International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 12-No.1 (2026) pp. 1092-1099
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Digital Twin Enablement through Automated PLM Deployments: Accelerating Hardware Product Development

Swami Venkatesh Mandepu*

Süleyman Independent Researcher, USA
* **Corresponding Author Email:** swamivmandepu@gmail.com - **ORCID:** 0000-0002-0247-5530

Article Info:

DOI: 10.22399/ijcesen.4973
Received : 03 January 2026
Revised : 20 February 2026
Accepted : 22 February 2026

Keywords

Digital Twin Technology,
Automation Of Product Lifecycle Management,
Infrastructure-as-Code,
Containerized Deployments,
Continuous Integration And Deployment

Abstract:

The use of digital twin technology radically revolutionizes hardware product development through the real-time synchronization of physical and virtual equivalents. Seamless integration of Product Lifecycle Management platforms with sophisticated engineering and manufacturing workflows is necessary to achieve the full potential of digital twins. This material describes an automation-centric approach to facilitating digital twins through optimized PLM deployments that utilize infrastructure automation and config management to accelerate environment setup, minimize integration complexity, and maintain consistency between development, QA and Production. The article utilizes infrastructure-as-code processes, containerization platforms, and declarative config management to normalize deployment processes while removing the need for manual configuration bottlenecks. Deployment on a big-ticket aerospace hardware program shows significantly faster environment setup times, accelerated iteration cycles for product verification, and improved integration between engineering and operations staff. The findings indicate that auto-deployed PLM is a key enabler of digital twin adoption, enabling more agile and cost-effective development of hardware products. Companies that deploy the framework realize stunning reductions in configuration drift events, expedited design verification cycles, and enhanced simulation correlation to physical test results, making digital twin technology a strategic differentiator in competitive production settings.

1. Introduction

Adoption of digital twin technology is transforming hardware product development by allowing physical assets to be synchronized in real time with their virtual representations. This, however, relies on the successful integration of Product Lifecycle Management (PLM) platforms with intricate engineering and manufacturing procedures. In this paper, an automation-centric framework for empowering digital twins through optimized PLM deployments is introduced. With the use of infrastructure automation and configuration management, the framework speeds up the provisioning of PLM environments, minimizes integration complexity, and provides consistency on the development and production systems. A large-scale hardware program case study illustrates quantifiable returns in the form of significant decreases in environment setup time, accelerated iteration cycles for product validation, and enhanced collaboration between engineering and operations teams. The findings emphasize the way in which automated PLM implementations act as a key enabler for digital twin uptake, facilitating more agile and cost-effective hardware product development.

The intersection of digital twin technology and automated PLM solutions is a paradigm shift in the way organizations conduct hardware product development. Traditional deployments of PLM commonly involve large amounts of manual configuration, bespoke integration scripts, and rigorous validation processes before they can become productive. Such long deployment cycles introduce serious bottlenecks in developing the underlying infrastructure required to support digital twin capability. Recent research has shown that organizations lacking automated PLM provisioning while developing digital twins encounter severe challenges in attaining operational readiness within distributed manufacturing environments. Manual

PLM deployment processes usually take several weeks, with configuration discrepancies arising frequently between development and production environments, especially in organizations that have large product portfolios with deep bill-of-materials hierarchies and multi-disciplinary engineering interdependencies [1]. The problem is more acute in distributed engineering organizations where several teams need coordinated access to product data, simulation models, and manufacturing specifications. Companies with more than one geographic location indicate that coordination overhead on hand deployments of PLM stretches out timelines significantly, while synchronization problems with environments lead to rework cycles consuming large segments of the first development sprints.

Contemporary hardware development initiatives require unprecedented degrees of collaboration among mechanical engineers, electrical designers, software developers, and manufacturing experts. Digital twins are the key integration nexus for these varied stakeholders, delivering a single virtual representation that develops during the product life cycle. Organizations report that PLM deployment and configuration timelines directly impact their ability to operationalize digital twin capabilities for design validation, performance analysis, and manufacturing optimization activities. Companies that have used infrastructure-as-code practices for PLM deployments cite revolutionary enhancements in operations efficiency and time-to-value metrics. Empirical evidence from business deployments shows that automated provisioning significantly decreases environment setup time in contrast with conventional manual cycles, reflecting significant cycle time gains [2]. This speed-up allows product design iteration more frequently and enables greater correspondence between virtual models and physical prototypes. Engineering groups using automated PLM platforms have reported being able to carry out design validation cycles at much higher frequencies compared to conventional methods and reducing configuration drift incidents by substantial percentages in the development, staging, and production environments.

The technical sophistication of integrating PLM systems with digital twin platforms goes beyond initial deployment to include maintaining configuration management, versioning, and environment synchronization. Enterprise PLM solutions usually communicate with computer-aided design software, simulation environment tools, manufacturing execution systems, and enterprise resource planning solutions, forming integration scenarios that can involve many different system interfaces based on organizational scale and product complexity. Each integration point provides opportunities for inconsistencies and configuration drift that undermine the fidelity of digital twin representation. Automated deployment platforms overcome these challenges using declarative configuration specifications, versioned infrastructure definitions, and continuous validation systems [2]. These features allow PLM environments to be consistent throughout their lifecycle and ensure changes propagate securely through development to production systems. Organizations using end-to-end automation frameworks indicate that environment consistency validation, which otherwise involves a significant manual verification effort per deployment cycle, can be performed by automated test suites running in quick cycles with extensive coverage of key configuration parameters.

2. Digital Twin Technology in Contemporary Hardware Product Development

Digital twin technology has grown from concept visualization tools into advanced platforms that facilitate predictive analytics, real-time monitoring, and closed-loop optimization throughout hardware product lifecycles. Modern digital twins combine various streams of data from physical sensors, simulation models, manufacturing system operational telemetry, and other sources to produce active virtual replicas continuously projecting the state and behavior of physical objects. Studies have shown that organizations that adopt end-to-end digital twin strategies realize significant product development cycle time reductions and significant manufacturing ramp-up first-pass yield rate improvements [3]. These advantages result from being able to verify design choices, maximize manufacturing processes, and forecast product performance before investing resources in physical prototype construction and production tooling. Organizations indicate that digital twin simulation-based early-stage design validation detects a high percentage of manufacturing issues before the physical prototype is built, allowing for corrective measures that greatly minimize late-stage engineering change orders and related cost overruns.

The technical foundations for successful digital twins are based on a sound data management infrastructure that is able to consume, process, and keep synchronized data from disparate sources throughout the product lifecycle. PLM platforms are the single source of truth for product definitions such as geometric models, bill-of-materials structures, engineering specifications, and change management workflows. With proper integration into digital twin platforms, PLM systems offer the structural framework necessary to guarantee virtual representations remain aligned with approved product configurations and are traceable throughout design development and production updates [3]. Organizations that develop robust PLM-digital twin integration demonstrate enhanced decision-making ability, as engineering teams can analyze design alternatives against performance specifications and manufacturing limitations in near-real time. Performance benchmarking research suggests that combined PLM-digital twin architectures significantly decrease design iteration cycle times over conventional methods while keeping simulation fidelity levels high that strongly correlate with physical test outcomes. Cross-functional coordination metrics report that coordinated PLM-digital twin environments significantly cut down inter-departmental overhead in coordination and lower design-to-manufacturing handoff faults throughout the product development process [4].

The deployment of digital twins within hardware design brings large-scale computational and data management needs that grow with product complexity and organizational scale. High-fidelity physics-based simulation, computational fluid dynamics analysis, and finite element modeling create large volumes of data that need to be versioned, stored, and made available to geographically dispersed engineering teams. Latest PLM implementations that enable digital twin functionality necessitate scalable storage infrastructures, high-performance computing, and streamlined data pipeline mechanisms to process simulation results and sensor telemetry [4]. Industry estimates indicate that large aerospace and automotive programs can produce significant amounts of simulation data per year, with each high-fidelity computational analysis generating large datasets based on mesh resolution and temporal granularity. The potential to quickly provision and scale these infrastructure pieces has a direct relation to how feasible it is to achieve digital twin implementation for intricate hardware products, especially those that have multi-physics interactions and coupled system behaviors.

Organizational and technical challenges to enterprise deployment of digital twin technology go beyond sole technology deployment. Cultural resistance to new business workflows, advanced analytics, and simulation skills gaps, and corporate data security concerns are all typical barriers to successful digital twin initiatives. The inherent complexity in traditional PLM deployments further complicates these challenges by making timelines between strategic decisions to adopt digital twins and capability delivery to operations. PLM provisioning frameworks automate this adoption resistance by allowing for speedy proof-of-concept deployments, supporting iterative improvement of digital twin designs, and easing the technical hurdles to creating production-ready environments. Organizations indicate that minimizing PLM deployment cycles greatly shortens organizational learning curves and creates momentum for digital twin take-up across engineering departments.

3. PLM Deployment Infrastructure Automation Framework

The PLM deployment automation framework utilizes infrastructure-as-code practices, containerization technologies, and declarative configuration management to automate and streamline environment provisioning to standardize and speed up the provisioning of the environment. The framework fundamentally approaches PLM infrastructure as versioned software artifacts that can be deployed identically to development, testing, and production environments. This method is a stark contrast to legacy manual deployment processes that depend on written runbooks, personal knowledge, and linear configuration steps that are subject to human error and environmental variances. The model uses configuration management tools to author PLM system topologies, such as application servers, database clusters, file repositories, and integration middleware, as executable code that is versionable, peer-readable, and deployable automatically. Version control systems offer critical functionality in managing infrastructure definitions to allow for collaborative development of deployment configurations with a complete historical record of all changes [5]. Infrastructure-as-code solutions minimize deployment error rates by several orders of magnitude over manual processes, while configuration drift incidents dramatically decrease due to automated state reconciliation functionality continuously monitoring and correcting environmental drifts.

Container orchestration systems are a key part of the automation infrastructure, allowing PLM applications and their dependencies to be packaged into a consistent, portable unit that can run on a variety of computing environments. Such deployments under traditional PLM have customarily been plagued by environment-specific configuration issues around operating system versions, library dependencies, and network configurations that produce subtle differences in behavior between development and production environments. Containerization solves these problems by bundling PLM applications with all their runtime dependencies, and this allows validated development environment configurations to map consistently into production deployments [6]. Organizations having containerized PLM installations report dramatic reductions in defects caused by environments and increased troubleshooting speed when problems do occur. Container images support reproducible diagnostic environments, with image build steps completing quickly and deployment propagation to numerous environments being done in an efficient manner per target environment. Microservices architecture facilitated by containerization permits PLM systems to be broken up into loosely coupled, decoupled components that may be separately developed, tested, and deployed, improving overall system responsiveness and minimizing configuration change blast radius [6].

The framework includes continuous integration and continuous deployment pipelines to mechanize validation, testing, and promotion of PLM configurations across environment levels. These pipelines run thorough test suites that ensure PLM functionality, integration, connectivity, and performance characteristics before promoting configurations to production systems. Automated testing includes unit tests for custom integrations, functional tests for core PLM workflows, integration tests for external system connectivity, and performance tests that ensure system behavior under representative load conditions. Studies prove that organizations adopting automated testing for PLM rollouts identify configuration problems earlier in rollouts, with defect detection shifting entirely away from production to pre-production testing stages [5]. Production issues due to environmental inconsistencies are reduced significantly after adopting automation frameworks. The capability to automatically validate PLM configurations against defined acceptance criteria gives assurance that digital twin infrastructure has the necessary levels of reliability and performance, with automated testing execution running complete validation suites much faster than manual test procedures.

Automation of infrastructure provisioning is not limited to the initial deployment but also includes ongoing configuration management, patching, and capacity scaling activities. The design utilizes declarative state management that continually synchronizes real PLM infrastructure configurations with desired state specifications stored in version-controlled stores. This makes it possible to automatically detect and rectify drift so that human changes or environmental irregularities cannot create inconsistencies that weaken digital twin capability. Organizations that utilize declarative infrastructure management for PLM systems show enhanced operational reliability with significant mean-time-between-failures increases, significantly decreased mean-time-to-recovery during incident handling, and stronger audit capabilities in support of compliance needs. The system also supports capacity management by allowing programmatic scaling of PLM resources according to workload levels, where scaling operations run quickly and optimize the cost of infrastructure while preserving high-performance service levels and availability goals.

4. Implementation Analysis and Operational Impact

Large-scale implementation of the automated PLM deployment framework in a large-scale aerospace hardware program gives empirical affirmation of the benefits of the approach and lays bare pragmatic considerations for corporate adoption. The program entailed designing next-generation aircraft components with strict performance demands and intricate multi-disciplinary design optimization requirements. Historically, traditional PLM deployment methodologies had taken several weeks to implement operating environments, resulting in huge delays to implement digital twin functionality for engineering teams. The automated framework considerably shortened this lead time, which reflected huge enhancements in deployment time, directly contributing to the program's capability to harness digital twin technology to validate designs and plan manufacturing. The deployment included several individual PLM modules servicing many simultaneous concurrent engineering users in a variety of geographic locations, with database instances handling large product lifecycle histories and current design states [7].

The deployment produced considerable environment consistency and configuration accuracy improvements over past manual deployment methods. Automated validation testing identified configuration mismatches that would have created slight behavioral variations in development versus production PLM environments and, in turn, compromised digital twin fidelity. The framework's continuous integration pipeline ran high volumes of automated tests in functional, integration, and performance aspects, delivering thorough verification of PLM capabilities prior to production promotion [7]. Test run coverage attained high percentages of key configuration parameters, with regression test suites running effectively per deployment cycle. Post-deployment monitoring reports showed that environments provisioned by the automated framework had fewer configuration drift incidents, with incident rates reducing significantly under automated deployment methods compared to manual. Engineering teams were more confident in digital twin outcomes because development environments in which models were developed had better consistency with production systems in which validated designs were being deployed, with simulation correlation coefficients significantly better in terms of accuracy.

Operational data gathered over long durations of framework use proves lasting advantages beyond the acceleration of initial deployment. The program performed many PLM environment upgrades during this period of observation, such as software patches, integration improvements, and capacity increases. Automated deployment processes largely cut the average time for these updates in half, limiting interruption to engineering activities and allowing for more frequent inclusion of capability enhancements [8]. Frequency of updates went up from rare planned maintenance windows to constant continuous improvement cycles, with system downtime while updating decreasing significantly per update incident. The rollback features of the framework were beneficial in cases where updates brought unforeseen problems, enabling quick return to known-good states as opposed to prolonged periods needed for manual rollback processes. These performance enhancements translated into quantifiable productivity benefits, with the engineering teams wasting less time troubleshooting environment problems and more time engaged in product development work. Time-motion studies documented that environment-related troubleshooting tasks reduced significantly after framework implementation.

The economic value analysis yields a compelling return on investment for the automated PLM deployment framework. Initial framework setup took significant committed effort from infrastructure automation experts and PLM designers and was a major upfront cost. Yet, simply the decrease in deployment time alone created savings equal to the investment in the first year of operation through reduced labor costs and faster time-to-value for digital twin features [8]. Each saved interval of manual deployment work represented large amounts of engineering labor dollars across the distributed team. Additional gains from enhanced environment stability, lower incident response time, and greater capability to scale PLM assets created recurring operating cost savings. Organizations weighing analogous automation investments must include both direct labor savings and indirect gains such as enhanced engineering productivity, shorter innovation cycles, and lower risk of expensive environment-related defects in digital twin deployments.

5. Future Directions

The ongoing advancement of digital twin technology and PLM platforms also opens the door for opportunities to take automation frameworks further than merely deployment activities into intelligent orchestration, self-healing properties, and predictive capacity management. New studies look into using machine learning methodologies to process PLM system telemetry and optimize configurations automatically against usage patterns, workload properties, and performance targets. These smart automation features could make PLM environments dynamically responsive to shifting engineering workflows, provision and de-provision resources automatically in anticipation of spikes in demand, and automatically mitigate performance degradation before affecting digital twin functions. Early implementations of machine learning-based optimization show promising gains in resource utilization efficiency, with predictive scaling algorithms delivering high levels of forecast accuracy for anticipating workloads over long time horizons. Early adopter companies indicate that self-healing mechanisms can directly fix large percentages of typical configuration flaws automatically, all on their own, without intervention by human personnel, dramatically lowering mean-time-to-recovery for normal problems [9]. Companies leading the charge in PLM automation are just starting to test these sophisticated capabilities, although widespread implementation awaits the growth of supporting technologies and the creation of operating expertise in managing autonomous systems infrastructure.

The adoption of digital twin platforms with other enterprise architectures holds both opportunities and challenges for automated PLM deployment strategies. Organizations are more interested in bridging digital twin capabilities with enterprise data platforms, advanced analytics systems, and artificial intelligence frameworks to provide cognitive insights and autonomous decisions. These integration needs add complexity to the automation of PLM deployments since frameworks need to support a wide variety of data exchange protocols, security needs, and performance limitations [10]. Surveys in the industry show that the deployments of enterprise digital twins usually interface with many different enterprise systems, such as manufacturing execution systems, enterprise resource planning systems, supply chain management systems, and quality management databases. Every integration point must be meticulously choreographed to ensure data integrity, with integration middleware handling high volumes of transactions during high-usage operational periods. Upcoming automation platforms will have to integrate extensible integration patterns supporting emerging tech but preserving the consistency and reliability that define successful digital twin infrastructure [9].

Industry cooperation in standardized interfaces and best practices for PLM-digital twin integration may speed up the creation of end-to-end automation solutions, with standardization activities potentially cutting integration development effort significantly, according to pilot program evaluations. The development of containerized microservices architectures holds promising avenues for attaining higher modularity in PLM installations, allowing individual functional pieces to be upgraded, scaled, and optimized individually without impacting larger system operations [10]. Future research areas involve investigating service mesh technologies for the handling of complicated inter-service communications, examining declarative policy frameworks for automated security and compliance enforcement, and creating adaptive resource allocation algorithms balancing performance demands against infrastructure cost. The intersection of these technical advancements with established DevOps practices puts organizations in a position to attain never-before-seen levels of agility in the management of PLM infrastructure, ultimately speeding the delivery of digital twin capabilities that deliver a competitive edge in hardware product development.

Table 1: *Digital Twin Capabilities and Integration Requirements in Hardware Development [3, 4]*

Digital Twin Component	Integration Requirement	Operational Benefit
Physics-based simulations and computational fluid dynamics	Scalable storage architectures and high-performance computing resources	Early-stage design validation and reduced engineering change orders
PLM platforms with geometric models and bill-of-materials	Data management infrastructure for heterogeneous sources	Improved traceability throughout design evolution and production changes
Real-time monitoring and operational telemetry	Efficient data pipeline mechanisms for sensor data	Predictive analytics and closed-loop optimization capabilities
Multi-disciplinary collaboration environments	Synchronized PLM-digital twin architectures	Reduced inter-departmental coordination overhead and handoff errors

Table 2: *Infrastructure Automation Framework Components for PLM Deployment [5, 6]*

Automation Component	Technical Capability	Implementation Approach
Infrastructure-as-code methodologies	Version-controlled deployment configurations with peer review	Declarative configuration specifications for system topologies
Container orchestration platforms	Portable application packaging with runtime dependencies	Microservices architecture for loosely coupled components
Continuous integration and deployment pipelines	Comprehensive automated testing across multiple dimensions	Unit, functional, integration, and performance test suites
Declarative state management systems	Automatic drift detection and correction mechanisms	Continuous reconciliation against version-controlled repositories

Table 3: *Implementation Outcomes in Aerospace Hardware Program [7, 8]*

Implementation Aspect	Transformation Achieved	Operational Impact
PLM environment deployment timeline	Substantial reduction from traditional extended periods	Accelerated digital twin capability enablement for engineering teams
Environment consistency and configuration accuracy	Automated detection of discrepancies across development and production	Higher confidence in digital twin results with improved simulation correlation
PLM environment update procedures	Transition from infrequent scheduled maintenance to regular continuous cycles	Minimized disruption to engineering workflows with rapid rollback capabilities
Return on investment profile	First-year savings equivalent to initial framework development investment	Ongoing operational cost reductions from improved stability and reduced incidents

Table 4: *Future Automation Capabilities and Enterprise Integration Challenges [9, 10]*

Emerging Capability	**Technical Foundation**	**Integration Challenge**
Intelligent orchestration with machine learning	PLM system telemetry analysis for configuration optimization	Maturation of underlying technologies and operational expertise requirements
Self-healing infrastructure mechanisms	Automatic resolution of configuration anomalies without human intervention	Balancing autonomous operations with reliability and safety constraints
Enterprise architecture integration	Connectivity with data platforms, analytics systems, and AI frameworks	Diverse data exchange protocols, security requirements, and performance constraints
Containerized microservices architectures	Modular PLM deployments with independent component management	Service mesh technologies for complex inter-service communications

6. Conclusions

Infrastructure automation is a key enabler for the adoption of digital twins in hardware product development, with automated PLM deployment frameworks yielding quantifiable value in deployment speed, environment consistency, and operating stability. The case deployment confirms that accelerating PLM provisioning cycles from long timeframes to short timelines has a dramatic impact on organizational capacity to capitalize on digital twin capabilities for design validation, manufacturing optimization, and collaborative product development. The ongoing integration, declarative management of configurations, and containerization design of the framework resolve inherent difficulties that traditionally obstruct digital twin rollouts, such as inconsistency in environments, long deployment cycles, and intricate change management demands. Organizations launching digital twin programs should place the highest priority on automation capability as the underlying infrastructure that enhances payback from investments in digital twin technology. The deeper implications involve more extensive considerations than individual technical deployments, such as strategic thought for hardware product creation in a more digital and integrated manufacturing world. Those that effectively combine automated PLM deployment with digital twin capabilities are poised to address market opportunities more quickly, iterate better on product designs, and optimize manufacturing operations with more accuracy. Rapid provisioning and the ability to change PLM environments allow for innovation around new digital twin topologies, encourage the uptake of new technologies, and lower obstacles to scaling successful pilots to enterprise programs. As hardware products grow more sophisticated and development cycles are subject to ongoing pressure to speed up, the blending of digital twin technology and automated PLM infrastructure increasingly characterizes competitive excellence in product development.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.
- **Use of AI Tools:** The author(s) declare that no generative AI or AI-assisted technologies were used in the writing process of this manuscript.

References

[1] Igiri Onaji, "Digital twin in manufacturing: conceptual framework and case studies," International Journal of Computer Integrated Manufacturing, 2022. Available: https://www.tandfonline.com/doi/full/10.1080/0951192X.2022.2027014

[2] Zara Raza, "PLM Implementation Best Practices: A Complete Guide," Converged, 2024. Available: https://converged.propelsoftware.com/blogs/plm-implementation-best-practices-a-complete-guide

[3] Angira Sharma, et al., "Digital Twins: State of the art theory and practice, challenges, and open research questions," Journal of Industrial Information Integration, 2022. Available:

https://www.sciencedirect.com/science/article/pii/S2452414X22000516

[4] Jan Molter, et al., "The Digital Twin - a production-related review," Procedia CIRP, 2024. Available: https://www.sciencedirect.com/science/article/pii/S221282712400684X

[5] Diomidis Spinellis, "Version control systems," IEEE Software, 2005. Available: https://www2.dmst.aueb.gr/dds/pubs/jrnl/2005-IEEESW-TotT/html/v22n5.pdf

[6] David Jaramillo, et al., "Leveraging microservices architecture by using Docker technology," IEEE Xplore, 2016. Available: https://ieeexplore.ieee.org/document/7506647

[7] M. Lokesh Gupta, et al., "Continuous Integration, Delivery and Deployment: A Systematic Review of Approaches, Tools, Challenges and Practices," ResearchGate, 2024. Available:https://www.researchgate.net/publication/380062081_Continuous_Integration_Delivery_and_Deployment_A_Systematic_Review_of_Approaches_Tools_Challenges_and_Practices

[8] Lianping Chen, "Continuous Delivery: Huge Benefits, but Challenges Too2015," ResearchGate, 2015. Available: https://www.researchgate.net/publication/271635510_Continuous_Delivery_Huge_Benefits_but_Challenges_Too

[9] Mikko Pirhonen, "The Pains and Gains of Microservices Revisited," ACM Digital Library, 2024. Available: https://dl.acm.org/doi/10.1007/978-3-031-78386-9_16

[10] Muhammad Waseem, et al., "A Systematic Mapping Study on Microservices Architecture in DevOps," Journal of Systems and Software, 2020. Available: https://www.sciencedirect.com/science/article/abs/pii/S0164121220302053