*, Cloud Native Applications (CNA), was therefore identified as a potential success factor from the experts of strategic and operational perspective. *"In cloud computing, you should not just do lift-and-shift. You have to be aware of the potentials of the cloud. There should be cloud native development and nothing should be done as it used to be done in the traditional data center." (Interview KL3D)*

With the use of modern technology *(TEC-3)* the security level of cloud applications can be increased and thus represents a success factor from a strategic and operational perspective, according to the experts. One expert explained that modern technologies like Google Angular - using the programming language TypeScript - do not allow certain attack vectors to be granted by default. Thus, a certain level of security is achieved without further activity. In the literature, this is also referred to as "taking the human out of the loop" in general and "taking the developer out of the loop" in particular.*"Processes and technology are a success factor. A carefully defined process must be chosen in which colleagues can work and also feel comfortable." (Interview G3YN, 43:06)*

A cloud application is typically based on open-source software and therefore uses program libraries from the community. The libraries used may contain vulnerabilities that are inevitably incorporated into the cloud application to be created. According to the strategic and mixed-form perspective, tracking and use of secure libraries *(TEC-4)* is a success factor.

According to Mell and Grance (2011), a distinction can be made in the three service models of IaaS, PaaS and SaaS. The use of a higher abstraction layer has advantages, as they can already be consumed by the organization as a 'managed service'. In addition to the previously mentioned service models, another higher abstraction layer, has developed: Function-as-a-Service (FaaS). FaaS is referred to as serverless computing, and it enables software developers to run event-driven functions in the cloud without having to

manage resources or configure the runtime environment (Jangda et al. 2019). In this case, the cloud provider takes over the secure provisioning of the services. According to the mixed-form and operational perspective, the use of server-less functions *(TEC-5)* can be a success factor.

According to the experts, the use of tools *(TEC-6)* can be a success factor from all perspectives, provided that the introduction to and use of tools accompany such use. A tool supports software and system life cycle processes (ISO/IEC/IEEE 2017). Tools for discovering and correcting vulnerabilities in program code are referred as security tools. OWASP strongly advocates the use of security tools within the development process. As already mentioned in *(TEC-1)* DAST and SAST tools can be incorporated into the build pipeline to detect easily identified security issues. *"Tools help. Especially if they are automated. But you also have to have the right tools. There are too many tools." (Interview ZQ1A, 31:13)*

Cluster: Human (aggregates psychological aspects and attitudes from individuals). According to the strategic and mixed-form perspective, employee self-responsibility *(HUM-1)* plays a crucial role. According to one CISO, self-responsibility is the decisive success factor. *"For me, self-responsibility is the decisive success factor. It means that the developer is responsible for programming and understands that he is building a product where the well-being of the company and other people also depend on it." (Interview G3YN, 42:00)*

According to mixed-form and operational perspective, identification with the project *(HUM-2)* is a success factor for the security level of cloud applications. The more the software developers can identify with the software product, the more likely they are to contribute to its quality. *"If a software developer works for something he cannot identify with, then it is much less likely that he will work 50 or 60 hours a week or that he will deliver his best work. Then you're just doing a job and that certainly has disadvantages for security." (Interview QL9F)*

According to the experts, motivation must be distinguished between extrinsic and intrinsic motivation. Motivation (extrinsic) *(HUM-3)* reveals a contradictory picture. Experts from a strategic perspective consider extrinsic motivation to be less suitable up to the assessment that extrinsic motivation even has an opposite effect and reduces the security level of cloud applications in the long term. From a mixed-form and operational perspective, extrinsic motivation is perceived as a suitable means.

Besides to that, there is agreement among the experts from all perspectives, that motivation (intrinsic) *(HUM-4)* is a success factor. The experts from the strategic and operational perspective confirmed that Organizational Commitment (HUM-5) can be crucial to the success towards secure software development.

Cluster: Project (aggregates factors from the project management triangle: time, quality, budget). According to an expert of the mixed-form perspective the collection of error statistics in projects *(PRJ-1)* is necessary in order to draw a baseline for security measurement**.** If software is developed with a defined scope of the final project *(PRJ-2)*, this can have a positive effect on the security level of the cloud application to be developed. This is the opinion of an expert from an operational perspective. Sufficient resources *(PRJ-3)* were also mentioned as a success factor from the strategic and operational perspectives. To be able to deal with secure software development in depth, sufficient time *(PRJ-4)* is indispensable and considered as success factor.

5. Conclusion

This study revealed 38 potential success factors classified into five clusters (see Table 2). The work thus contributes to answering the research question of *(RQ1) what potential success factors exist in secure software development of cloud applications in Germany*. The work also contributes to answering the research question of *(RQ2) what role does strategic and operational aspects play in secure software development of cloud applications:* From a strategic perspective exclusively, the success factors management attention *(ORG-8)*, management commitment *(ORG-9)* and analysis and measures for protection against foreign activities/sovereign acts *(ORG-3)* have been mentioned. From an operational perspective exclusively, alignment with standards/ best practices *(ORG-2)* and attention to the scope of the final project *(PRJ-2)* are success factors. Further distinctions from a strategic, mixed-form and operational perspective are elaborated for each success factor mentioned. However, the study has the following limitations: The study is based on qualitative-explorative expert interviews from different economic sectors and organizations of different sizes. The results can offer approaches for further research, but do not enable reliable statements about all organizations in Germany. The results of the present study generates new and relevant insights for theory

and practice. From the theoretical point of view, on the basis of the identified potential success factors a further selection and validation can be made. From a practical point of view, knowing and considering success factors in secure software development of cloud applications may help organizations in Germany to increase effectiveness and efficiency of their secure software development of cloud applications. It is aimed to validate the results of this study, named as study (I), through two additional studies: (II) a quantitative-validation study using hypotheses drawn from the theoretical foundations of Evans (1970) and House (1971) path goal theory of leadership and (III) a field experiment study using objective indicators and measures (e.g. identified vulnerabilities) relying on the developed source code. The studies (II) and (III) also aim to shed some light on the applicability of the success factors beyond Germany and to non-cloud secure software development processes and practices.

Table 2. Potential success factors in secure software development of cloud applications

C.	Potential success factor *In alphabetical order per dimension*	Perspective			Interviews	References
		S	*M*	*O*	*Pseudonyms*	*Not exhaustive*
Organization (ORG)	(1) (Advanced) security training for software developers	**X**	**X**	**X**	BN7W, DR4E, L9HK, NY1C, V2OP, QL9F	SKP4, MS-SDL, SAMM, BSIMM
	(2) Alignment with standards/best practices			**X**	HZ9L, O2CP	MS-SDL, SAMM, BSIMM, SSDF
	(3) Analysis and measures for protection against foreign activities/sovereign acts	**X**			NY1C	
	(4) Establishing a security culture	**X**		**X**	NY1C, L9HK	SSA1, SSA3, SSA4, SAMM
	(5) Expectation management on the topic of technical debt		**X**		BN7W, B9WX	
	(6) Free space/time	**X**		**X**	DR4E, G3YN, L9HK, V2OP, 5RNY	
	(7) Knowledge exchange/lessons learned	**X**	**X**	**X**	G3YN, L9HK, PR9Y, 5RNY	SAMM, SSDF
	(8) Management attention	**X**			V2OP,5RNY	BSIMM
	(9) Management commitment	**X**			V2OP, 5RNY	
	(10) Platform strategy for cloud computing	**X**	**X**		BN7W, V2OP	
	(11) Security as a duty/license to operate	**X**	**X**	**X**	B9WX, NY1C, QL9F	
	(12) Security champions/experts		**X**	**X**	KL3D, L9HK	SSA5, SAMM, BSIMM
	(13) Transparency towards the employees		**X**		PR9Y	
Process (PRC)	(1) Automated code, platform and container scanning	**X**			G3YN, NY1C	SAMM, BSIMM, SSDF
	(2) Development procedure with rules	**X**		**X**	G3YN, O2CP	SKP3, SSA2, SAMM, SSDF
	(3) Internal inspection process/quality gate/security audits / code reviews	**X**	**X**	**X**	B9WX, DR4E, G3YN, L9HK, ZQ1A	AB7, AB8, MS-SDL, SAMM, BSIMM, SSDF
	(4) Measurement by means of penetration testing	**X**	**X**	**X**	NY1C, PR9Y, DR4E	AB9, MS-SDL, BSIMM
	(5) Secure architecture and design of a cloud application		**X**	**X**	BN7W, O2CP, ZQ1A	AB2, SAMM, BSIMM, SSDF
	(6) Secure construction kit/development platform	**X**	**X**	**X**	BN7W, HZ9L, L9HK, NY1C, V2OP	MS-SDL, SAMM, BSIMM
	(7) Security awareness	**X**		**X**	H79L, L9HK, 5RNY	SSA6, SAMM, BSIMM
	(8) Security guidelines for the development	**X**			V2OP	SAMM, BSIMM
	(9) Threat modelling (with focus on cloud app. architecture)	**X**		**X**	G3YN, KL3D	AB3, MS-SDL, SAMM, BSIMM
	(10) Vulnerability identification and remediation	**X**	**X**	**X**	B9WX, G3YN, ZQ1A	SAMM, BSIMM, SSDF
Technology (TEC)	(1) Use of an automated CI/CD pipeline	**X**	**X**	**X**	BN7W, QL9F, V2OP	MS-SDL, SAMM, BSIMM, SSDF
	(2) Use of cloud native technology	**X**		**X**	DR4E, V2OP	
	(3) Use of modern technology	**X**		**X**	NY1C, G3YN, L9HK	SAMM
	(4) Use of secure libraries	**X**	**X**		NY1C, ZQ1A	MS-SDL, SAMM, BSIMM, SSDF
	(5) Use of serverless functions		**X**	**X**	DR4E, KL3D	
	(6) Use of tools	**X**	**X**	**X**	G3YN, L9HK, QL9F, ZQ1A	AB5, MS-SDL, SA-MM, BSIMM, SDF
Human (HUM)	(1) Self-responsibility	**X**	**X**		G3YN, KL3D, NY1C	
	(2) Identification with the project		**X**	**X**	PR9Y, QL9F	
	(3) Motivation (extrinsic)		**X**	**X**	B9WX, T3PI, 5RNY	
	(4) Motivation (intrinsic)	**X**	**X**	**X**	BN7W, B9WX, H79L, L9HK, PR9Y, 5RNY	
	(5) Organizational commitment	**X**		**X**	G3YN, KL3D	SKP1, SAMM
Project (PRJ)	(1) Collection of error statistics in projects		**X**		KL3D	MS-SDL, SAMM, SAMM, BSIMM,
	(2) Scope of the final project			**X**	DR4E	SKP2, MS-SDL, SAMM
	(3) Sufficient resources	**X**		**X**	L9HK, V2OP	
	(4) Sufficient time	**X**		**X**	DR4E, L9HK, V2OP	

Acknowledgement: This work is part of the project InnoProm Security, funded by the European Regional Development Fund (ERDF) and the Ministry of Science and Health of Rhineland-Palatinate, Germany and sapite GmbH, a medium-sized IT security company in Klein-Winternheim, Germany.

References

Alghamdi, Fatimah. 2020. 'Motivational Company's Characteristics to Secure Software'. Pp. 1–5 in *2020 3rd ICCAIS*. Riyadh, Saudi Arabia: IEEE.

Assal, Hala, and Sonia Chiasson. 2018. 'Security in the Software Development Lifecycle'. Pp. 281–96 in 14. Symposium on Usable Privacy and Security.

Baumgarth, Carsten, Martin Eisend, and Heiner Evanschitzky. 2009. 'Empirische Mastertechniken'. Pp. 3–26. Wiesbaden: Gabler Verlag.

Bogner, Alexander, Beate Littig, and Wolfgang Menz. 2014. *Interviews Mit Experten: Eine Praxisorientierte Einführung*. Springer-Verlag.

Chen, D., and H. Zhao. 2012. 'Data Security and Privacy Protection Issues in Cloud Computing'. Pp. 647–51 in *2012 IC-CEE*. Vol. 1.

Chow, Tsun, and Dac-Buu Cao. 2008. 'A Survey Study of Critical Success Factors in Agile Software Projects'. *Journal of Systems and Software* 81(6):961–71. doi: 10.1016/j.jss.2007.08.020.

Dhir, Saru, Deepak Kumar, and V. B. Singh. 2019. 'Success and Failure Factors That Impact on Project Implementation Using Agile Software Development Methodology'. Pp. 647–54 in Software Engineering. Vol. 731, Springer Singapore.

Dodson, Donna, Murugiah Souppaya, and Karen Scarfone. 2020. *Mitigating the Risk of Software Vulnerabilities by Adopting a Secure Software Development Framework (SSDF)*. NIST. doi: 10.6028/NIST.CSWP.04232020.

Evans, M. G. (1970). The effects of supervisory behavior on the path-goal relationship. https://doi.org/10.1016/ 0030-5073(70)90021-8

Fehling, Christoph, Frank Leymann, Ralph Retter, Walter Schupeck, and Peter Arbitter. 2014. *Cloud Computing Patterns*. Vienna: Springer Vienna.

Freiling, Felix, Rüdiger Grimm, Karl-Erwin Großpietsch, Hubert B. Keller, Jürgen Mottok, Isabel Münch, Kai Rannenberg, and Francesca Saglietti. 2014. 'Technische Sicherheit und Informationssicherheit: Unterschiede und Gemeinsamkeiten'. *Informatik-Spektrum* 37:14–24. doi: 10.1007/s00287-013-0748-2.

Geer, David. 2010. 'Are Companies Actually Using Secure Development Life Cycles?' *Computer* 43(6):12–16. doi: 10.1109/MC.2010.159.

Heidkamp, Peter, Marko Vogel, and Lukas Gentemann. 2021. 'Bitkom Cloud Monitor 2021'.

Heidkamp, Peter, Marko Vogel, and Axel Pols. 2020. 'Bitkom Cloud Monitor 2020'.

House, R. J. (1971). A path goal theory of leader effectiveness. Admin. Science Quarterly, 321–339.

ISO. 2013. 'ISO/IEC 27001 Information Security Management'.

ISO/IEC/IEEE. 2017. 'ISO/IEC/IEEE Systems and Software Engineering Vocabulary'. ISO/IEC/IEEE 24765:2017(E), doi: 10.1109/IEEESTD.2017.8016712.

Jangda, Abhinav, Donald Pinckney, Yuriy Brun, and Arjun Guha. 2019. 'Formal Foundations of Serverless Computing'. Proceedings of the ACM on Programming Languages 3(OOPSLA):1–26. doi: 10.1145/3360575.

Kissel, Richard. 2019. 'NIST Special Publication 800-64 Revision 2, Security Considerations in the System Development Life Cycle'. 68.

Kratzke, Nane. 2018. 'A Brief History of Cloud Application Architectures'. Applied Sciences 8(8):1368. doi: 10.3390/app8081368.

Kratzke, Nane, and Peter-Christian Quint. 2017. 'Understanding Cloud-Native Applications after 10 Years of Cloud Computing - A Systematic Mapping Study'. Journal of Systems and Software 126:1–16. doi: 10.1016/j.jss.2017.01.001.

Lipner, Steve. 2004. 'The Trustworthy Computing Security Development Lifecycle'. Pp. 2–13 in 20th Annual Computer Security Applications Conference. IEEE.

Mayring, Philipp. 2014. 'Qualitative Content Analysis: Theoretical Foundation, Basic Procedures […]'.

Mell, Peter, and Timothy Grance. 2011. 'NIST Definition of Cloud Computing'.

Meuser, Michael, and Ulrike Nagel. 2009. 'Das Experteninterview — Konzeptionelle Grundlagen Und Methodische Anlage'. Pp. 465–79 in Methoden der vergleichenden Politik- und Sozialwissenschaft. Wiesbaden: VS Verlag für Sozialwissenschaften.

OWASP. 2021. 'OWASP Security Knowledge Framework

Rangnau, Thorsten, Remco v. Buijtenen, Frank Fransen, and Fatih Turkmen. 2020. 'Continuous Security Testing: A Case Study on Integrating Dynamic Security Testing Tools in CI/CD Pipelines'. Pp. 145–54 in 2020 IEEE 24th EDOC. Eindhoven, Netherlands: IEEE.

Suryateja, P. S. 2018. 'Threats and Vulnerabilities of Cloud Computing A Review'. International Journal of Computer Sciences and Engineering 6(3):297–302. doi: 10.26438/ijcse/v6i3.297302.

Waidner, Michael. 2013. 'Entwicklung sicherer Software durch Security by Design'. 76.

Williams, Paul. 2007. 'Executive and Board Roles in Information Security'. Network Security 2007(8):11–14. doi: 10.1016/S1353-4858(07)70073-9.