Microsoft

mavim
change it

From Process to Delivery: Why You Start in Mavim Before You Scale in Azure DevOps

Bridging Business Alignment and Technical Delivery with Mavim and Azure DevOps in Dynamics 365 Implementations

Contents

mavim

Executive Summary

Why Mavim Comes Before Azure DevOps—Especially in an AI World

Executive takeaway: If you can't govern and agree on the process, you can't reliably scale Dynamics 365 delivery, or agentic AI, no matter how strong your DevOps engine is.

Most organizations fail in Dynamics 365 delivery for one reason: they start building before they agree on how the business should run. Azure DevOps is excellent at execution, but it assumes the process is already clear.

Mavim provides that clarity.

Mavim establishes the governed, business-owned **process foundation**, aligned to Microsoft's Business Process Catalog—before a single user story is created. As the **control plane above execution**, those approved processes then flow directly into Azure DevOps as epics, features, and stories.

The result: less rework, faster delivery, and full traceability from business intent to technical execution.

This order matters even more with agentic AI. AI agents can only operate safely and at scale when processes, ownership, and controls are explicit… **this is not optional once you're operating at enterprise scale**. Mavim defines the rules. Azure DevOps executes them. AI works inside that structure—not around it.

Start with Mavim. Execute in Azure DevOps. Enable D365 and AI with confidence.

mavim

Two Different Strengths: Process Management vs. Project Execution

Although Mavim and Azure DevOps are both built on the Microsoft technology stack, they serve fundamentally different purposes.

Function	Mavim	Azure Devops
Primary Lens	Business process and governance	Application lifecycle management
Audience	Business stakeholders, process owners, governance teams	Developers, engineers, project managers
Approach	Process discovery → modeling → visualization	Backlog → sprint → test → deploy
Core Output	Standardized, governed processes and roles	Executable work items and releases
Goal	Align the organization on how business should operate	Deliver and track technical changes efficiently

Azure DevOps answers "how fast can we build?"
Mavim answers "what should we build, and who is accountable for it?"

Successful D365 transformations require both questions to be answered—continuously, and in sync.

A complete D365 or enterprise transformation needs both, but sequence is everything.

If you try to execute before aligning on process, teams are forced to make assumptions, coding what they believe the business wants instead of what the business actually designed.

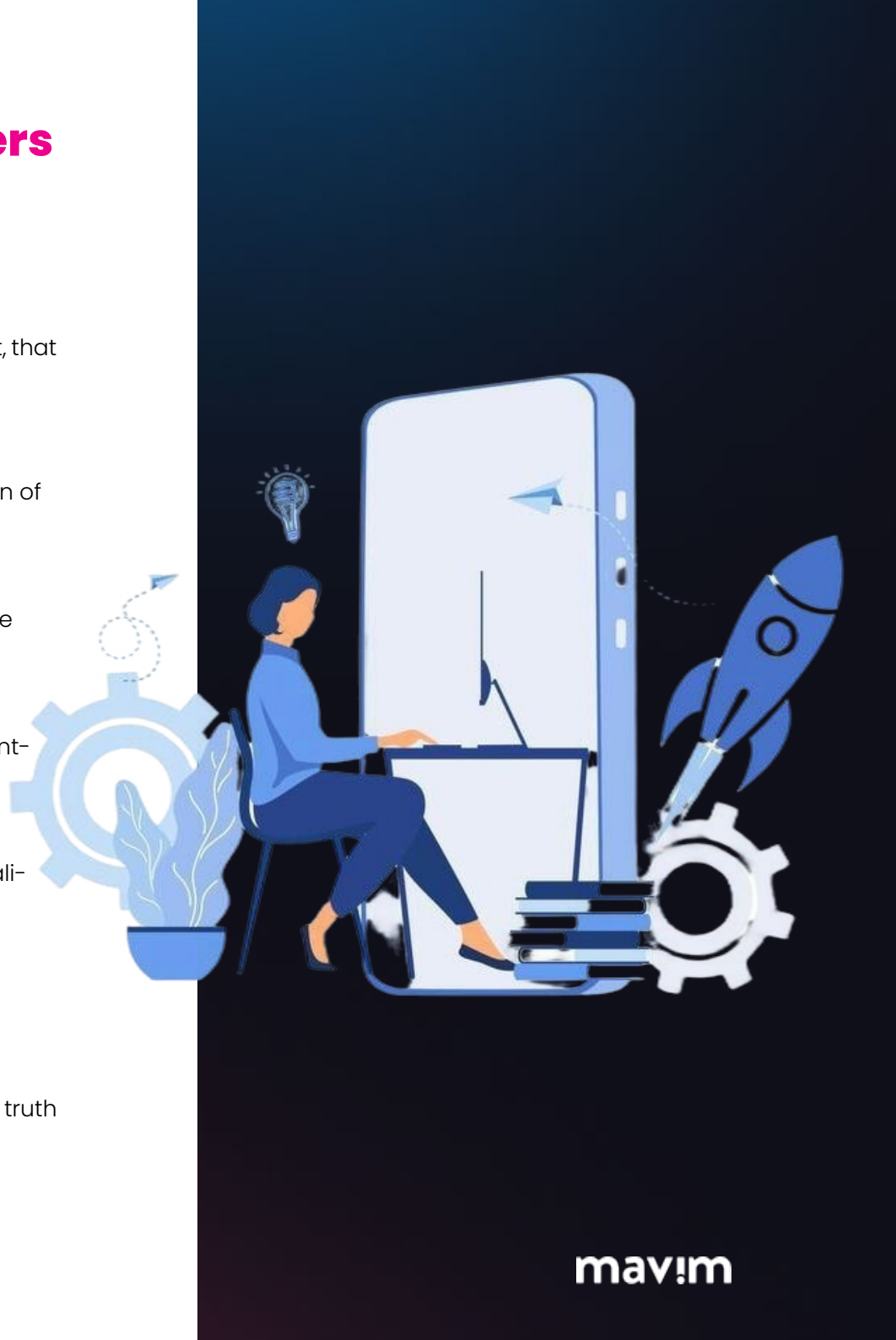
Why a Process-First Mindset Matters

The Problem with Starting Inside ADO

Teams that jump straight into Azure DevOps often cite speed or agility. Yet, that speed hides serious long-term risk:

- **Inconsistent story structures**
  Without an agreed process model, every team defines its own version of "order to cash" or "project to profit."
- **Lost process context**
  User stories lack mapping to actual business functions or compliance obligations.
- **Unclear ownership**
  Multiple teams may change overlapping areas with no single accountable owner.
- **Rework and technical debt**
  Fixing misaligned functionality later costs exponentially more than validating it early.
- **Compliance challenges**
  Audit evidence and controls are scattered across boards and repos rather than tied to process-level governance.

All of these issues trace back to one root cause: a missing single source of truth that sits above execution.



Why a Process-First Mindset Matters

The Mavim Approach

Mavim creates that single source of truth.
It captures how your organization intends to operate: visually, structurally, and governed. Mavim pulls directly from the Microsoft Business Process Catalog (BPC), giving your enterprise an intelligent blueprint of standardized D365 processes aligned with global APQC benchmarks.

Within Mavim, these best-practice models become living process maps enriched with your specific metadata:

- Organizational ownership and RACI roles
- Links to related policies, controls, and documentation in Microsoft 365
- Performance indicators and KPIs
- Dependencies with applications, data, and DevOps work items

Once the organization approves those models, they become the authoritative design layer for all future execution. When teams later build in ADO, they're building *against validated processes*, not interpretations of them.

Build in Mavim First: Designing the Foundation

Establish the Single Source of Truth

Every transformation begins with understanding the current state ("as-is") and defining the desired state ("to-be"). Mavim gives teams:

- End-to-end process maps that visualize every handoff and dependency.
- Clear ownership models to assign who is accountable for every process step.
- Managed metadata, ensuring documentation, risks, and metrics remain connected.

Instead of abstract requirements documents, Mavim provides *living governance*.

Executives see what changes are coming. Business users see how their work will shift. Developers see precisely where and why functionality fits.

Align People, Process, and Policy

While ADO organizes technical work, Mavim organizes operational accountability. It defines:

- Business processes and sub-processes
- Owners and contributors across departments
- Internal controls and compliance checkpoints
- Artifacts and documentation linked through Microsoft 365

By mapping this landscape first, organizations reduce translation errors between departments. Legal, finance, HR, operations, and IT can all view the same model through different lenses while maintaining one structured repository.

Govern Change Before It Proliferates

Every process model in Mavim becomes the reference point for change management. When a modification request arises, the team asks three questions inside Mavim:

1. Which processes and roles are impacted?
2. What upstream or downstream dependencies exist?
3. Does this alter compliance, policy, or reporting requirements?

Only once those answers are approved does the request flow into ADO as a technical work item.

That's "first-time-right" development in action.

Then Move to Azure DevOps: Operationalize and Deliver

Translating Process into Execution

Once validated, the process structure in Mavim feeds directly into Azure DevOps.
Each process or subprocess maps to specific hierarchy levels:

Mavim Artifact	Azure DevOps Equivalent
End-to-end process area	**Epic**
Sub-process or capability	**Feature**
Step or configuration	**User Story** or **Task**
Owner and governance rules	**Work item links and tags**

This mapping ensures that every sprint in ADO directly traces back to a business-approved process. When requirements change, the relationship remains bidirectional, updates in ADO automatically reference the originating process model in Mavim.

Deliver with Clarity and Efficiency

Once inside ADO, technical teams can focus on:

- **Sprint planning and work tracking** using the Agile hierarchy
- **Pipeline automation** for builds, tests, and deployments
- **Integration with Teams and Power BI** for transparency and reporting
- **Governance dashboards** showing status per business process area

Because Mavim supplied the structure, ADO becomes cleaner. Every feature and story is traceable, measurable, and aligned to objectives. Team throughput increases not through more speed, but through less confusion.

Continuous Synchronization

The integration between Mavim and ADO is **bidirectional**:

- When a process step updates in Mavim, the linked ADO user stories are refreshed automatically.
- When a developer updates a work item status in ADO, the connected process in Mavim reflects progress.

Business and IT remain synchronized in real time. This keeps steering committees informed without manual reporting and gives engineers direct visibility into business context.

What Happens If You Go Azure DevOps First

Many organizations attempt ADO-first implementations, assuming they can "connect the dots" later with process documentation. The consequences usually appear mid-project:

ADO-only: A team starts with "Order to Cash" user stories, but each workstream interprets steps and ownership differently. Mid-build, leaders realize approvals, exceptions, and controls weren't aligned—so stories are rewritten, integrations reworked, and UAT becomes the discovery phase.

Mavim + ADO: The business aligns and signs off on the "Order to Cash" process (steps, owners, controls) in Mavim first. Those approved processes generate the backlog in ADO, so delivery focuses on execution—changes stay traceable to process decisions, and AI agents can operate within explicit rules.

Fragmentation and Rework

Different teams define similar processes differently. When harmonization finally starts, it requires extensive refactoring.

Lack of Business Traceability

User stories accumulate without being traceable to validated processes or objectives. During audits or upgrades, there's no clarity on why a change was made or who approved it.

Governance Gaps

Without a process framework, it's impossible to prove policy adherence or segregation of duties. Remediation efforts slow delivery and inflate overhead.

Cultural Misalignment

Developers speak in "features and sprints," while business leaders speak in "processes and outcomes." Without an intermediary like Mavim, those conversations rarely align.

Update Paralysis

Every major D365 update or migration becomes a discovery project. With no end-to-end process view, understanding impacts across thousands of work items is nearly impossible.

The verdict: **you can recover from going ADO-first—but you'll pay for it in delays, rework, and governance debt.**

Mavim and Azure DevOps Together: The Complete Lifecycle

When orchestrated correctly, Mavim and Azure DevOps operate as a single, circular system:

1. **Define and Design in Mavim**
   Capture processes, roles, and governance structures.

2. **Validate and Approve**
   Gain cross-functional alignment and official sign-off.

3. **Push to Azure DevOps**
   Translate processes into epics, features, and user stories.

4. **Execute**
   Development teams run sprints, build functionality, and manage releases.

5. **Sync Back to Mavim**
   Completed items and change logs update process models and KPIs.

6. **Measure, Improve, and Govern**
   Operational performance feeds process mining dashboards to identify next opportunities.

This feedback loop transforms project management into a continuous transformation cycle. Mavim illuminates why and what; ADO delivers how and when.

Practical Benefits of "Mavim-First, ADO-Second"

Improved Communication and Transparency

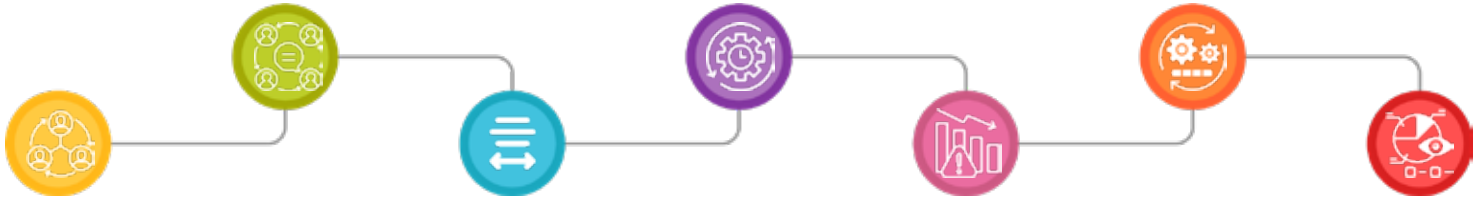
Every requirement, dependency, and deliverable ties directly to a visual process model. Stakeholders can see progress toward specific operational goals rather than isolated task completion.

Increased Efficiency and Fewer Hand-Offs

Automatic synchronization eliminates manual data entry between process teams and developers. Updates propagate instantly, accelerating cycle times.

Streamlined Upgrades

When new D365 versions arrive, Mavim's models highlight impact areas automatically, guiding ADO teams exactly where changes must occur.



Enhanced Collaboration

Business and IT finally share one operating language. Each team views the same source information in the tool best suited to their role—Mavim for context, ADO for execution.

Goal Alignment and Measurable Impact

KPIs defined in Mavim connect to status data in ADO, turning progress tracking into business value tracking.

Reduced Risk and Rework

KPIs defined in Mavim connect to status data in ADO, turning progress tracking into business value tracking.

Business-Readable Visualization

Unlike technical DevOps boards that speak only to engineers, Mavim visualizations are designed for executives and process owners—supporting communication, training, and audit readiness.

A Realistic Implementation Path

A best-practice roll-out typically follows four stages:

1. **Discovery and Benchmarking (Weeks 1–4)**
   Use Mavim to import baseline processes from the Microsoft Business Process Catalog and identify variances from current operations.

2. **Design and Governance Setup (Weeks 5–8)**
   Define ownership, link documentation, and establish governance workflows directly in Mavim.

3. **Integration and Handoff to ADO (Weeks 9–12)**
   Generate epics/features/user stories from validated processes and push to Azure DevOps for sprint planning.

4. **Continuous Improvement (Ongoing)**
   Operate in a closed feedback loop—business feedback and KPI results update process models in Mavim, which inform future backlogs in ADO.

This progression ensures the organization is not just executing a transformation—but governing it effectively.

Future-Proofing: Designing for Continuous Change

Agility doesn't end at go-live. D365 environments evolve constantly with updates, regulatory shifts, and new business priorities.

With Mavim's dynamic modeling and ADO's continuous delivery, organizations gain sustainable adaptability:

- Mavim models evolve as business conditions change.
- Those updates flow to ADO backlogs automatically for technical realization.
- Post-implementation monitoring feeds back into Mavim's process mining layer for optimization.

This creates a continuous digital twin of the organization—a live connection between how the business runs and how the systems deliver.

Conclusion: Start with Process, End with Confidence

Azure DevOps is a world-class engine for execution.
Mavim is the GPS that tells it where to go.

In an agentic AI world, that sequencing becomes a control question—not a preference. Mavim acts as the **control plane above execution**, making processes explicit, governed, and safe for AI agents to operate at scale—while serving as the **bridge between business intent, Dynamics 365 delivery, and AI agents**. Azure DevOps operationalizes that execution through disciplined delivery.

Without the GPS, even the most powerful engine drives fast in the wrong direction.

By starting in Mavim, your teams define how the organization operates, gain consensus across departments, and ensure governance before code is written. Then by implementing in Azure DevOps, they manage precise delivery, maintain traceability, and drive continuous improvement at scale.

The winning formula is simple:

Process-first with Mavim. Execute next in Azure DevOps. Govern and evolve continuously.

Enterprises that embrace this order consistently report faster adoption, lower rework, stronger compliance, and higher user satisfaction—proving that modern digital transformation begins not with development, but with design.

GET & STAY CONNECTED

youtube.com/mavimtv
1.9 M

linkedin.com/company/mavim
3.1 K

x.com/mavimintl
2.6 K

mavim
change it

www.mavim.com