

Implementation of Control Activities in Scaled Agile Environments at Financial Service Providers to Improve Risk Management

M.B. Becks (s1927620)

Supervisors: prof. dr. J. van Hillegersberg and dr. ir. M.J. van Sinderen
Léon Korf de Gidts, Enterprise Architect at Achmea

Final Thesis

Master Business Information Technology
University of Twente

ABSTRACT

Globalization and digitization have driven large financial service providers to adopt scaled agile methodologies like the Scaled Agile Framework (SAFe), Large Scale Scrum (LeSS), and Disciplined Agile 2.0 (DA 2.0) to enhance speed and flexibility. However, this shift introduces new challenges, particularly for large financial service providers traditionally reliant on rigid control frameworks such as the Control Objectives for Information and Related Technology (COBIT) and the Committee of Sponsoring Organizations of the Treadway Commission (COSO). Agile practices often lack explicit risk management mechanisms, leading to potential gaps in managing risks within solution architecture documentation—a key component of Enterprise Architecture Management (EAM). This research examines the integration of control activities into the agile solution architecture documentation process to improve risk management. Through a qualitative field study at a large financial service provider, key components, risks, and effective control activities were identified. The findings offer practical insights to improve risk management in scaled agile environments, bridging the gap between traditional control frameworks and agile methodologies, and ensuring agility in solution architecture documentation while maintaining robust risk management.

Keywords

Control Framework (CFW), Control Activities, Scaled Agile Framework (SAFe), Solution Architecture Documentation (SAD), Solution Intent (SI), Enterprise Architecture Management (EAM), Quality Assurance (QA)

1. INTRODUCTION

Globalization and digitization have significantly influenced financial service providers[1][2]. Customer expectations have increased, technology has progressed, and new regulatory obligations have been introduced[3][4]. The ability of organizations to sense environmental change and respond readily, is an important determinant of organization success[5]. The challenge for financial service providers lies in adapting to these changes and satisfying the evolving demands related to user experience, performance, privacy, security, and interoperability[6]. Achieving this necessitates smooth interaction both within and among organizations [1].

Therefore, an increasing amount of financial service providers is adopting (scaled) agile methodologies to increase speed, responsiveness to change and flexibility[7][8][9][10][11][12][13].

There is a noticeable trend towards scaled agile frameworks throughout organizations. The most popular scaling agile frameworks are Scaled Agile Framework (SAFe), Large Scale Scrum (LeSS), and Disciplined Agile 2.0 (DA 2.0) [14]. Scaling agile methods introduce new and significant challenges, such as inter-team coordination and distribution of work without a defined architecture or properly defined requirements[15][14]. Agile methods have been criticized for lack of attention to architecture[1]. The greater the agility of the projects, the higher the risk to the enterprise architecture function[16]. The importance of architecture frequently goes unrecognized, leading to agile teams facing difficulties due to the absence of appropriate architectures [1]. Agile methodologies do not explicitly model and manage risk[17][18][19].

This shift, while beneficial in many aspects presents challenges, especially in large financial service providers traditionally reliant on extensive documentation and rigid control frameworks, such as COBIT and COSO[13]. Agile methodologies do not explicitly model and manage risk[17][18][19]. Risk management is crucial for ensuring that organizations can respond adequately to environmental changes and uncertainties. Traditional control frameworks like COBIT and COSO emphasize structured processes and comprehensive documentation to mitigate risks[20][21]. However, agile methodologies focus on flexibility and rapid response, which can sometimes lead to insufficient attention to risk management[18][19]. This gap necessitates the integration of effective risk management practices within agile frameworks to ensure that organizations can maintain agility while managing potential risks effectively[17][18].

This research focuses on the applicability of control activities in an agile environment that can be used on solution architecture documentation process, which is one of the sub-processes of Enterprise Architecture Management (EAM)[22]. EAM deals with capturing, modeling, analyzing, and defining the current, planned, and future architecture in conjunction with the roadmap leading from the "as-is", emergent design[23] or baseline architecture to the "to-be", intentional architecture[23] or target architecture[24]. Architecture is essential for managing the complexity of large organizations or systems[25].

This research explores the intersection of risk management, (scaled) agile, and enterprise architecture—areas where current literature is limited. A systematic literature review showed that not much research has looked at where the areas of 'control', 'agile', and 'enterprise architecture' overlap. This research aims to fill that gap and is interesting in the field of Accounting Information Systems (AIS)

domain. The research examines how abstract control activities from literature can be implemented within an agile solution architecture documentation process. Conducted and validated at a large financial service provider, an organization with 16,500 employees serving 10 million customers, the findings offer practical insights that are especially valuable for similar organizations. By bridging the gap between theory and practice, this research provides actionable insights that enhance agile practices in the context of risk management and enterprise architecture, contributing to both academic and industry knowledge.

The goal of this research is to identify control activities known for use in agile environments, the key components of agile solution architecture documentation and to examine the relevant risks and control activities associated with this process. Specifically, the study explores how risk management can be improved by applying control activities within an agile solution architecture documentation process at a large financial service provider. The ultimate objective is to provide practical insights into how these control activities can be effectively used within (scaled) agile environments to improve risk management for financial service providers.

The design problem addressed in this research is formulated in Figure 1:

Improve risk management
by designing control activities for the agile solution architecture documentation process
that satisfies mitigation of relevant risks for financial service providers
in order to enable Enterprise Architects to effectively demonstrate control over the solution architecture documentation process

Figure 1. Research problem formulated according to Wieringa’s template [26]

In particular the following questions will be investigated:
Sub research questions:

- What control activities are known for use in an (scaled) agile environment to improve risk management?
- What are key components of solution architecture documentation within an agile environment?
- What are the relevant risks associated with agile solution architecture documentation for financial service providers?
- Which control activities are most applicable to managing these risks in solution architecture documentation at financial service providers?
- How can these control activities be effectively implemented, and visualized in financial service providers?
- To what extent can the automation of evidence delivery for key controls be achieved?

In Section 2, key concepts of Enterprise Risk Management (ERM), (Scaled) agile and Enterprise Architecture Management are described.

In Section 3, it is explained how the design science research is performed to answer the main research question.

In Section 4, a summary of a systematic literature review is presented.

In Section 5, a qualitative field study is performed at a large financial service provider, giving insights in solution architecture key components, risks, and control activities relevant to the financial service provider.

In Section 6, designed control activities on the solution architecture documentation process are applied in a controlled environment.

In Section 7, the relevancy of the control activities, the implementation of control activities, and the visualization in a dashboard is validated by a demonstration and survey.

In Section 8, implications of the research are discussed and limitations of the study are described in Section 8.3, and future work is proposed in Section 8.4.

In Section 9 the study is concluded.

2. BACKGROUND

This section provides background information on Enterprise Risk Management (ERM) and (scaled) agile frameworks.

2.1 Large Financial Service Providers

A financial service provider is an entity that offers a wide array of services related to finance and banking, encompassing activities such as cash accounts, savings, money management, investment management, money transfer and payments (e.g., credit cards), portfolio management, financial advice, money loaning and lending (e.g., mortgages, consumer loans, credits), foreign currency exchange, equity trading, brokerage, insurances, and pension planning[27]. These services are often regulated by national legislations and can include traditional institutions like banks, asset management companies, insurance companies, and brokerage firms[27]. Additionally, financial service providers may also encompass established institutions that have digitalized their services and offer online offerings, as well as newer entities that focus on financial technology (FinTech) solutions[27]. The categorization procedure of the EU recommendation 2003/361 is used to define a large organization[28]. Any organization with more than 249 employees, more than 50 million turnover or more than 43 million balance sheet turn over is categorized as a large organization.

2.2 Enterprise Risk Management standards

Enterprise risk management deals with risks and opportunities affecting value creation or preservation. Enterprise risk management can be defined as follows: *a process, effected by an entity’s board of directors, management and other personnel, applied in strategy setting and across the enterprise, designed to identify potential events that may affect the entity, and manage risk to be within its risk appetite, to provide reasonable assurance regarding the achievement of entity objectives.*[21]

2.2.1 Three Lines of Defense

The Three Lines of Defense (3LOD) model is a comprehensive framework used to manage performance and risks within an organization[29]. It delineates responsibilities across three distinct lines to ensure effective risk management and governance.

The first line implements and carries out the day-to-day tasks to manage performance and risks taken to achieve strategy and business objectives[30]. Second line provides guidance on performance and enterprise risk management requirements, and evaluates adherence to defined standards[30]. The third line, most commonly internal audit, often provides the last line of accountability by performing

audits or reviews of enterprise risk management practices, identifying issues and improvement opportunities, making recommendations, and keeping the board and executive management up-to-date on matters requiring resolution[30]. External auditors provide management and the board of directors with a unique, independent, and objective view that can contribute to an entity's achievement of its strategy and business objectives[30].

2.2.2 COSO

The Committee of Sponsoring Organizations of the Treadway Commission (COSO) is a joint initiative established in the United States dedicated to providing thought leadership through the development of frameworks and guidance on enterprise risk management, internal control, and fraud deterrence[31]. While internal control is integral to enterprise risk management, enterprise risk management is part of the governance process [32]. Therefore, it is considered essential for every organization to establish an internal control system [32]. COSO's widely recognized Internal Control-Integrated Framework provides a comprehensive structure for organizations to design, implement, and evaluate their internal control systems, ensuring compliance with financial regulations and promoting efficient operations. The framework outlines five components of internal control: control environment, risk assessment, control activities, information and communication, and monitoring activities, which work together to help organizations achieve their objectives[33].

2.2.3 COBIT

Control Objectives for Information and Related Technologies (COBIT) is a framework created by ISACA (Information Systems Audit and Control Association) for the governance and management of enterprise IT. COBIT provides a comprehensive framework that assists organizations in achieving their objectives for the governance and management of enterprise IT. It helps organizations create optimal value from IT by maintaining a balance between realizing benefits and optimizing risk levels and resource use. COBIT 2019, the latest iteration, includes design factors and governance and management objectives that help customize the governance system to align with enterprise strategy and goals, ensuring robust IT governance and management practices.

2.2.4 ITIL

The Information Technology Infrastructure Library (ITIL) is a set of detailed practices for IT service management (ITSM) that focuses on aligning IT services with the needs of business. ITIL describes processes, procedures, tasks, and checklists which are not organization-specific, but can be applied by an organization for establishing integration with the organization's strategy, delivering value, and maintaining a minimum level of competency. ITIL allows organizations to establish a baseline from which they can plan, implement, and measure improvement. It is used to demonstrate compliance and to measure improvement. ITIL 4, the latest version, introduces the concept of the service value system, which provides a holistic approach to the co-creation of value through service management.

2.2.5 ISO

The International Organization for Standardization (ISO) is an independent, non-governmental international organization with a membership of 165 national standards bodies. ISO develops and publishes international standards, such as ISO 27001 for information security management and ISO 9001 for quality management systems. These

standards provide frameworks and best practices that organizations can use to ensure that their products and services consistently meet customer requirements and that quality is consistently improved. ISO standards are designed to be universally applicable to organizations of all types and sizes, and they play a critical role in facilitating international trade and ensuring the reliability and quality of products and services across global markets.

2.2.6 Internal control

Internal controls are processes put into place by management to help an organization operate efficiently and effectively to achieve its objectives. ISACA defines internal controls as the policies, procedures, practices and organizational structures designed to provide reasonable assurance that business objectives will be achieved and undesired events will be prevented or detected and corrected[34]. The Committee of Sponsoring Organizations of the Treadway Commission (COSO) defines internal control and control systems as the specific measures that provide assurance that an enterprise's operations are effective and efficient, its financial reporting is reliable, and the enterprise is in compliance with all regulatory requirements. COSO describes these objectives further and provides detail on control components (risk assessment, control environment, control activities, information and communication, and monitoring) that are used to accomplish them[30].

Managers often think of internal controls as the purview and responsibility of accountants and auditors. The fact is that management at all levels of an organization is responsible for ensuring that internal controls are set up, followed, and reviewed regularly. The purposes of internal controls are to protect assets, ensure that records are accurate, promote operational efficiency, achieve organizational mission and goals, and ensure compliance with policies, rules, regulations, and laws[35].

2.3 (Scaled) Agile

Agile methodologies emerged in the 1990s as a response to the limitations of traditional, plan-driven software development methods. These methodologies prioritize flexibility, collaboration, and customer satisfaction through iterative development and continuous feedback. Agile processes, such as Scrum, Kanban, and Extreme Programming (XP), enable small, self-organizing teams to deliver high-quality software quickly and efficiently. The Agile Manifesto[36], published in 2001, outlines the core values and principles that guide agile practices, emphasizing individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan.

As organizations grew and projects became more complex, the need to extend agile principles beyond small teams to larger, enterprise-level projects became apparent. This led to the development of various scaled agile frameworks that address the challenges of coordinating multiple teams and managing dependencies in large-scale environments.

2.3.1 Scaled Agile Framework (SAFe)

The Scaled Agile Framework (SAFe) was introduced by Dean Leffingwell in 2011 as a structured approach to scaling agile practices across large organizations[23]. SAFe integrates principles from lean and agile methodologies to provide a comprehensive framework that supports multiple teams working on complex projects. It includes four levels: team, program, value stream, and portfolio, each with specific roles, responsibilities, and processes. At the team level, SAFe incorporates Scrum and Kanban practices, while the program level introduces the concept of Agile Release Trains (ARTs) to synchronize the work of multiple teams. The value stream level focuses on delivering end-to-end value, and the portfolio level aligns strategic objectives with execution through lean portfolio management. SAFe emphasizes alignment, collaboration, and delivery of high-quality products through a combination of agile and lean practices.

In the context of the SAFe, a Solution is referred to as the outcome of a development effort. A Solution is a product, system, or service that provides value to internal or external customers[23].

One of the key concepts in SAFe is the solution intent, which is the repository for storing, managing, and communicating the future vision and current state of the solution under development. Solution intent ensures that all stakeholders have a clear understanding of the desired outcome and provides a basis for decision-making and adjustments as the project evolves. When solution intent, future vision and current state are coherent and consistent, Enterprise architects can use the solution intent to align technical decisions with business goals, ensuring coherence and consistency across the entire organization.

Verification and validation (V and V) are processes used to assure a product, service, or system is designed in accordance with the solution intent and is fit for its intended purpose [23].

2.3.2 Large Scale Scrum (LeSS)

Large Scale Scrum (LeSS) is a framework developed by Craig Larman and Bas Vodde in 2008 to extend Scrum principles to large-scale projects. LeSS retains the simplicity of Scrum while adding minimal rules and guidelines to address the complexities of large-scale development. It introduces two frameworks: basic LeSS for up to eight teams and LeSS Huge for more than eight teams. LeSS emphasizes feature teams that are cross-functional and self-managing, working together to deliver a single product increment. Key practices include regular coordination meetings, joint sprint planning, and retrospectives to ensure alignment and continuous improvement. LeSS focuses on maintaining the integrity of Scrum while providing the necessary structure for large-scale projects.

2.3.3 Disciplined Agile (DA)

Disciplined Agile (DA), previously known as Disciplined Agile Delivery (DAD), was developed by Scott Ambler and Mark Lines in 2012. DA is a process decision framework that provides a toolkit for scaling agile practices across an organization. It encompasses a wide range of life cycles, including agile/basic, lean/advanced, continuous delivery, and exploratory, allowing teams to choose the most appropriate approach for their context. DA emphasizes disciplined agile delivery, enterprise awareness, and continuous improvement. It addresses the full delivery lifecycle, including inception, construction, and transition phases, with a focus on integrating agile and lean practices to de-

liver high-quality solutions. DA provides flexibility and guidance to tailor agile practices to the specific needs of large-scale, complex projects.

2.4 Enterprise Architecture Management

Enterprise architecture (EA) involves defining and representing a high-level view of an enterprise's business processes and IT systems, their interrelationships, and the degree to which these processes and systems are shared across the enterprise. EA's primary responsibility is to translate strategy into systems and processes that enable the organization to achieve its goals. The core objective of EA is to define the desired future state of the organization's business processes and IT systems (often referred to as the "to-be", intentional architecture[23] or target architecture) and to create a roadmap for transitioning from the current state ("as-is", emergent design[23] or baseline architecture) [37]. EA is composed of two key components: the planning process (the "definition") and the tangible outputs of that process (the "representation"), such as EA documentation, including architecture diagrams, roadmaps, and other artifacts.

2.4.1 Agile architecture

The agile movement, with its emphasis on "just enough documentation, just in time," introduces a paradigm shift towards simplicity and efficiency in EA[25]. This approach allows for changes to requirements at any stage of the IT project and aligns with one of the twelve principles of the Agile Manifesto—"Simplicity"—which advocates for the use of minimal project documentation[36].

Adopting agile enterprise architecture can significantly enhance communication, quality, and functionality in distributed agile software development, potentially leading to on-time project completion[38]. Agile architectures are designed to accommodate ad-hoc changes in business processes, making them particularly valuable in dynamic environments[39].

Frameworks like Scrum can be utilized to create architecture deliverables and facilitate collaboration between enterprise architects and agile development teams, ensuring that architecture practices are integrated with agile methods [40].

Furthermore, DevOps practices enable continuous exploration, requiring agile architecture to meet quality and security standards, align with value stream performance objectives, and manage tangible configurations under version control. This process should generate backlog items and non-functional requirements (NFRs) that support Agile planning and emergent design. The CALMR (Culture, Automation, Lean, Measurement, and Recovery) mindset is essential in guiding architectural decisions to maximize delivery speed and solution value [23].

An integration of agile methodologies with architecting processes can be observed, such as the Scaled Agile Framework (SAFe), and the incorporation of agile elements into traditional frameworks like The Open Group Architecture Framework (TOGAF)[41][42][23].

2.4.2 TOGAF

TOGAF, The Open Group Architecture Framework, is a comprehensive framework for developing, managing, and implementing enterprise architecture [42]. It provides a structured approach for organizations to design, plan, implement, and govern enterprise information architecture. The TOGAF Content Framework offers detailed guidelines and best practices for creating architectural deliver-

ables, including models, documents, and artifacts, ensuring consistency and completeness in architectural work. Central to TOGAF is the Architecture Development Method (ADM) cycle a step-by-step process that guides architects through the stages of defining an architecture, from preliminary planning and visioning to implementation and governance. This cyclical process ensures that architecture evolves in response to business needs, enabling continuous improvement and alignment with organizational goals.

2.4.3 IT4IT

IT4IT, developed by The Open Group, is a comprehensive reference architecture designed to manage the business of IT effectively. It provides a holistic framework that integrates all aspects of IT management, aligning IT services with business goals and optimizing the entire IT value chain. By leveraging IT4IT, organizations can streamline their IT operations, improve service delivery, and enhance overall IT efficiency. The framework covers key areas such as strategy, portfolio, requirement-to-deploy, request-to-fulfill, and detect-to-correct, ensuring that IT functions cohesively support business objectives and deliver value consistently[24].

3. RESEARCH DESIGN

This research follows Wieringa’s design science research methodology to investigate the problem, to design and validate the artifact[26]. Wieringa’s design science research methodology was chosen because it offers a comprehensive collection of tools and guidelines that are specifically suited for the design and validation activities. The research approaches used in this study have been reviewed and approved for ethical standards by the Ethics Committee BMS of the University of Twente, a committee that oversees and ensures ethical research involving human subjects in the Humanities & Social Sciences within the BMS Faculty and other UT faculties.

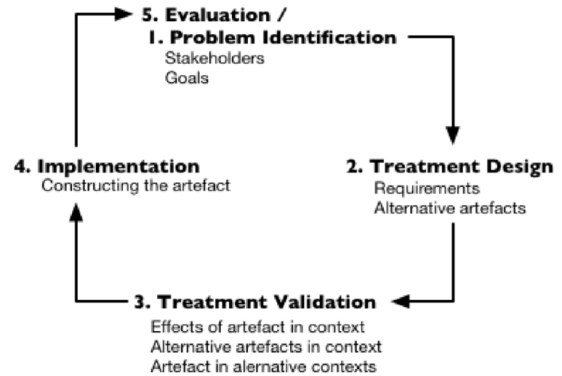


Figure 2. Wieringa Design Science Cycle[26]

In Section 1, the identified problem context is introduced and motivated. A systematic literature review is presented in Section 4, which was performed to assess what treatments already exist. These insights were used for the next phase of treatment design. A qualitative field study, described in Section 5, was performed to identify key components and risks of solution architecture documentation. Insights from this qualitative field study were used for the treatment design. The designed artifact in this study, presented in Section 6, is a set of control activities on solution architecture documentation which can be used by practitioners to implement on an agile enterprise architecture management process. As described in Section 7, the control activities were validated by a live demonstration of their practical implementation in a controlled environment, afterwards the participants filled in a survey and were asked to clarify and discuss the answers given to get qualitative insights. In Section 8, contributions, the relevancy, limitations and future work are discussed. The paper is concluded in Section 9.

In Figure 2 and Table 1 Wieringa’s design cycle phases are outlined. By iterating through these phases, the design cycle provides a systematic approach to designing and evaluating solutions, ensuring they are effective and aligned with organizational goals. Wieringa’s engineering cycle, step 4 and 5 of Figure 2, is not in the scope of this research.

Design Cycle	Methodology	Section(s)
Problem investigation	Literature review	1
Treatment design	Systematic literature review, Qualitative field study	4, 5, 6
Treatment validation	Expert opinions, Demonstration, Survey	7

Table 1. Application of Wieringa’s design science research methodology

4. SYSTEMATIC LITERATURE REVIEW

In this section, a systematic literature review is presented that outlines control activities known for use in (scaled) agile environments to improve risk management.

4.1 Research methodology

For the systematic literature review the five-grounded theory[43] is used and reported using the PRISMA statement[44]. Only scientific articles are included. Only papers published after 2001 are included, since The Agile Manifesto was published in February 2001 [36]. Databases used are Science Direct, Scopus, Web of Science and IEEE Xplore. Only papers written in English were included. The areas "Computer Science", "Engineering", "Business, Management and Accounting" and "Decision Sciences" were included. Terminology for these areas can differ between databases.

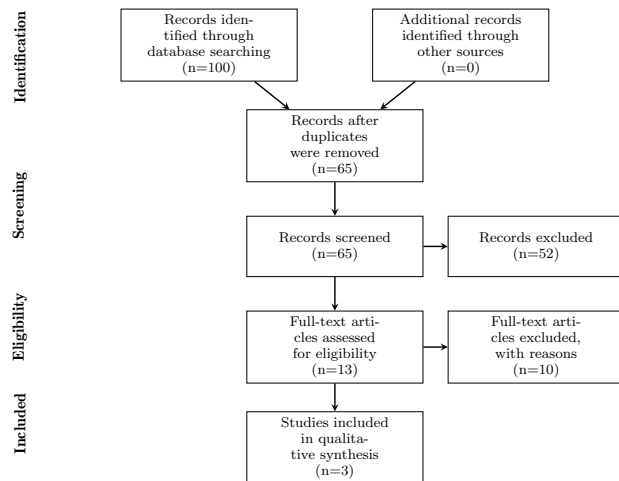


Figure 3. PRISMA statement SLR research process [44]

The goal of the systematic literature review is to identify control activities that are known for use in a (scaled) agile environment to improve risk management. Therefore the following search query was used: ("control activities" OR "control practices" OR "control measures" OR "IT audit") AND "agile".

As illustrated in Figure 3 the systematic literature review (SLR) identified 100 records through database searching, which were reduced to 65 after removing duplicates. After screening and eligibility assessment, 3 studies were included in the qualitative synthesis[45][46][47]. 10 full-text articles that were excluded for reason that these papers mentioned the concept of control activities, but did not explicitly gave examples to include in this systematic literature review.

Control activities are categorized into preventive and detective/corrective control activities[46]. Preventive controls are those intended to avoid errors or issues from occurring, while detective and corrective controls are designed to identify and rectify issues that arise during the development process. Preventive controls include enforce-

ment, standardization, and prioritization, aimed at establishing guidelines and preventing deviations. Detective and corrective controls, such as active monitoring, personnel interactions, and remediation, focus on identifying problems and implementing solutions to ensure the project remains on track. The found control activities in literature are detailed in Section 4.2.

4.2 Results

4.2.1 Preventive Controls

Enforcement

Consequences that project team members face if they break the rules set by management for the development project[46].

Standardisation

The definition of a series of established and accepted development activities that are conducted on each project [46].

Prioritisation

The establishment of a defined precedence of outstanding development project activities [46].

Automated change controls and thresholds

Higher impact changes need to be approved by independent units or a change advisory board[48][45].

Version Control

To quickly roll back deployments and trace changes, companies should always integrate version control and associated version control procedures[49][50][51][52][45]

Peer Review

Always require a product to be peer reviewed before deploying[48][53][54][45].

Testing: Control activities include test case writing and execution, unit testing, system testing, integration testing, and regression testing[47].

Security

Static code analysis[48][54], automated security tests[48][53][55][52][56], security training[57][58][56], configuration management[49][51][59][60][56], inventory management[51][59][60], separation of application and databases[61], penetration testing [45] and Risks analysis/Threat modelling[53][61][56] were mentioned as control activities to enhance security.

Training

Provide regular training sessions [45].

Soft Governance

Defining clear roles and responsibilities[62][61], communication and knowledge sharing[62][50][63][64][55] and autonomous teams[57][62] were mentioned in literature to improve soft governance. Soft governance incorporates both hard controls (content and process) and soft controls (culture and relations) [45].

Control Categories	Control Dimension	Definitions	Source
Control Activities - Preventive Controls	Enforcement	The disciplinary repercussions faced by project participants who contravene defined development project guidelines set by management.	[46]
	Standardisation	The definition of a series of established and accepted development activities that are conducted on each project.	[46]
	Prioritisation	The establishment of a defined precedence of outstanding development project activities.	[46]
	Change control	Higher impact changes need to be approved by independent units or a change advisory board.	[45]
	Version control	In order to quickly roll back deployments and trace changes, companies should always integrate version control and associated version control procedures.	[45]
	Peer review	Always require a product to be peer reviewed before deploying.	[45]
	Testing	This includes test case writing and execution, unit testing, system testing, integration testing, and regression testing.	[47]
	Security	Automated security tests like penetration testing.	[45]
	Training	Provide regular trainings.	[45]
	Soft governance	Takes into account both hard controls (content and process) as well as soft controls (culture and relations).	[45]
Control Activities - Detective/Corrective Controls	Active monitoring	The periodic inspections and confirmations conducted by project stakeholders to verify the execution and progress of development activities.	[46], [45]
	Personnel interactions	The ongoing exchange of project information, activities and development status among project stakeholders.	[46]
	Remediation	The identification and rectification of issues preventing the effective and efficient execution of systems development project activities.	[46]
	Compliance	Regular auditing, item tracking, standard templates in tools, automated compliance testing and reporting, isolation of test and development system from production.	[45]

Table 2. Control Categories, Dimensions, Definitions and Sources

4.2.2 Detective and Corrective Controls

Active Monitoring

Periodic inspections and confirmations conducted by project stakeholders to verify the execution and progress of development activities [46][45]. Control activities mentioned were logging [48][53][59][55], continuous monitoring [49][50] and reporting[49][50].

Personnel Interactions

Ongoing exchange of project information, activities, and development status among project stakeholders [46].

Remediation

Identification and rectification of issues preventing the effective and efficient execution of systems development project activities [46].

Compliance

Regular auditing[49][55], Item tracking[65], standard templates in tools[65][60], automated compliance testing and reporting[49][66], isolation of test and development system from production[53][61] were mentioned as control practices to enhance compliance.

5. QUALITATIVE FIELD STUDY

5.1 Research methodology

A qualitative field study was conducted at a large financial service provider in the Netherlands, referred to hereafter as "the organization". The organization employs more than 16,500 people and serves over 10 million customers globally. The organization has adopted the SAFe framework and is applying SAFe concepts to the enterprise architecture management process. The goal of the qualitative field study was to verify the outcome of the systematic literature review, identify key components and risks of agile solution architecture documentation and give insights in how the control activities from the literature review can be implemented on an agile solution architecture documentation process. In the following subsections it is briefly explained which methodology has been used to achieve this goal.

5.1.1 Identifying key components of solution architecture documentation

Within this qualitative field study the key components of agile solution architecture documentation have been identified with Enterprise Architect and Solution Architect by mapping contents from TOGAF Content framework[42] and Archimate to key components of Solution Intent concept from the SAFe framework[23].

5.1.2 Identifying relevancy of key components risks

The identified key components are then used as input to identify risks groups and assess relative relevancy to control/mitigate those risk groups for financial service providers by performing the Analytic Hierarchy Process (AHP) method[67] in semi-structured interviews. The interview, with an approximate duration between 30 to 60 minutes, was structured to get insights into perceived risks linked to Solution Architecture documentation. Participants were first briefed on the interview recording for research purposes and provide verbal consent. They then outline their current roles, experience with agile solution architecture documentation and provide a brief organizational overview. The concept of Solution Architecture documentation was clarified to ensure mutual understanding. Using the An-

alytic Hierarchy Process (AHP)[67], participants assesses the relative importance of eight key risk factors of Solution Architecture documentation to do systematic comparison. Participants compared risks across various components, discuss existing control activities, and propose enhancements. The interview concludes with validation of AHP results, assessing participants' expectations and perceptions of the relevance and realism of identified relative risks by showing them the result chart.

5.1.3 Applicability of control activities on solution architecture documentation process

A proposed set of control activities for the solution architecture documentation process was developed in collaboration with a Enterprise Architect and an IT quality advisor.

The control activities known for use in (scaled) agile environments presented in Section 4 are abstract. A set of control activities have been mapped to the organization's solution architecture documentation process following TOGAF ADM Cycle[39], Scaled Agile Framework [33] and COBIT Manage Architecture APO 03.04[30] in collaboration with an Enterprise Architect and 2 IT Quality Advisors.

5.1.4 Data collection

The data collection process for this qualitative field study involved multiple methods, as detailed below:

- Observations were done at the organization for 8 months and have been used as input for this research. During this period, the researcher joined the Enterprise Architecture team. Insights have been documented and used as input for this research. The researcher was involved in weekly sessions with a Enterprise Architect and Solution Architect to construct a Solution Architecture documentation template by mapping SAFe Solution Intent components to TOGAF content framework and Archimate Viewpoints. The researcher was also involved in monthly sessions with a IT Quality Advisor and 2 Enterprise Architects to reassess Key Risks, Key Controls, and Control Activities after the organization has shifted from traditional to an agile enterprise architecture management process.
- 7 semi-structured have been conducted, in which the AHP method[67] has been performed to pairwise compare the risks on solution architecture documentation. An Enterprise Architect, 2 Solution Architects, Control Tester of the Enterprise Architecture Management process, and 2 IT Quality advisors participated.
- The control activities known for use in (scaled) agile environments presented in Section 4 are abstract. A set of control activities have been mapped to the organization's solution architecture documentation process following TOGAF ADM Cycle[42], Scaled Agile Framework [23] and COBIT Manage Architecture APO 03.04[34] in collaboration with an Enterprise Architect and 2 IT Quality Advisors. The mapped control activities are described in Section 5.2.10.

5.2 Results

This chapter presents an overview of key components of solution architecture documentation, identified risk groups,

and the relative relevancy of control/mitigation of those risk groups.

In Table 3, an overview is given of key components of solution architecture documentation that have been identified by mapping TOGAF Content framework[42], SAFe Solution Intent concept [23] and Archimate layers and viewpoints[68].

In Table 4, an overview is given of the risk groups that have been identified based on the key solution architecture documentation components and process.

The aggregated relative relevancy scores of risk control/mitigation are presented in Figure 4, which highlights the average relevance of each risk group.

The results show that certain risk groups are perceived as more important to control or mitigate than others. The following sections detail the average outcomes for each risk group and mention the differences found in the interviews between the perspectives of the various roles.

5.2.1 Document Data (2.33%)

Document data risks were considered the least significant by the interviewees. Control testers and IT quality advisors indicated that the completeness and accuracy of document data are relatively well-managed within the organization. While recognized as important, this risk group did not emerge as a primary concern compared to others.

5.2.2 Vision and Context (10.65%)

Vision and context were highlighted as critical by many interviewees, particularly enterprise architects and the SAFe solution architect. Ensuring that the solution architecture documentation aligns with the organization's strategic goals and the broader business context is seen as essential. The SAFe solution architect mentioned that without a strong vision and context description, stakeholders like product managers and development teams are unable to set the right specifications due to a limited vision and time frame for making decisions. The SAFe solution architect emphasized that without a vision extending 5 to 10 years into the future, the specifications will be based on a shorter-term horizon, typically around one year or less, which might not always be optimal. Still, solution architects and control testers, placed less emphasis on this risk group, focusing more on implementation details.

5.2.3 Specifications (13.10%)

Specification risks were consistently rated as highly significant across all roles. The interviewees stressed the importance of having detailed and up-to-date specifications to ensure that the solution architecture documentation meets all functional and non-functional requirements. The enterprise architect mentioned that when specifications are missing, incorrect or not up-to-date you have a high risk that the Solution will not correctly be developed not conforming to specifications and standards. This risk group is critical for maintaining the accuracy of the solution architecture documentation, as highlighted by both strategic and operational roles.

5.2.4 Design (12.95%)

Design risks were identified as the most significant, slightly edging out specifications. The design phase is crucial as it directly impacts the overall architecture and implementation of the solution architecture. Enterprise architects, placed particularly high importance on design risks, emphasizing the need for thorough documentation and sys-

tematic reviews. Enterprise architect mentioned that design describes how the solution will be constructed and to which aspects a solution is compliant to security, CIA-classification, which are critical components to document.

5.2.5 Implementation Governance (8.98%)

Implementation governance risks were considered moderately significant. The interviewees, especially those responsible for oversight and compliance, such as IT quality advisors, highlighted the necessity of strong governance practices to document and evaluate design decisions systematically. This ensures compliance with standards and maintains the integrity of the solution architecture documentation. The enterprise architect mentioned that the documentation of deviations on standards are critical in the total enterprise architecture management process.

5.2.6 Tests (11.42%)

Testing risks were another high-priority area, especially for the control tester and IT quality advisor. Comprehensive test plans, including unit, functional, system, and security tests, are essential to verify that the solution meets all requirements. This risk group is vital for identifying potential issues early in the implementation process. The tests documentation are used to deliver evidence that the solution is in line with the goals, and specifications of the solution, and is therefore a critical component.

5.2.7 Process (7.09%)

Process risks were considered less significant compared to other areas but still noteworthy. Ensuring that processes for maintaining the solution architecture documentation are correctly executed and adhered to is essential for consistency and accuracy. Operational roles, such as solution architects, placed more emphasis on process risks. The enterprise architect mentioned that maintaining the solution architecture documentation is critical to make sure information is up-to-date, for example in case of calamities up-to-date information must be present in the solution architecture documentation. The Enterprise Architect mentioned that in practise it is hard to keep all solution architecture documentation up-to-date, and therefore emphasised the importance of controls on solution architecture documentation maintenance process.

5.2.8 Security Process (6.59%)

Security process risks were moderately significant, with a focus on ensuring robust information security measures to protect the Solution Intent. This risk group was particularly emphasized by roles involved in control and quality assurance, highlighting the need to balance security with other critical aspects of the Solution Intent. It was mentioned that security risks on the documentation process is very important, but already mechanisms have been implemented such as security training at employee onboarding and mandatory e-learning. So expected is that this already mitigates critical security risks on the process. The control tester mentioned that the probability that security issues arise in the current situation is low, partly thanks to existing controls in place, but an the impact can be very high.

Section	Description
1	General
1.1	Introduction
1.2	Scope and Purpose
1.3	Management Summary
2	Document Data
2.1	Distribution List
2.2	Author(s)
2.3	Approval
2.4	Change History
2.5	Related Documents
2.6	Definitions, Terms, and Abbreviations
3	Vision and Context
3.1	Solution Vision
3.2	Solution Context
3.3	IA - Market Research
4	Specifications
4.1	Introduction
4.2	<i>Intentional Architecture</i>
4.3	<i>Emergent Design</i>
5	Design
5.1	Introduction
5.2	Business Architecture <i>Intentional Architecture & Emergent Design</i>
5.3	Information Systems Architecture <i>Intentional Architecture & Emergent Design</i>
5.4	Technology Architecture <i>Intentional Architecture & Emergent Design</i>
5.5	Implementation Governance <i>Intentional Architecture & Emergent Design</i>
6	Tests
6.1	<i>Emergent Design</i>

Table 3. Solution Architecture Documentation Components

Risk group	Description
Document Data	The risk of incomplete, incorrect, outdated or missing document data. Document data includes distribution lists, responsible architects and ARTs, change history, abbreviations used, related documents, and the status (Concept, Under Review, Approved, Rejected, Obsolete) of the solution architecture documentation.
Vision and Context	The risk of incomplete, incorrect, missing, or outdated vision and context. The vision includes alignment with the organization’s strategic goals and IT architecture, and capabilities. The context includes the description of the solution’s context, such as market research, external factors, and regulations.
Specifications	The risk of incomplete, missing, or outdated specifications. Specifications include functional and non-functional requirements of the intentional architecture and emergent design. This includes the CIA classification, data requirements (DRS), features, user stories, and the use of architectural standards and coding guidelines.
Design	The risk of incomplete, incorrect, missing, or outdated design. The design includes links and visualizations of models, frameworks, principles, and the use of application(s), data, and technology. Design can also include information security, threat models, assessments, and possible execution of penetration tests.
Implementation Governance	The risk of incomplete, missing, or outdated documentation of decisions regarding deviations from standards that are not recorded in waivers. For example, the risk is that design decisions are not systematically evaluated.
Tests	The risk of incomplete, missing, incorrect, or outdated test plans. Test plans include tests for unit, functional, system, and security tests to verify non-functional requirements, such as performance and security.
Process	The risk of not executing, not timely executing, or incorrectly executing the process for maintaining the solution architecture documentation. The process includes registering the solution architecture documentation file for each solution in the catalogs using the most current template, keeping the solution architecture documentation up-to-date (after each Program Increment), performing peer reviews, and going through the quality checklist.
Information Security Process	The risk of insufficient information security around the registration and maintenance of the solution architecture documentation. Information security includes the set of policies, measures, procedures, and processes that ensure the availability, integrity, and confidentiality (CIA classification) of information.

Table 4. Description of risk groups Solution Architecture Documentation

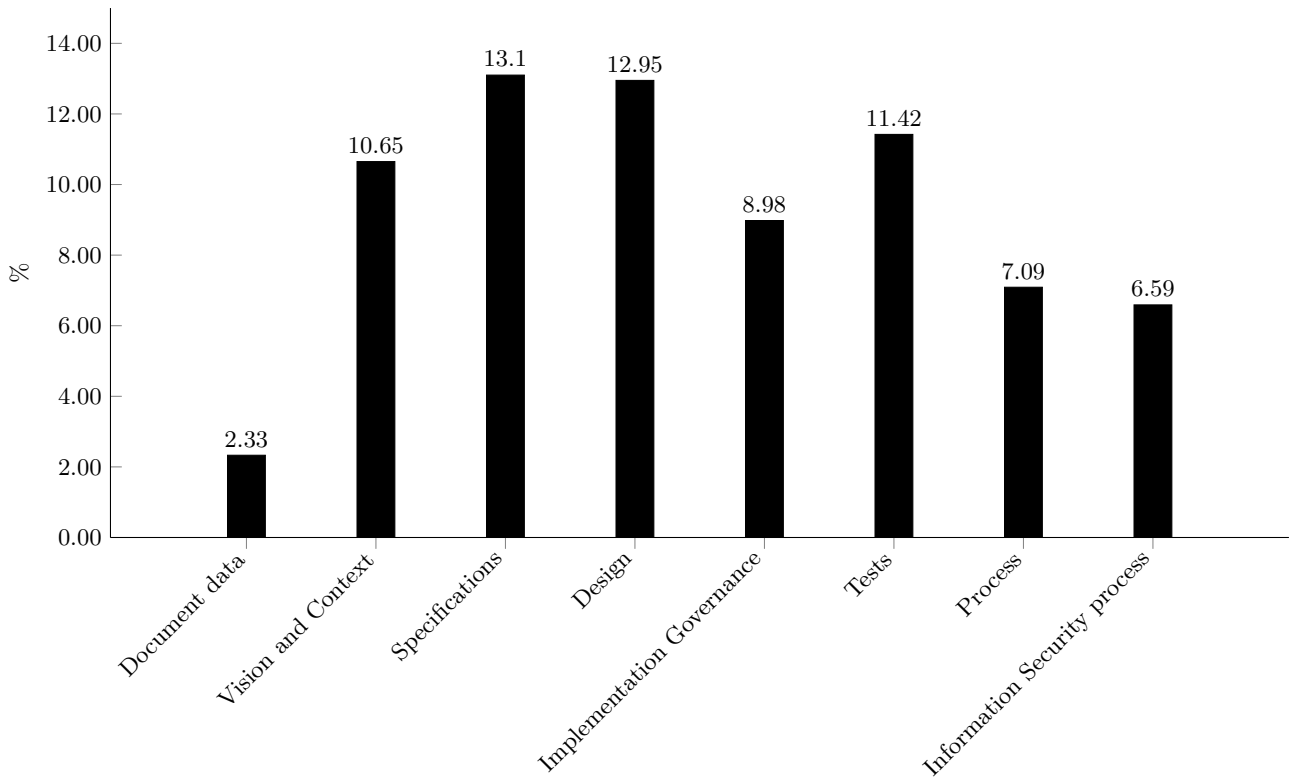


Figure 4. Relevancy of risk groups Solution Architecture Documentation

5.2.9 Control Framework

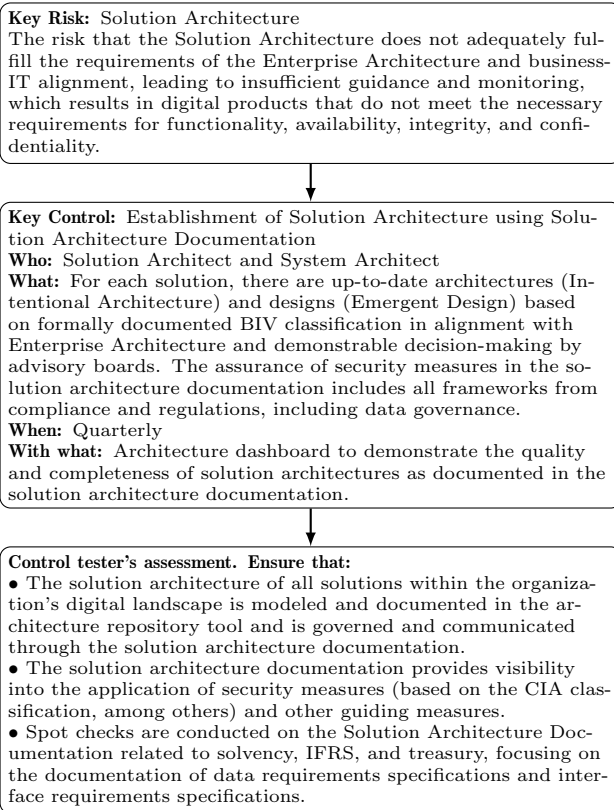


Figure 5. Organization's Key Risk, Key Control, and Work Program for solution architecture documentation process

The organization's control framework ensures that Solution Architecture Documentation aligns with business needs, mitigating key risks and allows Control Tester to demonstrate that they are in control of the solution architecture documentation process.

As shown in Figure 5, key risks, controls, and work programs are defined. Work Programs are utilized to operationalize compliance with Key Controls by outlining the evidence required from first-line departments. Periodically, typically quarterly or yearly, a control tester or approving manager produces an Audit Report, detailing the adherence to Key Risks, Key Controls, and corresponding Work Programs. This report includes validated evidence demonstrating compliance with Key Controls.

Quarterly and yearly, control testers from the first line will deliver the evidence that all controls are effective. The scope, depth, and frequency of second-line monitoring are outlined annually in the Compliance Year Plan.

5.2.10 Applicability of control activities on solution architecture documentation process

Table 5 presents a set of controls specifically tailored to the solution architecture documentation process. The set of control activities for the solution architecture documentation process was developed in collaboration with a Enterprise Architect and an IT Quality Advisor. The control activities in Table 5 aim to provide the necessary evidence that the organization is in control of the solution architecture management process, described in Section 5.2.9.

5.3 Discussion

Interviewees consistently highlighted the interrelated nature of Solution Intent components, stressing that completeness across specifications, design, and testing is cru-

cial. A deficiency in one area can significantly impact the others.

A key insight from the control tester, who also serves as a SAFe Enterprise Architect, was the importance of balancing flexibility with quality. While vision and context are essential for maintaining agility, ensuring robust specifications, design, and testing processes is critical for upholding standards. Given regulatory expectations from bodies like the Dutch National Bank (DNB), maintaining control over architecture is crucial, with the documentation process playing a central role in this oversight.

The distinction between "information security" for the documentation process versus the security of the developed solution was clarified, with the former being particularly relevant to specifications, design, and testing. The solution architect pointed out that while document data may pose lower risks due to the collaborative nature of agile release trains, this means that if this component is inaccurate the impact is not high. On the other hand, missing information or out-dated information in vision and context, or design can disrupt the value delivery stream, making these areas particularly important.

Although the Control Framework doesn't specify who is responsible for creating documentation, the Solution Architect mentioned that ideally, responsibility should be distributed, with each stakeholder accountable for the documentation relevant to their role. This approach aligns with agile principles, emphasizing the need to minimize unnecessary documentation while ensuring that critical information is well-maintained.

Different roles within the organization approach risks from varying perspectives: enterprise architects focus on vision and design, while solution architects and control testers concentrate on specifications and testing. Despite these differences, there was a shared agreement on the necessity of accurate and complete specifications and design to maintain the integrity of the Solution Intent.

6. TREATMENT DESIGN

6.1 Research methodology

A controlled environment is used to implement the control activities. To implement the control activities, a controlled environment was established using SharePoint to store the Solution Intents and Solutions. This environment integrates seamlessly with Microsoft Power Automate for automating workflows and Power BI for visualizing status and performance metrics. Additionally, Power Apps was utilized to create a user-friendly interface, enhancing accessibility and interaction for users. Word templates were employed to include the key components of the Solution Intent structure. These Word documents are equipped with content controls that synchronize directly with the SharePoint site, ensuring a 1:1 alignment and real-time updates.

6.2 Requirements

The control activities must accurately address the identified risks in the solution architecture documentation process and be relevant and applicable to the specific risks and context of the financial service provider. The effort required to implement these control activities should be reasonable and justifiable relative to the benefits they provide. Additionally, the control activities should be easy for stakeholders to understand and apply within existing processes, and they must integrate seamlessly with exist-

ing tools, systems, and workflows. Furthermore, the control activities must align with the key controls and key risks identified in the organization's control framework. Stakeholder acceptance is crucial, so the control activities must be positively received by stakeholders, including Enterprise Architects, Solution Architects, IT auditors, and control testers.

In Section 6.2.1, the implementation of CA1 through CA9, as well as CA11, is explained through the addition of specific SharePoint attributes. Section 6.2.5 details the implementation of CA10, which involves the creation of a quality dashboard. Finally, in Section 6.2.2, the placement of a link to the CA12 work instruction is described.

6.2.1 SharePoint lists and attributes

Solution Catalogue Lists.

The SharePoint environment includes four lists, which serve as catalogues for storing all solutions within the organization. A Target Operating Model is employed, within which four distinct layers are identified for the positioning of the organization's digital IT products and solutions. The focus is placed on the deployment and application of products that are composed of technology and application components, aligned with the technology and application layers as defined in TOGAF/Archimate. Each list organizes solutions according to their respective layers, providing a systematic and structured organization of the solution catalogue lists.

Solution Architecture Documentation Document Library.

The Solution Architecture Documentation Document Library houses all solution architecture documentation files (in .docx format) and attributes related to the solutions. The attributes are used to deliver evidence to the control activities.

In 6.2.1, the most important SharePoint attributes are outlined and it is indicated which attribute support which control activities.

6.2.2 Registration page

A SharePoint page titled "Solution Intent Registration" has been created to facilitate the creation and maintenance of solution architecture documentation. This page includes a button linking to the work instructions for managing documentation and provides access to the dashboard, as detailed in Section 6.2.5.

6.2.3 Workflow Creation Solution Architecture Documentation document

The solution architecture must be documented in a Microsoft Word document, following the structure outlined in Table 3. Three workflows – manual, semi-automatic, and automatic – have been implemented and illustrated in Figure 6, Figure 7 and Figure 8, respectively.

A Microsoft Word template for Solution Architecture Documentation was created and is used across all workflows to produce consistent documentation. This template includes several built-in Content Controls, which serve as user input fields that can update corresponding SharePoint attributes.

The three workflows for creating Solution Architecture Documentation—manual, semi-automatic, and automatic—are designed to streamline the document creation process. In

Control Activity	Control Activity from SLR	Applicability on Agile Solution Architecture Documentation Process
CA1	Active monitoring	Solution Architecture Documentation is present for each solution
CA2	Active monitoring	For each Solution Architecture Documentation, it is clear which ART(s) are maintaining it
CA3	Active monitoring	For each Solution Architecture Documentation, it is clear who the responsible architect is
CA4	Active monitoring	For each Solution Architecture Documentation, it is clear who performed the peer review (name; date; approval)
CA5	Peer review	A quality field is available and filled for each Solution Architecture Documentation
CA6	Change control	The Solution Architecture Documentation is updated to the production situation at the end of each Program Increment
CA7	Peer review	A peer review takes place at the end of each Program Increment
CA8	Version control	Version control is set up for the Solution Architecture Documentation
CA9	Version control	It is clear which Solution Architecture Documentation version corresponds to which production version and/or release; version number is included
CA10	Active monitoring	Requested Solution Architecture Documentation information is available such that it can be displayed in a dashboard and used for automated controls
CA11	Peer review, Personnel interactions, Security tests	The content of the quality field is based on the review checklist
CA12	Standardization	There is a work instruction for the use and standardization of the Solution Architecture Documentation

Table 5. Mapping of control activities on solution architecture documentation process

	Solution	Agile Release Train(s)	Architect	Status	Peer review done by	Peer review feedback	Peer review date	Release date
Control Activity	CA1	CA2	CA3	CA5	CA4, CA7	CA4	CA4, CA11	CA6, CA7, CA8, CA9
What	Solution from Catalogus	Responsible ART(s)	Responsible Architect	Status of documentation	Date of last peer review	Feedback from peer review	Date of last peer review	End date of Program Increment
Type	Lookup	Lookup	Person	Choice	Date	Text	Date	Date

Table 6. Solution Architecture Documentation List Attributes

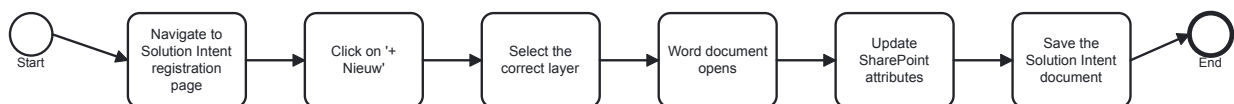


Figure 6. Manual Solution Intent creation



Figure 7. Semi-Automatic Solution Intent creation

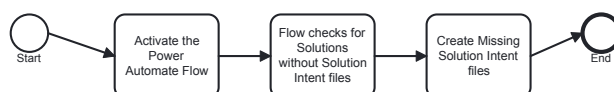


Figure 8. Automatic Solution Intent creation

the manual workflow, a document including the template is created on the registration page. All data entry and content generation are performed by the user directly within the Microsoft Word template or SharePoint attribute editor, requiring full manual effort. The semi-automatic workflow introduces a combination of manual input and automated processes, where some content is automatically pre-filled or synchronized with SharePoint attributes based on already known information about the Solution. This work-flow is reducing the manual workload while allowing user control. The automatic workflow fully automates the creation of the documentation, automatically populating the document template with data pulled from integrated systems and SharePoint, thereby minimizing user intervention. The solution architecture documentation is initially created with the status 'Empty', indicating that no content has yet been provided by an architect.

The Microsoft Flow for automatic Solution Architecture Documentation files creation operates by retrieving data from a SharePoint list of all solutions, using this information to create new documents based on a predefined template, and updating the associated metadata. The flow can be set to run on a schedule and iterates through each solution in the solutions list, ensuring that all necessary documents are created and their properties are accurately maintained. This automation reduces manual effort and ensuring consistency across the organization's solution intents.

6.2.4 Data model

In Figure 9 the underlying data model is illustrated using an UML Class Diagram. The model illustrates the structure and relationships within the SharePoint environment. The 'Solution' class is designed to store information about individual solutions, while the 'SolutionIntent' class is responsible for storing the associated Solution Architecture Documentation files. Each documentation file has a related 'SolutionIntentReview' class, which captures details such as the peer reviewer, review date, feedback, and potentially a maturity level score assessing the quality of the documentation. Additionally, the 'SolutionIntentChecklist' class is used to store the results of quality checks related to the documentation.

6.2.5 Dashboard

A dashboard, presented in Figure 10 and 11, was built in Microsoft Power BI to visualize the controls. The dashboard features an overview screen with multiple filter options, including IT4IT layer, responsible ART(s), and responsible architect. The status of the documentation is visualized using a ring chart, while the quality maturity level is displayed via a radar chart, indicating the five quality dimensions of the documentation. Additionally, the dashboard reports the number of solutions documented and highlights any solution that have missing documentation.

A "Solution Details" button is provided for deeper exploration. When a specific solution is selected, this button leads to a detailed overview screen. As shown in Figure 11, this detailed view includes information such as the solution's name, peer review details, CIA classification, responsible architect, quality maturity levels with textual feedback, status of documentation (Accepted, Under Review, Declined, In Playback, Empty), and the release date. The dashboard also includes buttons to edit these attributes and to open the documentation file directly.

7. TREATMENT VALIDATION

7.1 Research methodology

The research methodology involved relevancy assessment of control activities and validation of control activities and dashboard in the controlled environment. First, the relevancy of the control activities within the Solution Architecture Documentation process was assessed through a survey using a Likert scale. The responses were converted into normalized relevancy scores to assess the perceived importance of each activity.

Following this, a demonstration was given by the research in a controlled environment to showcase how the control activities can be implemented. After the demo, the controls were evaluated against criteria such as effectiveness, efficiency, usability, and stakeholder acceptance using a likert scale, with the option to provide additional comments.

Next, a second demonstration focused on the dashboard that integrates these controls. Participants evaluated the dashboard using a Likert scale, assessing usability, effectiveness, and overall satisfaction, while also providing feedback and suggestions for improvement.

The relevancy score for each control activity was calculated using a weighted average of the Likert scale responses. Each response category was assigned a numerical value: Not at all relevant = 0, Not relevant = 1, Neutral = 2, Relevant = 3, Very relevant = 4. The weighted average was calculated using the formula. Additionally, after the demonstration, the control activities and dashboard were validated using a criteria assessment based on another Likert scale. This scale ranged from Strongly Disagree = 1, Disagree = 2, Neutral = 3, Agree = 4, to Strongly Agree = 5, with participants also having the option to provide additional comments.

$$\text{Relevancy Score} = \frac{(n_0 \times 0) + (n_1 \times 1) + (n_2 \times 2) + (n_3 \times 3) + (n_4 \times 4)}{N} \quad (1)$$

where n_0, n_1, n_2, n_3, n_4 represent the number of responses in each category, and N is the total number of responses for that activity. The scores were then normalized to a range between 0 and 1 by dividing by the maximum possible score of 4:

$$\text{Normalized Relevancy Score} = \frac{\text{Relevancy Score}}{4} \quad (2)$$

The closer the score is to 1, the higher the perceived relevance or the degree to which someone agrees.

7.1.1 Data Collection

The data collection process for the treatment validation involved a demonstration and survey, as detailed below:

- **Demonstration and Survey:** A detailed demonstration was performed by the researcher with 2 IT Quality Advisors, 9 Enterprise Architects, and 1 Control Tester. During the demonstration, participants were provided with a survey to complete, allowing them to evaluate the control activities in real-time. This interactive session also facilitated questions and discussions, enabling participants to elaborate on their survey responses and provide in-depth insights.

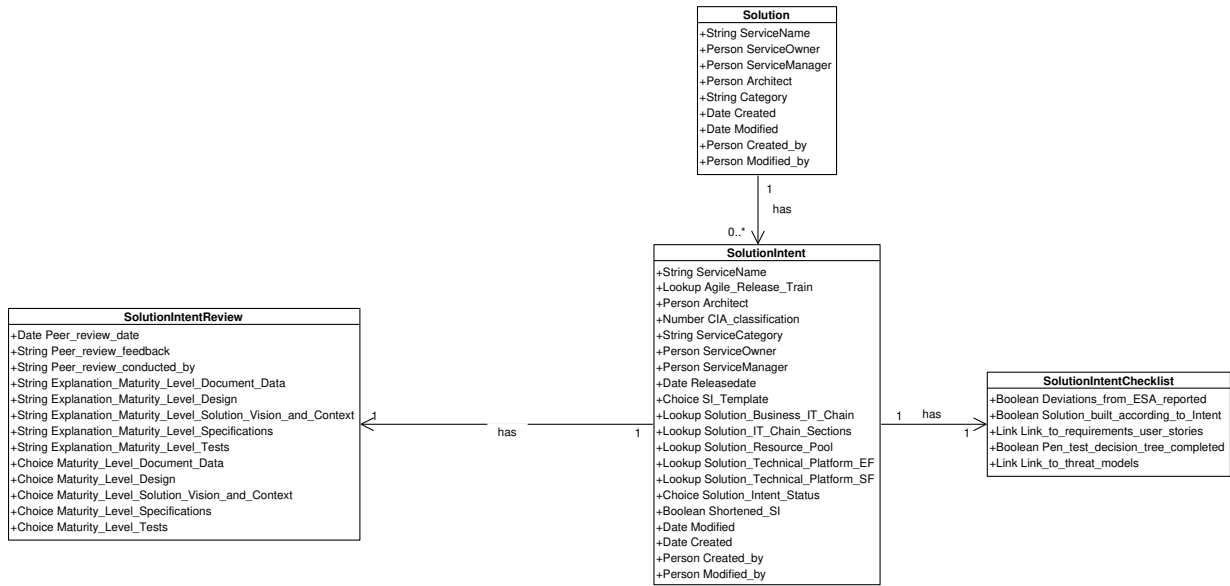


Figure 9. Solution Data Model UML Class Diagram

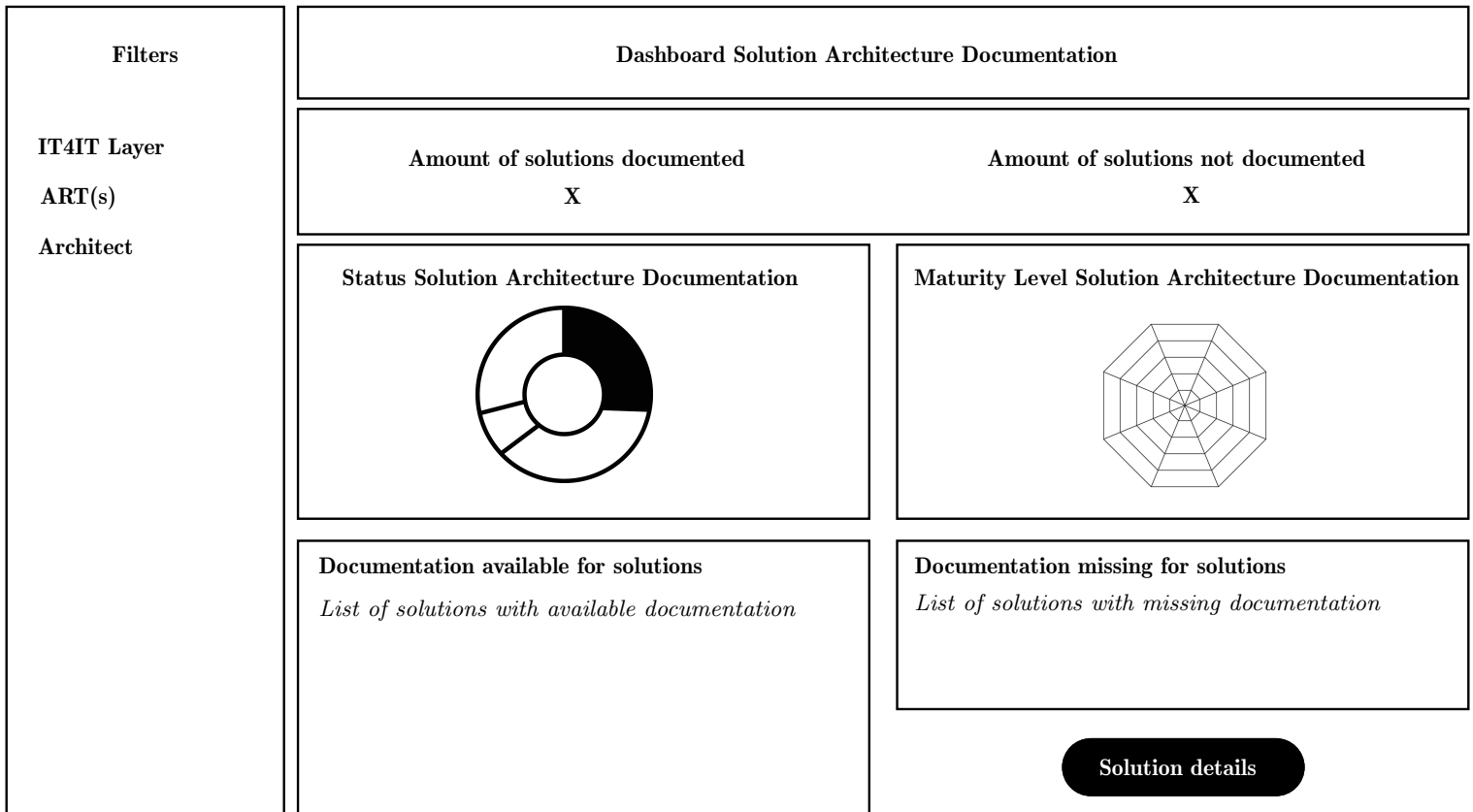


Figure 10. Dashboard Solution Architecture Documentation Overview

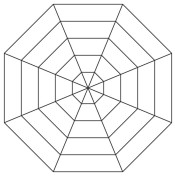
Filters IT4IT Layer ART(s) Architect	<div style="display: flex; justify-content: space-between;"> Edit attributes Name of Solution Open documentation </div>		
	Status documentation <i>Accepted, under review, declined, in playback, empty</i>	Release date <i>Date of release date corresponding to solution</i>	Maturity Level Solution Architecture Documentation 
	Responsible architect(s) <i>List of Architect(s) (Name, function, department of responsible architect)</i>		
	Peer review done on <i>Date</i>	Peer review done by <i>Name of peer reviewer</i>	
	Peer review feedback <i>Comments peer review</i>		Document data <i>Comments on dimension</i>
			CIA classification score <i>Based on availability, integrity, and confidentiality</i>
			Specifications <i>Comments on dimension</i>
			Design <i>Comments on dimension</i>
			Tests <i>Comments on dimension</i>

Figure 11. Dashboard Solution Architecture Documentation Details

7.2 Assessment Criteria

To validate the proposed control activities aimed at mitigating risks in the solution architecture documentation process, several key evaluation criteria were employed. Effectiveness was assessed based on the accuracy and relevance of the controls in addressing the identified risks and their applicability to the specific context of the financial service provider. Efficiency was measured by evaluating the implementation effort required, ensuring it was reasonable and justifiable in relation to the benefits. Usability focused on the ease with which stakeholders could understand and apply the controls within existing processes, as well as the seamless integration of these controls with current tools, systems, and workflows. Finally, expected stakeholder acceptance was estimated by asking key stakeholders for expectations, including Enterprise Architects, Quality advisors to get indication if the controls will be well-received.

7.3 Results

7.3.1 Participant profiles

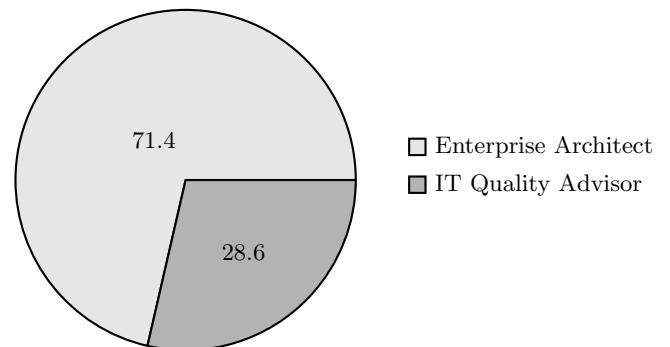


Figure 12. Distribution of Roles

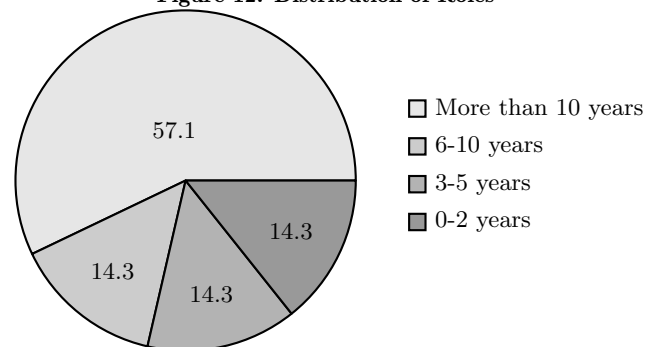


Figure 13. Distribution of Experience

7.3.2 Relevancy control activities

The average relevancy scores for each control activity related to Solution Architecture Documentation were assessed through a survey conducted after the demonstration. These scores, depicted in Figure 14, indicate how relevant each control activity was perceived in the context of the documentation process. The control activities are listed as CA1 through CA12, with each bar representing the average relevancy score on a scale from 0 to 1. The results suggest that control activities such as CA3, CA1, CA8, and CA10 were considered most relevant, with scores closer to 0.8 or higher, while CA12 was perceived as the least relevant, with a score of 0.43.

7.3.3 Applicability control activities

The validation of the control activities related to the Solution Intent revealed several key insights. The results are presented in Table 8 and Table 9. Regarding effectiveness (Q1), participants had mixed views, with some emphasizing the need for clarity on how these controls contribute to key controls tied to significant risks, while others highlighted the importance of integrating risk mitigation within the Solution Intent. For relevance and applicability (Q2), most respondents agreed that the quality controls are relevant to the organizations's specific risks. When considering efficiency (Q3), feedback highlighted concerns about the balance between effort and benefits, warning of potential bureaucratic overload that could lead to resistance. Usability (Q4) was generally well-received, with most finding the controls understandable and applicable, though the need for skilled architects was noted due to the complexity of the controls. In terms of integration (Q5), participants were optimistic about incorporating these controls into existing tools and workflows. Finally, stakeholder reception (Q6) was expected to be positive overall, but concerns were raised about the significant effort required. General suggestions included thorough testing of the implementation, maximizing automation, and clearly defining peer review responsibilities.

7.3.4 Dashboard

The validation of the Dashboard Solution Intent demonstrates strong overall approval from the participants, particularly in terms of its usability and effectiveness. The average score given was an 8.7. The results are shown in Table 9 and Figure 16. Respondents consistently rated the dashboard as user-friendly and practical, with many praising its clear design and ease of navigation. Key functionalities, such as the filters for the organizations' defined Target Operating Model layers, and responsibilities, were highlighted as particularly valuable and easy to use. The dashboard's ability to effectively visualize quality controls was also recognized, with participants noting that it provided meaningful insights into the quality of Solution Intent documents. This was further supported by the positive feedback regarding the quality maturity model integrated into the dashboard, which respondents found useful for assessing the quality level of the documentation. Despite the overall positive reception, there were some suggestions for minimizing the effort required to maintain the dashboard and improving its operationalization.

The feedback on the Dashboard was generally positive, with respondents appreciating its "look and feel," ease of use, and the clear, quick insights it provides into the status of projects. Many highlighted the dashboard's aesthetic appeal and its utility in connecting with current software, as well as its overall ability to present information effec-

tively. Suggestions for improvement included the need to reduce effort and clarify the distinction between peer reviews and formal assessments. Additionally, there was a call to focus on operationalizing the dashboard and ensuring that actions prompted by the dashboard are effectively followed up, emphasizing that the dashboard should go beyond just providing insights to driving actionable outcomes.

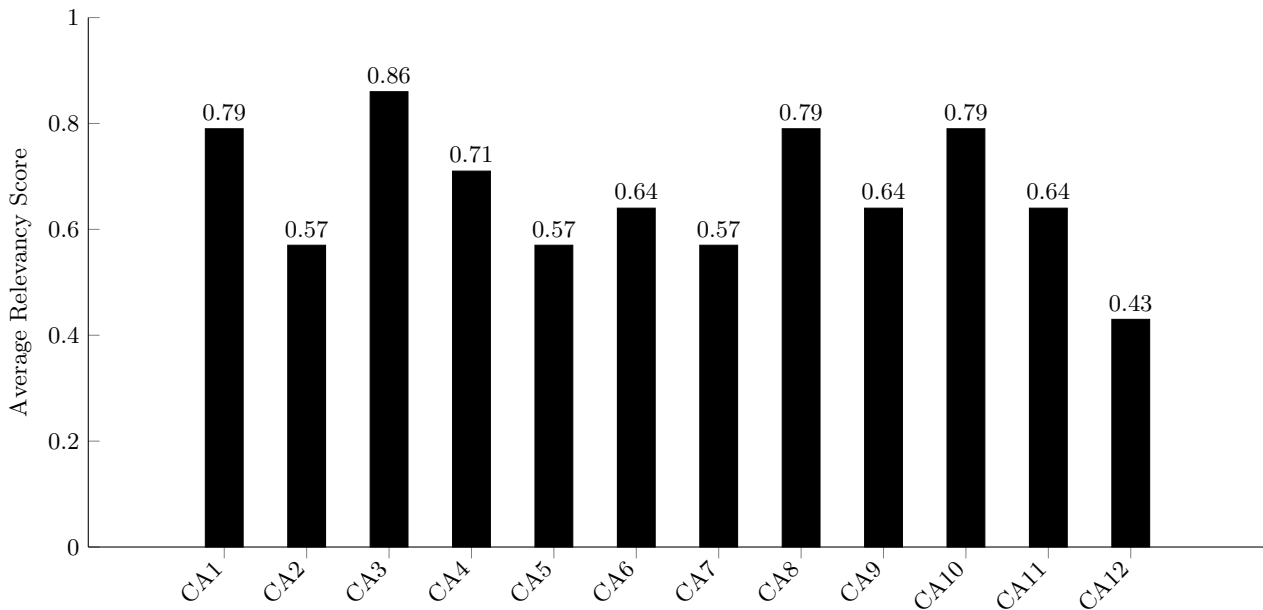


Figure 14. Average Relevancy Score for Solution Architecture Documentation Control Activities

Control Activity (CA)	Applicability on Solution Architecture Documentation Process	Calculated Relevancy Score
CA1	Solution Architecture Documentation is present for each solution	0.79
CA2	For each Solution Architecture Documentation, it is clear which ARTs are developing / maintaining it	0.57
CA3	For each Solution Architecture Documentation, it is clear who the responsible architect is	0.86
CA4	For each Solution Architecture Documentation, it is clear who performed the peer review (name; date; approval)	0.71
CA5	A quality field is available and filled for each Solution Architecture Documentation	0.57
CA6	The Solution Architecture Documentation is updated to the production situation at the end of each Program Increment	0.64
CA7	A peer review takes place at the end of each Program Increment	0.57
CA8	Version control is set up for the Solution Architecture Documentation	0.79
CA9	It is clear which Solution Architecture Documentation version corresponds to which production version and/or release; version number is included	0.64
CA10	Requested Solution Architecture Documentation information is available such that it can be displayed in a dashboard and used for automated controls	0.79
CA11	The content of the quality field is based on the review checklist	0.64
CA12	There is a work instruction for the use and standardization of the Solution Architecture Documentation	0.43

Table 7. Relevancy scores of control activities on solution architecture documentation process

Question	Question	Normalized Score (0-1)
Q1	The quality controls align with the key control for the creation and maintenance of the solution intent.	0.75
Q2	The quality controls are relevant and applicable to the specific risks and context of the organization.	0.7857
Q3	The effort required to implement the quality controls is reasonable and justified compared to the benefits.	0.7857
Q4	The quality controls are easy to understand and apply within the solution intent maintenance process.	0.8214
Q5	The quality controls integrate with existing tools, systems, and workflows.	0.8214
Q6	The quality controls are well received by stakeholders, including Enterprise Architects, Solution Architects, IT auditors, and control testers.	0.8214

Table 8. Results Control Activities Criteria Assessment

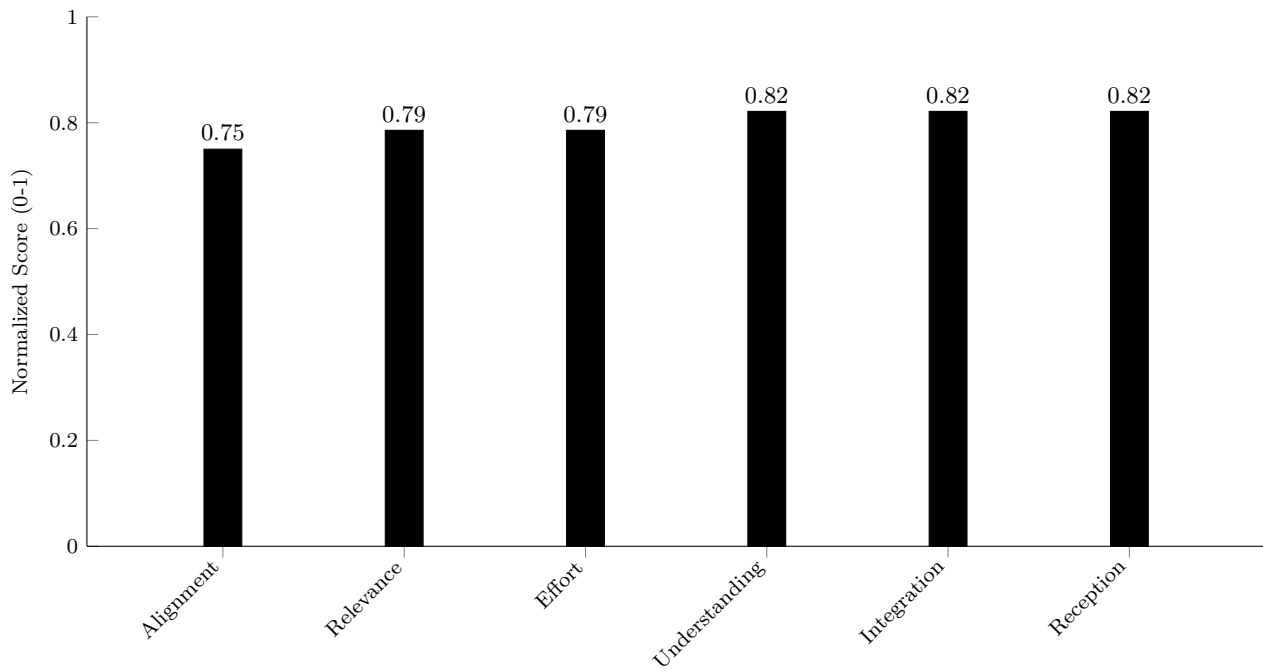


Figure 15. Results Control Activities Criteria Assessment

Question	Question	Normalized Score (0-1)
Q1	The filters are useful and easy to use	0.7857
Q2	The dashboard effectively visualizes the effectiveness of quality controls	0.75
Q3	The quality level provides useful insights	0.75
Q4	The dashboard is user-friendly and practical	0.8929

Table 9. Results Dashboard Features

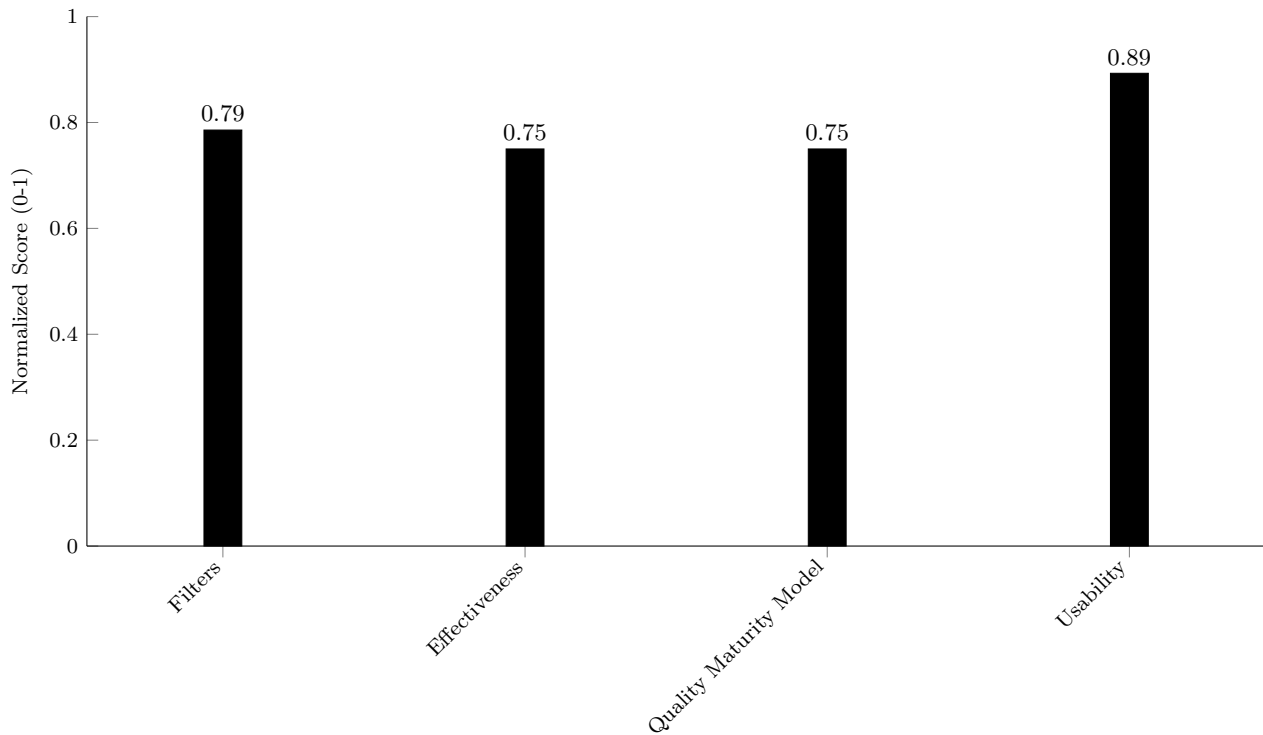


Figure 16. Results Dashboard Features Assessment

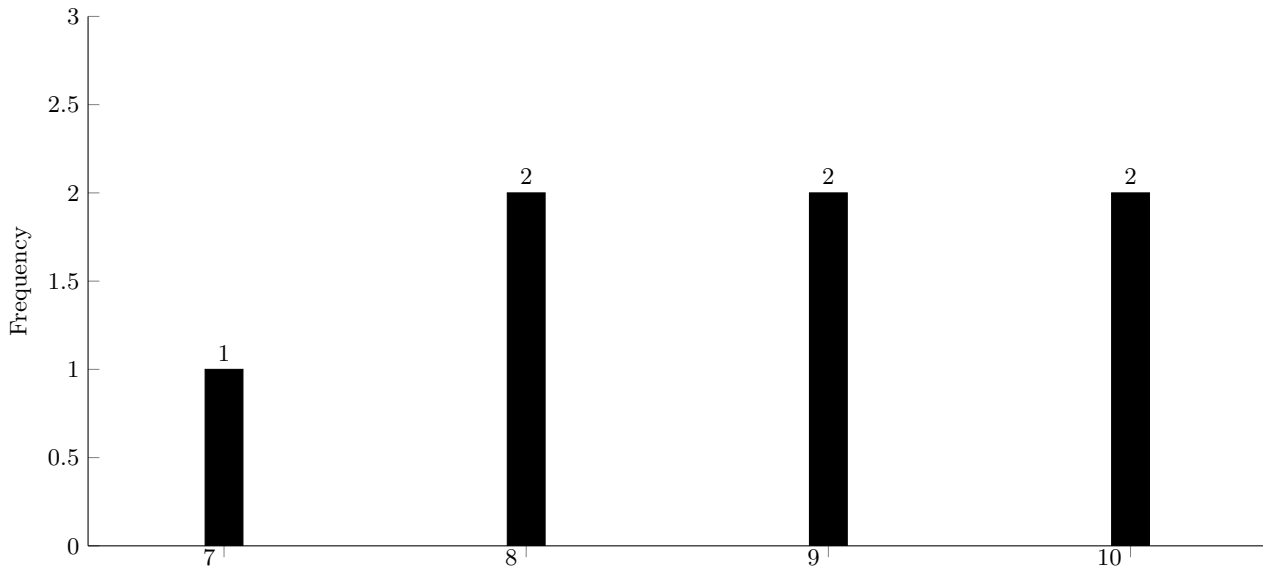


Figure 17. Distribution of Scores for Dashboard Solution Intent

8. DISCUSSION

8.1 Contributions to literature and practice

The systematic literature review revealed a gap in research concerning the overlap between control activities, agile practices, and enterprise architecture and is interesting in the field of Accounting Information Systems (AIS) domain. While agile methodologies emphasize flexibility and rapid response, traditional control frameworks like COBIT and COSO prioritize structured processes and comprehensive documentation. This research contributes to filling this gap by exploring how control activities can be adapted to agile environments, particularly in the documentation of solution architecture.

The qualitative field study further reinforces these findings by providing empirical evidence from a large financial service provider. The study identified key components of solution architecture documentation and associated risks, highlighting the critical areas where control activities are most needed. The integration of control activities like active monitoring, peer reviews, and version control within an (scaled) agile framework demonstrates how organizations can maintain agility while still ensuring risk management practices.

8.2 Relevance of Control Activities

The implementation of control activities within the solution architecture documentation process was validated through a demonstration and survey. The results indicate that certain control activities, such as version control and active monitoring, are particularly relevant in managing risks associated with agile documentation processes. These activities ensure that documentation remains up-to-date and accurate thereby supporting both agile practices and improving risk management.

The positive feedback from stakeholders, including enterprise architects and IT quality advisors, suggests that the proposed control activities are not only relevant but also feasible within the existing scaled agile framework. This aligns with the research goal of providing practical insights that can be readily applied in similar organizational contexts.

8.3 Limitations

8.3.1 Literature review

There are limitations that should be noted. The review focused primarily on scientific papers, which may have excluded relevant industry reports, conference papers, books, white papers, and practitioner insights, potentially limiting the breadth of perspectives on control activities in agile environments. Additionally, the inclusion criteria required the explicit mention of the term "agile," which might have overlooked relevant control activities not explicitly labeled as such, narrowing the scope of the review. The search strategy, while maintaining relevance, may have inadvertently excluded broader studies due to the specific terms used. The limited number of primary studies identified suggests that empirical research on control activities in agile environments remains scarce, highlighting the need for further research that incorporates a wider range of sources and considers control activities applicable to agile settings, even if not explicitly labeled as such.

8.3.2 Qualitative Field Study

The generalisability of the results in this study is subject to certain limitations. Due to the time constraints of this research, only one organization could be examined. The participant group, consisting of IT Quality Advisors, Enterprise Architects, and a Control Tester, while providing valuable feedback, may not capture the full spectrum of perspectives. Notably, for the validation process, only Enterprise Architects and IT Quality Advisors were involved, although the inclusion of Solution Architects and System Architects would have provided a more diverse range of perspectives. The study primarily utilized qualitative feedback, which, while detailed and rich in context, lacks the quantitative data that could further validate the findings. Future research should aim to incorporate both qualitative and quantitative data and expand the scope to include a wider range of organizations. Additionally, while this research effectively addresses Wieringa's design cycle, it does not extend to the implementation and validation stages of the engineering cycle, which would be valuable areas for future exploration.

8.4 Future work

This study primarily focused on the Design Cycle as defined by Wieringa, covering the phases of problem identification, treatment design, and treatment validation. Future research should extend this work by addressing Wieringa's Engineering cycle for treatment implementation and further validation. To further validate the control activities the inclusion of Solution Architects and System Architects would provide a more diverse range of perspectives.

Therefore, future research should focus on implementing the proposed solution in real-world settings to assess its practical applicability, followed by thorough validation through empirical studies. Also, investigating the long-term impact on risk management outcomes. This process will help identify operational challenges, refine the treatment based on real-world feedback, and ensure it meets the needs of diverse contexts.

Additionally, the quality maturity model proposed for peer review feedback, used to systematically assess the quality of the solution architecture documentation, was a proof of concept and requires empirical validation to ensure its effectiveness and reliability.

9. CONCLUSION

Through a systematic approach guided by Wieringa's design science research methodology, the study addressed the research questions by identifying relevant control activities, key components of solution architecture documentation, associated risks, and practical implementation possibilities.

This research aimed to explore the applicability of control activities within the agile solution architecture documentation process to improve risk management for financial service providers. By systematically addressing the research questions, the study has provided valuable insights.

An overview of control activities that are known for use in an agile environment to improve risk management is given. The identified control activities, which include preventive, detective, and corrective measures, were applied to the solution architecture documentation process.

By mapping the TOGAF content framework and Archimate to the solution intent concept from the SAFe framework, the study has identified the key components of a solution architecture documentation document. This mapping facilitated the identification of the most relevant risks associated with solution architecture in an agile context.

The implementation of control activities within a controlled SharePoint environment demonstrated how these controls can be designed, implemented, and visualized using a dashboard.

The use of SharePoint, along with integrations with Power BI and Power Automate, provided a practical example of how control activities can be automated and reported, offering valuable insights for financial service providers. The validation of this implementation within a financial service provider context showed, unlike critical feedback, positive reception, indicating the practical relevance and applicability of the proposed solution.

This research contributes to a deeper understanding of how control activities can be effectively integrated into agile environments to improve risk management. The findings offer an example for financial service providers to leverage agile methodologies while maintaining robust control and oversight over their solution architecture documentation processes.

The study revealed several key findings:

- **Identification of control activities:** Through a systematic literature review and field study, control activities such as active monitoring, peer review, change control, and version control were identified as crucial for improving risk management in an agile environment. These activities were mapped to the solution architecture documentation process, making them more concrete for practical implementation.
- **Key Components of solution architecture documentation:** The study identified the core components of solution architecture documentation within an agile context by mapping concepts from TOGAF, Archimate, and SAFe frameworks. These components include vision and context, specifications, design, and implementation governance and tests.
- **Relevance of risks:** The research identified specific risks associated with each component of the solution architecture documentation process. The most significant risks were linked to the design and specifications.
- **Treatment design:** The study demonstrated the practical application of control activities within a controlled environment using SharePoint, Power BI, and Power Automate. This integration facilitated monitoring, reporting, and automation of control activities, thereby improving the efficiency and effectiveness of risk management processes.
- **Treatment validation:** The validation phase, which included feedback from enterprise architects and IT quality advisors confirmed the relevance and applicability of the proposed control activities. The positive reception of the implementation highlights the potential for these control activities to be adopted by financial service providers to maintain control over agile solution architecture documentation processes.

The research provided an example for financial service providers to integrate control activities into their agile practices, thereby improving risk management and ensuring alignment with regulatory requirements. The findings contribute to both academic knowledge and practical application, offering a pathway for organizations to maintain agility while upholding control standards.

10. ACKNOWLEDGMENTS

I am deeply grateful to Achmea for the opportunity to conduct this research as an intern. Special thanks to Léon Korf de Gidts, Enterprise Architect at Achmea, for his consistent guidance. His insights, along with the contributions of others at Achmea, greatly enriched this research and made my experience enjoyable. I also sincerely thank my supervisors from the University of Twente, Prof. Dr. J. van Hillegersberg and Dr. Ir. M.J. van Sinderen, for their excellent mentorship and the pleasant meetings throughout this project.

11. REFERENCES

- [1] R. M. van Wessel, P. Kroon, and H. J. de Vries, "Scaling agile company-wide: The organizational challenge of combining agile-scaling frameworks and enterprise architecture in service companies," vol. 69, no. 6, pp. 3489–3502. Conference Name: IEEE Transactions on Engineering Management.
- [2] M. Hauder, D. Munch, F. Michel, A. Utz, and F. Matthes, "Examining adaptive case management to support processes for enterprise architecture management," in *2014 IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations*, pp. 23–32, IEEE.
- [3] P. Weill and S. L. Woerner, "Thriving in an increasingly digital ecosystem,"
- [4] B. Sherehiy, W. Karwowski, and J. K. Layer, "A review of enterprise agility: Concepts, frameworks, and attributes," vol. 37, no. 5, pp. 445–460.
- [5] E. Overby, A. Bharadwaj, and V. Sambamurthy, "Enterprise agility and the enabling role of information technology," vol. 15, no. 2, pp. 120–131. Publisher: Taylor & Francis _eprint: <https://doi.org/10.1057/palgrave.ejis.3000600>.
- [6] Lee and Xia, "Toward agile: An integrated analysis of quantitative and qualitative field data on software development agility," vol. 34, no. 1, p. 87.
- [7] M. West and D. Norton, "Market guide for enterprise agile frameworks."
- [8] D. K. Rigby, J. Sutherland, and A. Noble, "How to go from a few teams to hundreds," vol. 96, no. 3, pp. 88–96.
- [9] T. Dingsøy and N. B. Moe, "Towards principles of large-scale agile development: A summary of the workshop at XP2014 and a revised research agenda," in *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation* (T. Dingsøy, N. B. Moe, R. Tonelli, S. Counsell, C. Gencel, and K. Petersen, eds.), vol. 199, pp. 1–8, Springer International Publishing. Series Title: Lecture Notes in Business Information Processing.
- [10] T. Dingsøy, N. B. Moe, T. E. Fægri, and E. A. Seim, "Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation," vol. 23, no. 1, pp. 490–520.
- [11] Z. Zhang and H. Sharifi, "Towards theory building in agile manufacturing strategy—a taxonomical approach," vol. 54, no. 2, pp. 351–370. Conference Name: IEEE Transactions on Engineering Management.
- [12] F. O. Bjørnson, J. Wijnmaalen, C. J. Stettina, and T. Dingsøy, "Inter-team coordination in large-scale agile development: A case study of three enabling mechanisms," in *Agile Processes in Software Engineering and Extreme Programming* (J. Garbajosa, X. Wang, and A. Aguiar, eds.), vol. 314, pp. 216–231, Springer International Publishing. Series Title: Lecture Notes in Business Information Processing.
- [13] S. Nilsson Tengstrand, P. Tomaszewski, M. Borg, and R. Jabangwe, "Challenges of adopting SAFe in the banking industry – a study two years after its introduction," vol. 419 LNBIP, pp. 157–171. ISBN: 9783030780975.
- [14] O. Uludag, M. Kleehaus, X. Xu, and F. Matthes, "Investigating the role of architects in scaling agile frameworks," in *2017 IEEE 21st International Enterprise Distributed Object Computing Conference (EDOC)*, pp. 123–132, IEEE.
- [15] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review," vol. 119, pp. 87–108.
- [16] K. Petersen and C. Wohlin, "A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case," vol. 82, no. 9, pp. 1479–1490.
- [17] S. Coyle and K. Conboy, "A case study of risk management in agile systems development,"
- [18] M. Mousaei and T. Javdani, "A new project risk management model based on scrum framework and prince2 methodology," vol. 9, no. 4.
- [19] J. Nyfjord and M. Kajko-Mattsson, "Communicating risk information in agile and traditional environments," in *33rd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO 2007)*, pp. 401–408, IEEE.
- [20] W. van Grembergen, "COBIT 5 FOR IT RISK MANAGEMENT."
- [21] C. of Sponsoring Organizations of the Treadway Commission, "Enterprise risk management - integrated framework."
- [22] "COBIT5 enabling processes,"
- [23] Dean Leffingwell, "SAFe 6.0 framework."
- [24] T. O. Group, "An overview of TOGAF® version 9.,"
- [25] M. Lankhorst and H. Proper, "Agile architecture," in *Enterp. Eng. Ser.*, pp. 41–57, Springer. Journal Abbreviation: Enterp. Eng. Ser.
- [26] R. Wieringa, *Design science methodology: For information systems and software engineering*. Design Science Methodology: For Information Systems and Software Engineering. Pages: 332.
- [27] P. Gomber, J.-A. Koch, and M. Siering, "Digital finance and FinTech: current research and future research directions," vol. 87, no. 5, pp. 537–580.
- [28] "Commission recommendation of 6 may 2003 concerning the definition of micro, small and medium-sized enterprises (text with EEA relevance) (notified under document number c(2003) 1422)." Code Number: 124 Code: OJ L Legislative Body: COM.
- [29] H. Davies and M. Zhivitskaya, "Three lines of defence: A robust organising framework, or just lines in the sand?," vol. 9, pp. 34–42. _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/1758-5899.12568>.
- [30] COSO, "Enterprise risk management integrating with strategy and performance,"
- [31] COSO, "Committee of sponsoring organizations of the treadway commission enterprise risk management - integrated framework."
- [32] R. R. Moeller, "COSO enterprise risk management_ establishing effective governance, risk, and compliance processes."
- [33] COSO, "Committee of sponsoring organizations of the treadway commission enterprise risk management - integrating with strategy and performance."
- [34] ISACA, *COBIT | Control Objectives for Information Technologies*.
- [35] U. Department, "Implementing the five key internal controls."
- [36] K. Beck and Agile Alliance, "Manifesto for agile

- software development.”
- [37] J. A. Zachman, “A framework for information systems architecture,” vol. 26, no. 3, pp. 276–292.
- [38] Y. I. Alzoubi and A. Mishra, “Enterprise architecture contribution in distributed agile software development,” p. sys.21739.
- [39] A. Gromoff, N. Kazantsev, and J. B. (Stavenko), “An approach to knowledge management in construction service – oriented architecture,” vol. 96, pp. 1179–1185.
- [40] S. Hanschke, J. Ernsting, and H. Kuchen, “Integrating agile software development and enterprise architecture management,” in *2015 48th Hawaii International Conference on System Sciences*, pp. 4099–4108, IEEE.
- [41] E. R. Poort, “Driving agile architecting with cost and risk,” vol. 31, no. 5, pp. 20–23. Conference Name: IEEE Software.
- [42] “The TOGAF® standard, 10th edition (90-day evaluation license).”
- [43] J. F. Wolfswinkel, E. Furtmueller, and C. P. M. Wilderom, “Using grounded theory as a method for rigorously reviewing literature,” vol. 22, no. 1, pp. 45–55. Publisher: Taylor & Francis eprint: <https://doi.org/10.1057/ejis.2011.51>.
- [44] A. Liberati, D. G. Altman, J. Tetzlaff, C. Mulrow, P. C. Gøtzsche, J. P. A. Ioannidis, M. Clarke, P. J. Devereaux, J. Kleijnen, and D. Moher, “The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: explanation and elaboration,” vol. 62, no. 10, pp. e1–e34. Publisher: Elsevier.
- [45] O. Plant, J. van Hillegersberg, and A. Aldea, “Rethinking IT governance: Designing a framework for mitigating risk and fostering internal control in a DevOps environment,” vol. 45. Publisher: Elsevier Inc.
- [46] W. Cram and M. Brohman, “Controlling information systems development: A new typology for an evolving field,” vol. 23, no. 2, pp. 137–154.
- [47] M. Aamir and M. Khan, “Incorporating quality control activities in scrum in relation to the concept of test backlog,” vol. 42, no. 7, pp. 1051–1061. Publisher: Springer India.
- [48] I. Revolution, *IT Revolution An unlikely union: DevOps and audit - Information security and compliance practices. Technical Report. DevOps Enterprise Forum.*
- [49] B. Farroha and D. Farroha, “A framework for managing mission needs, compliance, and trust in the DevOps environment,” pp. 288–293.
- [50] P. Nielsen, T. Winkler, and J. Nørbjerg, “Closing the IT development-operations gap: The DevOps knowledge sharing framework,”
- [51] A. Robinson, “Continuous security: Implementing the critical controls in a DevOps environment. technical report. SANS institute.”
- [52] J. Morales, H. Yasar, and A. Volkman, “Implementing DevOps practices in highly regulated environments,” vol. Part F147763.
- [53] J. Michener and A. Clager, “Mitigating an oxymoron: Compliance in a DevOps environments,” vol. 1, pp. 396–398.
- [54] I. DeLuccia, “DevOps audit defense toolkit. technical report. IT revolution.”
- [55] V. Mohan, L. Ben Othmane, and A. Kres, “BP: Security concerns and best practices for automation of software deployment processes: An industrial case study,” pp. 21–28.
- [56] A. Rahman and L. Williams, “Software security in DevOps: Synthesizing practitioners’ perceptions and practices,” pp. 70–76.
- [57] P. Frijns, R. Bierwolf, and T. Zijderhand, “Reframing security in contemporary software development life cycle,” pp. 230–236.
- [58] V. Mohan and L. Ben Othmane, “SecDevOps: Is it a marketing buzzword? mapping research on security in DevOps,” pp. 542–547.
- [59] D. Shackelford, “A DevSecOps playbook,”
- [60] M. Lie, M. Sanchez-Gordon, and R. Colomo-Palacios, “DevOps in an ISO 13485 regulated environment: A multivocal literature review,”
- [61] M. Muñoz and O. Díaz, “DevOps: Foundations and its utilization in data center,” pp. 205–225.
- [62] A. Wiedemann, “IT governance mechanisms for devops teams – how incumbent companies achieve competitive advantages,” vol. 2018-January, pp. 4931–4940.
- [63] R. Bierwolf, P. Frijns, and P. Van Kemenade, “Project management in a dynamic environment: Balancing stakeholders,” vol. 2017-January, pp. 1–6.
- [64] P. Frijns, F. Van Leeuwen, and R. Bierwolf, “Risk management - from risk log to risk dialogue,” vol. 2018-January, pp. 699–703.
- [65] T. Laukkarinen, K. Kuusinen, and T. Mikkonen, “Regulated software meets DevOps,” vol. 97, pp. 176–178.
- [66] M. Abrahams and J. Langerman, “Compliance at velocity within a devops environment,” pp. 94–101.
- [67] T. L. Saaty, “How to make a decision: The analytic hierarchy process,” vol. 48, no. 1, pp. 9–26.
- [68] H. Jonkers, H. A. Proper, M. M. Lankhorst, D. Quartel, and M. E. Iacob, “ArchiMate® for integrated modelling throughout the architecture development and implementation cycle,” Publisher: IEEE.

12. APPENDIX

12.1 Controls Applied in Controlled Environment

The following figures illustrate the implementation of control activities within the SharePoint environment, supporting control activities CA1 through CA11.

Figure 18 presents the SharePoint list used to store all solution architecture documentation files including attributes. Figure 19 displays the overview of the Solution Architecture Documentation dashboard. This dashboard provides a high-level view of all documented solutions, visualizing key metrics such as the quality maturity levels across all solutions and the status of documentation. Lastly, Figure 20 zooms in on the solution details level within the dashboard. This figure shows the detailed view of individual solutions, where specific attributes like peer review status, CIA classification, and quality maturity levels are displayed. The quality maturity levels are visualized at both the aggregate level (across all solutions) and the individual solution level, providing comprehensive insights into the documentation quality.

Solution Intents		Solution Resourcepool	Agile Release Train(s)	Architect	Status Solution Intent	Collegiale review feedback	Collegiale review datum	PI einddatum
Open Solution Intent	Gegevens Bijwerken >	SolutionName 1	Responsible ART(s)	Responsible Architect	Goedgekeurd	Heldere visie en context. Modellen in orde. Specificaties helder. Tests sluiten goed aan op specificaties.	17-10-2024	27-11-2024
Open Solution Intent	Gegevens Bijwerken >	SolutionName 2	Responsible ART(s)	Responsible Architect	Concept	Modellen up-to-date	5-9-2024	3-12-2024
Open Solution Intent	Gegevens Bijwerken >	SolutionName 3	Responsible ART(s)	Responsible Architect	Goedgekeurd	Alles in orde	13-12-2024	3-1-2025
Open Solution Intent	Gegevens Bijwerken >	SolutionName 4	Responsible ART(s)	Responsible Architect	Onder review	Ziet er goed uit! Modellen in orde.	8-11-2024	14-11-2024
Open Solution Intent	Gegevens Bijwerken >	SolutionName 5	Responsible ART(s)	Responsible Architect	Algekeurd	Ketenplaat ontbreekt	22-11-2024	4-1-2025

Figure 18. Control activities SharePoint list

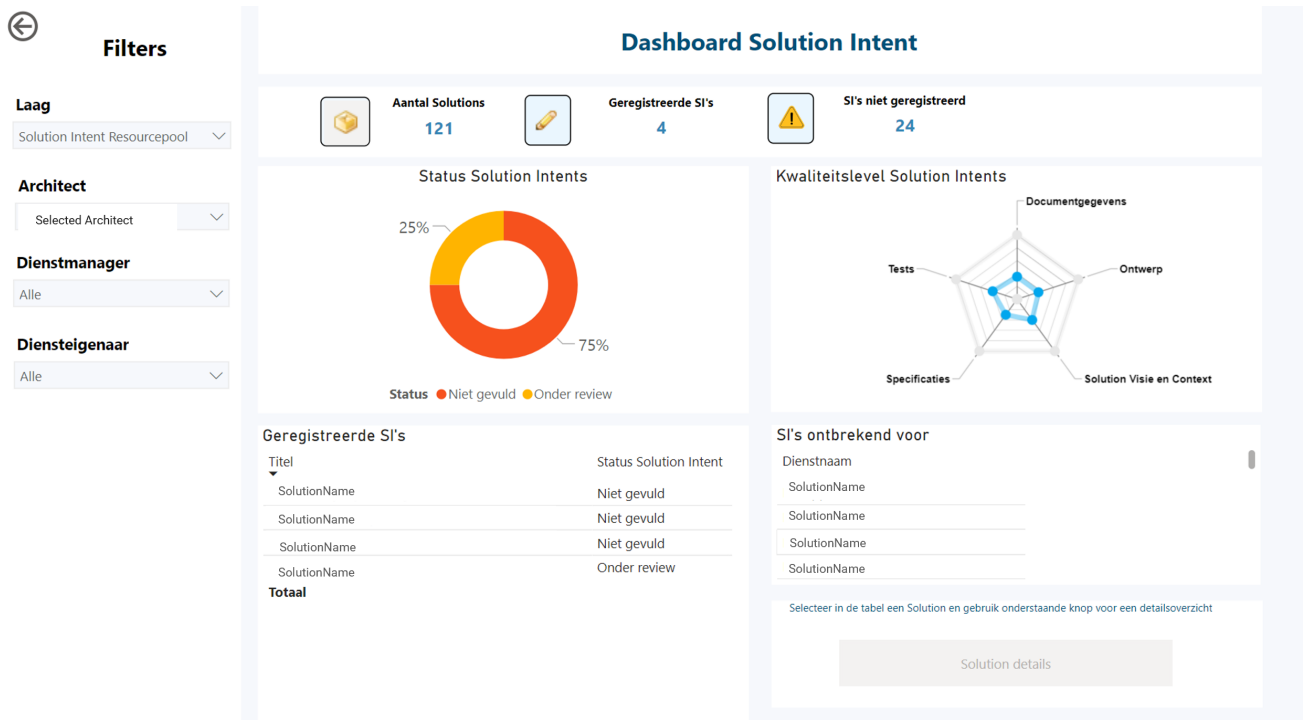


Figure 19. Dashboard Overview Solution Architecture Documentation

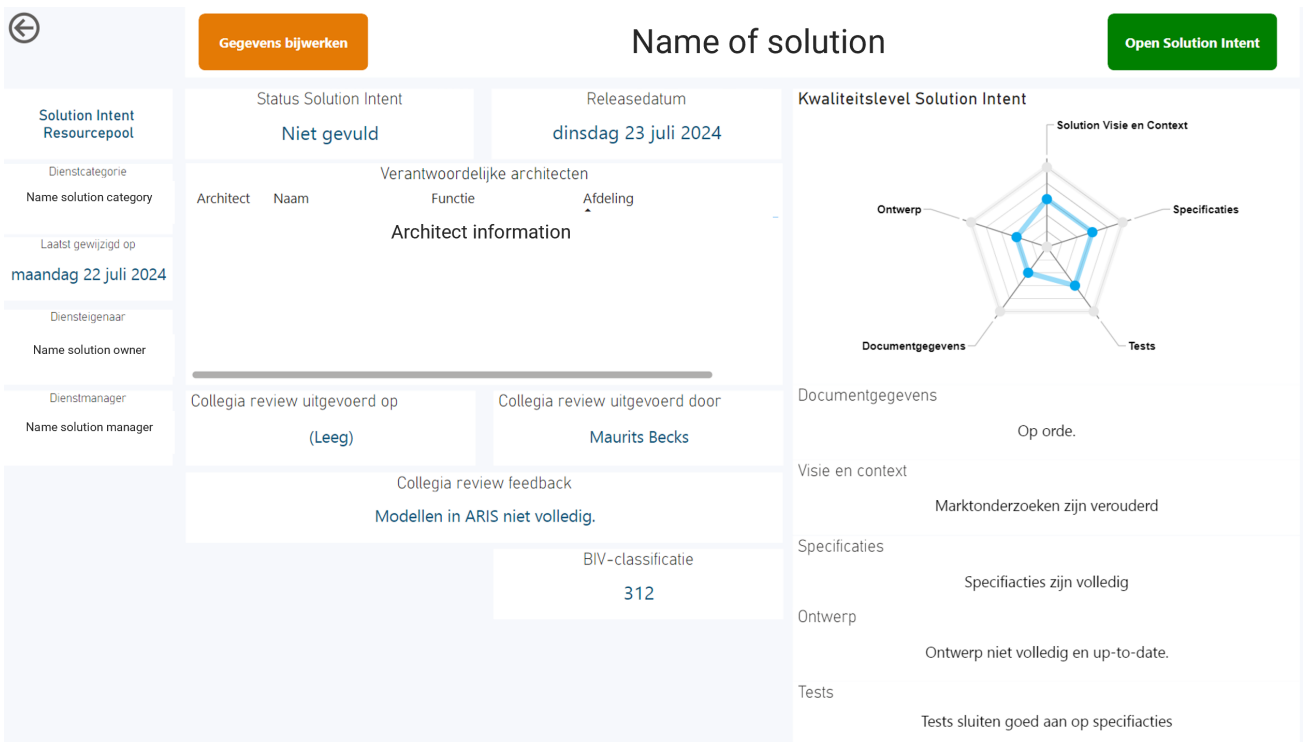


Figure 20. Dashboard Overview Solution Architecture Documentation Details