Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Governing Cybersecurity in Digital Business Ecosystems: A Maturity Model Integrating Enforceable Minimum Security Baselines

Nuno Seixas

naseixas@dei.uc.pt

University of Coimbra

José D'Abruzzo Pereira

University of Coimbra

João R. Campos

University of Coimbra

Marco Vieira

University of North Carolina at Charlotte

Paulo R. da Cunha

University of Coimbra

Research Article

Keywords: Cybersecurity governance, Maturity model, Digital business ecosystems, Minimum security baseline, Design science research, Information security management

Posted Date: November 2nd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10933744/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Governing Cybersecurity in Digital Business Ecosystems: A Maturity Model Integrating Enforceable Minimum Security Baselines

Authors

Nuno Seixas (corresponding author) University of Coimbra, CISUC/LASI, Department of Informatics Engineering (DEI), Coimbra, Portugal Email: naseixas@dei.uc.pt ORCID: 0000-0002-3744-5746

José D'Abruzzo Pereira University of Coimbra, CISUC/LASI, Department of Informatics Engineering (DEI), Coimbra, Portugal Email: josep@dei.uc.pt

João R. Campos University of Coimbra, CISUC/LASI, Department of Informatics Engineering (DEI), Coimbra, Portugal Email: jrcampos@dei.uc.pt

Marco Vieira University of North Carolina at Charlotte, Charlotte, NC, USA Email: marco.vieira@charlotte.edu

Paulo R. da Cunha University of Coimbra, CISUC/LASI, Department of Informatics Engineering (DEI), Coimbra, Portugal Email: rupino@dei.uc.pt

Author biographies

Nuno Seixas is a researcher at the Centre for Informatics and Systems of the University of Coimbra (CISUC), Portugal, where he is pursuing his PhD. His research interests are in software engineering, maturity models, and software engineering team optimization. He is also an Invited Assistant Professor at the Department of Informatics Engineering of the University of Coimbra, teaching software engineering. He has over 20 years of experience in the software engineering industry, spanning healthcare informatics, telecommunications, and other domains, and is a Certified Instructor for CMMI-DEV.

José D'Abruzzo Pereira holds a PhD in Informatics Engineering (Architectures, Networks and Cybersecurity) from the University of Coimbra and is an Assistant Professor there and a member of the Software and Systems Engineering (SSE) group at CISUC. His research interests include security and vulnerability detection, static code analysis, software project management, databases, software quality, cloud computing, and self-adaptive systems. He holds an MSc in Information Technology and Software Engineering from the University of Coimbra and Carnegie Mellon University and a BSc in Computer Science from the State University of Campinas (Unicamp), Brazil.

João R. Campos holds a PhD in Informatics Engineering from the University of Coimbra, where he is an Assistant Professor, a researcher in the SSE group of CISUC, and co-coordinator of its Cybersecurity Laboratory (CS-Lab). His research interests include intelligent dependability and security, failure prediction, the safety and trustworthiness of

AI-based and LLM-enabled systems, dependable and secure software development, software testing, vulnerability detection, and automated patching.

Marco Vieira is a Professor of Computer Science at the University of North Carolina at Charlotte. He earned his PhD in Informatics Engineering from the University of Coimbra. His research focuses on dependability and security assessment, fault injection, and software testing, with over 250 publications. He chairs IFIP WG 10.4 on Dependable Computing and Fault Tolerance, serves as Associate Editor for IEEE TDSC, and holds leadership roles in DSN, ISSRE, and LADC.

Paulo R. da Cunha is an Associate Professor of Information Systems (with Habilitation) and former head of the IS Group at the Faculty of Science and Technology of the University of Coimbra. He holds a PhD (2001) in Informatics Engineering from the University of Coimbra, has been Adjunct Associate Teaching Professor at Carnegie Mellon University, and Visiting Associate at Brunel University, UK. He serves on several editorial boards and has published more than 160 papers, including in ICIS, ECIS, AMCIS, HICSS, and ISD, and in journals such as Electronic Markets, Business & Information Systems Engineering, Requirements Engineering, and the Journal of Enterprise Information Management.

Governing Cybersecurity in Digital Business Ecosystems: A Maturity Model Integrating Enforceable Minimum Security Baselines

Abstract

Maturity models are widely used instruments for assessing and guiding organizational capability, yet they assume a single organization advancing along a single-stage path. That assumption is difficult to reconcile with cybersecurity governance in digital business ecosystems, where capabilities develop unevenly across autonomous organizations, and an enforceable baseline of protection must coexist with progressive capability growth. Drawing on design science research, we present a maturity model that integrates Minimum Security Baselines (MSBs) as binding, measurable requirements that every participant must meet, directly into a multi-dimensional maturity structure. We evaluate the model in a seaport ecosystem pilot of 35 transport and logistics partners, combining a six-week field assessment with an expert evaluation using an adapted System Usability Scale. Practitioners judged the model usable, learnable, and adaptable. We position the model against established cybersecurity maturity models and recent maturity-and-resilience frameworks, and show that none couples an enforceable participation baseline with a configurable, multi-dimensional progression at ecosystem scale. The contribution is a governance mechanism that pairs an enforceable participation baseline with configurable, non-linear improvement trajectories across heterogeneous ecosystem members.

Keywords: Cybersecurity governance; Maturity model; Digital business ecosystems; Minimum security baseline; Design science research; Information security management

1 Introduction

Maturity models have become one of the most widely adopted instruments for assessing and improving organizational capabilities in information systems and software engineering [1]. Their appeal rests on a straightforward logic: complex organizational development can be decomposed into discrete stages, each representing a more advanced set of practices, thereby enabling benchmarking and structured improvement [2]. Empirical studies have reinforced this view, linking maturity advancement to reduced cost overruns and improved software quality [3]. However, a growing body of research questions whether staged progression remains viable in environments where capabilities emerge across organizational boundaries, technologies shift continuously, and regulatory demands evolve faster than centralized model updates can accommodate [4, 5, 6, 7]. Recent research makes the same point specifically for cybersecurity: studying resource-constrained organizations, Haj Mohammadi et al. [8] find that cybersecurity capability develops through non-linear, path-dependent learning rather than through the staged progression that compliance-focused models such as C2M2 [9] and the NIST [10] /NICE [11] frameworks assume.

These concerns are especially pressing in cybersecurity. Protection in digital business ecosystems depends on coordinated action across organizations with uneven capabilities, different governance structures, and divergent regulatory obligations. In settings such as transport and logistics platforms, where federated applications and shared infrastructure create mutual dependencies, cybersecurity is a prerequisite for participation rather than an internal technical concern. Information systems research has long recognized that security in inter-organizational settings is a systemic property rather than the sum of individual postures: breaches propagate across shared knowledge networks [12], security investments are interdependent and require incentive alignment across participants [13], and shared technology platforms can amplify a single vulnerability across an entire population of connected actors [14]. Traditional maturity models, built for single-organization contexts with stable boundaries, offer limited guidance in this context [5]. At the ecosystem level, cybersecurity governance requires both a mandatory baseline of protection that all participants must meet and a structured path toward progressive capability growth, two requirements that existing models do not jointly address.

This paper uses and evaluates a maturity model architecture that integrates Minimum Security Baselines (MSBs) into its structure and applies the resulting model in the project NEXUS, a large-scale, multi-actor ecosystem [15]. The architecture builds on Mettler's [6] design principles for dynamic and evolvable models and Wulf et al.'s [7] configurational insights, and it supports two extension mechanisms: adding new maturity levels and integrating new capabilities across domains. Our starting point is a next-generation maturity model architecture proposed in prior work [16]; the present study advances that architecture by operationalizing it for ecosystem-scale cybersecurity governance. We integrate MSBs, defined as binding and measurable security requirements that establish an enforceable minimum posture for ecosystem participation, into the maturity structure. This integration allows baseline controls and progressive capability development to function together rather than as disconnected efforts.

We evaluate the integrated model in a seaport ecosystem pilot conducted as part of a digital transformation initiative involving 35 transport and logistics partners [15]. The setting suits this evaluation because capabilities develop across autonomous organizations at different paces and under different constraints. We classify MSBs into security domains and subdomains, then map each MSB to maturity-level guidance along three analytical dimensions: scope, knowledge level, and degree of management. These dimensions distinguish the extent to which a practice is adopted (scope), the depth of teams' understanding of it (knowledge level), and the degree to which the organization plans, monitors, and enforces it (degree of management). Following design science research [17, 18] and Becker et al.'s [19] maturity model development framework, the empirical work involved two distinct activities: (i) a six-week field assessment of the consortium's central platform, conducted by a team of cybersecurity consultants and resulting in a structured audit deliverable; and (ii) an expert evaluation of the integrated maturity model, conducted with four practitioners using an adapted System Usability Scale (SUS). The evaluation focused on two requirements: a theoretically grounded maturity model and configurable improvement trajectories that overcome linear, one-dimensional progression [5, 7].

This work makes three contributions to cybersecurity research and practice. First, it extends design principles for next-generation maturity model evolution: extensibility, multi-dimensional characterization, and reflective evolution, to the context of ecosystem-scale cybersecurity governance. We state the architecture together with the structural conditions under which it is expected to apply (Section 3.2), positioning it as generalizable to ecosystems that share those conditions rather than claiming it has been observed to hold across multiple organizations. Second, it introduces an integrated MSB–maturity mechanism that links enforceable security baselines to progressive capability growth: MSBs serve as a participation gate that every member must meet to access shared infrastructure, while the multi-dimensional maturity structure governs how members improve beyond that floor at their own pace. We position this contribution explicitly beyond existing cybersecurity compliance-focused maturity models such as C2M2 [9], the NIST CSF implementation tiers [10], OWASP SAMM [20], SSE-CMM [21], and CMMC [22], and beyond the recent cybersecurity maturity and resilience frameworks reviewed in Section 2.5, none of which couples an enforceable participation baseline with a configurable, multi-dimensional progression at ecosystem scale. Third, it reports empirical results from a seaport pilot in which experienced practitioners rated the model as usable, learnable, and adaptable through a structured expert evaluation, results that are directionally consistent with an earlier evaluation of the same architecture in a structurally distinct setting.

The remainder of the paper is organized as follows. Section 2 reviews related work on the evolution, successes, and critiques of maturity models; the cybersecurity-specific models against which we position our contribution; the information systems literature on security governance in digital business ecosystems; and recent cybersecurity maturity and resilience frameworks. Section 3 details the research context and design while introducing the conceptual mechanism linking MSBs to maturity-based capability development. Section 4 presents the integrated model and reports its application to the consortium's central

platform. Section 5 presents the evaluation and results. Section 6 discusses implications for research and practice. Section 7 concludes with limitations and future work.

2 Theoretical Background and Related Work

This section reviews the evolution of maturity models and their application to cybersecurity, organized around five themes: (2.1) the conceptual foundations of maturity models; (2.2) their empirical record and adoption drivers; (2.3) critiques that expose their limitations; (2.4) design responses toward next-generation models; and (2.5) cybersecurity-specific models, recent maturity and resilience frameworks, and the ecosystem gap. Section 2.6 then situates the contribution within the literature on security governance in digital business ecosystems.

2.1 Conceptual foundations and early development

Andersen et al. [2] conceptualize maturity as being "complete, perfect, or ready," drawing on a biological metaphor in which growth unfolds through identifiable stages toward an ultimate form. This metaphor is seductive because it promises a clear developmental arc: assess where you stand, follow the prescribed stages, and arrive at the destination. But it also embeds a strong assumption that there is a single destination and that all organizations move toward it along the same path. Much of the critique surveyed in Section 2.3 traces back to this assumption.

The metaphor was first operationalized in quality management. Crosby's [23] Quality Management Maturity Grid and Nolan's [24] staged theory of information systems growth established the core idea: organizations could map complex socio-technical phenomena onto ordered developmental levels that combined diagnostic value with prescriptive guidance [6]. The Capability Maturity Model (CMM) and its successor, Capability Maturity Model Integration (CMMI) [25], became the most influential instantiations in software engineering, codifying maturity as a process discipline characterized by repeatability, predictability, and managerial control. The ISO/IEC 33000 family (SPICE) [26] and COBIT [27] extended these concepts to process assessment, IT service management, and corporate governance. Despite their differences in scope, all these frameworks shared a process-centric, staged view of organizational growth.

2.2 Adoption drivers and empirical evidence

Maturity models have become widely adopted in research and practice [1], driven by three factors. First, in regulated industries such as defense and aerospace, CMMI appraisals served as mandatory signals for supplier qualification, lending institutional legitimacy [25]. Second, the structured levels provided managers with a practical vocabulary for identifying capability gaps and prioritizing improvements [28]. Third, empirical studies reported tangible benefits: Kandt [3] documented reductions in cost overruns and schedule deviations in flight software development, as well as improvements in quality and customer satisfaction. More recent work has correlated digital-transformation maturity with financial performance, reporting profitability increases of 9% to 26% and market-

valuation premiums around 12%, though these figures come from cross-sectional studies that do not establish causation and vary substantially by industry and sample [1].

These results helped consolidate the belief that structured capability development leads to competitive advantage. They also help explain why maturity models have diffused beyond software engineering into domains like cybersecurity, where several specialized instruments have emerged. C2M2 [9] provides Maturity Indicator Levels per practice in the energy sector; the NIST Cybersecurity Framework [10] offers four-tier descriptions of cybersecurity risk management practices; OWASP SAMM [20] supplies three maturity levels per practice for software assurance; the SSE-CMM [21] adapts the CMM five-level structure to systems security engineering; and CMMC [22] establishes staged levels for defense supply chain compliance. Each introduced staged or tiered capability assessments in its target domain. However, they all inherit the same structural assumptions as their predecessors: staged progression, single-organization scope, and update cycles centralized by the model creator. They thus face the same conceptual challenges discussed next.

2.3 Critiques and conceptual limitations

Researchers have questioned the theoretical robustness, methodological rigidity, and contextual sensitivity of traditional maturity models from several angles. As the following paragraphs show, many of these critiques trace back to the single-destination, single-path assumption embedded in the biological metaphor introduced in Section 2.1.

Regarding theoretical grounding, Mettler [6] argues that many maturity models aggregate best practices heuristically rather than ground their levels in validated theory, thereby limiting their explanatory power and adaptability. Bach [4] raised similar concerns earlier, criticizing CMM's narrow focus on processes and its tendency to marginalize human, cultural, and contextual factors.

On practical feasibility, early adoption data tell a pointed story. During the initial diffusion of CMM across U.S. defense contractors in the late 1980s and early 1990s, roughly 73% of assessed organizations remained stuck at Level 1 [1]. Researchers attributed this stagnation to procedural complexity and bureaucratic overhead that imposed costs on project teams without proportional benefits.

Regarding structural assumptions, Wulf et al. [7] argue that organizational development rarely follows a uniform sequence, since capability growth often emerges through iterative, path-dependent processes shaped by local constraints. Forsgren et al. [5] push this further, noting that staged models assume stable contexts and underestimate environmental volatility. By implying a final state of maturity, these models risk treating maturity as an endpoint rather than a continuing capability. Empirical studies reinforce the point: organizations have achieved comparable performance outcomes through diverse practice configurations that diverge from prescribed maturity sequences [3, 7], while others that closely followed model prescriptions did not obtain the expected improvements. This concern is not merely conceptual. In a recent qualitative study, Haj Mohammadi et al. [8] examined how micro and small enterprises develop cybersecurity capability and found that it emerges through experiential learning, social-capital mobilization, and values-aligned

adaptation rather than through the predefined maturity stages that models such as C2M2 [9] and the NICE [11] framework assume; they argue that stage-based logic inadequately captures the non-linear, path-dependent character of capability development in resource-constrained settings.

Beyond structure, the evolutionary mechanisms of maturity models also attract criticism. Mettler's [6] "reflect evolution" design step highlights that most models evolve through centralized, infrequent, creator-driven updates that favor standardization over contextual fit. In fast-changing environments, these slow revision cycles leave models increasingly misaligned with organizational realities [5].

2.4 Design responses: toward next-generation maturity models

Several authors have proposed design principles to address existing shortcomings. Mettler [6] calls for dynamic, evolvable models that adapt through empirical feedback and user-driven updates rather than functioning as static artifacts. Wulf et al. [7] argue that model architectures should accept that organizations reach comparable performance through different improvement strategies, breaking with the assumption that staged progression is the only valid path. Recent information systems work continues this trajectory: Breiter et al. [29], for example, develop a capability maturity model for twin (digital and sustainability) transformation using design science research, structuring 45 dynamic capabilities across six dimensions and identifying multiple pathways rather than a single sequence, evidence that multi-dimensional, path-sensitive maturity models are an active concern in the information systems field.

Building on Mettler's and Wulf et al.'s recommendations, prior work [16] proposes a maturity model architecture organized around four structural elements. First, *domains* group related capability areas (for example, in cybersecurity: governance, protection, defense, resilience). Second, within each domain, *capabilities* describe the high-level skills that an organization develops. Third, each capability is realized through a set of *practices*, as observable and assessable activities. Fourth, each practice is aligned with one or more *maturity levels*, which describe progressively more advanced ways of implementing that practice.

The defining feature of this architecture is that maturity levels are not characterized by a single composite descriptor. Instead, each level is described along multiple analytical dimensions, and the specific dimensions can be selected to fit the application context. The architecture accommodates any number of dimensions, and the same practice can sit at different levels across them simultaneously. This makes it possible to represent organizations whose capability profiles are uneven: strong in one aspect of a practice and weak in another, without forcing them onto a single linear scale.

The architecture also supports two explicit extension mechanisms. First, new maturity levels can be added without restructuring existing domains, because guidance elements are anchored to the dimensions rather than to the levels themselves. Second, new capabilities can be integrated into any domain without disrupting the rest of the model, enabling it to absorb emerging requirements as they arise. Together, these mechanisms operationalize

the dynamic and evolvable design principles called for by Mettler [6] and the configurational flexibility argued for by Wulf et al [7].

This architecture was designed and validated in single-organization settings [16]. It did not address the broader challenge of ecosystem-scale governance, where a mandatory protection baseline must coexist with progressive capability growth across organizations with heterogeneous starting points, nor did it integrate enforceable minimum-security requirements into the maturity structure itself. These gaps motivate the present study.

2.5 Cybersecurity maturity models and the ecosystem gap

No existing cybersecurity maturity model meets both requirements identified above: an enforceable participation baseline and a configurable, multi-dimensional progression structure. We review the five most widely cited models.

CMMC [22], developed for the U.S. defense supply chain, organizes cybersecurity practices into staged levels and ties certification at a given level to eligibility for defense contracts. The certification mechanism makes CMMC's baseline genuinely enforceable: organizations that do not certify cannot participate. The progression structure, however, is a single-staged ladder with one composite descriptor per level, which forecloses the heterogeneous improvement paths that ecosystem participants typically follow. Updates are infrequent and centrally controlled through formal federal rulemaking.

C2M2 [9] assigns each practice a Maturity Indicator Level (MIL 0–3), allowing different practices to sit at different MILs within the same assessment. This per-practice granularity is a step toward configurability, but C2M2 is delivered as a self-assessment instrument with no enforceable baseline for participation, and it offers no mechanism to coordinate assessments across organizations that share infrastructure.

OWASP SAMM [20] provides three maturity levels for each of its software assurance practices, again allowing progression per practice. SAMM is community-driven and revised through periodic public releases that address some of the standardization-versus-fit tension affecting more centralized models. It was designed for single-organization software assurance programs; however, it does not address ecosystem-level governance or participation conditions.

SSE-CMM [21] adapts the CMMI's five capability levels to systems security engineering. It inherits CMMI's strengths, clear progression semantics and mature assessment methods, but also its limitations: a single descriptor per level, a staged path, and a slow standards-body revision cycle. Like SAMM, it targets organizations rather than ecosystems.

NIST CSF [10] describes four tiers of cybersecurity risk management posture, intended to characterize an organization's overall approach rather than to drive per-practice improvement. The tiers are widely used as a vocabulary for self-description and benchmarking, but they neither prescribe an enforceable baseline nor support per-practice configurability.

These five models are the most widely cited, but they are not the whole landscape, and recent systematic reviews of cybersecurity maturity models reach the same conclusion

from which this paper starts. A steady stream of recent work confirms both the demand for cybersecurity maturity models and the specific gap this paper addresses. Büyüközkan and Güler [30] combine a systematic literature review with a bibliometric science-mapping analysis and propose a consolidated cybersecurity maturity model of five dimensions, fifteen factors, and five levels, demonstrated on three companies. Their synthesis is valuable evidence of what the field looks like in aggregate, and it confirms two persistent features: the models it consolidates remain single-organization and staged (a linear five-level ladder), and none pairs an enforceable participation requirement with a configurable progression. The proposed model advances consolidation and analytical grounding; it does not target ecosystem governance or an enforceable baseline.

Papachristofis et al. [31] develop a modular cyber-resistivity maturity-and-scoring framework (CRMSF) spanning seven domains, three security grades (Baseline, Advanced, Maximal), and per-grade maturity levels, with weighting, complexity, and criticality attributes. CRMSF is genuinely configurable by allowing an organization to hold different grades in different domains, and it defines a "Baseline" grade that every organization should attain. Two differences are decisive for the present contribution. First, CRMSF's baseline is a self-assessed minimum grade recommended to a single organization, not an externally audited condition that gates participation in a shared ecosystem; CRMSF is explicitly a self-assessment instrument. Second, by the authors' own statement, CRMSF "does not include empirical validation" and rests on conceptual synthesis, with a pilot planned as future work. The present work supplies both the enforceable, ecosystem-gating baseline that CRMSF's self-assessed grade does not, and a field application that CRMSF has yet to undertake.

Closest to our setting, Tabansky and Lichterman [32] present PROGRESS, a sectoral cyber-capability maturity model that operates explicitly above the firm. Grounded in complex-systems theory, it maps four Dimensions of Operation (key entities, sector supervision, IT/OT supply chain, and national cybersecurity) against five practice domains to produce twenty maturity-level scores, and it has been applied through focus-group assessments in a dozen countries. PROGRESS and our work share the premise that resilience is a property of interacting organizations rather than of individual firms, and PROGRESS is the more broadly validated of the two. They differ in function. PROGRESS is a diagnostic instrument: it produces multi-dimensional scores to explain why systemic gaps persist, such as regulatory arbitrage, coordination failure, and labor scarcity, and to guide policymakers and regulators, and it deliberately imposes no enforceable baseline. Our contribution is an operational governance instrument for a platform-mediated consortium: an enforceable, auditable minimum that gates access to shared infrastructure, coupled to a per-practice progression that members traverse non-linearly. The two are complementary, a sector-level diagnostic and an ecosystem-level participation mechanism, and neither subsumes the other. A related governance-oriented proposal, Ababneh et al. [33], integrates legal and ethical dimensions into a cybersecurity governance model, but likewise does not couple governance to a measurable, enforceable maturity progression.

Table 1 summarizes how these models, together with the recent frameworks discussed above, relate to four properties: an enforceable participation baseline, a configurable multi-dimensional progression structure (both introduced above), an ecosystem or multi-actor

scope, and whether the model was empirically evaluated in its introducing work. The architecture proposed in this paper is the only entry that combines an enforceable baseline (via MSBs as a participation requirement) with a configurable, multi-dimensional progression (via the level-by-dimension matrix described in Section 4) at ecosystem scope, and reports a field evaluation. Each existing model or recent framework addresses only a subset of these properties.

Table 1 - Positioning against existing and recent cybersecurity maturity models and frameworks.

Model / framework	Enforceable baseline	Configurable multi-dimensional progression	Ecosystem / multi-actor scope	Empirical evaluation
NIST CSF tiers	No	No	No	n/a (standard)
SSE-CMM	No	No	No	n/a (standard)
OWASP SAMM	No	Partial	No	n/a (standard)
C2M2	No	Partial	No	n/a (standard)
CMMC	Yes (certification)	No	Partial	n/a (standard)
Büyüközkan & Güler (2025)	No	Partial	No	3 companies
CRMSF (2025)	No (self-assessed)	Yes	No	None (conceptual)
PROGRESS (2024)	No	Yes	Yes	Multi-country
MSB–Maturity (this work)	Yes (auditable gate)	Yes	Yes	Field + expert evaluation

2.6 Security governance in digital business ecosystems

The paper's contribution is at the ecosystem level, so it engages a stream of research on information security across, rather than within, organizational boundaries. Three observations from that literature motivate the design.

First, security in inter-organizational information systems is interdependent: one participant's posture affects the others. Soper et al. [12] model this directly, proposing an inter-organizational knowledge-sharing security model that detects and monitors breach propagation across a business network, an early recognition that the relevant unit of protection is the network, not the firm. Fang et al. [13] formalize the incentive side of the same problem, showing that interdependent security risks in inter-organizational

information systems require mechanisms. In their case, a Security Compliance Consortium with signaling and screening, to align participants on a common protection level. The MSB mechanism proposed here can be read as a governance-instrument counterpart to that economic argument: where Fang et al. use incentives to induce compliance, MSBs make a defined minimum a non-negotiable condition of participation and render it auditable. Security-domain work reinforces the point: reviews of cyber supply-chain risk management show that a participant's exposure is largely inherited from its suppliers and integrators and cannot be managed by any single firm in isolation [34].

Second, shared platforms concentrate risk. Brennecke et al. [14] show that when smart devices reuse shared technology platforms, a single vulnerability can propagate across an entire population of connected products, and they derive governance implications spanning individual, organizational, and regulatory levels. Our setting exhibits the same structure: a shared Open Data Collaboration Platform on which consortium members depend, and the MSB participation gate is precisely a mechanism for governing the platform-level risk that Brennecke et al. characterize. Dependency-aware risk models make this propagation explicit: Faraji et al. [35] quantify how risk cascades across interdependent critical infrastructures through linkage strength and operational importance, the very dynamic an enforceable participation baseline is intended to contain.

Third, security governance for managers increasingly means reconciling multiple overlapping standards into a usable instrument. Herath et al. [36] integrate five governance and control frameworks (COBIT, SABSA, ISG, ITIL, and ISO 27000) into an information security balanced scorecard, developed through design science and validated with an expert panel and a case application. Our MSB catalog performs an analogous integration by reconciling requirements from ISO, ENISA, NIST, OWASP, CIS, and other bodies into a single auditable set. Then, it embeds the result in a maturity structure and extends it from a single-organization condition to an ecosystem participation condition.

Positioned against this literature, the present work is distinct in three respects: it targets the ecosystem rather than the firm; it couples an enforceable baseline with a configurable, multi-dimensional maturity progression rather than treating compliance and improvement as separate instruments; and it delivers a design artifact evaluated in a field setting rather than an analytical model or a measurement framework alone.

3 Research Context and Design

This section describes the research context in which the integrated MSB–maturity model was developed and applied (Section 3.1), specifies the structural conditions under which the architecture is expected to generalize (Section 3.2), introduces the conceptual mechanism linking Minimum Security Baselines to maturity-based capability development (Section 3.3), and states the research method (Section 3.4).

3.1 The NEXUS ecosystem

The work took place within project NEXUS [15], a digital transformation initiative in the transport, logistics, and multimodal mobility sector. NEXUS brings together a consortium of 35 partners: port authorities, terminal operators, carriers, logistics providers, technology suppliers, and research institutions. It centers on an Open Data Collaboration Platform that integrates federated applications, smart hardware, and digital twins from different consortium members. The platform supports cross-organizational data sharing and offers a portfolio of digital products and services that target decarbonization and operational efficiency in port and logistics operations.

For the purposes of this paper, what matters about NEXUS is not the breadth of its product portfolio but its structure as a multi-actor ecosystem. Capability development in NEXUS does not occur within a single organizational boundary. Consortium members from different sectors develop their cybersecurity practices at varying paces, under different regulatory obligations, and from diverse starting points. Improvements emerge asynchronously across organizations rather than through a centrally coordinated rollout. These conditions are precisely what traditional maturity models, designed for single-organization use, struggle to accommodate, making NEXUS a relevant testbed for the architecture this paper develops.

In this setting, cybersecurity is treated as a prerequisite for participation rather than an internal concern for any individual partner. Because consortium members share infrastructure and exchange data through a platform, each participant's security posture affects the others. Governance, therefore, needs both a baseline of protection that all participants must meet and a structured path toward progressive improvement.

3.2 Generalization of the ecosystem structure

The architectural argument of this paper does not depend on the specifics of the maritime and logistics setting. We specify the empirical circumstances under which the architecture is expected to hold, rather than claiming that it has been observed to hold in each case. In presenting these conditions, we follow Lee and Baskerville's [37] account of generalizing from an empirical description to a theory: a contribution established in one setting can be stated as a theory, with the claim that settings sharing the specified conditions allow practitioners to test it. What the architecture requires structurally is: (i) multiple autonomous organizations sharing infrastructure or data through which their security postures interact; (ii) heterogeneous starting capabilities and regulatory exposures across those organizations; and (iii) a need to combine an enforceable participation baseline with progressive, organization-specific improvement. Several other settings exhibit the same structure, for example:

- National healthcare data exchanges, where hospitals, primary care providers, laboratories, and insurers share patient data through a federated infrastructure. Postures vary widely between large hospital networks and small clinics, and participation is conditional on minimum security guarantees.

- Financial open banking ecosystems, in which banks and licensed third-party providers exchange data via regulated APIs (e.g., PSD2 in Europe). Participation requires baseline controls such as strong customer authentication and secure communication, while individual providers' broader maturity varies.
- Public-sector shared service platforms, where central government, agencies, and municipalities share infrastructure for services such as national identity and e-procurement. Capability heterogeneity is large, and participation depends on baseline compliance with national cybersecurity frameworks.

What is specific to NEXUS is the maritime and logistics regulatory context, the integration of IT and operational technology into port operations, and the inclusion of digital twins among the shared assets. The architectural choices reported here, with the MSB integration on multi-dimensional levels and configurable trajectories, are general; the threshold calibration, the specific MSBs, and the choice of three dimensions are NEXUS-specific.

3.3 Minimum Security Baselines as a bridge between strategy and execution

Translating a cybersecurity strategy into coordinated practice among heterogeneous actors is not something policy documents alone can achieve. High-level frameworks such as the NIST Cybersecurity Framework [10] and the ISO/IEC 27000 family [38] define intent and align participants on shared concepts, but they leave significant room for interpretation in implementation. At the other extreme, isolated technical controls produce coherence within individual organizations but cannot ensure traceability or comparable protection at the ecosystem level.

Maturity models partially address this gap. They provide a structured way to assess current cybersecurity practices, identify gaps, and plan improvement over time, accommodating organizations with different starting points. What maturity models cannot do, on their own, is enforce a consistent baseline of protection across an ecosystem of participants. By design, a maturity model recognizes that organizations are at different levels, some of which may be inadequate for participation. In a setting like NEXUS, where one participant's weak posture can compromise the others, this flexibility becomes a liability unless it is paired with non-negotiable minimum requirements.

Minimum Security Baselines (MSBs) provide that complement. An MSB is a binding, measurable security requirement that all consortium members must meet to access and operate within the NEXUS platform. MSBs cover concrete practice areas, including encryption standards, authentication, secure configuration, and incident reporting. Unlike general guidelines, MSBs are auditable: they specify what must be in place and provide enough detail for verification by audit and validation services within the consortium. For example, an MSB on encryption of data in transit specifies the minimum required protocol version and documented cipher-suite restrictions, which an auditor can verify against deployed configurations; an MSB on incident reporting specifies a maximum reporting window and a defined recipient, both directly verifiable against incident logs.

The integration of MSBs into the maturity model then serves as the conceptual bridge between strategy and execution. MSBs anchor the minimum number of participants below

which no participant can fall, while maturity levels describe how organizations can grow beyond it. The next section details how this integration is realized and how the resulting model was applied in the NEXUS pilot.

3.4 Research method

The study follows the design science research (DSR) methodology, whose purpose is to build and evaluate purposeful artifacts that address a class of problems [17, 18]. The artifact here is the integrated MSB–maturity model; the class of problems is cybersecurity governance in digital business ecosystems that must combine an enforceable participation baseline with heterogeneous, non-linear capability development. We structured artifact construction using Becker et al.'s [19] maturity model development framework, which prescribes systematic problem identification, iterative design, evaluation, and formalization. We adopted a qualitative expert evaluation approach, consistent with the evaluation strategies recognized in Design Science Research. Venable et al. [39] classify expert opinion under "artificial evaluation" within their FEDS framework and treat it as appropriate for early-stage artifacts where the goal is to gather informed feedback before broader empirical exposure. Prat et al. [40] similarly cataloged informed expert evaluation as a recognized evaluation method for IS design artifacts.

Evaluation combined two activities, consistent with a formative-to-summative progression. First, a field application demonstrated the artifact in use: over six weeks, a team of cybersecurity consultants applied the integrated model to assess the organization responsible for the ecosystem's central platform, producing a structured audit deliverable (Section 4.2). Second, an artificial, expert-based evaluation elicited informed practitioner judgment on the artifact's coherence and adaptability using an adapted System Usability Scale [41, 42], with subgroup analysis reported in Section 5. This design privileges depth of informed judgment and demonstrated utility in a real setting over statistical generalization, an appropriate posture for an artifact whose intended contribution is a theory with stated boundary conditions rather than a population estimate [37].

4 The Integrated MSB–Maturity Model and Its Application

This section presents the integrated MSB–maturity model and its application to NEXUS. Section 4.1 defines the MSB structure, describes how the MSB catalog was derived, and explains how the MSBs were mapped to the maturity-level dimensions. Section 4.2 reports the application of the resulting model to the assessment of the platform's developing organization, including the structured audit deliverable produced.

4.1 Defining the Minimum Security Baselines

Each MSB is characterized by a fixed set of attributes: a sequential identifier (MSB Number), a brief descriptive label (MSB Name), a type (preventive, corrective, or detective), a stated purpose, the requirement(s) needed for compliance, summary guidance to support implementation, and additional information pointing to source documents and references. MSBs were derived from a range of sources, including standards, frameworks,

guidelines, existing maturity models, and regulations. These sources come from organizations with different mandates: some focus on cybersecurity (e.g., the European Union Agency for Cybersecurity, ENISA), while others address general standardization (e.g., the International Organization for Standardization, ISO). As a result, the level of detail varies across sources, and some overlap among them is unavoidable. Overlap was reconciled during MSB definition by deduplicating requirements that addressed the same control objective, retaining the variant with the most operationally specific phrasing, and recording cross-references to the alternative formulations. Table 2 lists the entities.

Table 2 - Entities used as sources for MSBs.

Acronym	Full name	Online address
NIST	National Institute of Standards and Technology	https://www.nist.gov
ISO	International Organization for Standardization	https://www.iso.org
OWASP	Open Worldwide Application Security Project	https://owasp.org
CIS	Center for Internet Security	https://www.cisecurity.org
CSA	Cloud Security Alliance	https://cloudsecurityalliance.org
IIoTC	Industry IoT Consortium	https://www.iiconsortium.org
W3C	World Wide Web Consortium	https://www.w3.org
EU Data	European Data Portal	https://data.europa.eu/en
IoTSF	IoT Security Foundation	https://iotsecurityfoundation.org
OTA	Internet Society: Online Trust Alliance	https://www.internetsociety.org/ota
ENISA	European Union Agency for Cybersecurity	https://www.enisa.europa.eu
DoE	U.S. Department of Energy	https://www.energy.gov
DoD	U.S. Department of Defense	https://www.defense.gov
ETSI	European Telecommunications Standards Institute	https://www.etsi.org
CISA	Cybersecurity & Infrastructure Security Agency	https://www.cisa.gov

We extracted candidate requirements from these sources and, where relevant, transformed them into MSBs. In some cases, the wording of an MSB closely follows its source; for example, *MSBP014 — Identify and manage security-by-design components and services* draws on several sources [38, 43, 44] and uses a shared vocabulary across all MSBs in the catalog. Each defined MSB was then assigned to one of four security domains: Governance and Ecosystem, Protection, Defense, or Resilience. These four domains align with the cybersecurity domain taxonomy used in European cybersecurity governance and are

already familiar to NEXUS partners from prior compliance work. Table 3 reports the number of identified MSBs by domain and subdomain.

Table 3 - Number of identified MSBs by security domain and subdomain.

Security domain	Security subdomain	Number of identified MSBs
Governance and Ecosystem	1.1 Security Governance & Risk Management	15
	1.2 Ecosystem Management	1
Protection	2.1 IT Security Architecture	8
	2.2 IT Security Administration	17
	2.3 Identity and Access Management	9
	2.4 Security in Maintenance, Reuse or Disposal	3
	2.5 Physical and Environmental Security	8
	2.6 Systems Secure Development	8
	2.7 Secure Systems Testing	2
Defense	3.1 Detection	5
	3.2 Computer Security Incident Management	4
Resilience	4.1 Continuity of Operations	1
	4.2 Crisis Management	1

The catalog is heaviest in the Protection domain, particularly IT Security Administration (17 MSBs), reflecting the operational density of administrative controls relative to higher-level governance or resilience requirements. The Resilience subdomains contain few MSBs because the requirements at this level are general and we consider that a continuity-of-operations capability either exists or does not, rather than being decomposable into many fine-grained controls.

The practices associated with each MSB were then aligned with the maturity-model levels by mapping each MSB to the maturity-level dimensions. For the NEXUS application, the consortium agreed that each maturity level should be characterized along three analytical dimensions: scope, knowledge level, and degree of management. These three dimensions were selected because they separate concerns that a single composite level conflates: how widely a practice is adopted (scope), how well the team understands it (knowledge level), and how systematically it is planned, monitored, and enforced (degree of management). This separation is what allows the assessment to expose the uneven profiles: strong on one dimension and weak on another, which motivates the multi-dimensional architecture. The minimum threshold for each dimension is as follows:

- **Scope:** practices should be consistently implemented at the team level.

- **Knowledge level:** the team should possess a proficient level of knowledge to apply the practices in day-to-day operations.
- **Degree of management:** practices should be executed with a managed perspective, meaning work is planned, monitored, and goals are set.

The threshold values reflect the consortium's calibration of a defensible ecosystem-wide minimum, given members' heterogeneity. "Team level" was chosen for scope because individual-level adoption is too brittle for shared infrastructure, while organization-wide adoption was judged unrealistic to mandate as a participation precondition for smaller members. "Proficient" knowledge, as the ability to apply a practice without external supervision in routine cases, was selected over "advanced" for the same reason: advanced expertise is a goal, not a floor. "Managed" perspective was selected because unmanaged execution provides no audit traceability, and NEXUS participation requires auditability. These thresholds are properties of the NEXUS agreement, not of the architecture itself; a different ecosystem could adopt different thresholds along the same dimensions, and Section 5 reports a field-assessment finding suggesting that refinements to this calibration may be warranted for some MSBs.

To illustrate how MSBs align with maturity-level guidance, we present two worked examples from different security domains.

Example 1: MSBP014 — Identify and manage security-by-design components and services (Protection — 2.6 Systems Secure Development):

- **Level 1:** Technical solutions for security and security-by-design needs are defined on a case-by-case basis.
- **Level 2:** Teams have internal definitions for implementing the security requirements. These definitions are consistently used within the team.
- **Level 3:** The organization defines and shares an inventory of components and techniques used to implement security features. The use of these components is mandatory and enforced organization-wide.

Example 2: MSBP090 — Information security incident assessment and decision-making (Defense — 3.2 Computer Security Incident Management):

- **Level 1:** Incidents are handled reactively when noticed. No predefined escalation, role assignments, or communication channels exist.
- **Level 2:** Teams maintain documented incident-handling procedures and a defined on-call rotation. Incidents are logged in a shared tracker, but post-incident review is informal.
- **Level 3:** The organization operates a CSIRT or equivalent function with documented playbooks, defined SLAs for triage, structured post-incident reviews feeding back into preventive controls, and exercises (tabletops or red-teaming) at least annually.

The two examples illustrate that the level-guidance structure adapts across different practice types: artifact-bound practices (security-by-design components) and operational practices (incident response). In both cases, the level progression captures increasing

organizational integration, from ad hoc to team-consistent to organization-wide, without dictating a single implementation path. Each organization can choose a target level based on its current state and improvement goals; as with any organizational change, building a supportive culture takes time, so progress is expected to be incremental rather than abrupt. A sample of identified MSBs from each security domain is provided in Appendix 1.

Figure 1 shows the resulting architecture obtained by applying the maturity model to this context.

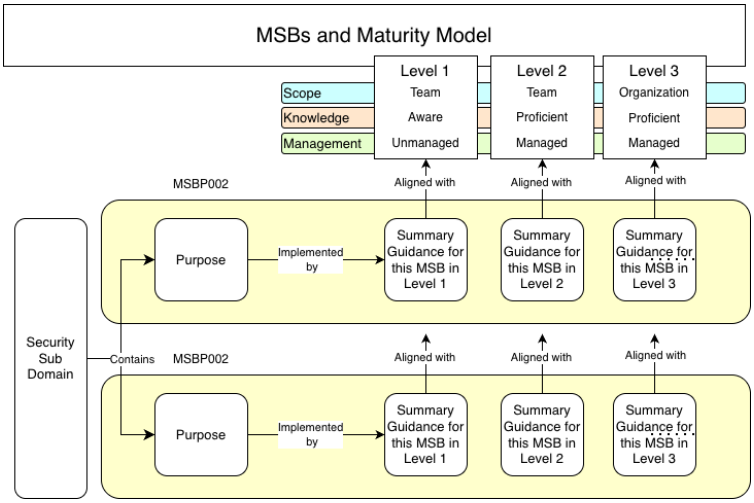


Figure 1 - MSBs integrated with the maturity model architecture.

The architecture in Figure 1 integrates MSBs directly into the maturity model structure, applying the design principles for dynamic and evolvable models discussed by Mettler [6] and the configurational insights of Wulf et al. [7]. Each security subdomain contains capabilities expressed as MSBs, and each MSB is decomposed into purpose-driven guidance aligned with maturity levels, which are characterized by combinations of the three dimensions: scope, knowledge level, and degree of management. This structure realizes the two extension mechanisms of the base architecture. First, additional maturity levels can be added without restructuring existing domains, because the guidance elements remain anchored to the level-defining dimensions. Second, new capabilities (MSBs) can be integrated into any subdomain without disrupting the overall model, enabling it to absorb emerging security requirements as they arise.

The integration also operationalizes Mettler's [6] "reflect evolution" idea. MSBs are not static compliance artifacts; they are intended to be revised in response to empirical validation, audit feedback, and practical experience gained from field application. The alignment between MSBs and maturity levels gives domain experts a structured way to update guidance, refine practices, and recalibrate the level-defining dimensions as conditions change. The MSBs and the maturity model are designed to evolve together rather than as separate artifacts maintained on separate cycles.

Finally, the architecture supports non-linear capability configurations. Different subdomains can sit at different maturity levels for the same organization, and different MSBs within a subdomain can be at different levels. For example, a consortium member

may reach organization-level maturity in incident response while still operating at team-level maturity in secure development. The three-dimensional characterization of each level captures this unevenness directly: an organization can be advanced in scope but underdeveloped in knowledge or degree of management for the same practice. This is a deliberate design choice that responds to Wulf et al.'s [7] argument that organizational improvement does not follow a uniform trajectory.

4.2 Applying the model in the Open Data Collaboration Platform

We began applying the integrated maturity model by assessing the organization responsible for the cybersecurity of the Open Data Collaboration Platform, as this platform is central to the NEXUS initiative.

The pilot proceeded in three stages. In the first, the maturity model with MSBs (Figure 1) was systematically evaluated over six weeks through interviews and evidence collection by a team of cybersecurity consultants from a consortium technology provider. The interviews involved four members of the assessed organization, drawn from technical and management roles, who provided process definitions, supporting documentation, and evidence of execution where available. In the second stage, the findings were evaluated against the three analytical dimensions of maturity levels (scope, knowledge level, and degree of management), and the assessed organization's maturity in each subdomain was classified as below baseline, aligned with baseline, or above baseline. In the third stage, the coded findings were analyzed to identify maturity gaps.

The application revealed an uneven maturity profile across security subdomains. Table 4 reports the percentage of MSBs in each state (below, aligned, or above the NEXUS-defined baseline) for each subdomain, broken down by the three maturity-level dimensions. Cell shading encodes the direction and intensity of each result relative to the NEXUS-defined baseline: cells in the *Below* columns are shaded on a red scale (deeper red indicates a larger share of MSBs falling short of the baseline, i.e. a deficit), while cells in the *Aligned* and *Above* columns are shaded on a green scale (deeper green indicates a larger share meeting or exceeding the baseline, an achievement). Shading is normalized within each dimension so that color intensity is comparable across subdomains for the same dimension; a cell with no measured MSBs in a state is left unshaded at 0%.

Security Subdomain	Scope			Knowledge Level			Degree of Management		
	Below	Aligned	Above	Below	Aligned	Above	Below	Aligned	Above
Part 1.1 - Security Governance & Risk Management	0%	57%	43%	7%	93%	0%	36%	64%	0%
Part 1.2 - Ecosystem Management	0%	100%	0%	100%	0%	0%	100%	0%	0%
Part 2.1 - IT Security Architecture	0%	100%	0%	25%	75%	0%	88%	13%	0%
Part 2.2 - Security Administration	0%	29%	71%	24%	76%	0%	53%	47%	0%
Part 2.3 - Identity and Access Management	0%	56%	44%	22%	78%	0%	22%	78%	0%
Part 2.4 - Security In Maintenance, Reuse or Disposal	0%	33%	67%	0%	100%	0%	0%	100%	0%
Part 2.5 - Physical and Environmental Security	0%	0%	100%	0%	100%	0%	13%	88%	0%
Part 2.6 - Systems Secure Development	0%	100%	0%	100%	0%	0%	100%	0%	0%
Part 2.7 - Secure Systems Testing	0%	100%	0%	100%	0%	0%	100%	0%	0%
Part 3.1 - Detection	0%	60%	40%	60%	40%	0%	80%	20%	0%
Part 3.2 - Computer Security Incident Management	0%	0%	100%	0%	100%	0%	0%	100%	0%
Part 4.1 - Continuity of Operations	0%	0%	100%	0%	100%	0%	0%	100%	0%
Part 4.2 - Crisis Management	0%	0%	100%	0%	100%	0%	0%	100%	0%

Table 4 - Maturity-level dimension results by subdomain (percentage of MSBs below, aligned with, or above the NEXUS-defined baseline for each dimension).

The small absolute MSB counts in some subdomains (notably Ecosystem Management, with one MSB, and the Resilience subdomains, each with one MSB) should be read as qualitative indicators of placement rather than as statistical measures of the underlying capability distribution.

The evaluated organization showed consistent alignment in the scope dimension, indicating that practices were established and communicated at the team level. Performance in the knowledge-level dimension was weaker: gaps were observed in the depth of understanding required to apply certain MSBs effectively. The degree-of-management dimension scored well below baseline, with no structured planning, monitoring, or goal-setting for MSB execution across several subdomains.

Within the subdomain of Security Governance and Risk Management, the organization had formal practices in place, including managing data center assets via a ServiceDesk application. However, no documented procedures existed for maintaining asset information specific to the assessed project, and information security requirements were not fully integrated into the project management process. Scope was aligned, but management maturity remained below baseline.

Results for Ecosystem Management followed a similar pattern. Policies for supplier evaluation existed and were linked to ISO 27001 [38] compliance requirements, but they were not tailored to the specific products used in the project. The scope dimension was

aligned, but knowledge and management were both deficient regarding the security implications of suppliers' products in this context.

In the IT Security Architecture assessment, more substantial weaknesses were exposed. Early risk assessments did not consistently include considerations of data ownership, privacy, and the exposure of personally identifiable information. The network topology placed development, staging, and production environments within the same VLAN, contrary to recommended practice for environment segregation. These gaps reflected both a knowledge shortfall in security architecture principles and the absence of systematic management mechanisms to enforce secure configurations.

Secure software development was one of the most critical areas of deficiency. Static application security testing (SAST) was integrated into the CI/CD pipeline, but practices such as threat modeling, security use-case definition, and system hardening were absent from earlier stages of the system development life cycle (SDLC). Reliance on non-authenticated APIs created foreseeable risks. The audit identified several high- and critical-severity CVEs in widely used components of the deployed software stack, along with specific version-upgrade guidance, suggesting that vulnerability management was reactive rather than proactive. Practices existed within scope but were underdeveloped in both knowledge and managerial enforcement.

In Secure Systems Testing, testing was limited to unit and integration levels, with no systematic penetration testing, which limited the organization's ability to validate solutions before deployment. In Detection, logging mechanisms were operational but incomplete; in particular, container-orchestration logging was not consistently configured, leaving gaps in event detection and evidence collection.

Taken together, these results show an uneven maturity profile: alignment in declarative practices (scope), partial alignment in knowledge (with gaps in application and awareness), and systematic weaknesses in managerial oversight (degree of management). This is exactly the kind of differentiated picture that the multi-dimensional architecture is designed to capture, and it would be obscured by a single composite maturity score.

The application produced a structured assessment deliverable: a detailed report containing the maturity ratings shown in Table 4, an identification of 79 applicable MSBs out of the catalog of 82 (three MSBs were determined to fall outside the assessed scope), and seven prioritized improvement recommendations spanning policy guidance, asset management, system-health monitoring, secure-design practices, vulnerability testing, secure-systems principles, and cryptographic key management. The recommendations included specific technical findings — six high- or critical-severity CVEs in the deployed software stack, each paired with version-upgrade guidance — alongside structural findings such as the absence of environment separation in the production VLAN and the lack of an asset inventory for the project. The report was delivered with explicit prioritization guidance: high-impact, low-cost actions could be addressed first, with more resource-intensive structural changes deferred. This prioritization was the audit team's own cost-and-impact judgment and was not derived from the model; the model's contribution at this stage was the differentiated, dimension-resolved maturity picture used as diagnosis. Whether and how the

recommendations are ultimately acted on falls outside the scope of this paper's empirical claims; what the application demonstrates is that the integrated model produced concrete, actionable output of the kind a practitioner audit would generate, and that the receiving organization engaged with that output through evidence provision, technical interviews, and acceptance of the prioritized remediation list. At this point, we demonstrate the instrument in the field on a single organization, which an ordinary maturity model could also assess. The ecosystem-level contribution lies not in this assessment but in the MSBs functioning as an enforceable participation baseline that gates access to shared infrastructure (Section 3.3) and in the architecture's capacity to accommodate the heterogeneous, organization-specific trajectories that ecosystem members follow above that baseline.

5 Evaluation and Results

This section reports the evaluation conducted within the NEXUS ecosystem. The evaluation targets two requirements: theoretical grounding of the maturity model, and support for configurable, non-linear improvement trajectories as advocated by Forsgren et al. [5] and Wulf et al. [7].

5.1 Evaluation design and instrument

The evaluation was conducted within a multi-actor setting involving technology producers, technology consumers, and shared digital infrastructure spanning IT and OT components. This heterogeneity made the setting a demanding testbed for both the model's conceptual coherence and its configurational flexibility.

We adopted a qualitative expert evaluation approach, consistent with evaluation strategies recognized in design science research. Venable et al. [39] classify expert opinion under "artificial evaluation" within their FEDS framework and treat it as appropriate for early-stage artifacts where the goal is to gather informed feedback before broader empirical exposure. Prat et al. [40] similarly catalog informed expert evaluation as a recognized evaluation method for IS design artifacts. In the maturity model literature specifically, small-panel expert evaluations have been used in design science contexts where the goal is depth of informed judgment rather than statistical inference [45].

The choice of a small panel also reflects a position on what kind of generalization this evaluation supports. As set out in Section 3.2, the target of generalization is the architecture as a theory with stated boundary conditions, not a population parameter. The expert evaluation accordingly corroborates, through informed practitioner judgment, the model's usability and adaptability; it is not a sample-based measurement. We are explicit throughout Section 5.2 about which statements the four-person panel can and cannot support.

We used the System Usability Scale (SUS) [42] as a structured elicitation instrument. The SUS consists of 10 items rated on a 5-point Likert scale and yields a composite score reflecting perceived usability [41]. We supplemented the SUS items with a

recommendation question (on a 0–10 scale) to capture overall endorsement, and we adapted the item wording to fit the maturity-model context. Because the panel comprised two role profiles with different relationships to the model, we used two parallel questionnaire versions: a consultant version, framed around portability across diverse client engagements, and a practitioner version, framed around application within the practitioner's own software development context. The two versions occupy the same SUS item positions and address the same underlying constructs (adaptability, complexity, learnability, internal coherence, and confidence). The adapted questionnaires are reproduced in Appendix 2.

The empirical work involved two distinct activities. **Field assessment:** the integrated model was applied by two cybersecurity consultants who had participated in the model's design, working as the assessment team within the consortium's audit work package. Over six weeks in mid-2025, they conducted a structured assessment of the organization responsible for the Open Data Collaboration Platform, drawing on workfile-based evidence collection, interviews with four members of the assessed organization, and a technical review of the deployed software stack. The assessment produced the report described in Section 4.2. **Expert evaluation of the integrated model:** four practitioners completed the adapted SUS. Two were the consultants who had designed and applied the model (P1, P2); the other two were software development professionals from the consortium's platform development context with no involvement in the model's design (P3, P4).

The four evaluators represented two senior career profiles. The consultants were both over 55 years of age, with 10–20 and over 20 years of process-improvement consultancy experience in software development. The software development professionals were both aged 30–40 and had 10–20 years of software development experience in implementation roles. The panel therefore spans process consultancy and engineering practice, with substantial experience in both.

We acknowledge that two of the four evaluators are the same individuals who designed and applied the model. This is a constraint on the strength of the evidence: the SUS results from P1 and P2 cannot be characterized as design-naive. We address this constraint analytically by reporting ratings separately for the two subgroups in Section 5.2 and comparing them at the item level. The split allows convergence between the design-involved and design-naive subgroups to function as a partial internal control on familiarity effects. We did not attempt to measure longitudinal performance outcomes; organizational improvement depends on structural, procedural, and cultural factors that unfold over years, and isolating the causal contribution of any single instrument is difficult in field settings.

5.2 Results

Table 5 reports the composite SUS scores and the recommendation question; Table 6 breaks down responses by item, separating the two consultants who participated in the model's design and field application from the two software development professionals who had no involvement in design.

Table 5 - Composite SUS scores and evaluator profiles.

Participant	Role	Age band	Years of experience	SUS score	Rec. (0–10)
P1	Consultant	>55	10–20	80	8
P2	Consultant	>55	>20	80	8
P3	Software dev. professional	30–40	10–20	77.5	7
P4	Software dev. professional	30–40	10–20	75	8
Mean (consultants)			80	8	
Mean (developers)			76.25	7.5	
Overall mean			78.125	7.75	

Table 6 - Item-level SUS responses by subgroup.

Q	Item (summary)	Dir.	P1	P2	P3	P4	Cons.	Dev.	Gap
Q1	Would use frequently	+	4	4	4	4	4.0	4.0	0.0
Q2	Unnecessarily complex	–	2	2	2	3	2.0	2.5	0.5
Q3	Easy to adapt / use	+	5	5	4	4	5.0	4.0	1.0
Q4	Needs external support	–	2	2	3	3	2.0	3.0	1.0
Q5	Components fit together	+	4	4	4	5	4.0	4.5	0.5
Q6	Inconsistency across contexts	–	2	2	1	1	2.0	1.0	1.0
Q7	Quick to learn	+	5	4	4	4	4.5	4.0	0.5
Q8	Cumbersome to apply	–	2	1	1	2	1.5	1.5	0.0
Q9	Felt confident	+	4	4	4	4	4.0	4.0	0.0
Q10	Many concepts to learn first	–	2	2	2	2	2.0	2.0	0.0

Direction (Dir.): "+" indicates positively keyed items (higher = more favorable); "–" indicates negatively keyed items (lower = more favorable). Cons. and Dev. are the consultant and developer subgroup means; Gap is the absolute difference between them.

All four evaluators produced composite SUS scores in the favorable range. On Bangor et al.'s [41] adjective scale, scores above 68 are above average, and scores above 80.3 fall in the "excellent" range; all four scores sit between these thresholds. Given the panel size, we read these composites as a baseline indication of favorable reception rather than as a statistically interpretable result. The item-level data in Table 6 carries more analytical weight.

Across the ten SUS items, no item showed a response spread greater than one point on the 1–5 scale, and three items (Q1, Q9, Q10) elicited identical responses from all four

evaluators. This is an unusually tight convergence for a four-person panel spanning both the design-involved and design-naive subgroups. We highlight three observations.

First, the strongest single signal in the dataset is the near-unanimous disagreement that the model is cumbersome to apply (Q8: panel mean 1.5), the only negatively keyed item on which developers were rated at least as favorably as consultants. For a paper whose contribution centers on configurability and ease of adaptation, this is informative: the design-naive subgroup independently arrives at the model's adaptability claim through their own application experience, rather than inheriting it from the design context.

Second, three items show a one-point subgroup gap. On Q3 ("easy to adapt for organizations of varying sizes" / "easy to use") and Q4 ("would need significant external support"), consultants rated one point more favorably than developers. On Q6 ("too much inconsistency"), developers rated one point more favorably. The Q3 and Q4 gap is consistent with the difference in scope of judgment between the two questionnaire versions: consultants explicitly evaluated the model's portability across diverse client engagements, while developers evaluated its fit with their own context. The Q6 gap, where developers perceive less internal inconsistency, may reflect that the developers worked within a single, coherent application of the model, whereas the consultants calibrate against a broader set of usage scenarios where edge cases are more visible.

Third, the convergence holds on the items the architecture was designed to address. Q3 (adaptability) and Q7 (learnability) both stay in the favorable range across all four evaluators, with consultant means of 5.0 and 4.5 and developer means of 4.0 on both. The consultant–developer agreement on adaptability and learnability is the empirical statement that maps most directly to the configurability and non-linearity claims established in Section 4. Q3 is also the item where consultants give their unanimous maximum response (5/5), consistent with their cross-client perspective on a model designed to support cross-context application.

One small but informative pattern within the consultant subgroup deserves note. Although P1 and P2 produced identical composite SUS scores (80) and agreed exactly on eight of ten items, their two disagreements (Q7 and Q8) are in opposite directions: P1 was more optimistic about how quickly teams would learn the model, while P2 was more positive about cross-domain customization. This minor divergence indicates that the evaluators engaged with individual items rather than producing uniform endorsements, a small but real sign that the broader convergence reflects considered judgment rather than coordination. Recommendation scores (consultant mean 8.0, developer mean 7.5) indicate that all participants would endorse the model for use in professional practice.

5.3 Mapping results to requirements

The two requirements assessed here were derived in Sections 1 and 2 from established critiques of maturity models: the need for theoretical grounding [4, 6], and the need for configurable, non-linear improvement trajectories rather than uniform staged progression [5, 7].

Requirement 1: Theoretical grounding. The model was constructed through a methodological synthesis of design science research principles [17, 18] and Becker et al.'s [19] maturity model development framework, which ensured systematic problem identification, artifact design, iterative evaluation, and formalization. This synthesis directly responds to Mettler's [6] critique that many maturity models lack theoretical grounding. The expert evaluations support the practical payoff of this grounding: none of the four respondents rated the model as overly abstract or detached from operational reality, and across the panel, the items most closely related to internal coherence (Q5) and confidence in application (Q9) sit at or above 4 on the 1–5 scale. This suggests that the theoretical structure translates into operational comprehensibility for practitioners with relevant domain expertise.

Requirement 2: Configurable improvement trajectories. The seaport pilot tested this through the field assessment reported in Section 4.2, which produced the maturity-by-dimension matrix in Table 4. The architecture met this requirement by accommodating uneven maturity profiles within a single organization: the assessed organization aligned with the baseline in the scope dimension but remained below baseline in the degree-of-management dimension, and its profile varied markedly across subdomains. The three-dimensional characterization of each level enabled an assessment that captured this unevenness rather than collapsing it into a single stage label. The SUS evaluators independently rated the model favorably on adaptability and learnability (Q3 and Q7), consistent with the configurational flexibility the architecture was designed to provide. The pilot application provides the more direct evidence of this property; the SUS data triangulates the practitioner judgment behind it.

5.4 Cross-pilot comparison and internal controls

The seaport results are directionally consistent with those reported in an earlier evaluation of the same architecture [16], conducted in structurally different organizational settings. In both cases, evaluators rated the model favorably on usability and adaptability. This convergence across distinct environments is encouraging, though it must be interpreted with caution given the small number of respondents in each pilot, and we do not rest the generalization claim on it.

Three distinct forms of familiarity exposure affect this evaluation. Two of the four evaluators (P1, P2) were not only involved in designing the model, defining the three dimensions, the MSB catalog, and the baseline thresholds, but also conducted the six-week field assessment that produced the audit report. Their judgment on the model's usability is therefore shaped by an investment in both its design and its successful application. The other two evaluators (P3, P4) were design-naive: they were not involved in either the model's design or the field assessment and evaluated the integrated artifact from a user perspective.

The design-involved and design-naive split provides a partial internal control on familiarity effects. If the consultants' favorable ratings were primarily an artifact of their design-stage investment, we would expect the developers to rate the model less favorably across items. The data show partial support for this prediction for items framed around cross-context

portability (Q3, Q4), where consultants were rated more favorably than developers with one-point subgroup gaps. But on the item most central to the architecture's design claim, Q8, on whether the model is cumbersome to apply, developers rate it at least as favorably as consultants, and the developers' composite recommendation scores (7 and 8) overlap the consultants' (both 8). We do not interpret this as evidence that familiarity bias is absent; rather, on the qualities the architecture was specifically designed to deliver, the design-naive subgroup's judgment converges with that of the design-involved subgroup.

A separable form of external touchpoint exists outside the SUS panel. The four members of the assessed organization who provided evidence, sat for technical interviews, and received the audit report engaged substantively with the audit deliverable and accepted its prioritized remediation list. Their engagement is not a formal evaluation, but it provides additional evidence that the model produced output that an audited party found legible and actionable.

A further methodological observation surfaced through the field application itself. In several subdomains where the scope dimension showed alignment with the baseline (team-level practice), applying the model in a real organizational context revealed that some MSBs should have been assessed against an organization-wide baseline rather than a team-wide one, because the underlying practices only achieve their intended outcome when applied across the organization. This is not a finding about the assessed organization; it is a finding about the model's threshold calibration for the NEXUS context. It illustrates the "reflect evolution" mechanism the architecture is designed to support: a structured application produces feedback that refines the model's threshold definitions for the next iteration. We treat this as preliminary evidence that the integrated MSB–maturity instrument can co-evolve with its application context, as Mettler's [6] design principles require.

6 Discussion

The findings support two claims and qualify a third. The model is theoretically grounded in a way that practitioners find operationally comprehensible; it supports configurable, non-linear improvement trajectories that a single composite score would obscure; and its ecosystem-level governance contribution, the MSB participation gate, is established conceptually and demonstrated at the level of a single platform-operating organization rather than across the full membership. We draw out the implications of these findings for research and for practice.

6.1 Implications for research

For research on cybersecurity maturity models, the study offers a concrete design response to a long-standing critique. Where the literature has argued that staged, single-descriptor models cannot represent path-dependent, uneven development [5, 7], the multi-dimensional level structure demonstrated here shows one way to operationalize that argument in a security-governance setting and to surface the resulting unevenness in a field assessment. This directly engages a recent critique in the literature. Studying

resource-constrained organizations, Haj Mohammadi et al. [8] conclude that cybersecurity capability does not follow the linear maturity stages assumed by compliance-focused models such as C2M2 [9] or NICE [11], and instead pivot away from maturity models toward a dynamic-capabilities account. Our contribution shares their diagnosis but takes a different design stance: rather than abandoning the maturity model, we show that a multi-dimensional, extensible architecture can represent exactly the uneven, path-dependent profiles they observe. An organization can be advanced in scope yet underdeveloped in knowledge or degree of management for the same practice, while an enforceable baseline governs participation rather than developmental staging. The two accounts agree that linear staging fails and differ on the remedy, which we see as a productive point of contact between the design-oriented and dynamic-capabilities readings of cybersecurity capability. The account of generalization is deliberately theory-directed rather than sample-based [37]: we contribute an architecture together with the structural conditions under which it is expected to apply, which invites replication in the other settings that share those conditions (healthcare data exchanges, open banking, public-sector shared services) rather than a claim to have measured a population.

In security governance research within digital business ecosystems, the MSB mechanism supplements existing analytical and economic approaches. Past work on information systems has modeled how security risks and breach propagation depend on organizational boundaries [12], discussed the incentive mechanisms to align participants' protection levels [13], and examined how shared platforms distribute risk among connected actors [14]. This study introduces a governance instrument contrast: instead of incentivizing compliance or analyzing risk, it integrates a non-negotiable, auditable minimum into a maturity framework, ensuring both the baseline and improvement path are governed by a single artifact. Recent maturity and resilience frameworks highlight different roles: PROGRESS [32] offers a diagnostic, multi-actor maturity model that explains systemic gaps and guides policymakers without enforcing a baseline, while CRMSF [31] recommends a baseline grade for an organization based on self-assessment. This work positions itself between these approaches by providing an enforceable, auditable baseline that controls participation and allows customizable progression at the ecosystem level. This opens a research nexus on how enforceable baselines and adaptable progressions interact. Specifically, it reasons about whether making the baseline auditable influences how members prioritize improvements above it.

Methodologically, the study demonstrates a justifiable approach to reporting a design artifact evaluated by a small expert panel, some of whom are involved in the design. This approach involves explicitly stating the familiarity constraint and dividing the evaluators into design-involved and design-naive subgroups as a form of partial internal control. While it does not eliminate the constraint, it reveals its impact and can be adapted for other design science studies dealing with the balance between evaluator expertise and independence.

6.2 Implications for practice

For ecosystem orchestrators like platform operators, consortium governance bodies, and regulators of federated infrastructures, this model helps balance two often separate

objectives: ensuring a minimum level of protection that no participant can fall below, and enabling diverse members to progress at their own pace. The MSB catalog offers an auditable participation criterion, while the multi-dimensional maturity framework provides a diagnostic that discerns what is implemented (scope), what is understood (knowledge), and what is managed (degree of management).

For assessed organizations, the application can produce a tangible, practitioner-facing deliverable: a structured audit report with subdomain-level maturity ratings and seven prioritized recommendations, including specific technical findings paired with remediation guidance. The receiving organization engages with that output and accepts the prioritized remediation list, indicating that the model can function as a decision-support instrument rather than solely as a measurement instrument. For practitioners adopting the approach, two design choices are worth transferring: express baseline requirements in auditable, verifiable terms (a required protocol version, a maximum reporting window) rather than as general guidance, and calibrate dimension thresholds to the ecosystem's heterogeneity rather than importing a generic scale.

7 Conclusion, Limitations, and Future Work

This paper developed a maturity model that integrates Minimum Security Baselines into its structure and evaluated it in a multi-actor seaport ecosystem. The model's architecture builds on Mettler's [6] principles for dynamic and evolvable models and Wulf et al.'s [7] configurational insights, and it supports two extension mechanisms: adding new maturity levels and integrating new capabilities across domains. Our contribution was to operationalize this architecture for ecosystem-scale cybersecurity governance, where enforceable baseline controls and progressive capability development must coexist across organizations with heterogeneous starting points.

The seaport pilot provided initial evidence for two requirements. The model's theoretical grounding, rooted in a synthesis of design science research [17, 18] and Becker et al.'s [19] framework, translated into operational comprehensibility for the practitioners who evaluated it. And the architecture supported configurable, non-linear improvement trajectories: the field assessment revealed that different subdomains could sit at different maturity levels simultaneously, and the three-dimensional characterization of each level captured organizational unevenness rather than collapsing it into a single stage. The expert evaluation, conducted with four practitioners using an adapted SUS and reported separately for the design-involved and design-naive subgroups, showed convergence on the qualities the architecture was specifically designed to deliver: configurability, learnability, and ease of adaptation. The integrated MSB–maturity mechanism addresses a gap left open by existing cybersecurity maturity models (Table 1): models such as C2M2 [9], the NIST CSF tiers [10], OWASP SAMM [20], SSE-CMM [21], and CMMC [22] each provide staged or tiered assessments; recent proposals such as Büyüközkan and Güler's [30] consolidated model, CRMSF [31], and PROGRESS [32] advance consolidation, configurable scoring, and sector-level diagnosis respectively; but none combine an enforceable participation baseline with a multi-dimensional maturity structure that accommodates heterogeneous improvement paths at ecosystem scale.

Several limitations qualify these conclusions. The most significant is the scale of empirical validation. The expert evaluation involved four practitioners and should be read as structured qualitative feedback rather than statistical evidence; four respondents cannot establish population-level usability or generalizability. As noted in Section 5.1, sampling-based generalizability is not the claim being made. The panel corroborates the model as a theory, and the usability scores remain a qualitative signal rather than a population estimate. We note that this evidentiary posture is not unusual for artifact proposals in this area: comparable recent frameworks have been published on the basis of conceptual synthesis with validation deferred to future pilots [31] or demonstrated on a small number of companies [30]. This locates the field application and expert evaluation reported here within, rather than below, current practice.

A related limitation is that two of the four evaluators participated in the model's design and conducted the field assessment; their ratings cannot be considered design-naive. We addressed this analytically through a design-involved versus design-naive subgroup comparison (Sections 5.2 and 5.4), but this only makes the constraint's effects visible rather than eliminating them. Broader validation with larger, more diverse, and fully independent panels is needed.

A second limitation concerns the unit of empirical analysis. The pilot covers the assessment of the organization responsible for the Open Data Collaboration Platform, which is the architectural keystone of NEXUS, but does not extend to multiple consortium members. The platform operator was selected because it is the shared component on which the other members depend, and a six-week audit at that depth was the access the consortium's work plan permitted; assessing additional members at comparable depth was outside the resourcing available. The ecosystem-level claim that the architecture supports heterogeneous, non-linear trajectories across organizations is therefore demonstrated at the level of a single organization with heterogeneous subdomain maturity, not yet across multiple organizations with heterogeneous organizational maturity. Following Lee and Baskerville's [37] distinction between sampling-based and theory-directed generalization, a single rich field application warrants a theoretical contribution, which is, in this case, an architecture with stated boundary conditions, rather than a sample from which to infer population-level usability.

A third limitation concerns causal attribution. We evaluated perceived usability, adaptability, and the model's capacity to produce a structured assessment deliverable, not organizational outcomes. Whether the model improves cybersecurity posture over time remains untested; organizational improvement depends on structural, procedural, and cultural factors that unfold over years, and isolating the contribution of any single instrument from concurrent initiatives, regulatory pressures, and leadership decisions is difficult in field settings.

A fourth limitation concerns domain scope. The empirical work focused exclusively on cybersecurity within transport and logistics. The architecture was designed to be domain-agnostic, and Section 3.2 identified other settings where the structural conditions hold, but its behavior in those domains has not been tested. A fifth, methodological, limitation concerns threshold calibration: the field assessment revealed that the NEXUS-defined

scope threshold (team-level practice) is appropriate for some MSBs but understates the required level for others, for which organization-wide adoption is the appropriate floor. We treat this as an example of the architecture's reflective-evolution mechanism in action, but it is also a limitation of the present calibration to be addressed in subsequent iterations.

Future research will pursue four directions. First, we will expand validation across additional sectors and with independent evaluator panels, using professional networks (national CSIRTs, ISACA chapters) and a screening question to confirm the absence of prior contact with the model; comparison of external ratings against the design-naive subgroup (P3, P4) will indicate whether the favorable reception generalizes beyond NEXUS, while comparison against the design-involved subgroup (P1, P2) will indicate the magnitude of design-stage familiarity effects. Second, we will apply the architecture to other organizations across the domains identified above, namely healthcare organizations, safety-critical software producers, and software development as a service. Third, we will assess multiple consortium members to directly substantiate the cross-organizational heterogeneity claim. Fourth, we will explore how automated methods, including language models applied to regulatory and standards corpora, might support the "reflect evolution" mechanism by identifying emerging practices, flagging misalignments between MSBs and evolving threat environments, and proposing updates to maturity-level guidance. With this, we expect to reduce dependence on centralized, infrequent model updates that the literature identifies as a persistent weakness of traditional maturity models [5, 6].

Declarations

Funding: Content produced within the scope of the NEXUS "NEXUS - Pacto de Inovação - Transição Verde e Digital para Transportes, Logística e Mobilidade", financed by the Portuguese Recovery and Resilience Plan (PRR), with no. C645112083-00000059 (investment project no. 53).

Competing interests. The authors declare that they have no competing interests.

Ethics approval and consent to participate. All evaluation participants provided informed consent to take part and for their anonymized responses to be reported.

Materials and Data availability. The MSB catalog samples and adapted questionnaires are provided as appendices.

Consent for publication. The authors consent to the publication.

Author's Contributions. All authors have actively participated in the making of the presented manuscript.

References

[1] Thordsen, T., Bick, M.: A decade of digital maturity models: much ado about nothing? *Inf. Syst. e-Bus. Manag.* (2023). https://doi.org/10.1007/s10257-023-00656-w

[2] Andersen, K.N., Lee, J., Mettler, T., Moon, M.J.: Ten misunderstandings about maturity models. In: Proceedings of the 21st Annual International Conference on Digital Government Research, pp. 261–266 (2020). https://doi.org/10.1145/3396956.3396980

[3] Kandt, R.K.: Experiences in improving flight software development processes. *IEEE Softw.* 26(3), 58–64 (2009). https://doi.org/10.1109/MS.2009.66

[4] Bach, J.: The immaturity of the CMM. *Am. Program.* 7(9), 13 (1994)

[5] Forsgren, N., Humble, J., Kim, G.: Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations. IT Revolution Press, Portland (2018)

[6] Mettler, T.: Thinking in terms of design decisions when developing maturity models. *Int. J. Strateg. Decis. Sci.* 1(4), 76–87 (2010). https://doi.org/10.4018/jsds.2010100105

[7] Wulf, J., Winkler, T.J., Lowry, P.B.: Evolutional and transformational configuration strategies: a Rasch analysis of IT providers' service management capability. *J. Assoc. Inf. Syst.* 21(3), 574–606 (2020). https://doi.org/10.17705/1jais.00613

[8] Haj Mohammadi, B., Sadeghi, V.J., Sukumar, A., Mmadubuko, M.: Cybersecurity as a dynamic capability: how micro and small social enterprises build digital resilience. *Inf. Syst. Front.* (2026). https://doi.org/10.1007/s10796-026-10718-2

[9] U.S. Department of Energy: Cybersecurity Capability Maturity Model (C2M2) (2022). https://www.energy.gov/ceser/cybersecurity-capability-maturity-model-c2m2

[10] Pascoe, C., Quinn, S., Scarfone, K.: The NIST Cybersecurity Framework (CSF) 2.0. NIST CSWP 29, National Institute of Standards and Technology (2024). https://doi.org/10.6028/NIST.CSWP.29

[11] National Initiative for Cybersecurity Careers and Studies (NICCS): NICE Workforce Framework for Cybersecurity (2026). https://niccs.cisa.gov/tools/nice-framework

[12] Soper, D.S., Demirkan, H., Goul, M.: An interorganizational knowledge-sharing security model with breach propagation detection. *Inf. Syst. Front.* 9(5), 469–479 (2007). https://doi.org/10.1007/s10796-007-9055-2

[13] Fang, F., Parameswaran, M., Zhao, X., Whinston, A.B.: An economic mechanism to manage operational security risks for inter-organizational information systems. *Inf. Syst. Front.* 16(3), 399–416 (2014). https://doi.org/10.1007/s10796-012-9348-y

[14] Brennecke, M., Fridgen, G., Jöhnk, J., Radszuwill, S., Sedlmeir, J.: When your thing won't behave: security governance in the Internet of Things. *Inf. Syst. Front.* 27(4), 1471–1490 (2024). https://doi.org/10.1007/s10796-024-10511-z

[15] Nexus Consortium: Innovation Agenda for Digital and Green Transition, (2026). Online, available at: https://nexuslab.pt/

[16] Seixas, N.A., Rupino da Cunha, P., Vieira, M., Barata, J., and Ribeiro, V., "Rethinking Maturity in Next-Generation Maturity Models," in Computer, vol. 59, no. 4, pp. 60-80, April 2026, doi: 10.1109/MC.2025.3635020

[17] Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. *MIS Q.* 28(1), 75–105 (2004). https://doi.org/10.2307/25148625

[18] Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. *J. Manag. Inf. Syst.* 24(3), 45–77 (2007). https://doi.org/10.2753/MIS0742-1222240302

[19] Becker, J., Knackstedt, R., Pöppelbuß, J.: Developing maturity models for IT management. *Bus. Inf. Syst. Eng.* 1(3), 213–222 (2009). https://doi.org/10.1007/s12599-009-0044-5

[20] OWASP SAMM Project Team: OWASP SAMM: Software Assurance Maturity Model, Version 2 (2020). https://owaspsamm.org/

[21] ISO/IEC: Information technology — Security techniques — Systems Security Engineering — Capability Maturity Model (SSE-CMM). ISO/IEC 21827:2008 (2008)

[22] U.S. Department of Defense: Cybersecurity Maturity Model Certification (CMMC) Model Overview (2021)

[23] Crosby, P.B.: Quality Is Free: The Art of Making Quality Certain. McGraw-Hill, New York (1979)

[24] Nolan, R.L.: Managing the computer resource: a stage hypothesis. *Commun. ACM* 16(7), 399–405 (1973). https://doi.org/10.1145/362280.362284

[25] CMMI Institute: CMMI — Build capability, drive performance. ISACA (2025). https://cmmiinstitute.com/

[26] ISO/IEC: Information technology — Process assessment — Concepts and terminology. ISO/IEC 33001:2015 (2015)

[27] ISACA: COBIT: Control Objectives for Information Technologies (2025). https://www.isaca.org/resources/cobit

[28] de Bruin, T., Rosemann, M., Freeze, R., Kaulkarni, U.: Understanding the main phases of developing a maturity assessment model. In: Proceedings of the 16th Australasian Conference on Information Systems (ACIS) (2005)

[29] Breiter, K., Crome, C., Oberländer, A.M., Schnaak, F.: Dynamic capabilities for the twin transformation climb: a capability maturity model. *Inf. Syst. Front.* 26, 2205–2226 (2024). https://doi.org/10.1007/s10796-024-10520-y

[30] Büyüközkan, G., Güler, M.: Cybersecurity maturity model: systematic literature review and a proposed model. *Technol. Forecast. Soc. Change* 213, 123996 (2025). https://doi.org/10.1016/j.techfore.2025.123996

[31] Papachristofis, K., Vardoulias, G., Vavousis, K., Iordanou, C., Danidou, Y.: Developing cyber-resistivity maturity and scoring framework (CRMSF) for any size of organization. *Int. J. Inf. Secur.* 24(6), 242 (2025). https://doi.org/10.1007/s10207-025-01154-5

[32] Tabansky, L., Lichterman, E.: PROGRESS: the sectoral approach to cyber resilience. *Int. J. Inf. Secur.* 24, 18 (2024). https://doi.org/10.1007/s10207-024-00910-3

[33] Ababneh, J., Attar, H., Alghamdi, H., Obeidat, I., Sharrab, Y., Salameh, J.B., Ali, H. and Hafez, M.: Cybersecurity laws and ethical implications: integrated cybersecurity governance model (CLEI-ICGM). *J. Cloud Comput.* 15, 86 (2026). https://doi.org/10.1186/s13677-026-00902-9

[34] Latsiou, A., Lambrinoudakis, C.: Cyber supply chain risk management: from threats to treatment. *Int. J. Inf. Secur.* 25, 40 (2026). https://doi.org/10.1007/s10207-025-01207-9

[35] Faraji, F., Mouratidis, H., Al Zakwani, A.: Risk impact pathway analysis (RIPA): a dynamic risk assessment framework for interdependent critical infrastructures. *Int. J. Inf. Secur.* 25(1), 26 (2026). https://doi.org/10.1007/s10207-025-01201-1

[36] Herath, T.C., Herath, H.S.B., Cullum, D.: An information security performance measurement tool for senior managers: balanced scorecard integration for security governance and control frameworks. *Inf. Syst. Front.* 25(2), 681–721 (2023). https://doi.org/10.1007/s10796-022-10246-9

[37] Lee, A.S., Baskerville, R.L.: Generalizing generalizability in information systems research. *Inf. Syst. Res.* 14(3), 221–243 (2003). https://doi.org/10.1287/isre.14.3.221.16560

[38] ISO/IEC: Information security, cybersecurity and privacy protection — Information security controls. ISO/IEC 27002:2022 (2022)

[39] Venable, J., Pries-Heje, J., Baskerville, R.: FEDS: a framework for evaluation in design science research. *Eur. J. Inf. Syst.* 25(1), 77–89 (2016). https://doi.org/10.1057/ejis.2014.36

[40] Prat, N., Comyn-Wattiau, I., Akoka, J.: A taxonomy of evaluation methods for information systems artifacts. *J. Manag. Inf. Syst.* 32(3), 229–267 (2015). https://doi.org/10.1080/07421222.2015.1099390

[41] Bangor, A., Kortum, P., Miller, J.: Determining what individual SUS scores mean: adding an adjective rating scale. *J. Usability Stud.* 4(3), 114–123 (2009)

[42] Brooke, J.: SUS: a retrospective. *J. Usability Stud.* 8(2), 29–40 (2013)

[43] European Union Agency for Cybersecurity (ENISA): Guidelines for Securing the Internet of Things. ENISA (2020)

[44] OWASP Smart Contract Project Team: OWASP Smart Contract Top 10 (2025). OWASP (2025). https://owasp.org/

[45] Salah, D., Paige, R., Cairns, P.: An evaluation template for expert review of maturity models. In: Product-Focused Software Process Improvement (PROFES 2014). LNCS, vol. 8892, pp. 318–321. Springer (2014). https://doi.org/10.1007/978-3-319-13835-0_31

Appendixes

Appendix 1 — Sample of MSBs. Presents sample MSB records from each security domain, illustrating the MSB attribute structure.

I. DOMAIN: GOVERNANCE & ECOSYSTEM

Sub-domain: Part 1.1 – Security Governance & Risk Management

MSBP002 Security Roles and Duties

Recommended NG Maturity Level

Team, Proficient, Managed)

Purpose

Provide a clear, accepted and well understood structure for use by the organisation in implementing, operating, and managing information security.

Requirement(s)

Establish and assign the information security roles and responsibilities required by the organisation considering any competing and conflicting responsibilities.

Summary Guidance

Roles, duties and authority for information security should be clearly outlined, assigned and understood. Specific management positions or work functions (e.g. CEO, CISO, IT Service Manager, system owners) should be given accountability for information security. It's crucial to define who owns or belongs to which group of information assets.

When assigning roles and duties, consider that competing and conflicting responsibilities and tasks need to be segregated in order to lessen the possibility of unintended or unauthorized change or misuse of the organisation's assets. In smaller organisations, the concept should also be followed to the greatest extent practicable, especially considering the governance and management of the most valuable or at higher risk organisation’s assets.

The role allocation should be formalized and documented, for example, in a table or organisation chart.

Additional Information

Further implementation guidance may be derived from the following sources:

• ISO/IEC 27002:2022 “5.2 Information security roles and responsibilities”;

• ISO/IEC 27002:2022 “5.3 Segregation of duties”;

• NIST SP 800-82R3 “Establish OT Cybersecurity Governance”.

II. DOMAIN: PROTECTION

Sub-domain: Part 2.1 IT Security Architecture

MSBP012 Information Classification

Recommended NG Maturity Level:

Team, Proficient, Managed

Purpose

To identify and understand the protection requirements for information in accordance with its significance to the organisation.

Requirement(s)

An information classification scheme, based on the confidentiality, integrity and availability and other relevant requirements, should be used to classify information.

Summary Guidance

There should be an organisation-wide classification of information scheme, with classification levels and criteria, allowing the classification of information assets. Information asset owners are ultimately accountable for the implementation of a classification scheme.

Organisations should take a topic-specific approach, understanding the information requirements of each business unit and assessing the level of sensitivity and criticality of the information before implementing a comprehensive and consolidated information classification scheme that can integrate these requirements across the organisation. For that it should be considered:

1) The applicable legal obligations;

2) The requirements for information exchange between the parties;

3) The use of a risk-based strategy to understand the sensitivity of the information and how its disclosure to non-authorised parties might affect the organisation.

The asset owner should consider the business needs, the level of impact a breach of information might have on the organisation, and the value and sensitivity of the information when classifying it.

The classification levels and criteria should be reviewed and updated, if necessary, on a regular basis.

Consider also that each organisation has its how methods and names for classifying information, and due to these differences, there are risks of misinterpretation in the classification of information. In those cases, an agreement with those parties should be considered to safeguard the correct interpretation of the classification levels.

Additional Information

Further implementation guidance may be derived from the following sources:

• ISO/IEC 27002:2022 "5.12 Classification of information"

.

Sub-domain: Part 2.6 – Systems Secure Development

MSBP014 Security in the Development Life Cycle

Recommended Maturity Level

Team, Proficient, Managed

Purpose

To ensure that information security is integrated into the entire system development lifecycle.

Requirement(s)

Procedures to assure a secure systems development life cycle should be defined and implemented.

Summary Guidance

The creation of services, architectures, software, and systems is all covered under the systems development life cycle.

The creation of secure development environments should take into account personnel working in the environment, applied development methodologies, software and data handling processes, use of third party code and products, outsourced services, physical and network environment, and coexistence with other development and operational efforts.

The development environment that supports the specificities of the system under development (IT systems, OT systems, IoT systems, IoT devices, and ML systems), and related practices should be defined and all involved in the development activities should be informed of the defined practices.

Information security should be taken into account as early as the specification and design stages, by incorporating checkpoints into the project plan and scheduled testing. Introducing security on the early development stages of the system (principle of "shifting left") is less expensive and allows seamless incorporation of the measures, contributing to a more secure system, architecture, product, or service.

The following are guidelines should be followed in order to provide secure (and safe) software systems, products, and architectures:

1) Environments for development, testing, and production should be separated;

2) The staff involved in the development of the system should have the necessary knowledge of system and application security and receive the necessary training in order to identify and avoid security risks;

3) Secure SDLC should be considered. Information security criteria that relate to the specification and design stage of the system (IT system, OT systems, IoT devices, IoT systems and IoT applications, ML systems, and Smart Contracts) should be created and put into practice. Threat modelling and the definition of security use cases involving each system component, the system component's logic, and the way the system components interconnect, should be used in that stage (principle of "shifting left") to help developers find vulnerabilities and become aware of the security implications of their architecture design, code, and configuration decisions. A verification/test plan should be created considering the gathered information. Security testing should be conducted in the remaining stages, according to the specifics of the system under development, until the deploy stage and the results recorded and reviewed and considered in terms of the undergoing system development;

4) Information security related configurations and settings, where applicable, should be considered in the architecture, design and implementation of the system and service so that they can be consistently enforced during installation and operation. Relevant settings and configurations can include software versions and patches, open ports and network services, access restrictions to the service's functionalities and data, and more, depending on the characteristics of the OT device, IoT device or IoT service. If those settings and functionality are protected by an initial password, that password should be specific to the device or system and its initial value provided securely to the user;

5) Security and system testing such as code scanning, penetration tests, and other vulnerability tests more suitable to the system architecture (IT systems, OT systems, IoT devices, IoT systems and IoT applications, ML systems, and Smart Contracts) that is under validation should be executed;

6) Integrated development environment (IDE) tools and secure coding methodologies considering the applicable programming languages and application architecture should be used;

7) Organisations should use industry well known and maintained development frameworks, components and code, in order to reduce risks related with the supply chain;

8) Organisations should establish secure repositories for storing source code and configurations and should implement and maintain version control of that information.

Additional Information

Further implementation guidance may be derived from the following sources:

• ISO/IEC 27002:2022 “8.25 Secure development life cycle”;

• ISO/IEC 27400:2022 “Control-08 Secure development environment and procedures”;

• ISO/IEC 27400:2022 “Control-11 Verification of IoT devices and systems design”;

• ISO/IEC 27400:2022 “Control-15 Secure settings and configurations in delivery of IoT devices and services”;

• ISO/IEC 27400:2022 “Control-19 Security measures adapted to the life cycle of IoT system and services”;

• ENISA “Good Practices For Security of IOT - Secure Software Development Lifecycle”;

• ENISA Securing Machine Learning Algorithms “Security Controls - Ensure ML projects

• follow the global process for integrating security into projects”;

• OWASP “OWASP Smart Contract Top Ten – SC03:2025: Logical Errors”;

• NIST SP 800-82R3 “Build and Train a Cross-Functional Team to Implement the OT Cybersecurity Program”);

• NIST SP 800-82R3 “Vulnerability Scanning (DE.CM-8)”;

• NIST SP 800-82R3 “Configuration Change Control (Configuration Management) (PR.IP-3)”.

III. DOMAIN: DEFENSE

Sub-domain: Part 3.1 – Detection

MSBP087 Logging

Recommended NG Maturity Level:

Team, Proficient, Managed

Purpose

To generate evidence in the investigation of information security incidents, and to identify events that can lead to the confirmation of the existence of those incidents.

Requirement(s)

Activity information, exceptions, errors, and other security significant event information should be generated, logged, protected, and analysed.

Summary Guidance

Cybersecurity logging involves the creation, collection, storage, analysis, and management of log data generated by the various information systems and applications in an

organisation. Properly implemented logging can help identify and respond to security incidents, track user activities, and ensure compliance with regulations.

Monitoring and logging in IoT should be considered to maintain record of all states, events, communications, and other security relevant aspects associated with the IoT device (IoT sensors,

IoT actuators and other equipment). But, IoT devices may have limited resources and functions or IoT devices located off-premises can be vulnerable in maintaining log information. Self-monitoring and logging may not be feasible, so monitoring and logging may need to be performed on servers or network nodes of the IoT system.

The following are guidelines that can be used in the cybersecurity logging creation and management:

1) Select the systems, applications and network devices that will generate logs: servers, firewalls, intrusion detection systems and other information assets that are considered critical should be included;

2) Identify the specific events and activities that need to be logged, considering the applicable legal, regulatory and organisational compliance requirements and industry best practices, and the processing and retention requirements necessary to produce the logs. Logged events may include:

a) Successful and rejected system access (user authentication) attempts;

b) Successful and rejected data and other resource access attempts;

c) Changes to system configuration;

d) Access and changes performed by privileged users;

e) Use of utility programs and applications;

f) Files accessed and the type of access, including deletion of important data files;

g) Alarms raised by the access control system;

h) Activation and de-activation of security systems, such as anti-virus systems and intrusion detection systems;

i) Creation, modification or deletion of identities;

j) Transactions executed by users in applications.

Additional Information

Further implementation guidance may be derived from the following sources:

• ISO/IEC 27002:2022 “8.15 Logging”;

• ISO/IEC 27400:2022 “Control-12 Monitoring and logging”;

• ENISA Securing Machine Learning Algorithms “Security Controls - Include ML applications into detection and response to security incident processes”;

• ENISA Securing Machine Learning Algorithms “Security Controls - Ensure ML projects follow the global process for integrating security into projects”;

• NIST SP 800-82R3 “Logging (PR.PT-1)”;

• NIST SP 800-82R3 “Anomalies and Events (DE.AE)”.

IV. DOMAIN: RESILIENCE

Sub-Domain: Part 4.1 – Continuity of Operations

MSBP092 Information Processing Facilities Redundancy

Recommended NG Maturity Level

Team, Proficient, Managed

Purpose

To ensure availability of the necessary resources for business continuity.

Requirement(s)

Business critical facilities should be implemented considering redundancy to meet the necessary availability requirements.

Summary Guidance

The organisation should design and implement systems with appropriate redundancy to achieve the service availability required by the business. The design should ensure the integrity and confidentiality of replicated data and configurations.

The organisations may consider duplicate in part or entirely their information processing facilities.

The following may be done to implement redundancy:

1) Duplicate the access to the Internet (by using multiple internet service providers (ISP)), and other critical information assets;

2) Implement network redundancy;

3) Implement geographic redundancy - geographically separated data centres;

4) Implement redundant power sources;

5) Duplicate components in systems or networks;

6) Implement parallel instances of software components with automatic load balancing

7) between them (between instances in the same data centre or in different data centres).

The redundant information processing facilities may be always active or may be activated only in case of an emergency. Procedures and mechanisms should be defined and implemented to:

1) Maintain or achieve data synchronisation between the primary and the replicated information processing facilities;

2) Activate the redundant components in production when required (failover) and restore production to the primary components. These procedures can be automated or manual;

3) Monitor the information processing facility and, in the event of a failure, alert the organisation to carry out the procedure to activate the redundant information processing facility in production, thereby maintaining service availability while the original information processing facility is being repaired;

4) Periodically test the procedures for activating the redundant information processing facility (failover) and returning to the original information processing facility.

The redundant components and information processing facilities should have the same security level as the primary ones.

Redundancy and information and communication technology (ICT) readiness for business continuity are related. Some of the redundancy measures may be part of the ICT continuity strategy, especially if the organisation requires short recovery times.

Additional Information

Further implementation guidance may be derived from the following sources:

• ISO/IEC 27002:2022 "8.14 Redundancy of information processing facilities"

Appendix 2 — Full SUS-adapted questionnaire. Presents both Consultant and practitioner questionnaire versions.

The complete questionnaire (see Annex) comprised four sections:

- Section 1 – Contextual Overview: Introduces the methodology, maturity model architecture, maturity levels, and associated capabilities and practices.
- Section 2 – Demographic Information: Collects participant details (e.g., age, professional experience).
- Section 3 – Usability Evaluation: Presents an adapted SUS questionnaire with ten Likert-scale items (1—strongly disagree to 5—strongly agree). Scores above 70 indicate acceptable usability, 50–70 marginal, and below 50 unacceptable.
- Section 4 – Overall Evaluation: Includes a recommendation likelihood question (0–10 scale) and an open-ended question for qualitative feedback and improvement suggestions.

Questions for Practitioners

Table 7 presents the ten questions administered to the software development professionals. Questions with odd numbers were formulated from a positive perspective, whereas those with even numbers were formulated negatively.

Table 7. Questions for Software Development Professionals

Code	Question
Q1-DEV	I think I would like to use this maturity model frequently for improving our software development processes.
Q2-DEV	I found this maturity model unnecessarily complex.
Q3-DEV	I thought this maturity model was easy to use in the context of our organization's improvement efforts.
Q4-DEV	I would need significant external support (e.g., from consultants or coaches) to effectively apply this maturity model.
Q5-DEV	I found the different components of this maturity model to be well integrated.
Q6-DEV	I thought there was too much inconsistency in how this maturity model is structured or presented.
Q7-DEV	I believe most people in our software development organization would learn to use this maturity model very quickly.
Q8-DEV	I found it cumbersome to apply this maturity model to our specific improvement needs.
Q9-DEV	I felt very confident when applying the maturity model to guide our improvement efforts.
Q10-DEV	I needed to learn a lot of new concepts or skills before I could effectively utilize this maturity model.

The Overall evaluation questions presented to Practitioners were:

- On a scale of 0 to 10, how likely are you to recommend the proposed maturity model to a colleague in your organization or other organizations?
- What specific improvements or additions could help you apply this maturity model more effectively across diverse organizations?

Questions for Consultants

Table 8 presents the 10 questions provided to consultants in the SUS-based questionnaire. They continue to uphold the structure of a positive perspective for odd-numbered questions and a negative one for even-numbered questions.

Table 8. Questions for Consultants

Code	Question

Q1-CONS	I think I would like to use this maturity model frequently when consulting for different software development organizations.
Q2-CONS	I found this maturity model unnecessarily complex when trying to apply it in multiple client contexts.
Q3-CONS	I thought the maturity model was easy to adapt for organizations of varying sizes or industries.
Q4-CONS	I would need significant external support (e.g., specialized training or additional resources) to effectively use this model with different clients.
Q5-CONS	I found that the model's components fit together well for a wide range of software development scenarios.
Q6-CONS	I noticed too much inconsistency in how the model addresses various client environments or business contexts.
Q7-CONS	I believe most client teams would learn to navigate and apply this maturity model very quickly.
Q8-CONS	I found it cumbersome to customize the maturity model for clients in different domains (e.g., finance, healthcare, tech startups).
Q9-CONS	I felt very confident using this maturity model as a guiding framework for improvement in diverse software development organizations.
Q10-CONS	I needed to learn a lot of new concepts or guidelines before I could effectively apply the model across different client contexts.

The Overall evaluation questions presented to Practitioners were:

- On a scale of 0 to 10, how likely are you to recommend the proposed maturity model to a colleague in your organization or other organizations?
- What specific improvements or additions could help you apply this maturity model more effectively across diverse organizations?