

A Gate-Based Release Management Framework for Multi-Site Robotic System Deployment

Livia Silveira Bezerra de Menezes

Engineering Project Manager
Efficient Plant Inc. (EPI)
Laval, Quebec, Canada
livasbm@hotmail.com

Rogério Teixeira Mâsih

Adjunct Professor II in the Department of Production Engineering
Universidade Federal do Ceará
Fortaleza, Ceará, Brazil
rogeriom@ufc.br

Abstract

Deploying autonomous robotic systems across multiple distribution center sites simultaneously requires a structured approach to managing software releases, hardware changes, and system updates without compromising operational continuity. This paper presents the Multi-Site Release Management Framework (MSRMF), a gate-based release management framework developed and applied during the rollout of an automated warehouse robotic platform across 26 distribution center sites. The framework integrates systematic ticket triage with formal inclusion/exclusion authority, enabling a 75% reduction in push-through incidents across the network. Key elements include a multi-dimensional triage protocol for classifying change requests, defined gate criteria for release inclusion, formal decision authority structures, and rollback protocols. The framework is discussed in the context of its design, implementation, and operational outcomes, with implications for practitioners managing multi-site robotic deployments.

Keywords

Release management, warehouse automation, multi-site deployment, supply chain, change management.

1. Introduction

The deployment of autonomous robotic systems in warehouse and distribution environments presents a class of engineering management challenges distinct from traditional software release management. Unlike conventional enterprise software, robotic systems operate in physical environments where a software defect, hardware misconfiguration, or unvalidated system update can cause immediate operational disruption — including product damage, system downtime, and safety incidents.

When a single robotic platform serves multiple client sites simultaneously, the complexity compounds: a release that performs correctly in one site configuration may behave differently in another, given variations in facility layout, product mix, and operational parameters. The question of what goes into a release, and who decides, becomes a critical operational and risk management function, not merely a technical one.

This paper describes MSRMF, a gate-based release management framework, applied during the multi-site deployment of an automated warehouse robotic system across 26 distribution center sites. The framework emerged from the need

to manage over 500 change requests across a geographically distributed network while maintaining operational continuity and minimizing incident rates.

1.1 Objectives

This paper pursues three objectives: (1) to present a gate-based structure — triage, validation, and release authorization — for managing change requests across a multi-site robotic deployment; (2) to identify which design element of that structure is most strongly associated with observed incident reduction; and (3) to validate the framework against a real production deployment dataset of 546 triaged tickets across 26 sites, establishing what outcomes it produced and under what limitations those outcomes should be interpreted. These objectives are revisited explicitly in the Conclusion.

2. Literature Review

Managing what changes are allowed into a system, and when, has been studied from three largely separate angles: software release management, engineering change management for physical products, and operations/supply chain coordination. This section reviews each before identifying the gap this paper addresses.

Software release management has been extensively studied within software engineering. Ruhe and Saliu (2005) frame release planning as a multi-criteria optimization problem balancing feature scope against schedule and resource constraints, while van der Hoek et al. (1997) provide an early formalization of release management as a distinct software engineering discipline. Humble and Farley (2010) and Kim et al. (2016) describe continuous delivery and DevOps practices that automate and accelerate the release pipeline. This literature is directly relevant to the gate structure proposed in this paper, but it universally assumes software-only systems, where a defective release can typically be rolled back without physical consequence — an assumption that does not hold for robotic systems operating on a physical warehouse floor.

A second, largely separate literature addresses engineering change management (ECM) for physical products. Jarratt et al. (2011) provide a widely cited categorized overview of the ECM literature, and Hamraz et al. (2013) extend this into a holistic categorization framework spanning product, process, and organizational perspectives on change. Reddi and Moon (2009) propose a framework specifically for managing the propagation of engineering changes across interdependent components — a concern closely related to the cross-site propagation risk addressed in this paper, but developed in the context of new product development rather than in-service robotic fleets. This ECM literature offers formal change-classification and propagation-analysis concepts, but has not been applied to the specific problem of a single robotic platform operating simultaneously across dozens of physically distributed, operationally live production sites.

A third stream situates change and release decisions within operations and supply chain management more broadly. Chase et al. (2018) provide the standard operations management treatment of process control and continuous improvement, and Zhang et al. (2003) examine the relationship between manufacturing flexibility and customer satisfaction, relevant to how a platform vendor balances release stability against client-specific enhancement requests. The warehouse automation literature — Boysen et al. (2019), de Koster et al. (2007), Fragapane et al. (2021), Roodbergen and Vis (2009), Tompkins et al. (2010), and Wurman et al. (2008) — documents the operational scale and configuration diversity of the multi-site networks within which release decisions of this kind must be made, but does not itself address change governance.

Taken together, existing literature provides release-planning methodology for software-only systems, change-propagation and categorization methodology for physical products in development, and operational context for multi-site warehouse networks — but no prior work combines these three elements into a governance structure for a single robotic platform undergoing continuous change while already live across a distributed physical network. This paper addresses that gap through a practitioner-derived, empirically validated gate structure.

3. Methods

The framework organizes the release management process into three sequential gates, each with defined entry criteria, evaluation activities, and formal decision authority, illustrated in Figure 1.

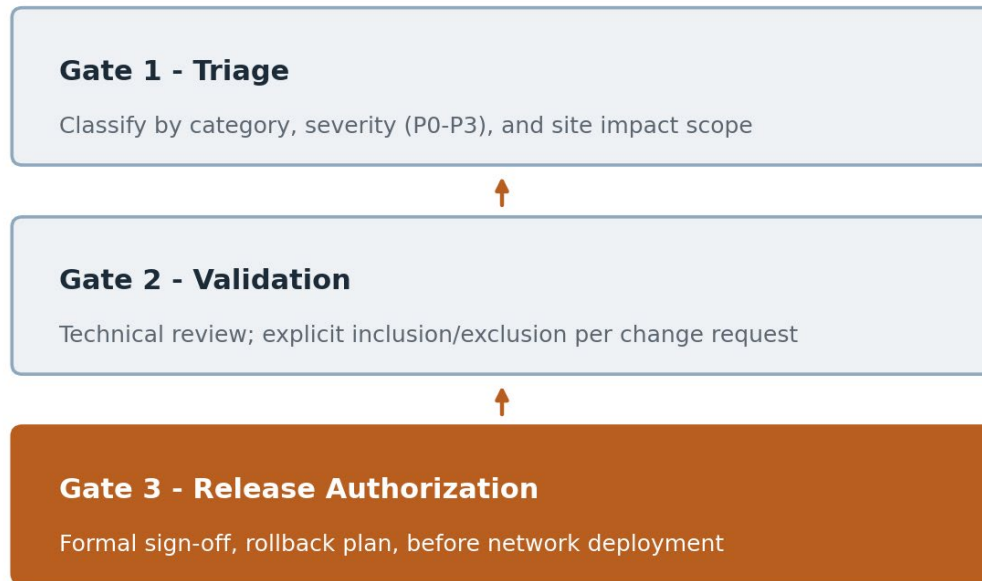


Figure 1. Three-gate release management structure: triage, validation, and release authorization.

Gate 1 — Triage classifies incoming change requests along three dimensions: category (software defect, hardware update, configuration change, enhancement request, or safety-critical item), severity (P0 operational blocker through P3 minor/cosmetic), and scope (network-wide, client-specific, site-specific, or single-unit).

Gate 2 — Validation is a technical review in which a designated Release Manager makes an explicit inclusion or exclusion decision for each change request, independent of the originating team's preference. This role requires cross-functional technical understanding spanning mechanical, electrical, software, vision, and safety engineering, in order to weigh trade-offs across the multi-disciplinary system.

Gate 3 — Release Authorization is a formal checkpoint requiring complete documentation of all included changes, sign-off from the Release Manager, confirmation of the rollback procedure, and operational readiness confirmation from site representatives. No release proceeds to network deployment without Gate 3 completion.

4. Data Collection

Data were collected from the production change-tracking system used during the deployment period, covering 26 operational distribution center sites. Every change request entering Gate 1 was logged with its category, severity, and scope classification, together with the Gate 2 inclusion/exclusion decision, requesting team, and Gate 3 authorization status. Incident data were collected separately through the network's operational incident reporting process, covering push-through incidents before and after the framework's implementation, and rollback events following each network release.

In total, 546 change requests were triaged and tracked through the three-gate structure over the deployment period. This dataset forms the basis for the results reported in Section 5.

5. Results and Discussion

5.1 Numerical Results

Table 1 summarizes the outcome metrics from the production deployment period. The reduction in push-through incidents correlated most strongly with the implementation of formal Gate 2 inclusion/exclusion authority, discussed further in Section 5.3.

Table 1. Outcome metrics from the production deployment period.

Metric	Value	Context
Total tickets triaged	546	Over the full deployment period
Sites covered	26	Operational distribution centers
Push-through incident reduction	75%	vs. pre-framework baseline
Rollback events required	Near-zero	Following Gate 3 formalization

5.2 Graphical Results

Figure 2 presents the push-through incident rate indexed to a pre-framework baseline of 100, showing the 75% reduction achieved following framework implementation. The index form is used because absolute incident counts are considered client-confidential; the relative reduction is the metric of engineering significance and is reported without alteration.

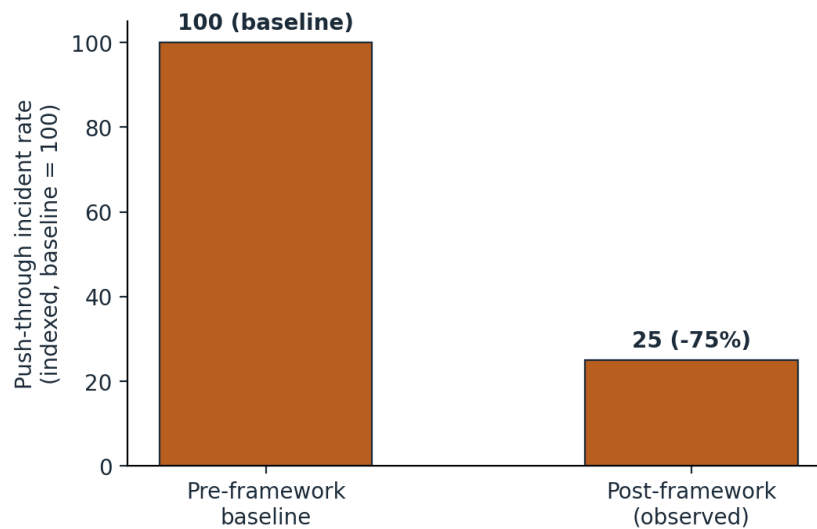


Figure 2. Push-through incident rate before and after framework implementation, indexed to a pre-framework baseline of 100.

5.3 Proposed Improvements

Analysis of incident patterns before and after framework implementation indicates that the structure of the decision process, rather than the content of individual releases, was the primary driver of outcome improvement. Consensus-based processes prior to framework implementation had consistently introduced changes that an individual technical reviewer would have excluded. The proposed improvement embedded in the framework — formalizing a single Release Manager's inclusion/exclusion authority, independent of the originating team's preference — directly targets this failure mode. This is consistent with the engineering change management literature's finding that ambiguous or distributed change authority is a recurring source of uncontrolled change propagation (Jarratt et al. 2011; Reddi and Moon 2009), and extends that finding from the product-development context in which it was originally documented to a live, multi-site operational context.

5.4 Validation

The framework was validated against the 546-ticket production dataset described in Section 4, covering the full 26-site deployment period. The 75% push-through incident reduction and near-zero rollback rate following Gate 3 formalization are the primary validation outcomes. This is a single-case, before/after comparison rather than a controlled experiment with a counterfactual network operating without the framework; other contemporaneous factors

at the deployment sites cannot be fully ruled out as contributing causes, and this limitation is stated explicitly rather than implied.

Positioned against prior work, the outcome is consistent with the software release management literature's general finding that structured release gating reduces defect escape rate (Ruhe and Saliu 2005; Humble and Farley 2010), while extending that finding to a context — a live physical multi-site robotic network — not covered by that literature. It is also consistent with the engineering change management literature's finding that formal, individual change authority reduces uncontrolled change propagation (Hamraz et al. 2013), providing a rare empirical data point for that claim outside the new-product-development context in which it is typically studied. Generalizability across different robotic platform architectures, regulatory environments, and network scales would benefit from further empirical study, and the framework's performance in environments with substantially higher ticket volumes or more complex multi-vendor architectures has not been evaluated.

6. Conclusion

This paper presented MSRMF, a gate-based release management framework for multi-site robotic system deployments, validated against a production deployment dataset of 546 triaged change requests across 26 distribution center sites. Returning to the three objectives stated in Section 1.1: the three-gate structure was presented in full, with explicit entry criteria and decision authority at each gate (Objective 1); the formalization of Gate 2 inclusion/exclusion authority was identified as the design element most strongly associated with the observed 75% incident reduction (Objective 2); and the framework was validated against real production data, with the single-case, before/after nature of that validation stated explicitly as a limitation (Objective 3).

For practitioners managing expanding autonomous system deployments, the central recommendation is structural: invest in defining decision authority before scaling, not after incidents occur. The gate-based approach provides a repeatable structure that can be adapted to different platform architectures while preserving the core benefit of formalized change control.

To support independent evaluation, replication, and pilot testing by practitioners and researchers, an open-source reference implementation of MSRMF has been made publicly available at <https://liviabm.github.io/msrmf-console/>.

References

- Boysen, N., de Koster, R. and Weidinger, F., Warehousing in the e-commerce era: a survey, *European Journal of Operational Research*, vol. 277, no. 2, pp. 396-411, 2019.
- Chase, R., Aquilano, N. and Jacobs, F., *Operations and Supply Chain Management*, McGraw-Hill Education, New York, 2018.
- de Koster, R., Le-Duc, T. and Roodbergen, K. J., Design and control of warehouse order picking: a literature review, *European Journal of Operational Research*, vol. 182, no. 2, pp. 481-501, 2007.
- Fragapane, G., de Koster, R., Sgarbossa, F. and Strandhagen, J. O., Planning and control of autonomous mobile robots for intralogistics: literature review and research agenda, *European Journal of Operational Research*, vol. 294, no. 2, pp. 405-426, 2021.
- Hamraz, B., Caldwell, N. H. M. and Clarkson, P. J., A holistic categorization framework for literature on engineering change management, *Systems Engineering*, vol. 16, no. 4, pp. 473-505, 2013.
- Humble, J. and Farley, D., *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*, Addison-Wesley, Boston, 2010.
- Jarratt, T. A. W., Eckert, C. M., Caldwell, N. H. M. and Clarkson, P. J., Engineering change: an overview and perspective on the literature, *Research in Engineering Design*, vol. 22, no. 2, pp. 103-124, 2011.
- Kim, G., Humble, J., Debois, P. and Willis, J., *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*, IT Revolution Press, Portland, 2016.
- Reddi, K. R. and Moon, Y. B., A framework for managing engineering change propagation, *International Journal of Innovation and Learning*, vol. 6, no. 5, pp. 461-476, 2009.
- Roodbergen, K. J. and Vis, I. F. A., A survey of literature on automated storage and retrieval systems, *European Journal of Operational Research*, vol. 194, no. 2, pp. 343-362, 2009.
- Ruhe, G. and Saliu, M. O., The art and science of software release planning, *IEEE Software*, vol. 22, no. 6, pp. 47-53, 2005.

- Tompkins, J. A., White, J. A., Bozer, Y. A. and Tanchoco, J. M. A., *Facilities Planning*, 4th ed., Wiley, Hoboken, 2010.
- van der Hoek, A., Hall, R. S., Heimbigner, D. and Wolf, A. L., Software release management, *ACM SIGSOFT Software Engineering Notes*, vol. 22, no. 6, pp. 159-175, 1997.
- Wurman, P. R., D'Andrea, R. and Mountz, M., Coordinating hundreds of cooperative, autonomous vehicles in warehouses, *AI Magazine*, vol. 29, no. 1, pp. 9-19, 2008.
- Zhang, Q., Vonderembse, M. A. and Lim, J. S., Manufacturing flexibility: defining and analyzing relationships among competence, capability, and customer satisfaction, *Journal of Operations Management*, vol. 21, no. 2, pp. 173-191, 2003.

Biographies

Livia Silveira Bezerra de Menezes is an Engineering Project Manager specializing in R&D program management for supply chain automation and warehouse robotics systems. She holds a Master of Industrial Engineering from Concordia University (Montreal) and a BSc in Mechanical Production Engineering from Universidade Federal do Ceara (Brazil). She is a PMP-certified Project Management Professional, a Lean Six Sigma Green Belt (McGill University), and a licensed engineer (Ordre des Ingenieurs du Quebec — OIQ).

Dr Rogério Teixeira Mâsih is an Adjunct Professor II in the Department of Production Engineering, Center of Technology, at the Universidade Federal do Ceará (UFC), Brazil. He holds a BSc in Mechanical Engineering from UFC (1997) and an MSc (1999) and PhD (2005) in Production Engineering from the Universidade Federal de Santa Catarina (UFSC), where his doctoral research focused on strategic competency mapping and its integration with the Balanced Scorecard for competency-based human resource management. He has taught and advised graduate and undergraduate students in production engineering, logistics, and operations management at UFC since 2006, and served as Associate Dean for University Extension (Pró-Reitor Adjunto de Extensão) from 2013 to 2023 and as Coordinator of UFC's Internship Agency from 2012 to 2023. His research and teaching interests include strategic cost management, performance measurement systems, logistics, and production planning and control.