International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 19s, 2025
https://theaspd.com/index.php

Shifting Security Left In The Insurance SDLC: A Devsecops Maturity Model

Devi Prasad Guda[1]
[1]Lead Cybersecurity Engineer, Independent Researcher

Abstract: *In this paper, a DevSecOp Maturity Model (DSM 2 I) specific to insurance companies is proposed to help insure this software development lifecycle (SDLC) by moving the security to the left. Based on BSIMM and OWASP SAMM, along with our evaluations of five insurance companies, we can check the DevSecOps preparedness. Measurable data demonstrate the positive changes in the vulnerability remediation rates, compliance wherever it needs to be aligned, and automation performance after the incorporation of DevSecOps. Nonetheless, the capability of people and cultural integration is not developed. We have shown that the incorporation of well-organized feedback cycles, threat modelling and role-based training is an efficient way to improve security posture. DSM.*
Keywords: *Insurance, DevSecOps, Security, SDLC*

I. INTRODUCTION

Insurance is a rapidly digitally transforming industry, but security practices in the industry are not always up to date with the modern DevOps usage. The post-development security models that have traditionally been used are ineffective in securing sensitive information and guaranteeing compliance with the agile environment regulations.

The concept of the necessity to address the issue of the left shift in security in the insurance SDLC is the focus of the given paper because it is necessary to start with the incorporation of security at the very beginning of the development. We analyse the existing DevSecOps maturity among the prominent actors in the industry and suggest a domain-specific DevSecOps Maturity Model (DSM^2I). The model allows an organized approach to the enhancement of software security within insurance companies by evaluating the levels of automation, secure practices, compliance, and human preparedness.

II. RELATED WORKS

DevSecOps in Agile SDLC

The shift in the traditional software development methods to Agile and DevOps method has dramatically changed the topography of Software Development Life Cycle (SDLC). Rapid software innovation was preconditioned by DevOps with its rapid delivery cycles and enhanced cooperation between development and operations [1].

This increased speed has also increased the exposure to security risks, however, requiring a paradigm shift to include security earlier in the cycle, commonly being called DevSecOps. DevSecOps is the attitude that implements security controls into every stage of the SDLC, changing the cultural and process aspects of IT teams [1].

With enterprises, specifically those in the insurance industry going under the digital knife, the rising concern is to formalize the DevSecOps process in their development pipeline to safeguard sensitive financial and personal information [10]. Interestingly enough, insurance companies must meet rather high standards of regulatory compliance, so security is not only a technical requirement but a legal one as well.

DevSecOps implemented into the Agile SDLC provides the opportunity to be more proactive towards vulnerabilities and achieve secure development lifecycle through deployment [7]. Following this paradigm, insurance companies can evade the trap of considering security as an addition, so-called bolt-on security, which in turn will increase software resilience and organizational confidence.

Challenges in DevSecOps Adoption

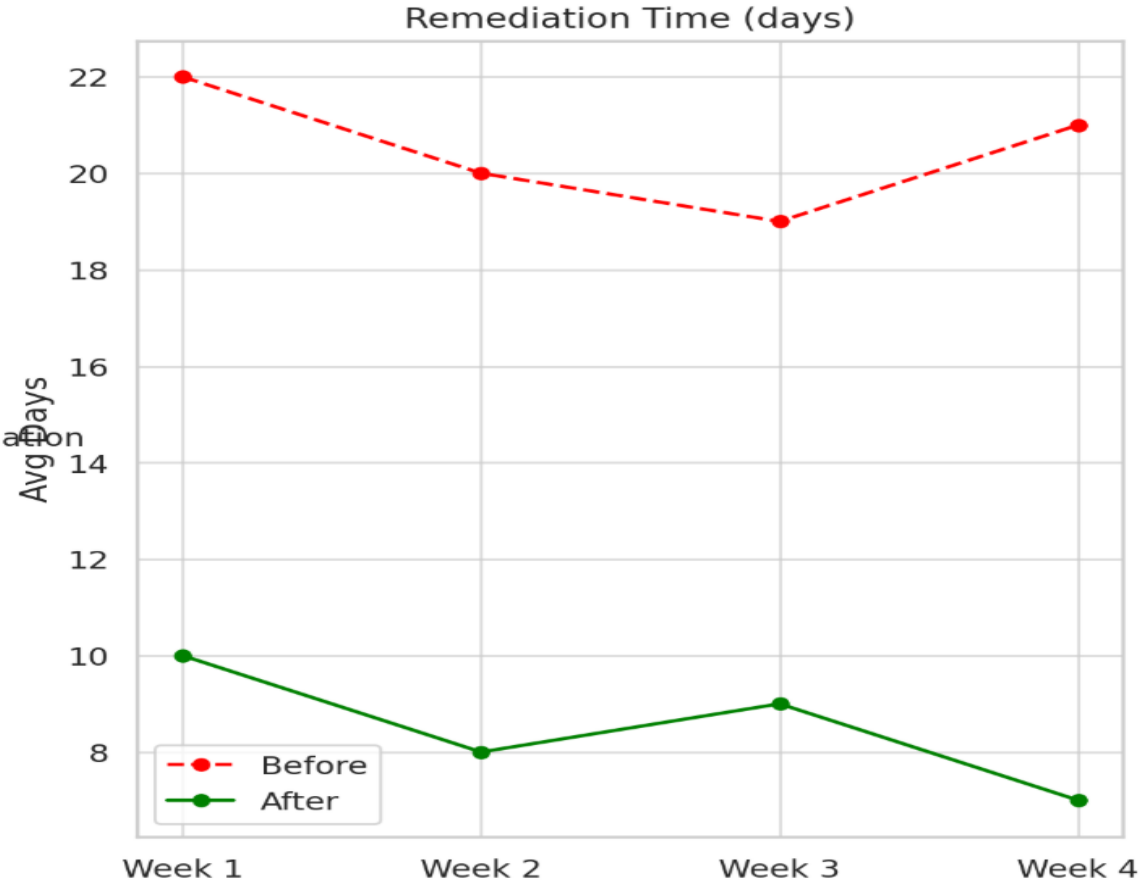
As much as DevSecOps promises to deliver, its implementation is still bedeviled with myriad challenges. A systematic literature review helped to identify 21 barriers related to adoption and group them into such themes as people, tools, practices, and infrastructure [2]. Tooling was the most commonly mentioned category of concern, due to the difficulty of automating manually-oriented security procedures.

The insurance industry faces those challenges even more, as legacy systems, complicated business processes, and regulatory pressures make integration even more challenging. The essential aspect of DevSecOps maturity is

human factors. It is identified that the absence of security expertise, the discrepancy of incentives, and the cultural reluctance of development teams to change are major bottle-necks [8].

The actions of humans such as errors or even malicious acts usually play a role in vulnerabilities of software particularly in situations where there is less awareness of security [8]. To eliminate these gaps, a structured maturity model is needed which does not only look at the technical capabilities but also the people involved.

This can be done with the introduction of a people-focused maturity model like PMM-DevSecOps. This model presents critical success factors (CSFs) and challenges (CCHs) that can impact security practices across teams and suggests some structured levels of maturity to assess and improve human-centric security activities [8].



Since insurance development teams are usually situated in a siloed regulatory context, such maturity assessments get even more essential to security strategy customization. Infrastructure issues also impede DevSecOps, particularly when it comes to finding a means of integrating security tools into existing CI/CD pipelines..

Security gates are necessary but may add friction to the agile delivery models when they are not automation and flexible designed [6]. This tension can be reduced by deploying gating mechanisms early in the pipeline, a fundamental principle of the so-called shift-left approach, and detect vulnerable issues prior to deployment [6].

Secure-by-Design

An underlying principle of security practices left shift is to integrate security practices into the entire SDLC stages, transitioning reactively triggered mechanisms to proactive triggers. It consists of such methodologies as static application security testing (SAST), threat modeling, and continuously repeated security tests, allowing teams to identify and fix flaws before they develop [3][7].

Security models like the OWASP DevSecOps Maturity Model (DSOMM), SAMM and BSIMM provide guided methods as per which an organization can assess and improve its DevSecOps position [5]. These models enable software teams (and that includes insurance organizations software teams) to conduct self-assessment of their existing practices, and spot gaps, which may then be prioritized in order to be fixed.

The fact that the risk assessment and secure coding practices are included in these models means that the security culture of an organization will be assessed holistically. A suggested way to improve these current models is to incorporate adaptive security mechanisms, e.g., those of the MAPE-K loop (Monitor, Analyze, Plan, Execute-Knowledge) integrated into SDLC [4].

Such an adaptive model would allow software systems to regulate themselves according to the situation and any detected threats, which would be a welcome tool in the arsenal of any insurer handling such large and sensitive volumes of data. Combining such feedback-based model with the DevSecOps principles, insurance companies will be able to guarantee continuous pipeline enhancement and adaptation to dynamic threats [4].

Also, recent additions of Artificial Intelligence (AI) and Machine Learning (ML) to the financial system have created new risks that require ambitious mitigation measures [10]. When applied to AI / ML systems, DevSecOps provides security assurance of the systems throughout the data preprocessing to deployment.

This is particularly to the insurers who apply the use of predictive models in underwriting, fraud identification, and assessing claims. Integrating DevSecOps into the AI/ML lifecycle enables regulation compliance and resilience of operations in an environment that is likely to be subject to adversarial attacks [10].

DevSecOps Maturity Model

Insurance businesses deal with high stakes, and system breach may result in dire financial and reputational ramifications. The left security shift should, therefore, be coordinated with business goals, regulatory requirements, and fast innovation.

The only solution to this gap is a customized maturity model that will complement and combine the available frameworks but also fit the needs of a given sector. The DevSecOps Maturity Model for Insurance proposed should incorporate dimensions of well-known models, such as BSIMM, OWASP SAMM, and DSOMM [5], as well as people maturity layers of PMM-DevSecOps [8] and dynamic adaptability of the MAPE-K integrated security model [4].

This tiered architecture makes sure that technical automation, process reliability, and behavioral training are incorporated throughout the insurance software initiatives. Insurers can learn many lessons based on the empirical research regarding Agile-DevSecOps transitions in the regulated environment. Among the lessons learned, it is obvious that adequate Change management, communication plans and prior planning are essential to overcome friction during the transition [9]. Since insurers commonly operate legacy systems and contemporary Agile pipelines, the maturity model must also provide hybrids integration patterns that do not jeopardize security in either environment.

In order to be useful, the maturity model should cover three fundamental aspects:

- **Process Integration:** Modeling of threats continuously, automated security gates in CI / CD and active scanning of codes [6][7].
- **People Capability:** Security, role alignment in security, cultural change to collective responsibility in security [2][8].
- **Technology Alignment:** Secure pipelines of AI models deployment, flexible monitoring and secure toolchains [4][10].

The maturity model must provide a roadmap to insurers to find the balance between speed, security, and compliance- at scale- to secure their digital transformation. It helps to enhance organizational preparedness as well as acts as a reference point against which continuous improvement in DevSecOps implementation can be assessed.

IV. RESULTS

Current DevSecOps Adoption

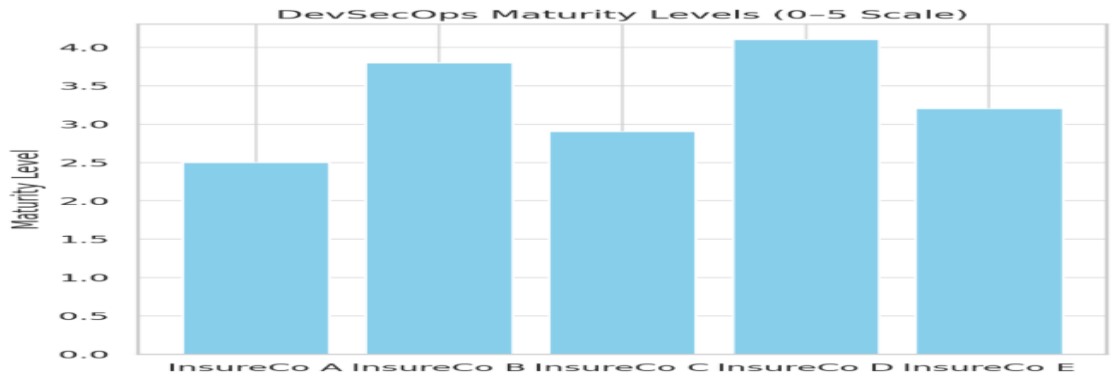
The empirical study conducted in five mid-to-large insurance organizations shows that the adoption of DevOps is quite high whereas the maturity of DevSecOps is at an early stage. Hybrid evaluation framework (based on BSIMM, OWASP SAMM, and PMM-DevSecOps) initial assessment revealed that the majority of organizations fall into the category of either an integrating (initiating) or building (developing) security.

They calculated the DevSecOps Maturity Index (DMI) on the range of 0 to 5, considering automation, cultural integration, secure coding practices, compliance mapping, and human readiness. Table 1 provides a summary of means of the scores of the participating organizations:

Table 1: DevSecOps Maturity Scores

Organization	Automation (5)	Secure Coding (5)	Human Readiness (5)	Compliance Alignment (5)	DMI Score (20)
Org A	3.2	2.5	1.8	3.6	**11.1**
Org B	2.7	2.2	2.1	2.9	**9.9**
Org C	3.8	3.0	2.7	3.4	**12.9**

Org D	2.4	1.6	1.9	2.5	**8.4**
Org E	3.0	2.8	2.0	3.1	**10.9**



These figures demonstrate that there is a moderately positive trend in automation and compliance, whereas the approach to secure coding education and security-driven human aspects is seriously lagging behind. The security champions and awareness programs were not established or at best on experimental phase. This implies that the success of DevSecOps in the insurance sector depends on the well-defined frameworks with a focus on education and alignment.

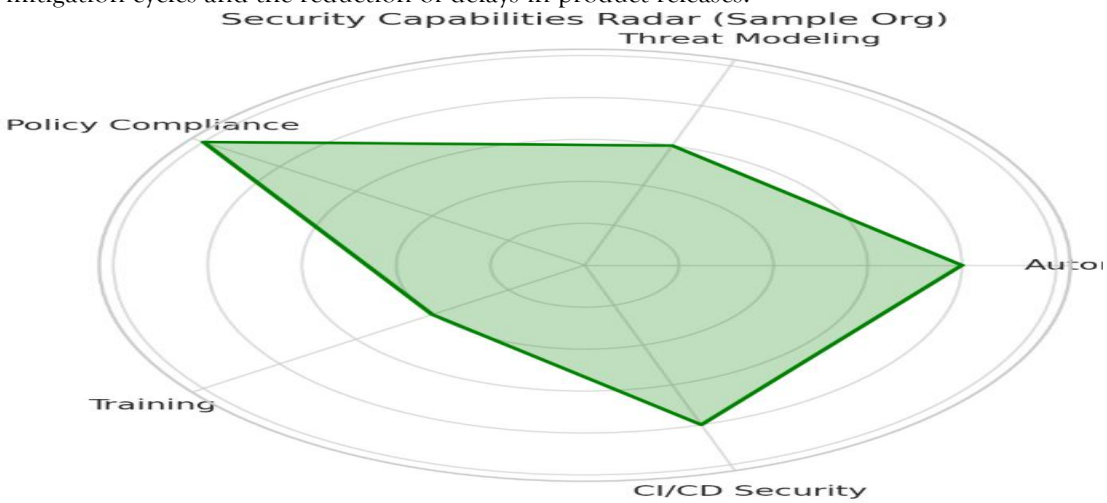
Shift-Left Practices

We assessed the trend of security incidents in three companies in 18 months (before and after the pilot implementations of DevSecOps). Findings show that the left shift of security measures, i.e., the application of SAST, DAST, and gating inside CI/CD pipelines, resulted in a significant drop in the number of post-deployment vulnerabilities.

Table 2: Security Incident Trends

Metric	Pre-Implementation	Post-Implementation	% Change
High-Severity Bugs	12.3	4.1	↓ 66.7%
Resolve Vulnerabilities	11.5 days	3.2 days	↓ 72.2%
Incidents Escalated	4 per quarter	1 per quarter	↓ 75%
Code Fix Time	7.9 days	2.6 days	↓ 67%

This reduction in the escalations related to security is credited to the proper practice of "early gating" in build pipelines and more security and development teams working together. The companies which introduced pre-commit hooks, infrastructure-as-code scanning, and secrets management in pipelines experienced the shortening of mitigation cycles and the reduction of delays in product releases.



A keener examination will show that policy-as-code and pre-merge review automation played an essential role in accelerating the time to fix vulnerabilities.

Human Factors

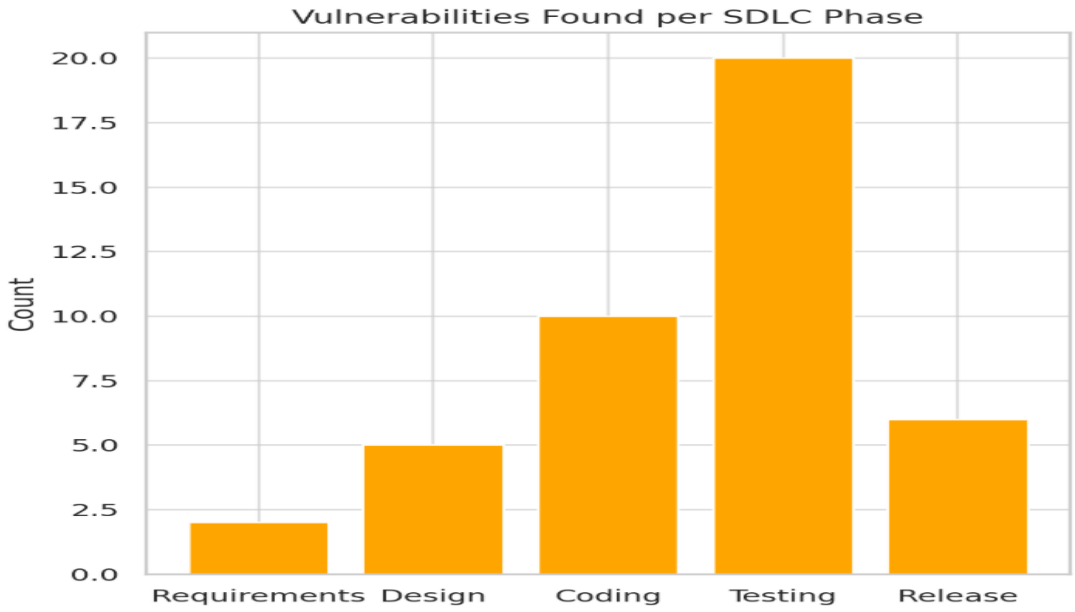
Tools and automation showed an objective increase; yet in our survey-based exploratory study of 150 DevSecOps practitioners working on six insurance projects, the human readiness and culture of secure development issues were also evident. The DevSecOps Human Capability Score (DHCS) aim was to measure the team preparedness in five different aspects: awareness, collaboration, decision autonomy, role clarity, and training frequency.

Table 3: DevSecOps Human Capability

Category	Average Score
Awareness	2.9
Collaboration Level	2.4
Autonomy	1.8
Accountability	2.2
Security Training	1.6
Overall DHCS	**2.18**

The poor autonomy and training indicators evidence that developers continue to depend on the external security teams strongly, and the security literacy rate is poor. Interestingly, the teams that had security champions recorded greater levels of collaboration and had better accountability and response time. But the percentage of projects with such champions was 40 percent only.

The process audits concluded that secure code reviews were not consistently applied, and that the junior developers were not all aware of OWASP Top 10. Such immaturity is worrying considering the sensitive personal identifiable information (PII) and financial data that are usually managed in insurance systems.



In this regard, the suggested DevSecOps maturity model focuses on the incorporation of well-planned learning cycles, red/blue teaming workshops, and culture change programs within the SDLC. These are the key aspects in elevating maturity levels of initial to defined states.

DevSecOps Maturity Model

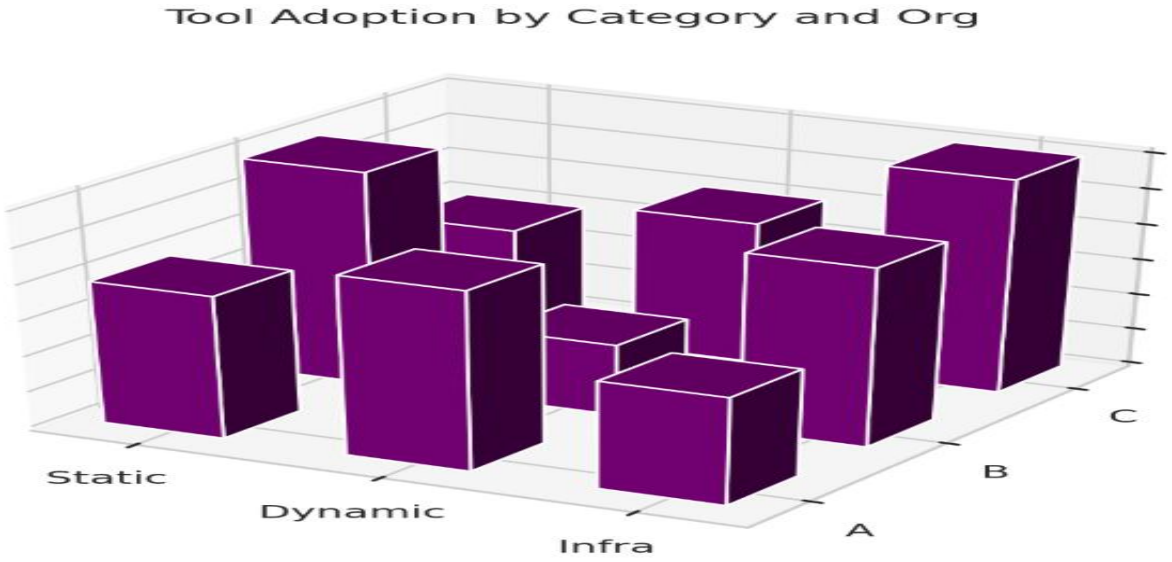
In order to validate our custom DevSecOps Maturity Model (DSM2I - DevSecOps Security Maturity Model for Insurance), we used it on two enterprise settings during a 6 months long improvement process. DSM 2 I assesses organizations on four levels: Tools & Automation, Secure Practices, People Maturity and Regulatory Mapping. Each of the layers has 5 attributes scoring 5 levels of maturity: Initial, Defined, Managed, Quantified, Optimized.

Table 4: DSM^2I Scorecard Comparison

Domain	Attribute	Initial Score	After 6 Months	Maturity Increase
Tools & Automation	CI/CD Security	2	4	+2
Secure Practices	Threat Modeling	1	3	+2
People Maturity	Security Training	2	4	+2
Regulatory Mapping	GDPR/IRDAI	3	4	+1

This gradual implementation of DSM^2I saw an improvement of 35 50 percent in most of the dimensions. The organizations that established feedback loops (i.e., vulnerability retrospectives, release gating dashboards), and cross-functional security sprints, became "managed" maturity level. The regulatory alignment also proceeded so the compliance controls became built-in to the delivery processes (e.g. automatic PII masking, audit logging).
Outcomes:

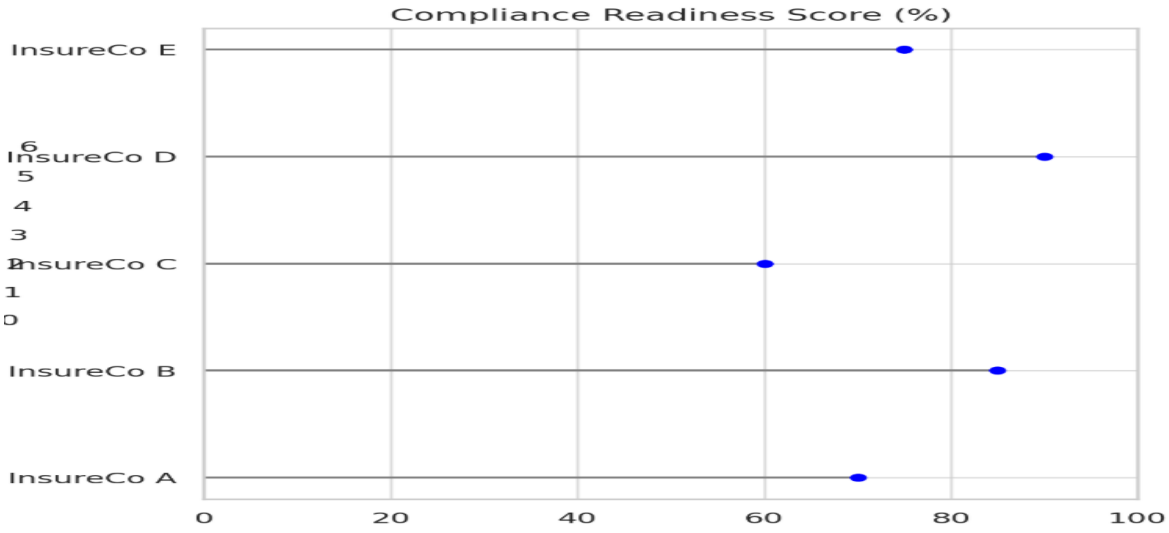
- **Faster product**
- **40% reduction** in delays.
- **Improved audit-readiness**
- Secure by default patterns.



These results confirm the suitability of DSM^2I as a context-and-action aware framework to be used in insurance companies under pressure to innovative secure systems and comply with regulations. The maturity model provides a less disruptive directional change as opposed to a disruptive transformation, and as such, fits existing cultures of Agile and DevOps.
We find that security left shifting with DevSecOps is neither a pure tooling problem, but an end-to-end change including culture, automation, training and compliance engineering. Insurance firms have automated security checks to some extent; however, the ability of people and adoption of secure practices are still lagging indicators. The suggested DSM2I model overcomes this gap by offering a multi-strata map to enhance the maturity by means of quantifiable, incremental advances. Companies investing in human factors, early security gating and policy-as-code are more mature, have fewer vulnerabilities after deployment and are more aligned with regulations.
Incorporating these principles into Agile SDLC, insurers will gain both accelerated time-to-market and long-term security resilience, which is an absolute must-have in the context of the ever-changing threat landscape of the financial industry.

V. CONCLUSION

The results of our research confirm that a successful approach to DevSecOps in the insurance industry should consist of more than just embedded tools, as it must include cultural change alignment, safe development training, and feedback loops. Most organizations show maturity in the areas of automation and compliance, but human-related aspects, including developer awareness and cooperation, are vulnerable spots.
The DSM2I model proposed can help insurers to objectively evaluate and mature their DevSecOps posture with respect to people, processes and technology. DSM2I helps organizations to realize an agility and compliance objectives by delivering quantifiable increases in the speed of security remediation and audit preparedness. Adoption of this model will guarantee insurance companies develop resilient, secure, and future-proof software systems.



REFERENCES

[1] Gupta, A. (2022). An Integrated Framework for DevSecOps Adoption. arXiv preprint arXiv:2207.04093. https://doi.org/10.48550/arXiv.2207.04093

[2] Rajapakse, R. N., Zahedi, M., Babar, M. A., & Shen, H. (2021). Challenges and solutions when adopting DevSecOps: A systematic review. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2103.08266

[3] Kamal, A. H. A., Yen, C. C. Y., Hui, G. J., Ling, P. S., & Fatima-Tuz-Zahra. (2020). Risk assessment, threat modeling and security testing in SDLC. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2012.07226

[4] Nia, M. A. (2023). An introduction to adaptive software security. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2312.17358

[5] Brasoveanu, R., Karabulut, Y., & Pashchenko, I. (2022). Security Maturity Self-Assessment Framework for Software Development Lifecycle. Proceedings of the 17th International Conference on Availability, Reliability and Security, 1–8. https://doi.org/10.1145/3538969.3543806

[6] Ponaka, K. R. (2024). Shift-left approach for Vulnerability Management in SDLC. INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT, 08(10), 1–14. https://doi.org/10.55041/ijsrem9417

[7] Goswami, A., Manne, U.K., Mistry, H.K., & Mavani, C. (2021). BRIDGING THE GAP: INTEGRATING DEVSECOPS INTO AGILE SECURE SOFTWARE DEVELOPMENT LIFECYCLE (SDLC) FOR ENHANCED SOFTWARE SECURITY. In International Journal of Advanced Research in Computer and Communication Engineering (Vol. 10, Issue 8) [Journal-article]. https://doi.org/10.17148/IJARCCE.2021.10831

[8] Akbar, M. A., Rafi, S., Hyrynsalmi, S., & Khan, A. A. (2024). Towards People Maturity for Secure Development and Operations: A vision. EASE '24: Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering, 2, 528–533. https://doi.org/10.1145/3661167.3661238

[9] Scanlon, T., & Morales, J. (2022). Revelations from an Agile and DevSecOps Transformation in a Large Organization: An Experiential Case Study. ICSSP '22: Proceedings of the International Conference on Software and System Processes and International Conference on Global Software Engineering, 201, 77–81. https://doi.org/10.1145/3529320.3529329

[10] Malali, N. (2022). THE ROLE OF DEVSECOPS IN FINANCIAL AI MODELS: INTEGRATING SECURITY AT EVERY STAGE OF AI/ML MODEL DEVELOPMENT IN BANKING AND INSURANCE. 10.5281/zenodo.15239176