

Deep Dive: Using the Deliverables Hierarchy and Delivery Plan in the Microsoft Business Process Catalog

Presenter:

Rachel Profitt – Principal Program Manager

Mandi Toves – Senior Program Manager

Leo Salomons – Value Engineer

Moderator:

Mandi Toves – Senior Program Manager



Mandi Toves

Senior Program Manager

Microsoft

manditoves@microsoft.com

<https://www.linkedin.com/in/mandi-mollerstuen-toves-8488097/>



Rachel Profitt

Principal Program Manager
Microsoft

Rachel.Profitt@microsoft.com

<https://Dynamics365Lady.com>

<https://twitter.com/rachelprofitt>

<https://linkedin.com/in/rachelprofitt>

<https://youtube.com/c/Dynamics365Unboxed>



Leo Salomons

Value Engineer

Mavim

leonard.salomons@mavim.com

<https://www.linkedin.com/in/leo-salomons-8a026a2/>



Agenda

Two major topics covering the Deliverables Tree and the Delivery Plan pillars of the Business Process Catalog



Part 1: The Deliverables Tree

- Overview
- Work item types and categorization
- Best practices and common pitfalls
- Real-world application and examples



Part 2: The Delivery Plan

- Framework overview and Success by Design alignment
- How workshops drive execution
- Guidance for partners (do's and don'ts)



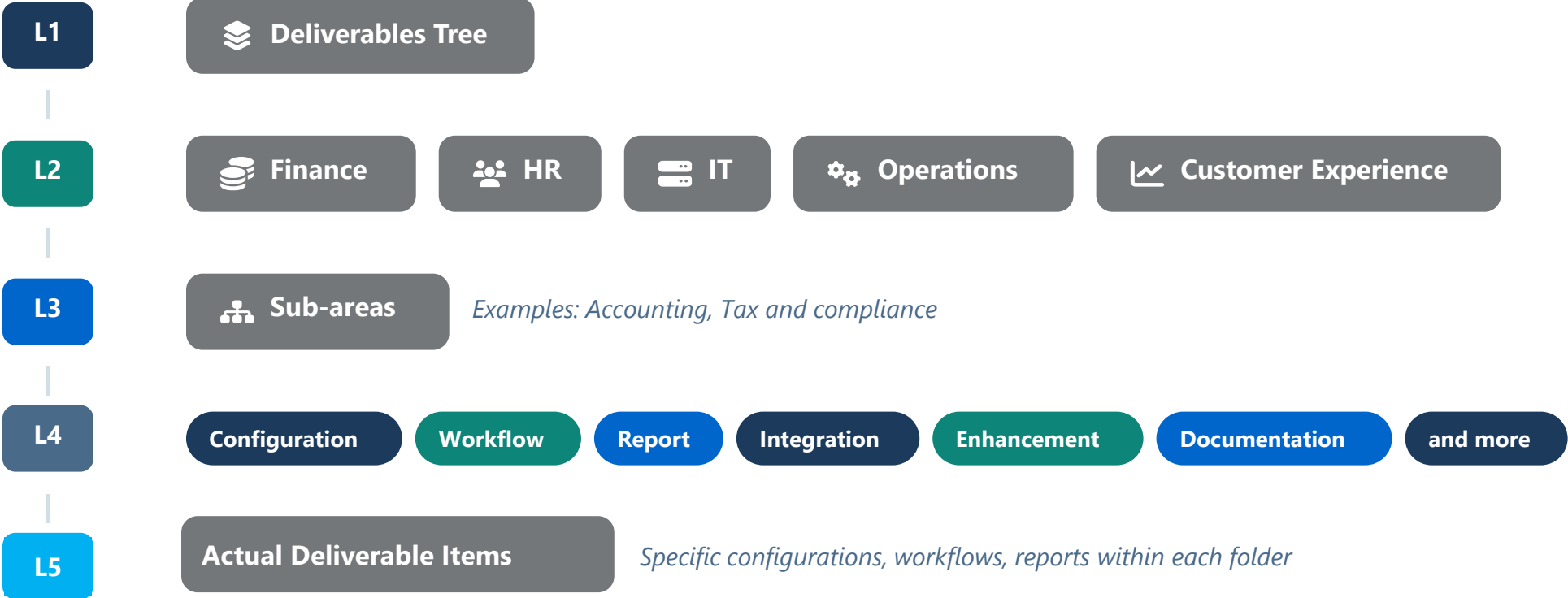
Previous TechTalks covered the BPC Hierarchy (6 levels) and the Objectives pillar

Deliverables Tree



Deliverables Tree Overview

The Deliverables Tree transforms flat task lists into a structured hierarchy with 7,000+ predefined deliverables



7,000+ Deliverables

▼ Filter by business process scope

+ Custom work items

🔗 Area paths in Azure DevOps

🔗 Scaled agile planning

L4 Work Item Types

Fourteen distinct work item types ensure every deliverable is properly categorized



Configuration

Standard system setup and parameters



Workflow

Approval chains and automated routing



Report

Reporting and analytics outputs



Integration

Data flows between systems



Enhancement

Custom code and extensions



Documentation

Written artifacts and guides



Job

Batch jobs and scheduled processes



Migration

Data migration tasks



Personalization

UI and user experience customizations



Security

Security roles and permissions



Workshop

Collaborative working sessions



Testing

Test strategies and execution



Requirement

Business requirements being defined



Deliverable

General deliverable items

Area Path Relationships

Business Processes L1-L3 map to end-to-end processes; L4-L6 map to functional areas — this is how processes connect to deliverables through Azure DevOps Area Paths

Business Process Levels 1-2-3

L1 → L2 → L3

These are cross-functional processes

Area Path:

End-to-End Process (e.g., "Source to Pay")



Business Process Levels 4-5-6

L4 → L5 → L6

These map to deliverables

Area Path:

Functional Area (e.g., "Finance / Accounting / Accounts Payable")

Deliverables Tree Structure

- L1 = Functional Area (Finance, Customer Experience, HR, IT, Operations)
- L2-L3 = Sub-areas (Accounting, Accounts Payable)
- L4 = Work item type folders
- L5 = Actual deliverable items

L4-L5-L6 business processes use same Area Path as deliverables functional areas — this enables cross-team visibility in Azure DevOps Boards

Do and Don't -- Work Item Types

Each work item should represent ONE distinct deliverable -- be specific, not vague or overly broad

✓ Do

- ✓ **"Invoice approval workflow"**
Workflow
- ✓ **"AP aging summary"**
Report
- ✓ **"Migrate open vendor balances"**
Migration
- ✓ **"Configure PO tolerances"**
Configuration

✗ Don't

- ✗ **"Set up everything for AP"**
Too broad
- ✗ **"Build vendor stuff"**
Vague
- ✗ **"Test the system"**
Too broad
- ✗ **"Write all the docs"**
Combines multiple artifacts

💡 Each work item should represent ONE distinct deliverable

Do and Don't — Choosing the Right Type

The right type ensures the right fields, governance, and reporting

✓ Do

- ✓ **Configuration** for standard setup and any screen out of the box
- ✓ **Enhancement** for code/extensions, also includes Power Platform
- ✓ **Integration** for data flows between systems; each inbound and outbound integration has its own
- ✓ **Workflow** for approval chains
- ✓ **Requirement** for items still being defined
- ✓ **Enhancement + Configuration** when custom code includes new setup screens

✗ Don't

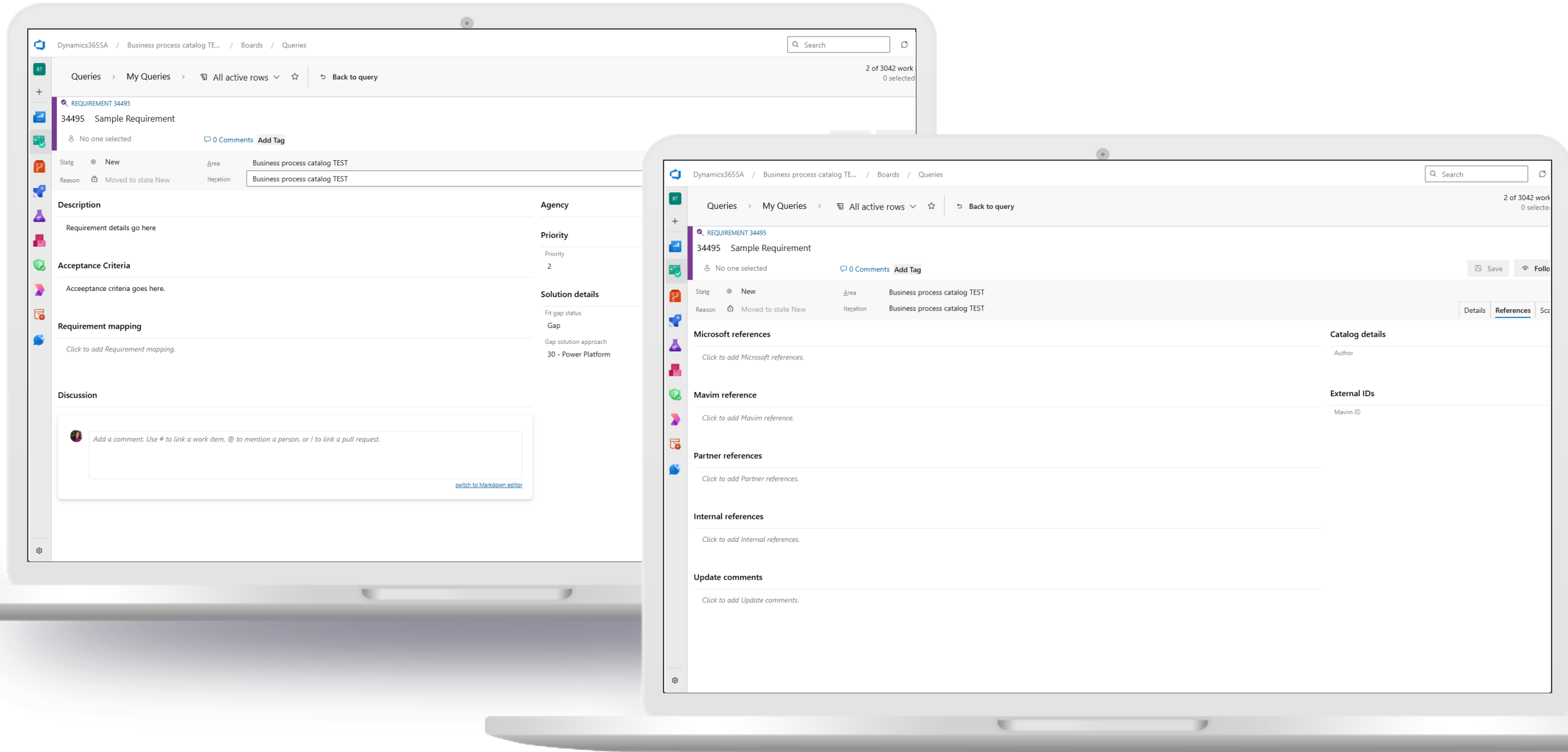
- ✗ **Configuration Misuse** Custom code belongs in Enhancement
- ✗ **Documentation Misuse** putting the actual documentation directly in the Documentation work item.
- ✗ **Combining Types** inappropriately — migration and integration have different lifecycles
- ✗ **Catch-all Misuse** Deliverable should not be used as a catch-all
- ✗ **Testing vs Test Case** confusion — Testing is strategy/planning, Test Cases are actual tests

Right type ensures right fields, governance, and reporting

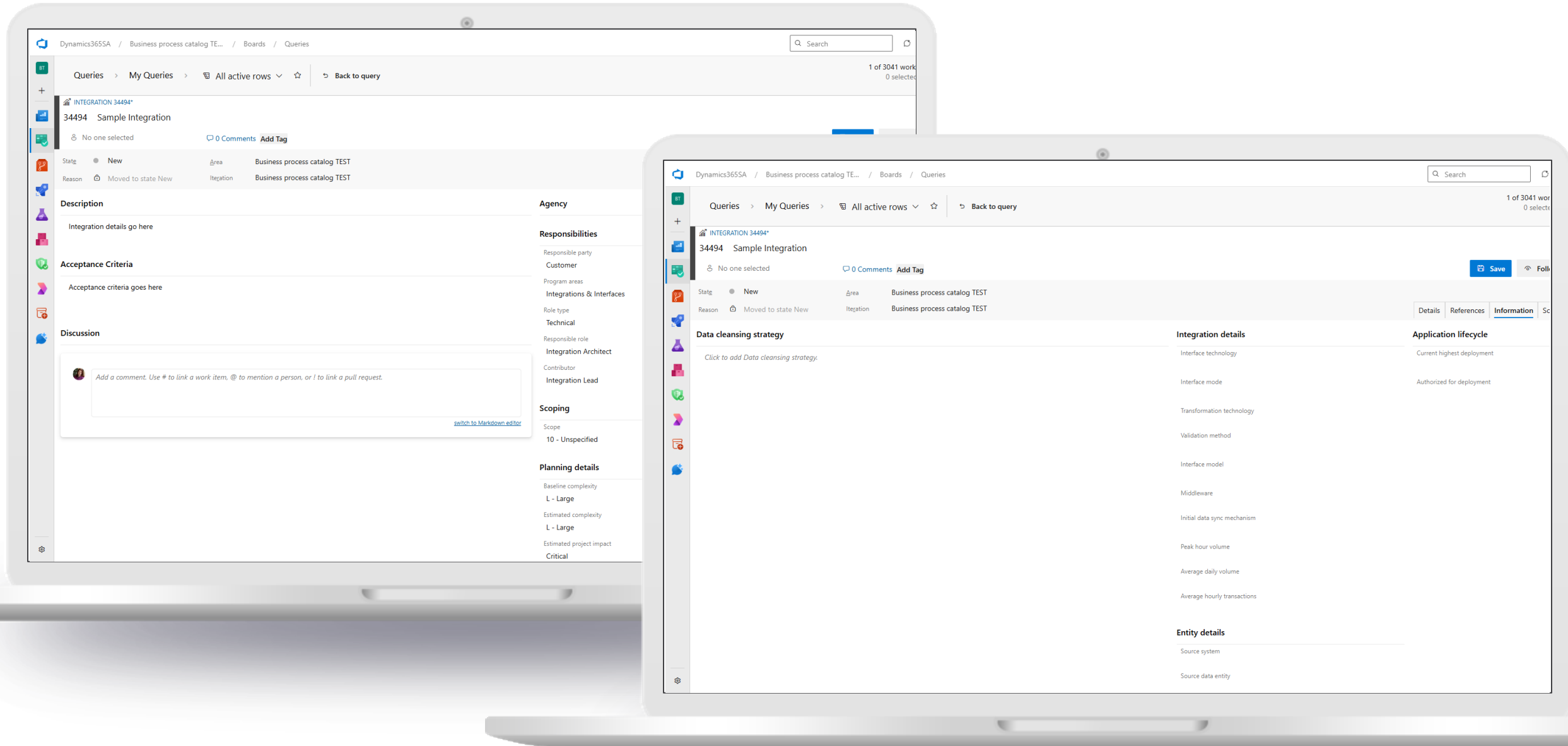
Deliverables in action



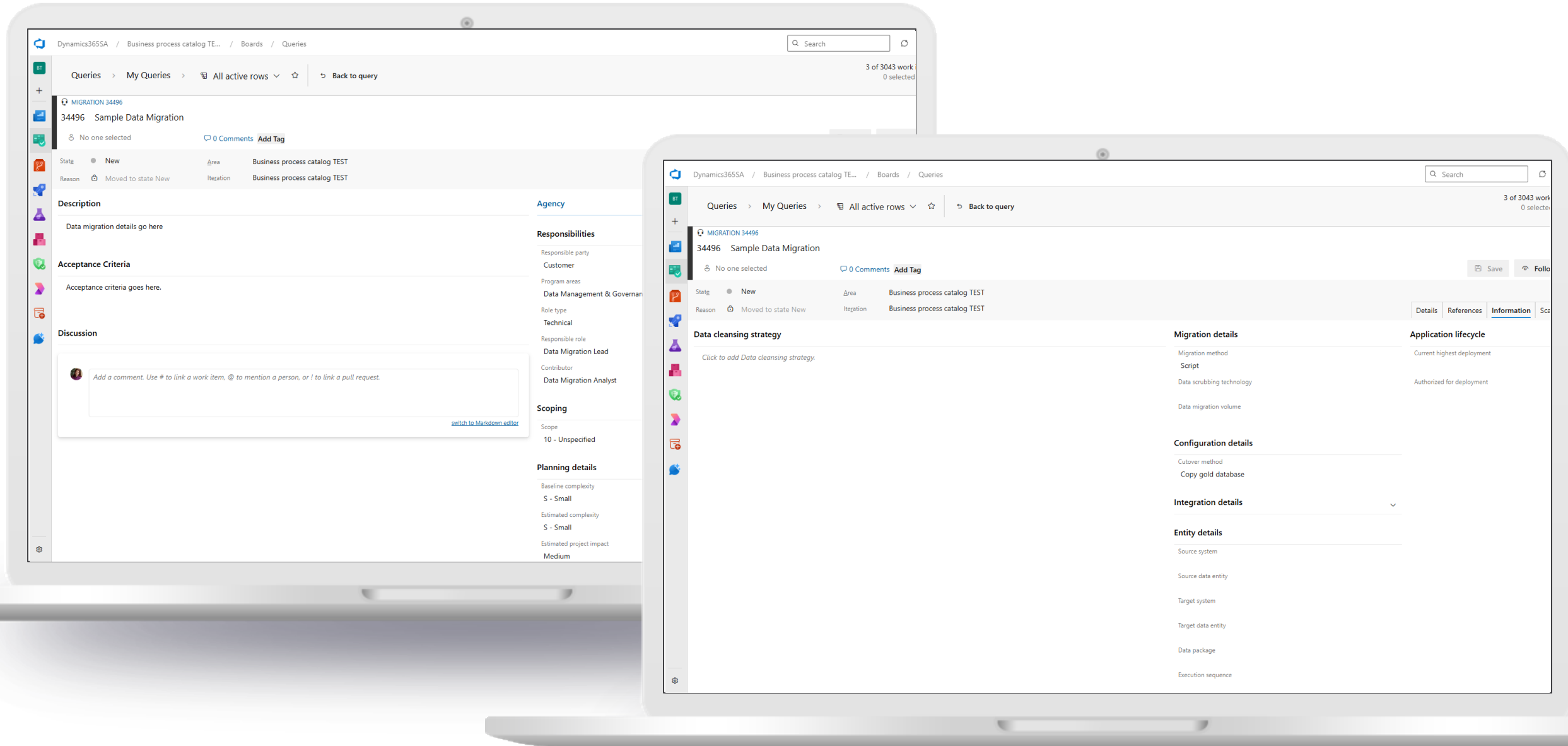
Managing Requirements in Azure DevOps



Managing Integrations in Azure DevOps



Managing Data Migrations in Azure DevOps



Demo: Mavim

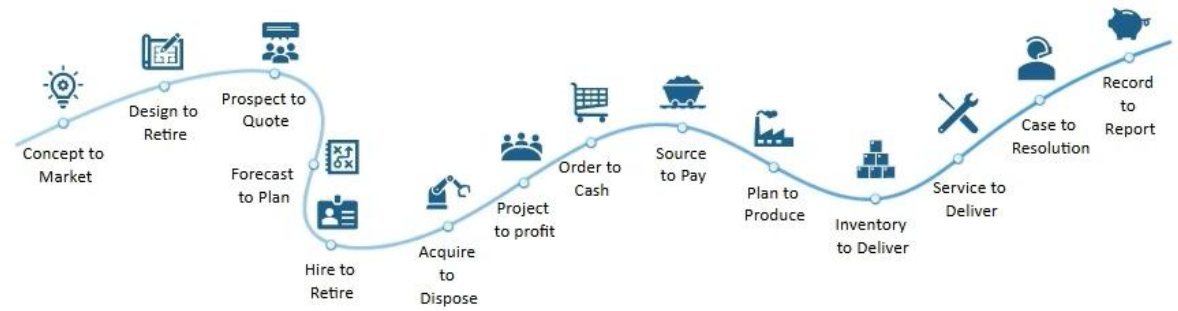
Business Process Catalog - End to end journeys

Charts | Description | Fields | Relationships | Subtopics | Discussion

Charts

Classic Modern

Generic Chart



Application family	Security roles	Deliverables
Products	Modules	Workshops
Success by Design Delivery plan	Objectives	

Delivery Plan



From Deliverables to Delivery Plan

Deliverables define what needs to be built; the Delivery Plan defines when and in what sequence the work happens — together they connect business process definition to actual project execution



Deliverables

What needs to be configured and built



Connects to project execution



Delivery Plan

When and in what sequence does the work happen



Key Message for Partners

The Delivery Plan is a framework, not a methodology — it provides direction, sequencing, and dependencies for you to map your own methodology onto



A Framework, Not a Methodology

Map Your Methodology to the Baseline — provides direction, sequencing, and dependencies



What the Delivery Plan Gives You

- Success by Design required activities
- Execution checklists
- Validation points



What You Add

- Delivery cadence
- Team organization
- Change management
- Industry-specific accelerators



Cross-team timeline view



Dependency visualization



Progress tracking



Dynamic sequencing

Delivery Plan Overview

The Delivery Plan defines how, when, and in what sequence deliverables get executed, aligned to Success by Design phases



How Workshops drive the Delivery Plan

The Delivery Plan is the connective tissue tying Objectives, Business Processes, and Deliverables into a coordinated, time-sequenced plan

End to End Processes

Process Areas

Scenario board Workshops
Case to Resolution Scenario Board



Storyline Design Review Workshops
Case to Resolution Storyline Design Review



Detailed Process Workshops
Intake Cases Deep Dive Discovery




Requirements




Scoping




Migration



Testing



Fit Gap



Enhancements



Security



Workshops



Integrations

Do and Don't — Delivery Plan

Treat the Delivery Plan as a living framework to align teams, not as a complete or static project plan

✓ Do

- ✓ **Align teams on sequencing across functional areas**
- ✓ **Map your methodology to Success by Design (SbD) phases**
- ✓ **Include all SbD required activities and milestones**
- ✓ **Use ADO timeline view for cross-team visibility**
- ✓ **Treat it as a living plan that evolves with the project**

✗ Don't

- ✗ **Treat it as your complete project plan — it's a baseline, not every task**
- ✗ **Build everything from scratch — ignoring the plan means missing critical SbD activities**
- ✗ **Create a disconnected plan — always tie back to business processes and deliverables**
- ✗ **Make it too granular — keep it outcome-oriented, not task-by-task**
- ✗ **Treat it as static — the plan should evolve continuously**

💡 **The Delivery Plan is a framework, not a methodology — adapt it to your approach**

Demo: Delivery Plan in Azure DevOps

Demo: Delivery Plan in Mavim

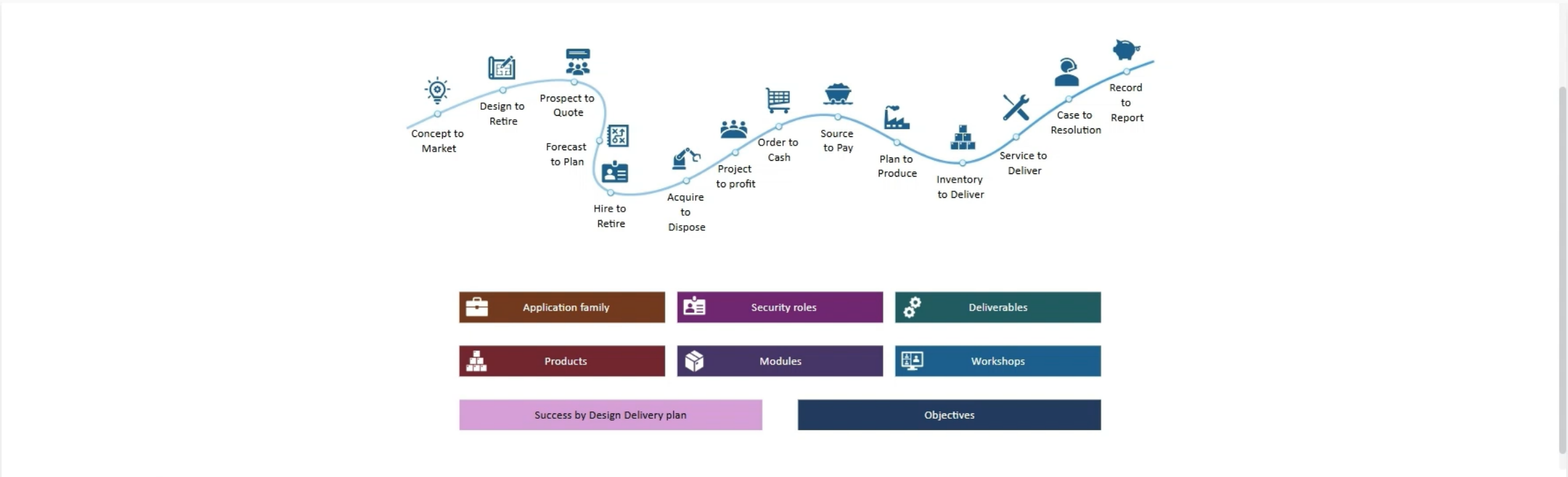
Business Process Catalog - End to end journeys

Charts | Description | Fields | Relationships | Subtopics | Discussion

Charts

Classic Modern

Generic Chart



Page Mavim

Resources and Recommendations



Resources and Next Steps

Explore these resources and share your feedback to help shape the Business Process Catalog



Business Process Catalog

aka.ms/businessprocesscatalog



ADO Template

aka.ms/BPEADOTemplate



Business Process TechTalks

aka.ms/businessprocesstechtalks



OneGuidance

aka.ms/oneguidance



Catalog Feedback

aka.ms/businessprocesscatalogfeedback



Contribute on Microsoft Learn

<https://learn.microsoft.com/en-us/dynamics365/get-started/contribute>



Mavim Free Preview


[Browse Apps](#) | [Microsoft Marketplace](#)



TechTalks Feedback

aka.ms/BPETechTalksFeedback

 **Contact:** bizprocessguides@microsoft.com

 **Feedback:** aka.ms/BPETechTalksFeedback

Microsoft and Mavim BPE Resources:

- [Microsoft Business Process Catalog in Mavim Free Preview](#)
- [Success by Design with the Dynamics 365 business process catalog in Mavim - Dynamics 365 | Microsoft Learn](#)
- [Navigate the business process catalog in Mavim - Dynamics 365 | Microsoft Learn](#)
- [Microsoft and Mavim Dynamics 365 Business Talks](#)
- [Unlocking Transformation Synergy: Mavim & Microsoft Process Catalog](#)
- [Mavim for Dynamics Finance and Operations](#)
- [Introduction to the business process catalog in Mavim - Dynamics 365 | Microsoft Learn](#)

Start your free trial today!



Next TechTalk...

Transforming Business Processes
with an Agentic Mindset for
Customer Engagement Apps

Thursday, May 14th

7:00-8:00am PST

<https://aka.ms/BPETechTalksOption1Join>

<https://aka.ms/BPETechTalksBlog>

Survey

Your feedback is important!

Please take a moment to complete our survey about today's event

<https://aka.ms/BPETechTalksFeedback>

Business Process - Dynamics 365 -
TechTalk Series





QUESTIONS

Dankie Faleminderit **Shukran** Chnorakaloutioun Hvala Blagodaria
Děkuji **Tak** Dank u **Tānan** Kiitos **Merci** Danke Ευχαριστώ A dank
Mahalo הודות. **Dhanyavād** Köszönöm Takk **Terima kasih** **Grazie** Grazzi

Thank you!

감사합니다 Paldies Choukrane Ačiū **Благодарам** ありがとうございます
谢谢 Баярлалаа **Dziękuję** Obrigado Mulțumesc **Спасибо** Ngiyabonga
Ďakujem **Tack** Nandri **Kop khun** Teşekkür ederim Дякую **Хвала** Diolch

