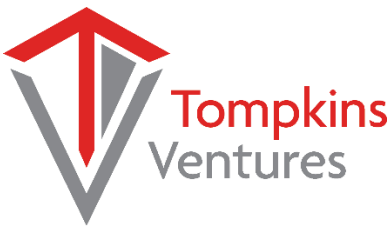
THE WAREHOUSE OPERATING SYSTEM OF THE FUTURE

Software, Optionality and Intelligent Orchestration

Author: Sachin Subhedar
Executive Director, Software Engineering, Tompkins Solutions

Co-author: Jim Tompkins
Chairman, Tompkins Ventures and Tompkins Solutions



Tompkinsventures.com

tompkinsinc.com

Table of Contents

Executive Summary

Warehouse automation has entered a new era – one defined by software-enabled optionality that gives operations the power to pivot instantly as conditions change. In a world shaped by SKU proliferation, labor variability, changing carrier-cutoff times and geopolitical disruptions, rigid "best-path" optimization becomes a liability, while optionality becomes essential for resilience.

Optionality means the ability to run multiple viable operational scenarios – sometimes radically different – without re-engineering systems, rewriting workflows or disrupting the floor. Modern material handling systems now rely on AI-driven orchestration and modular architectures to make decisions in real-time that dynamically reconfigure picking methods, batching rules, routing paths, wave structures and labor allocation. This shift places control back into the hands of warehouse operators, empowering them to respond to spikes, bottlenecks and new constraints quickly and accurately. Productivity gains can reach 20-25% over time.

Optionality reduces the cost of change to near zero.

As a result, software becomes the true differentiator, functioning as the warehouse brain that continuously adapts, predicts and selects the right scenario for the moment. Software that is designed for optionality becomes the key to building supply chains agile enough to handle disruption.

This white paper explores how software is reshaping material handling systems to support optionality, why it matters in practice and how organizations can leverage it to build more adaptive, profitable operations.

Optionality in Software: The Hidden Driver of Digital Agility

For decades, warehouse software was built around a single goal: optimize the flow – one picking method, one batching strategy, one wave structure. But today's environment features demand spikes, SKU proliferation, labor variability and changing carrier cut-off times that reshape order profiles overnight.

In supply chain operations, the most durable advantage isn't a single "optimal" flow – it's the ability to switch among several good options when conditions change. In warehousing, optionality means designing execution logic, integrations and data models to support multiple viable paths (e.g., alternate pick methods, routing strategies or fulfillment priorities) without rewriting core code.

Optionality gives operators levers – configuration, policy and plug-ins – to rebalance throughput, cost and service levels.

In this context, a single optimized flow becomes a liability; organizations need the ability to change flows quickly and safely.

What Optionality Looks Like in Real Operations

Optionality becomes real when software can dynamically reconfigure the warehouse.

No code rewrite. No downtime.

The following scenarios illustrate how this works in practice:

2.1 Diverting Single-SKU Orders to a Fast-Pick Station

A promotion hits; one SKU explodes in volume. Instead of overwhelming the main pick line, the system:

- Identifies all single-SKU orders
- Removes them from the standard order stream
- Routes them to a dedicated fast pick/pack station
- Rebalances labor accordingly

This is optionality: creating a new operational path instantly.

2.2 Changing the Order Picker's Workload

A picker normally handles six large orders at once. When speed trumps cube, the system transitions to:

- Twelve smaller orders per picker
- Different batching logic
- New routing and replenishment triggers

No hardware changes. No downtime. Just software-driven reconfiguration.

2.3 Adjusting Wave Lengths

A 45-minute wave works during steady state; at peak, a 75-minute wave can reduce congestion and increase throughput. Optionality means:

- Changing wave duration
- Re-sequencing order release
- Re-timing replenishment
- Adjusting labor allocation

All without rewriting the WMS.

2.4 Switching Among Picking Modes

As order profiles shift, the system may need to toggle among discrete order picking, batch picking, cluster picking and zone picking – based on real-time conditions and SLA requirements.

Why Optionality Is Hard to Achieve

Legacy systems were built for stability, not adaptability**.**

Older WMS/WES platforms often hard-code workflows; changing a process requires custom development, weeks of testing and increases risk in adjacent operations. Optionality requires modularity, not monoliths.

Optionality is easy to describe – and difficult to engineer.

Making decisions in real-time is complex. To pivot operational modes, software must ingest and reason over order profiles, inventory, labor capacity, equipment constraints, carrier cut-offs and throughput bottlenecks – continuously.

Automation adds constraints. MHE (conveyors, AMRs, AS/RS, sorters) often assumes fixed flows. Optionality demands dynamic routing, configurable logic, API-driven orchestration and software that can override mechanical defaults.

These challenges are why many teams now favor modular approaches with clean seams and only selective distribution.

Designing Software for Optionality

4.1 Scenario-based logic: Support multiple picking strategies, batching algorithms, routing paths and wave structures; activate scenarios via rules and triggers.
4.2 Real-time orchestration: Sense conditions → decide → execute instantly; this is where WES/WCS intelligence is critical.

4.3 Modular, API-driven architecture: Plug-and-play workflows, configurable rules and stable contracts; avoid hard-coded logic and over-distribution.
4.4 Human-centered adaptability: Dynamic labor allocation, skill-based routing, real-time task reprioritization and intuitive operator UIs – because optionality fails if humans can't keep up.

Early enthusiasm for microservices equated greater service granularity with more flexibility. Many teams later discovered the hidden "microservice premium": distributed synchronization, version orchestration and end-to-end observability overhead that slows delivery and increases failure modes.

Pragmatic teams now favor modular monoliths or "sliceable monoliths" – strict internal boundaries, clear module ownership and an evolution path to services only where distribution clearly pays for itself.

Modular architectures concentrate on separation of concerns, stable interfaces and domain-driven boundaries so that optionality (new flows, robots or order types) lands in well-defined extension points – not tangled into the product core. Many teams now favor monolith-first or modular monolith strategies to avoid distributed-systems complexity until scale demands it.

Every alternate execution path multiplies the test matrix.

New options increase complexity – unless the system is designed for them.

Optionality must therefore be disciplined – governed by configuration schemas, compatibility policies and automated test suites (unit, contract and scenario tests). Distributed topologies add network, latency and version-compatibility risks that teams should avoid unless there are clear benefits. Microservice boundaries are difficult to define up front and expensive to refactor later – another reason to develop optionality inside strong module boundaries first.

A practical rule: provide strategic hooks rather than endless toggles. Hooks include REST/GraphQL APIs, events and plug-in points with guardrails (validation, security scopes and idempotency), giving customers freedom to build unique flows without destabilizing the core.

Warehouse platforms must isolate product behavior (repeatable logic) from project customizations (client-specific flows). The WES (execution/orchestration) and WCS (equipment control) split is a natural seam: WES optimizes and sequences work across people, robots and assets; WCS drives real-time device control and routing. Clear boundaries keep core behaviors stable while allowing site-specific acceleration at the edge.

Integration across WMS/WES/WCS is where complexity spikes – especially as you add automation from different vendors with different interfaces. A clean contract between execution and control layers, plus adapters to each OEM, is critical to preserve optionality without fragility.

Strategic hooks in practice

- **APIs & events:** Versioned APIs and event topics for order lifecycle, task state and equipment telemetry enable extensions and analytics without core changes.
- **Plug-ins:** Site-specific rules (e.g., release logic, batching, exception handling) load as plug-ins constrained by contracts.
- **Protocol adapters:** OPC UA/MQTT adapters abstract vendor diversity and secure OT/IT data exchange, simplifying third-party equipment onboarding.

Optionality and Innovation in Warehouse Software

In warehousing, optionality takes the form of configurable workflows, digital-twin-driven scenario testing and integration frameworks for rapidly onboarding new automation.

Digital twins mirror operations and test "what-if" designs before deployment, reducing risk and accelerating change. Digital twins increasingly pair with AI to evaluate throughput, travel paths and labor/robot mixes. AI can also analyze order profiles and staffing conditions to trigger shifts between operating scenarios.

Clinging to rigid architectures is risky in a world of perpetual disruption. Market overviews of WMS/WES vendors note ongoing differentiation in usability, adaptability, decision support and cloud services – evidence that configurability and cloud delivery are now table stakes.

Broader market studies also confirm that cloud-based execution platforms are gaining share thanks to scalability, easier integration and reduced infrastructure burden. This is key to faster time-to-value and continuous innovation.

Engineer's Lens: Why Material Handling Engineers Select Tompkins Solutions

As an engineer designing a material handling solution, the selection criteria map directly to Gartner-observed buyer priorities (top purchase criteria for targeted customer group in NA, EU, retail, groceries and fashion from 2023 survey): time to market, cost, performance, maintenance (extendibility/sustainability), trust, usability, cloud, alliances and security. Below is how Tompkins Solutions addresses each – while preserving optionality through our WMS/WES/WCS stack.

1) Time to Market

- **Feature reuse by design:** A modular product core with strict separation of concerns lets us compose site solutions from proven modules (waves/release, batching, routing, exceptions) and add only the deltas via plug-ins.
- **Configuration-first:** Policy libraries (e.g., slotting, cartonization, release throttles) are data-driven; most behaviors are dialed in – not coded – accelerating commissioning.
- **Template integrations:** Prebuilt adapters for common OEMs and protocols (OPC UA/MQTT) compress equipment onboarding timelines.

2) Cost

- **Reuse > re-build:** Maximized reuse of core capabilities and test assets cuts engineering effort and lifecycle costs.
- **Right-sized distribution:** We keep local control loops near equipment (edge) and move planning/orchestration to the cloud where it makes sense – avoiding unnecessary microservice sprawl.

3) Performance

- **Edge orchestration + real-time control:** Deterministic control in WCS; near-real-time orchestration in WES with back-pressure and SLA-aware release.
- **Digital-twin-aided tuning:** Simulate candidate flows before go-live; validate cycle-time and throughput headroom under peaks.

4) Maintenance (Extendibility/Sustainability)

- **Stable contracts:** Versioned APIs/events and schema-managed configurations permit safe evolution.
- **Green upgrades:** Plug-in isolation and roll-forward/roll-back pipelines reduce downtime and waste during updates.

5) Trust

- **Evidence-based decisions:** Telemetry, audit trails and explainable AI policies (e.g., why tasks were interleaved or rerouted) build operator confidence.
- **Standards alignment:** OT security posture aligned to NIST SP 800-82 and IEC 62443 guidance.

6) Usability

- **Operator-first UX:** Role-based interfaces for supervisors, technicians and engineers with context-aware recommendations.

- **Assistive automation:** Copilot features for exception handling and fault recovery reduce cognitive load.

7) Cloud is a Must

- **Cloud-native where it counts:** Elastic planning, analytics and multi-site benchmarking in the cloud; low-latency control at the edge – an architecture pattern increasingly favored by modern WMS/WES leaders.

8) Alliances

- **Ecosystem-friendly:** Protocol adapters and API kits simplify interoperability with WMS, ERP and robotics vendors; our approach mirrors market demand for suites and convergence across execution domains.

9) Security

- **Defense-in-depth for OT/IT:** Zero-trust access, segmented zones, encrypted channels, least-privilege service accounts and vendor onboarding aligned with OT security standards.

Architecture and Integration

Shrinking time-to-market with reuse, configuration and modularity.

Design tenets

1. **Clear module boundaries:** Release planning, batching, routing, labor orchestration, exception handling and equipment abstraction live in distinct modules with narrow, stable APIs.
2. **Configuration over code:** YAML/JSON policy layers (e.g., pick path preferences, pack-out rules, wave throttle curves) drive behavior; “code last” only for non-standard logic.
3. **Evented spine:** Business events (examples include – OrderReleased, TaskAssigned, ToteDiverted, PLCFaultRaised) enable loose coupling and advanced analytics while preserving traceability.
4. **Sliceable evolution path:** Start modular; distribute selectively when clear scale/availability benefits outweigh complexity.

Testing strategy for optionality

- Contract tests for adapters, scenario tests for flows (baseline, peak, failure) and chaos drills for degraded modes (device offline, lane blocked).
- Twin-in-the-loop verification for high-impact changes before production.

Execution platforms touch both IT and OT. That means security patterns must respect real-time constraints and safety – while meeting enterprise governance.

OT security anchors

- **NIST SP 800-82 Rev.3 (OT Security):** guidance for segmenting ICS/OT networks, securing PLCs/SCADA and managing risk with safety and availability in mind.
- **IEC 62443:** defense-in-depth, zone/conduit segmentation and secure development lifecycle for industrial control components and systems.

Integration hardening

- OPC UA's security model (certificates, signing, encryption) and information modeling reduce adapter sprawl and help standardize semantics across OEMs.

The Strategic Value of Optionality

Software –not hardware – is now the primary source of advantage.

Resilience: Flexible systems bend under stress instead of breaking.

Profitability: Faster promo response, better labor utilization, higher peak throughput and lower cost per order.

Customer experience: Faster cycle times, more consistent SLAs – even in volatile conditions.

Competitive advantage: In a world where everyone has automation, optionality is the differentiator.

The Future: Software as the Brain of the Warehouse

As material handling hardware becomes increasingly standardized, competitive advantage is shifting to software. Fixed automation flows no longer define the warehouse. Instead, an intelligent software layer continuously senses conditions, evaluates alternatives and dynamically selects the best operating mode for the moment.

This software "brain" orchestrates labor, automation and inventory by enabling optionality – the ability to switch among multiple proven execution scenarios without re-engineering systems or disrupting operations. Rather than locking the business into a single optimized flow, it gives management control to adapt to demand variability, labor constraints, carrier cutoffs and operational disruptions.

This orchestration layer integrates external AI capabilities – from forecasting, robotics, vision systems and workforce analytics – without being tightly coupled to any single

vendor. The platform uses insights from these systems to drive real-time operational decisions while maintaining flexibility.

Optionality lowers the cost of change to near zero, making continuous optimization financially viable. Small daily improvements compound over time, driving 20-25% productivity gains through better labor utilization, higher peak throughput and faster response to variability – without major capital investments.

Organizations that invest in software-driven optionality will gain resilience, sustained productivity improvement and a clear competitive advantage.

Implementation Playbook: From Design to Value in Weeks – not Months

Optionality delivers value only when it is implemented quickly and safely. This playbook focuses on realizing rapid value through targeted changes, rather than large-scale, multi-year transformation programs.

Start with targeted optionality – not full transformation.

1. Discovery & Objectives: Focus on High-Leverage Optionality

Discovery should identify where adaptability creates immediate economic value, not try to define every requirement up front.

Key focus areas:

- Identify recurring operational stress points such as peak demand, promotions, labor shortages or automation congestion.
- Pinpoint decision moments that materially impact cost or service order release timing, picking strategy, batching logic, routing and labor allocation.
- Translate goals into business outcomes, such as reduced labor per order, improved peak throughput or lower overtime.

The output is a short, prioritized list of optionality points where multiple execution paths already exist – but are difficult to activate today.

2. Integration Plan: Enable Flexibility Without Risk

The integration strategy introduces optionality without destabilizing operations.

Core principles:

- **Maintain clear boundaries:** WMS as system of record, WES as orchestration layer, WCS as deterministic control.
- Integrate by capability, not by vendor, using APIs and events to connect automation, labor systems and external AI tools.

- Choose integration depth deliberately – control-level where appropriate, execution-level where value justifies it.
- Ensure all integrations are versioned, isolated and reversible.

This approach allows new capabilities to be added incrementally, keeping the core system stable.

3. AI-Assisted Tuning: Practical Intelligence, Not Theoretical Optimization

Apply AI where it improves day-to-day decisions, not as an abstract optimization layer.

Primary uses include:

- Short-horizon adjustments to batching, release timing and balancing workload.
- Early detection of emerging congestion or inefficiencies.
- Learning which operational scenarios perform best under specific conditions.

Humans should govern AI recommendations, enabling operators and engineers to refine policies rather than rewrite logic. The result is steady, compounding improvement.

4. Reliability Uplift: Flexibility With Operational Confidence

Optionality must increase resilience, not risk.

Key practices:

- Predictive monitoring of throughput, queues and equipment health.
- Graceful degradation – when a constraint appears, the system shifts to an alternate execution path.
- Decisions should be transparent so supervisors understand why the system changed modes.

Reliability goes beyond uptime – it is the ability to adapt without disruption.

5. Core KPIs: Measuring the Value of Optionality

Optionality requires KPIs that capture flexibility and flow, not just static efficiency.

Recommended metrics:

- **Productivity:** labor hours per order, picks per hour, hit-per-pick
- **Flow efficiency:** cycle time, queue depth, release latency
- **Reliability:** availability, recovery time, frequency of disruption
- **Service:** SLA adherence during peak, on-time shipment
- **Economics:** cost per order, overtime reduction, throughput per labor dollar

Tracked over time, these metrics reveal the compounding effect of optionality – supporting 20-25% productivity improvement through continuous, low-cost optimization.

6. Governance & Upgrades: Optionality at Enterprise Scale

Optionality requires disciplined governance to add value rather than unnecessary complexity.

Effective approaches include:

- Clear ownership of configurations, policies and AI decision boundaries.
- Controlled rollout of changes using feature flags, phased deployment and rollback mechanisms.
- Upgrade paths that preserve customer configurations and custom extensions without rework.

Strong governance ensures organizations can evolve continuously, adopt new AI capabilities and upgrade platforms without operational disruption.

Summary

This playbook turns optionality into a repeatable operating model. By focusing on targeted discovery, disciplined integration, AI-assisted tuning, built-in reliability and the right KPIs, organizations can move from design to measurable value in weeks – while building a foundation for sustained performance improvement.

Conclusion

The supply chain industry is shifting beyond a focus on efficiency alone. Organizations that build systems with resilience and optionality at their core will be better positioned to adapt. Software is the key – not because it optimizes a single flow, but because it enables many. Optionality transforms warehouses from rigid machines into adaptive ecosystems – providing a clear and lasting competitive advantage.