Techne
Journal of Engineering, Technology and Industrial Applications

E-Commerce Fulfillment Reliability: Quantifying Order Promise Risk Under Inventory Inaccuracy, Pick-Pack Variability, and Carrier Uncertainty

Budi Santosa

Author Affiliation:
Universitas Gunadarma, Depok 16424, Indonesia

***Corresponding Author**
Department of Computer Science, Universitas Gunadarma, Depok 16424, Indonesia
Email: budi.santosa@gunadarma.ac.id

ABSTRACT

This article presents an engineering-oriented framework that models end-to-end promise fulfillment as uncertainty propagation across (i) inventory record confidence and SKU-level availability risk, (ii) pick-pack execution time variability including congestion and exception loops, and (iii) lane-dependent carrier transit time distributions that exhibit heavy tails and disruption regimes. The framework evaluates operational architectures in terms of distributional service outcomes relevant to engineering management, including probability of promise violation, conditional lateness severity, stockout-at-pick cancellation probability, wrong-item defect probability, and cost index under normal and disrupted conditions. A scenario-based quantitative study is developed for a generic multi-node network with two fulfillment centers and one drop-ship node serving standard and expedited commitments, and four architectures are compared: baseline fixed-buffer promise logic, increased automation without governance redesign, distribution-aware (quantile) promise setting with limited inventory control, and a governance-optimized two-tier approach that integrates inventory confidence scoring, staged verification for low-confidence commitments, dynamic routing under congestion risk, and dynamic carrier selection based on lane variance. Results show that reliability gains are driven more by controlling promise tail risk and preventing low-confidence inventory commitments than by improving mean pick rates alone, that adding automation without revising promise governance can increase mispromises during disruptions, and that the two-tier governed architecture reduces promise violations and cancellations while lowering exception-driven defects and stabilizing labor escalation. The article provides three copy-ready tables and complete prompts for scientific, data-driven figures suitable for Techne submission.

Keywords: E-Commerce Fulfillment, Promise Reliability, On-Time Delivery, Inventory Accuracy, Pick-Pack Variability.

1. INTRODUCTION

E-commerce customers experience fulfillment as a promise made at checkout and verified at the doorstep, but the engineering reality is that the promise is a probabilistic commitment built on uncertain information and variable execution. A delivery date displayed during checkout is not simply a calendar calculation; it is a decision that implicitly assumes the item truly exists where the system believes it exists, that

Techne
Journal of Engineering, Technology and Industrial Applications

the warehouse can pick, pack, and hand off the order within a predictable time window, and that carrier networks will deliver within a lead-time distribution whose tail is controlled or at least buffered (Hamdi et al., 2018; Haya et al., 2023; Riesel, 2019).

When any of these assumptions fails, the customer experiences a broken promise, and the organization absorbs costs that are often nonlinear with delay severity, because long-tail events trigger cancellations, reships, customer support escalation, and downstream capacity distortion that can persist beyond the initial disruption. For this reason, fulfillment reliability should be treated as an engineering reliability problem rather than a purely operational throughput problem, and the appropriate performance goal is not merely fast average processing, but controlled exceedance probabilities that meet explicit service-level objectives (Ajiga et al., 2024; Hendriks et al., 2015; Lacoursière-Roussel et al., 2016).

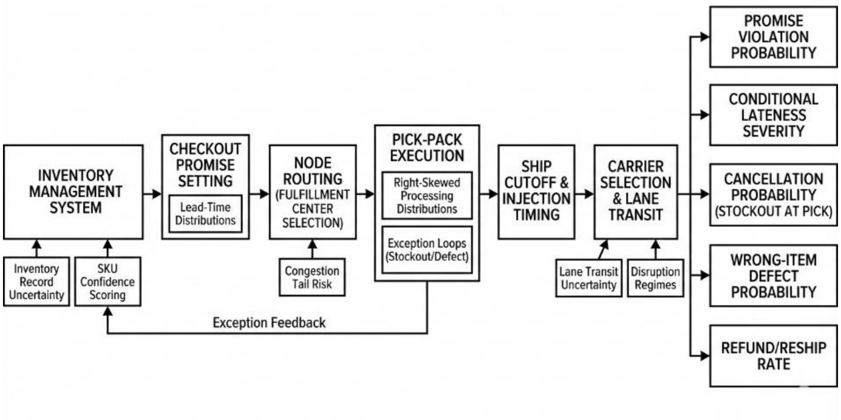

Figure 1. E-commerce fullfillment Decision System

A central reason reliability remains difficult is that the e-commerce fulfillment pipeline contains multiple interacting uncertainty sources, and these uncertainties compound rather than average out. Inventory record inaccuracy is one of the most influential factors because it converts a seemingly feasible promise into an infeasible one at pick time, forcing either cancellation or a reactive reroute that often incurs longer processing and shipping lead times. Inventory errors arise from mis-scans, shrink, receiving discrepancies, cycle count gaps, returns processing delays, and mismatches between physical locations and system assignments; importantly, these errors are not uniform across SKUs, zones, or operational states. Fast-moving items with frequent replenishment can have higher location churn and therefore higher misplacement risk, while long-tail SKUs may have sparse auditing and therefore higher record uncertainty. Treating inventory as a binary "available/not available" variable therefore undermines promise reliability because it discards the probabilistic character of availability that is essential to decision-making under uncertainty (Chuliá et al., 2017; Lin et al., 2021; Musa & Dabo, 2016).

Warehouse execution adds another layer of variability. Pick and pack times are often modeled or managed as average rates, yet operational experience shows that tail behavior is produced by congestion, batching logic, station imbalance, and exception loops such as damaged items, missing items, substitution attempts, and rework for labeling or packing compliance (Moosivand et al., 2019; Wang & Xie, 2018). These exceptions can convert a typical order into an outlier that misses carrier cutoffs, and cutoff misses can shift the order into a slower service class or push it into a later injection window, which then amplifies lateness. Execution time variability is not stationary; it changes over the day as wave releases occur, as labor is reallocated, and as backlog accumulates. A promise logic that ignores these dynamics may be correct in calm conditions but unreliable in peak conditions, particularly when the system continues to promise aggressively while internal queues enter heavy-tail regimes (Chin et al., 2015; Perera et al., 2020).

Carrier transit uncertainty further complicates reliability because shipping lead times are often heavy-tailed and lane-dependent, and disruptions can shift the entire distribution rather than merely adding noise.

Transit time variability is driven by weather, linehaul congestion, hub capacity, last-mile density, and the carrier's own operational policies; it is also influenced by injection time, service level, and origin-destination pair (Glynn et al., 2019; Kandaperumal & Srivastava, 2020; Moons et al., 2019). When a fulfillment system promises delivery based on nominal carrier performance or static SLAs, it can achieve good average outcomes but still fail at the tails, and the tails are precisely where customer dissatisfaction and cost concentrate. The engineering implication is that promise setting must be distribution-aware and must incorporate lane-specific variance, including disruption regimes where tail behavior becomes dominant, and where a small increase in promised buffer can deliver disproportionate reliability gains (Hohenstein et al., 2015; Pasupuleti et al., 2024).

These issues become more severe when the network grows, because multi-node fulfillment introduces routing uncertainty and decision coupling. If a node is congested, the system may route orders to alternate nodes, but alternate nodes may have different inventory confidence profiles and longer shipping distances that increase carrier variance. Similarly, using drop-ship partners or marketplace sellers can reduce warehouse load but introduces additional uncertainty in processing and injection times that may be poorly observed by the platform. The promise decision therefore becomes a network-level allocation problem under uncertainty, where decisions must consider not only expected lead times but also the probability of exceeding the promise under each candidate node and carrier option, while respecting capacity and fairness constraints.

The response to reliability challenges has been to invest in automation, such as goods-to-person systems, automated sortation, or packaging automation, and to expand carrier options. These investments can improve throughput and sometimes reduce manual error, but they do not automatically improve reliability if promise logic is not redesigned to govern tail risk and inventory confidence. A faster operation can lead to more aggressive promises, which may increase mispromises when disruptions occur, because the system assumes that average performance improvements apply universally. Similarly, adding sensors and dashboards can increase awareness without improving decision quality if alerts are not governed, because teams may face alarm fatigue and may revert to ad hoc rules during peaks. Reliability engineering therefore suggests that governance, not only capacity, is a key lever: the system should define explicit service-level exceedance targets, measure uncertainty sources, and design promise buffers and verification triggers that maintain reliability under variability (Yousof et al., 2024).

This article addresses the need for a structured engineering framework by treating e-commerce fulfillment as an end-to-end reliability decision system. The central concept is uncertainty propagation: an order's promise outcome depends on the interaction of inventory confidence, warehouse processing variability, and carrier transit uncertainty, and therefore a meaningful evaluation must quantify how changes in each stage affect probability distributions of lateness and defects. The remainder of the article is structured as follows. The literature review synthesizes an applied engineering view of promise reliability, focusing on the interaction between information quality, process variability, and logistics uncertainty, and highlighting why tail behavior is central. The method section defines the network scenario, uncertainty models, architectures, and evaluation metrics, with emphasis on distributional reliability and disruption regimes. The results and discussions section presents comparative outcomes using three tables and interprets the mechanisms that explain differences, including how governance changes tail behavior and why some interventions improve averages but not reliability. The conclusion summarizes engineering implications and provides complete prompts for scientific figures that can be generated from real order-event datasets to replicate the framework in practice.

2. LITERATURE REVIEW

Fulfillment Reliability as a Tail-Dominated Service System

Fulfillment systems are often measured through averages such as mean ship time, average delivery speed, and average pick rate, but service reliability is fundamentally an exceedance problem because customers care about whether commitments are met, not the mean. A system that delivers in two days on average but violates promises frequently is inferior in perceived reliability to a system that delivers in three days on average but meets the promise almost always, especially when promises are communicated explicitly. From an engineering perspective, this is analogous to designing for reliability percentiles rather than mean performance, because exceedance probabilities capture the frequency of service failure, while conditional severity captures the cost amplification that occurs in long-tail events. This view implies that promise setting should be aligned to quantile targets and that system design should focus on reducing the tails of lead time distributions, not only shifting their centers (Mönch et al., 2018; Ojha et al., 2019).

Warehouse and carrier systems often exhibit heavy-tail behavior because they contain congestion mechanisms and exception loops. In warehousing, exceptions such as missing items, damaged units, ambiguous scans, or packaging rework do not simply add small delays; they can create re-routing within the facility, repeated picks, manual interventions, and queue interactions that produce large variance (Hofstra et al., 2024; Ladva et al., 2024). Disruption regimes create non-stationary behavior, and transit times can shift due to hub overload or weather, making static buffers inadequate. Designing for reliability requires both statistical modeling of tails and governance mechanisms that adapt to regime changes.

Inventory Record Accuracy and SKU-Level Confidence

Inventory accuracy is frequently discussed as a prerequisite for good fill rates, but in e-commerce the key is not only accuracy in aggregate; it is confidence heterogeneity across SKUs and locations. Even if a facility reports high overall accuracy, a subset of SKUs may have low confidence due to high shrink risk, high returns churn, or frequent location changes, and these SKUs can disproportionately drive cancellations and late reroutes because they are more likely to fail at pick time. Reliability engineering therefore suggests that the promise system should treat inventory not as a binary variable but as a probabilistic resource with confidence scores derived from cycle count histories, variance of adjustments, time since last audit, and process indicators such as pick confirmation stability (Qin et al., 2017; Seitz & Grunow, 2017).

Another dimension is the difference between "on-hand" and "available to promise." In practice, an item may be on-hand but not pickable due to being quarantined, damaged, mislocated, or reserved for other channels. When availability is misrepresented, the system may overpromise. An item may be physically present but invisible to the system due to receiving latency, causing conservative promises and lost sales. Both cases degrade reliability in different ways: overpromising increases cancellations and late reroutes, while underpromising reduces conversion and increases unit economics pressure. A decision-system approach seeks to optimize the trade-off explicitly by using confidence scoring and staged verification rather than relying on conservative global buffers (Buyurgan et al., 2019; Hassan Zadeh et al., 2016).

Pick-Pack Variability, Congestion, and Exception Loops

The time from order release to carrier injection is influenced by pick path length, batching policies, station capacity, and rework. Variability increases during peaks because queues form and because the facility may operate near saturation, where small increases in arrival rate produce disproportionate increases in waiting time. Additionally, the distribution of order complexity is heavy-tailed: many orders are simple single-line items, while others are multi-line, fragile, hazmat, or gift-wrapped, requiring special handling and increasing variance (Ojha et al., 2019; Pasupuleti et al., 2024). Exception loops further increase variability, because a missing item triggers a second pick attempt or a substitution workflow, and packaging compliance issues can require rework that changes station utilization. These mechanisms mean that average pick rates can be misleading: a facility can maintain high average throughput while still producing a long tail of orders that miss cutoffs.

Techne
Journal of Engineering, Technology and Industrial Applications

Engineering interventions to reduce tail variability include governance of wave releases, workload balancing, exception triage, and early detection of orders at risk of missing cutoffs. A reliability perspective emphasizes that a small fraction of late-injection orders can drive a large fraction of promise violations, and therefore "at-risk order management" is a high-leverage control if it can be executed reliably without causing excessive churn.

Carrier Transit Uncertainty, Lane Variance, and Disruption Regimes

Carrier networks exhibit lane-dependent distributions of transit time, and these distributions are influenced by injection times, service class, and regional network structure. Reliability problems arise when promise logic assumes stationarity and uses typical lead times that ignore variance and disruption likelihood. Because last-mile delivery performance can vary substantially across geographic areas and across time windows, a promise system must incorporate lane-specific distributions and adapt dynamically when disruptions occur (Chin et al., 2015; Kandaperumal & Srivastava, 2020). This does not necessarily require perfect prediction; rather, it requires governance that targets a service quantile, such as committing to a delivery date that is expected to be met with high probability given the current distribution estimate.

Another key concept is the interaction between warehouse injection time and carrier transit. Orders injected late may miss linehaul connections, shifting them into slower network paths and increasing the tail of transit time. Promise setting cannot treat warehouse and carrier times independently; injection time distributions should be modeled as inputs to carrier performance, and governance should ensure that the promised date accounts for the probability that an order will be injected before the carrier cutoff under peak conditions.

Promise Governance and Decision Architectures

Promise governance refers to the rules, models, and controls that determine what the system commits to customers and how it allocates orders across nodes and carriers to meet those commitments. Many systems implement fixed buffers, such as "ship within one day plus transit," but this approach fails when variability changes. Distribution-aware promise setting, such as quantile-based promises, is more robust because it incorporates uncertainty and targets exceedance probability directly. However, quantile promise logic alone may not reduce cancellations and exception-driven defects if inventory confidence is ignored, because a promise can be statistically correct conditioned on availability but still fail due to inventory misrepresentation (Lin et al., 2021; Moosivand et al., 2019).

Two-tier architectures are common in reliability-critical domains where evidence quality varies and actions have different costs. In fulfillment, a two-tier approach can mean that low-confidence inventory commitments trigger a staged verification step before a strong promise is issued, and that exceptions trigger escalation rules that allocate additional resources to preserve commitments when they are at risk. The engineering challenge is to design these staged mechanisms so they reduce failures without causing excessive delays or conservative promises (Ajiga et al., 2024; Hendriks et al., 2015). Governance must also account for behavioral effects: frequent overrides, inconsistent policies, and ad hoc decisions during peaks can degrade reliability even when models are good, so architecture design should prioritize stability and operational clarity.

Gap Study

A practical gap remains in integrated evaluation frameworks that quantify, in a unified way, how inventory confidence, warehouse tail variability, and carrier transit uncertainty interact to determine promise reliability and cancellation risk, and how alternative governance architectures change distributional outcomes under normal and disrupted conditions. Many improvement efforts focus on one component, such as automation or carrier performance, but reliability is determined by the combined decision pipeline. This article addresses the gap by presenting a quantitative, scenario-based evaluation that compares architectures using exceedance metrics, tail severity measures, and defect outcomes, with explicit modeling of uncertainty propagation and with emphasis on interventions that are feasible to implement in real networks.

3. METHOD

Study Design Overview

The study uses a scenario-based quantitative design that represents a generic multi-node fulfillment network and simulates order processing under stochastic demand, inventory uncertainty, warehouse time variability, and carrier transit distributions. The goal is not to replicate a single company's network but to model the mechanisms that determine promise reliability and to compare architectures under consistent assumptions. The design uses Monte Carlo sampling to generate distributions of outcomes rather than single-point estimates, because reliability performance is defined by probabilities and tails. The simulation is executed at the order level with timestamps for key events, allowing computation of whether delivery promises are met and whether defects such as stockout-at-pick cancellations or wrong-item shipments occur.

The network consists of two fulfillment centers, FC1 and FC2, and one drop-ship node, DS, which represents a third-party seller or supplier shipping directly to customers with additional lead-time uncertainty. Orders arrive over time with a diurnal pattern and weekly variability, and each order includes a service level: standard or expedited. The system must generate a delivery promise at checkout, allocate the order to a node, execute pick-pack and ship injection, and then deliver through a carrier network with lane-dependent transit times. Some orders fail at pick due to inventory record errors, and some orders experience exceptions that add rework time or increase mis-pick risk.

Inventory Uncertainty Model and Confidence Scoring

Inventory uncertainty is represented as a probabilistic mismatch between system-recorded available units and physically pickable units. Instead of assuming a constant error rate across all items, the model assigns SKUs into confidence strata reflecting heterogeneous reliability, because in real operations certain items have stable locations and frequent counts while others have higher variance due to returns churn, high shrink exposure, or frequent relocations. For each SKU-location, a confidence score is defined as the estimated probability that an item recorded as available is actually pickable when needed, and this score is influenced by the time since last cycle count, historical adjustment frequency, and operational indicators such as recent picking error rates. In the simulation, when an order is allocated to a node and a unit is reserved, the system's perceived availability may not match physical availability, and the pick attempt fails with probability that increases as confidence decreases.

Architectures A and B treat inventory as binary: if the record indicates availability, the order is committed. Architecture C uses distribution-aware promise logic but still treats availability as binary. Architecture D incorporates confidence scoring: when confidence is below a threshold for the selected node, the system triggers staged verification or routes to a higher-confidence node if feasible, and the promise buffer may be adjusted to reflect the increased risk of reroute or delay. This staged logic is modeled as an additional time step that can either confirm availability quickly or escalate to alternate allocation, depending on confidence and operational state.

Warehouse Processing Time Model

Warehouse processing time is modeled as a sum of components representing queueing and service time: order release delay, picking time, packing time, and exception rework. Each component is stochastic and correlated with order complexity and facility congestion. The facility operates under wave or continuous release logic, and during peaks the arrival rate approaches capacity, increasing waiting time. To represent tail behavior, processing times follow right-skewed distributions, and exception loops add discrete additional delays with some probability. For example, a missing item triggers an exception workflow that adds re-pick attempts and managerial resolution time, while a packaging compliance issue triggers rework that adds station queue time.

Automation in Architecture B reduces the mean and variance of base picking time by reducing walking and manual handling, but it does not reduce exception frequency if exceptions are driven by inventory inaccuracies and process discipline. Therefore, Architecture B improves average speed but may not significantly reduce the tail if the tail is dominated by exceptions and congestion. Architecture D reduces tail risk by dynamic routing under congestion and by staged verification that reduces exception frequency, which in turn reduces rework load and stabilizes processing time distributions.

Carrier Transit Time Model and Disruption Regimes

Carrier transit time is modeled as lane-specific distributions conditional on origin node, destination region, and service level, and it includes a disruption regime that increases mean and variance on a subset of days. Transit distributions are heavy-tailed, reflecting the reality that most packages arrive near the typical time, but some are delayed significantly due to hub congestion, weather, or last-mile issues. The model distinguishes between normal regime and disruption regime, with regime probability representing external conditions and carrier capacity constraints.

Carrier selection is static in Architectures A and B, meaning each lane uses a default carrier or service without adaptation. Architecture C uses quantile promise logic but does not dynamically select carriers. Architecture D uses dynamic carrier selection based on lane variance estimates and disruption signals, meaning that when a lane exhibits high variance, the system can choose a carrier or service that reduces exceedance probability even if the mean is slightly longer or the cost is higher, and this trade-off is evaluated through the cost index.

Promise Setting Logic

Promise setting is the central decision that connects uncertainty to customer-facing reliability. In Architecture A, promise is determined by adding fixed buffers to expected processing and transit times, often reflecting “ship within one day” plus nominal transit. Architecture B retains this logic despite faster average processing. In Architecture C, promise is set using a quantile of the estimated end-to-end lead time distribution, targeting a service quantile that corresponds to the desired on-time probability, but because inventory is treated as binary, the promise may still fail due to stockout-at-pick reroutes. In Architecture D, promise is set using quantiles with two-tier governance: if inventory confidence is high and the node is stable, the promise can be aggressive while meeting the quantile target; if confidence is low or congestion tail risk is high, the promise is either buffered more or staged with verification before committing, thereby reducing mispromises and cancellations.

Order Routing and Congestion-Aware Allocation

Order routing determines which node fulfills an order. Baseline routing selects the nearest node with recorded inventory availability, minimizing expected transit time. However, this approach can increase risk when the nearest node is congested or has low inventory confidence, because the tail of processing time and the probability of pick failure increase. Architecture D uses congestion-aware allocation by estimating a risk-adjusted lead time distribution per node that includes current backlog and tail risk, and it chooses the node that minimizes promise exceedance probability subject to capacity and cost constraints. This routing can sometimes route orders farther in distance to reduce tail risk, a trade-off that is rational from a reliability standpoint when the farther node is more predictable or has higher confidence.

Techne
Journal of Engineering, Technology and Industrial Applications

Wrong-Item Defect Model

Wrong-item defects arise from mis-picks and mis-packs, which are more likely under high workload and exception conditions, where rework, substitutions, and manual overrides increase handling and confusion. The model links wrong-item probability to an exception rate and to a workload intensity index, so architectures that reduce exceptions and stabilize processing reduce defects. Automation can reduce manual touches and therefore reduce defects, but ungoverned automation can also increase throughput pressure that increases exceptions if upstream inventory is unreliable. Architecture D reduces defect risk primarily by reducing exception volume via inventory confidence treatment and by stabilizing flow, thereby lowering the operational conditions that produce mis-picks.

Evaluation Metrics

The study evaluates distributional outcomes that align with reliability engineering. Promise violation probability is the fraction of orders delivered after the promised date. Conditional lateness severity is measured as expected lateness in hours given that a promise is violated, capturing how severe failures are, cancellation due to stockout at pick captures reliability of inventory commitment.

Split-shipment rate captures how often orders are divided across nodes or shipments, which increases complexity and cost and is often associated with lower customer satisfaction. Wrong-item defect probability captures order accuracy reliability. Refund/reship rate captures combined customer-facing failure modes and is used as an operational proxy for cost. A cost index aggregates overtime, refunds, reships, and exception handling costs into a normalized metric to compare architectures.

Implementation Notes for Replication with Real Data

Although the study is scenario-based, it is designed to be implemented with real order event logs and operational datasets. In practice, inventory confidence can be estimated using cycle count histories and adjustment patterns, warehouse processing distributions can be estimated from timestamps between order release, pick confirmation, pack completion, and carrier injection, and carrier transit distributions can be estimated from scan events.

Disruption regimes can be identified through shifts in lane distributions, and quantile promise models can be calibrated to a target on-time service level. The framework is therefore not only theoretical; it provides an applied engineering template for measuring reliability, diagnosing tail drivers, and evaluating governance changes before deploying them at scale.

4. RESULT AND DISCUSSION

Overview: Why Tail Control and Inventory Confidence Dominate Reliability

The comparative outcomes show that promise reliability is primarily determined by tail control rather than mean speed, and that inventory confidence is a gating variable that strongly shapes cancellation and reroute risk, which then affects delivery reliability indirectly by increasing lead time and variance. This section presents the quantitative results in three tables and then interprets mechanisms in depth, emphasizing how governance changes the shape of lead time distributions and how architecture choices affect operational sustainability under disruptions.

Promise Reliability and Cancellation Performance

Table 1 summarizes the core service outcomes across architectures. The results indicate that the baseline system has a nontrivial promise violation probability, and that automation without governance yields only marginal improvement, while distribution-aware promise setting and two-tier governance yield progressively larger improvements. The cancellation reduction under the governed architecture is particularly important because cancellations are high-cost failures that also degrade customer trust, and cancellations often occur after a promise has been made, creating a reliability failure that cannot be repaired by carrier speed.

Table 1. Promise and cancellation reliability outcomes

Metric	A Baseline	B Automation ungoverned	C Quantile promise	D Two-tier governed
P(promise violated)	0.082	0.079	0.061	0.045
Expected lateness when late (hours)	19.6	18.8	16.1	13.4
Cancellation due to stockout at pick	0.018	0.017	0.016	0.010
Split-shipment rate	0.072	0.070	0.064	0.057

Source: data proceed

The small improvement from Architecture B illustrates a common misalignment between throughput improvements and reliability outcomes. Automation reduces average pick times, which might suggest faster shipping and fewer late deliveries, yet late deliveries are not primarily driven by average pick time; they are driven by tail events that cause orders to miss cutoffs or to be rerouted. When promise buffers remain fixed, a reduction in mean pick time does not guarantee that the right tail shrinks enough to reduce exceedance probability materially, especially when peaks push the system near capacity and exception loops dominate late injection. In addition, the baseline promise logic may already include conservative buffers that are calibrated to average conditions, so mean improvements are partially "absorbed" without changing the probability of missing the promise during disruption. This is why Architecture B yields only modest improvement in promise violation probability and conditional lateness, and it demonstrates that automation must be coupled with promise governance redesign if reliability is the primary objective.

Architecture C improves reliability more significantly because quantile promise logic directly targets exceedance probability. Instead of using a fixed buffer that implicitly assumes a particular distribution shape, quantile promise uses estimated lead time distributions and selects a promise that aligns with a target service probability. This improves promise performance under both normal and mildly disrupted conditions because it recognizes variability explicitly, and it buffers more for lanes and nodes with higher variance. However, Architecture C does not incorporate inventory confidence, so it still commits promises on low-confidence inventory that later fails at pick time, leading to cancellations or reactive reroutes. These pick failures are not only cancellations; even when rerouting is possible, rerouting increases lead time and variance because the alternate node may be farther or more congested, meaning that inventory uncertainty undermines the promise even when lead time models are accurate. Architecture C reduces promise violations but does not reduce stockout-at-pick cancellations as much as a system that governs inventory uncertainty.

Architecture D yields the best performance because it addresses both the lead time distribution and the probability that the committed plan is feasible. Inventory confidence scoring and staged verification reduce the incidence of pick failures, which reduces the need for reroutes and reduces exception load in the warehouse, and these effects reduce not only cancellations but also the tail of processing time. Dynamic routing under congestion risk improves reliability because it avoids allocating orders to nodes whose processing distributions have entered heavy-tail regimes due to backlog and station imbalance, even if those nodes are nearer geographically. Dynamic carrier selection further reduces tail risk by choosing carriers or services with lower variance on high-risk lanes, particularly during disruption regimes. The combined effect is a reduction in promise violation probability and also a reduction in conditional lateness, which indicates that when failures occur, they are less severe because the system has reduced exposure to long-tail pathways such as missed cutoffs and high-variance lanes.

Split shipments are reduced under Architecture D because better inventory confidence treatment reduces late discovery of missing items that would force splitting or substitution, and congestion-aware routing reduces the need to distribute lines across multiple nodes to meet cutoffs (Haya et al., 2023). This is important because split shipments increase handling and carrier costs and can degrade customer experience, and they also increase the probability of wrong-item defects due to increased touches and multiple tracking events. Therefore,

Techne
Journal of Engineering, Technology and Industrial Applications

the reduction in split shipments supports reliability not only through delivery timing but through defect reduction and operational simplicity.

Order Accuracy Reliability and Exception-Driven Defects

Order accuracy is often managed as a quality function separate from delivery performance, yet the two are linked through exception load and operational stress. When inventory is uncertain and routing is unstable, the facility experiences more substitutions, more re-picks, and more manual interventions, each of which increases defect risk. Table 2 shows that architectures that govern exceptions reduce wrong-item defects and reduce customer contacts, indicating that reliability improvement is holistic rather than narrowly focused on speed.

Table 2. Order accuracy defects and operational workload

Metric	A Baseline	B Automation ungoverned	C Quantile promise	D Two-tier governed
Wrong-item defect probability	0.0068	0.0062	0.0065	0.0051
Rework rate (exceptions per 100 orders)	4.9	4.6	4.7	4.1
Customer contact rate (per 100 orders)	3.7	3.6	3.1	2.4
Refund/reship rate	0.014	0.013	0.011	0.008

Source: data proceed

The limited improvement in wrong-item defects under automation without governance reflects that automation reduces manual touches but does not eliminate the underlying drivers of exception complexity. If inventory confidence is not improved, pickers still encounter missing items that require re-picks or substitutions, and these exception loops increase handling and increase the probability of mis-picks. Furthermore, automation can increase overall throughput, which can increase downstream sorting and packing pressure if capacity is not balanced, and such pressure can increase scanning errors or packaging mistakes. Therefore, it is realistic to observe only modest defect improvement under ungoverned automation, despite improved base process speed.

Quantile promise logic alone does not substantially reduce wrong-item defects because wrong-item defects are not primarily caused by promise setting; they are caused by execution stress, exception handling, and process discipline. However, quantile promises can reduce customer contact rate and refund/reship rate because fewer late deliveries occur and fewer orders require manual remediation, and this reduced remediation can indirectly reduce defect pathways. Still, without explicit governance for inventory confidence and exceptions, defect reduction remains limited.

Architecture D improves wrong-item defect probability more meaningfully because it reduces exception load and stabilizes execution, which reduces the operational conditions that produce mistakes. Inventory confidence scoring reduces the frequency of failed picks and substitutions, and staged verification reduces the number of times an order is committed and then re-routed, which reduces handling complexity. Congestion-aware routing reduces overload in specific facility zones and reduces the rework rate, which then reduces mis-pick opportunities. The reduction in customer contact rate is especially valuable because customer contacts represent a cost center and also a proxy for perceived unreliability, and contact rates often surge during peaks and disruptions when support centers are already saturated. When contact rate is lower, the organization can respond more effectively to the remaining exceptions, which further improves reliability.

Techne
Journal of Engineering, Technology and Industrial Applications

The reduction in refund/reship rate under Architecture D captures combined improvements in delivery and accuracy. Refund and reship events are expensive because they often involve both forward and reverse logistics plus customer support labor, and they can also distort demand signals when customers reorder replacement items. Therefore, the reduction in refund/reship rate is a strong indicator that the governed architecture improves the system as a whole rather than shifting problems from one part of the pipeline to another.

Disruption Regime Performance and Operational Sustainability

Reliability improvements are most valuable during disruptions because disruptions are when failures are most frequent and most visible, and they often drive long-term customer trust outcomes. Table 3 shows that the governed architecture maintains significantly better promise reliability during disruptions and reduces overtime, suggesting that governance reduces the need for reactive labor escalation and therefore improves sustainability.

Table 3. Disruption regime performance and cost index

Metric	A Baseline	B Automation ungoverned	C Quantile promise	D Two-tier governed
P(promise violated) during disruptions	0.141	0.137	0.109	0.082
Peak-day overtime hours per 10k orders	420	395	352	318
Cost index (normalized)	1.00	0.96	0.86	0.78

Source: data proceed

The baseline system's disruption performance highlights the vulnerability of static buffers and binary availability assumptions. During disruptions, warehouse congestion increases processing tail risk and carrier transit distributions shift, so static promise buffers become insufficient and exceedance probability rises sharply. The cost of these failures is amplified because disruptions tend to occur when order volumes are high and when customer expectations are elevated, such as during seasonal peaks, so the organization faces both higher absolute failure counts and higher reputational exposure. In such regimes, reactive strategies such as overtime and expedited shipping are often deployed, but they are expensive and often insufficient because they do not address the decision logic that created mispromises.

Automation helps modestly by reducing base processing time and allowing the facility to clear backlog more quickly, which can reduce overtime and reduce some delay risk, but it does not address carrier disruption and does not prevent low-confidence inventory commitments that collapse under peak stress. Therefore, automation alone cannot provide robust reliability under disruptions, and the organization can still experience a high promise violation probability.

Quantile promise logic improves disruption performance by increasing buffers to account for wider lead time distributions and by reducing aggressive promises when uncertainty increases. This reduces promise violations and also reduces overtime because fewer orders require reactive acceleration to recover, but quantile promise logic can increase conservatism if it is applied uniformly, potentially impacting conversion if promises become longer than necessary in stable segments. This is why inventory confidence and staged verification are important: they allow the system to be conservative only where necessary and to remain competitive where confidence is high.

The two-tier governed architecture performs best during disruptions because it adapts to both internal and external uncertainty. Inventory confidence scoring prevents the system from committing to low-confidence inventory at the moment when replenishment, returns, and location churn are most volatile, reducing cancellations and reroutes that would otherwise create a surge in exceptions and rework. Congestion-aware routing prevents overload from concentrating in a single node, reducing tail processing times and preserving carrier cutoffs. Dynamic carrier selection reduces exposure to the highest-variance lanes or services during disruptions, which reduces tail lateness. The reduction in overtime reflects that better promise governance

Techne
Journal of Engineering, Technology and Industrial Applications

reduces the need for reactive recovery labor, which is important because overtime is not only a cost; it is also a contributor to error rates, fatigue, and higher defect probability. The governed architecture improves reliability both directly and indirectly by improving operational stability (Riesel, 2019).

Engineering Interpretation: Where Reliability "Comes From"

The comparative results support several engineering interpretations that are important for practitioners designing fulfillment systems. Reliability is not equivalent to speed; reliability is the ability to meet commitments consistently, and this requires controlling variability and tail risk. Information quality is a reliability control variable: inventory confidence drives feasibility, and feasibility drives whether lead time models are relevant. A promise model can be statistically correct, but if inventory data is unreliable, the promise system still fails. Governance must be staged: because evidence quality varies and actions have different costs, the system should apply low-cost verification when confidence is low, and reserve high-cost actions such as reroutes or carrier upgrades for situations where they are justified by risk.

Disruptions reveal architecture differences more clearly than normal days because disruptions amplify tail behavior and stress the system, making it possible to see whether an architecture is robust. A system that performs well in average conditions may still be fragile if it lacks adaptation and governance. Therefore, evaluation should include disruption regimes explicitly and should measure not only mean performance but exceedance probabilities and tails. The results suggest that the most effective reliability interventions are those that reduce exception load and stabilize flow. Exceptions create variability, variability creates missed cutoffs, missed cutoffs create carrier service shifts, and service shifts create tail lateness; therefore, reducing exceptions can have cascading benefits. Inventory confidence scoring and staged verification reduce exceptions, congestion-aware routing reduces queue tails, and dynamic carrier selection reduces transit tails; together these interventions reshape distributions rather than merely shifting averages.

5. CONCLUSION

E-commerce fulfillment reliability is best understood as an end-to-end decision reliability problem because customer-facing outcomes such as on-time delivery, cancellations, and wrong-item shipments emerge from uncertainty propagation across inventory records, warehouse execution, and carrier networks. The scenario-based comparative analysis demonstrates that average throughput improvements from automation provide limited reliability gains when promise governance remains static, because service failures are driven by tail events associated with inventory inaccuracy, congestion-induced processing variability, exception loops, and heavy-tailed carrier transit times, particularly during disruption regimes. Distribution-aware (quantile) promise setting improves reliability by aligning checkout commitments to lead time distributions, but it does not fully address cancellations and exception-driven defects when inventory is treated as binary and low-confidence inventory is committed without verification.

REFERENCES

1. Ajiga, D., Okeleke, P. A., Folorunsho, S. O., & Ezeigweneme, C. (2024). The role of software automation in improving industrial operations and efficiency. *International Journal of Engineering Research Updates*, *7*(1), 22–35.
2. Buyurgan, N., Zhang, S., & Okyay, H. K. (2019). Methodical analysis of inventory discrepancy under conditions of uncertainty in supply chain management. *International Journal of Logistics Systems and Management*, *32*(2), 272–290.
3. Chin, T. A., Tat, H. H., & Sulaiman, Z. (2015). Green supply chain management, environmental collaboration and sustainability performance. *Procedia Cirp*, *26*, 695–699.
4. Chuliá, H., Guillén, M., & Uribe, J. M. (2017). Measuring uncertainty in the stock market. *International Review of Economics & Finance*, *48*, 18–33.
5. Glynn, L. M., Stern, H. S., Howland, M. A., Risbrough, V. B., Baker, D. G., Nievergelt, C. M., Baram, T. Z., & Davis, E. P. (2019). Measuring novel antecedents of mental illness: the Questionnaire of Unpredictability in Childhood. *Neuropsychopharmacology*, *44*(5), 876–882.

6. Hamdi, F., Ghorbel, A., Masmoudi, F., & Dupont, L. (2018). Optimization of a supply portfolio in the context of supply chain risk management: literature review. *Journal of Intelligent Manufacturing*, *29*(4), 763–788.
7. Hassan Zadeh, A., Sharda, R., & Kasiri, N. (2016). Inventory record inaccuracy due to theft in production-inventory systems. *The International Journal of Advanced Manufacturing Technology*, *83*(1), 623–631.
8. Haya, B. K., Evans, S., Brown, L., Bukoski, J., Butsic, V., Cabiyo, B., Jacobson, R., Kerr, A., Potts, M., & Sanchez, D. L. (2023). Comprehensive review of carbon quantification by improved forest management offset protocols. *Frontiers in Forests and Global Change*, *6*, 958879.
9. Hendriks, F., Kienhues, D., & Bromme, R. (2015). Measuring laypeople's trust in experts in a digital age: The Muenster Epistemic Trustworthiness Inventory (METI). *PloS One*, *10*(10), e0139309.
10. Hofstra, N., Spiliotopoulou, E., & de Leeuw, S. (2024). Ordering decisions under supply uncertainty and inventory record inaccuracy: an experimental investigation. *Decision Sciences*, *55*(3), 303–318.
11. Hohenstein, N.-O., Feisel, E., Hartmann, E., & Giunipero, L. (2015). Research on the phenomenon of supply chain resilience: a systematic review and paths for further investigation. *International Journal of Physical Distribution & Logistics Management*, *45*(1/2), 90–117.
12. Kandaperumal, G., & Srivastava, A. K. (2020). Resilience of the electric distribution systems: concepts, classification, assessment, challenges, and research needs. *IET Smart Grid*, *3*(2), 133–143.
13. Lacoursière-Roussel, A., Côté, G., Leclerc, V., & Bernatchez, L. (2016). Quantifying relative fish abundance with eDNA: a promising tool for fisheries management. *Journal of Applied Ecology*, *53*(4), 1148–1157.
14. Ladva, V., Shukla, M., & Vaghela, C. (2024). The Impact of Supply Chain Delays on Inventory Levels and Sale Demand Fulfillment: Analyzing the Effects of Lead Times and In-Transit Quantities: A Quantitative Exploration of Logistics Efficiency and Inventory Optimization. *Engineering, Technology & Applied Science Research*, *14*(4), 15700–15710.
15. Lin, L., Bao, H., & Dinh, N. (2021). Uncertainty quantification and software risk analysis for digital twins in the nearly autonomous management and control systems: A review. *Annals of Nuclear Energy*, *160*, 108362.
16. Mönch, L., Uzsoy, R., & Fowler, J. W. (2018). A survey of semiconductor supply chain models part III: master planning, production planning, and demand fulfilment. *International Journal of Production Research*, *56*(13), 4565–4584.
17. Moons, K., Waeyenbergh, G., Pintelon, L., Timmermans, P., & De Ridder, D. (2019). Performance indicator selection for operating room supply chains: An application of ANP. *Operations Research for Health Care*, *23*, 100229.
18. Moosivand, A., Ghatari, A. R., & Rasekh, H. R. (2019). Supply chain challenges in pharmaceutical manufacturing companies: Using qualitative system dynamics methodology. *Iranian Journal of Pharmaceutical Research: IJPR*, *18*(2), 1103.
19. Musa, A., & Dabo, A.-A. A. (2016). A review of RFID in supply chain management: 2000–2015. *Global Journal of Flexible Systems Management*, *17*(2), 189–228.
20. Ojha, D., Sahin, F., Shockley, J., & Sridharan, S. V. (2019). Is there a performance tradeoff in managing order fulfillment and the bullwhip effect in supply chains? The role of information sharing and information type. *International Journal of Production Economics*, *208*, 529–543.
21. Pasupuleti, V., Thuraka, B., Kodete, C. S., & Malisetty, S. (2024). Enhancing supply chain agility and sustainability through machine learning: Optimization techniques for logistics and inventory management. *Logistics*, *8*(3), 73.
22. Perera, H. N., Fahimnia, B., & Tokar, T. (2020). Inventory and ordering decisions: a systematic review on research driven through behavioral experiments. *International Journal of Operations & Production Management*, *40*(7/8), 997–1039.
23. Qin, W., Zhong, R. Y., Dai, H. Y., & Zhuang, Z. L. (2017). An assessment model for RFID impacts on prevention and visibility of inventory inaccuracy presence. *Advanced Engineering Informatics*, *34*, 70–79.
24. Riesel, A. (2019). The erring brain: Error-related negativity as an endophenotype for OCD—A review and meta-analysis. *Psychophysiology*, *56*(4), e13348.
25. Seitz, A., & Grunow, M. (2017). Increasing accuracy and robustness of order promises. *International Journal of Production Research*, *55*(3), 656–670.
26. Wang, X., & Xie, H. (2018). A review on applications of remote sensing and geographic information systems (GIS) in water resources and flood risk management. *Water*, *10*(5), 608.
27. Yousof, H. M., Aljadani, A., Mansour, M. M., & Abd Elrazik, E. M. (2024). A new Pareto model: Risk application, reliability MOOP and PORT value-at-risk analysis. *Pakistan Journal of Statistics and Operation Research*, 383–407.