



2026
DYNAMICS
CON
Vegas



2026 DYNAMICS CON



Foundations of D365FO Technology for Non-Technical Users

Logan consulting

Ajay Dua





ABOUT US



Founded in 1992
33+ Years ERP Expertise



Chicago-Based, Global Reach
125+ Consultants Deployed Annually



Certified Microsoft Business
Solutions Partner



User Group Contributor since
2014



Microsoft is our largest
practice



Over a dozen Microsoft
Certifications

Core Service

- Assessments
- Implementations/Migrations
- Change Management
- Optimization
- Ongoing Support

Extended Service

- AI Readiness
- Copilot Deployment
- Data Engineering
- Technology Selection
- Project/Program Management
- CRM Implementations
- EDI Management
- Technology Strategy



Why this session?

- Business users don't understand technical terminology; developers don't always understand business intent.
- Lack of Understanding of D365FO Architecture
- Confusion Around Environments
- Poor Visibility into Development Effort
- Frustration with Build & Deployment Delays
- Lack of Awareness of Source Control & Versioning
- Version mismatch issues
- Misaligned expectations



Agenda

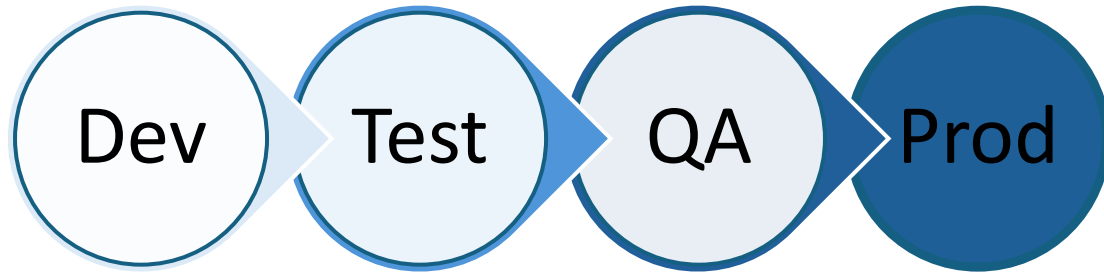
Essential understanding of Environments

Essential understanding of builds and deployment terminology

Essential understanding of data storage



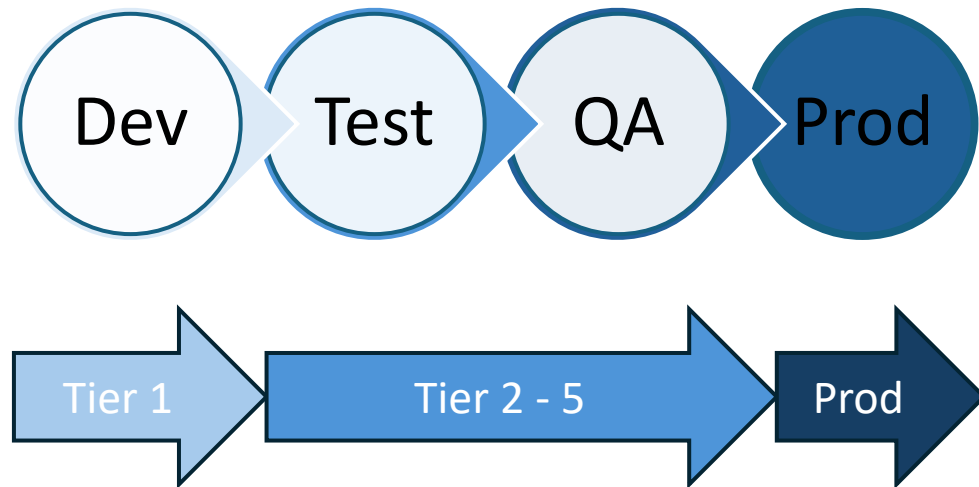
Environments



- Multiple development environments, one development environment per developer
- At least one test or QA environment, sometimes both (preferred if you have multiple development initiatives running at same time)
- Production environment



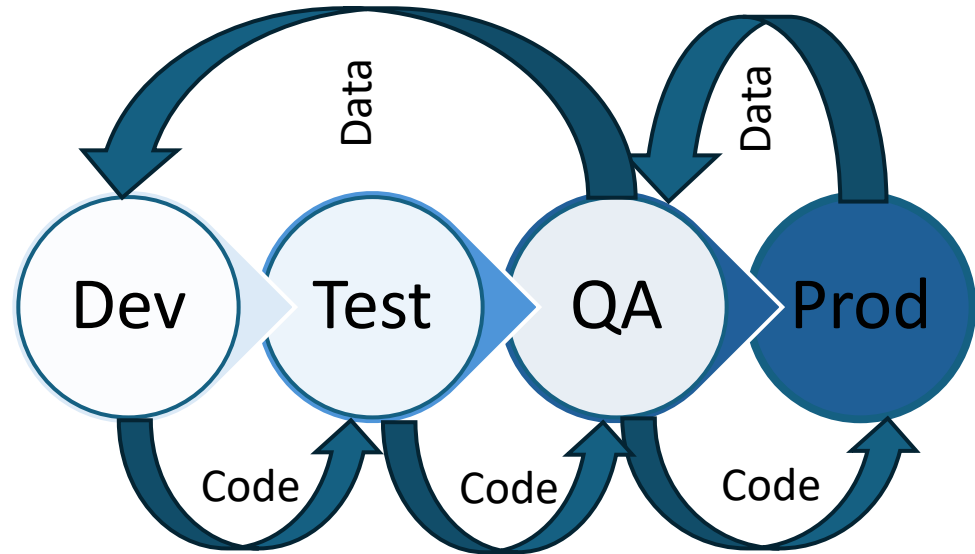
Environments



At times technical term used for these environments is referencing by tier. In simple terms tier is machine size.

- **Tier 1 environment** – Dev environments, self contained VM silos of business logic and database, not managed by Microsoft
- **Tier 2 environment** – Test/UAT/QA (a lot of names but serve same purpose)
- **Tier 3-5 environments** -Test/UAT/QA/Sandbox (better specs as you go higher, used for load/stress testing)
- **Production environment**

Environments

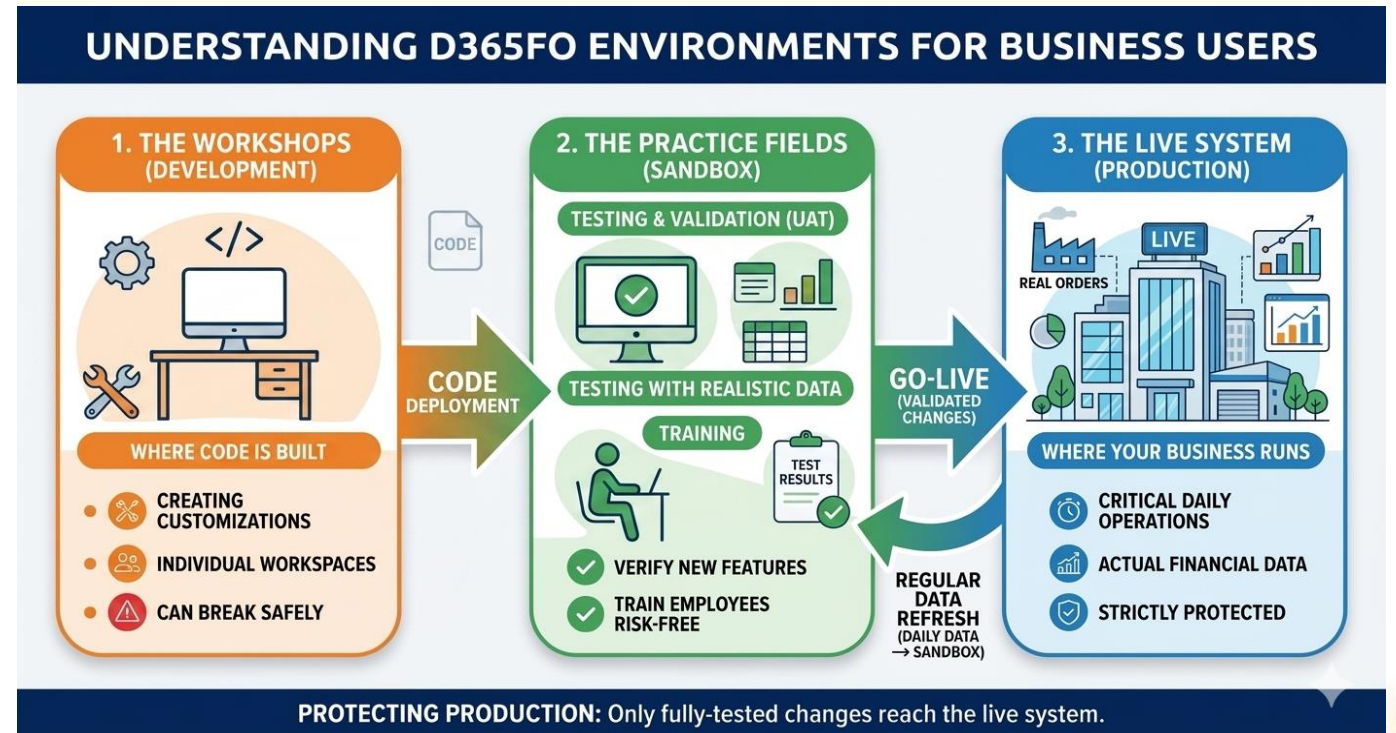


- Code moves from Lower tier to Higher tiers
- Data moves from Higher tier to lower tiers

Environments

Summary

- Dev environment – **Workshop**, for building new features/customizations
- Test/QA – **Sandbox**, for testing and training.
- Production – **Real world**



Builds and Deployments

User – Why are the latest changes not in Sandbox environment?

System administrator – I ran the build pipeline last night, but it failed because of NuGet version mismatch and models we received from ISV are missing reference so a package could not be generated to deploy to Sandbox.

User – Great 😞



Builds and Deployments

Why do we need to know something about this?

- **If you are having this conversation, you have a stake in process, as an IT person or Business analyst or a functional super user/consultant.**
- **To communicate and understand what is needed to get beyond this hurdle.**
 - **ISV model issue – Reach out to ISV and request update.**
 - **NuGet issue – System administrator can manage.**
 - **Label file reference – Let developer know that there is an issue with their checked in code.**



Builds and Deployments

Dev branch

Your model

Tax ISV Model

MS Model 1

Retail ISV Model

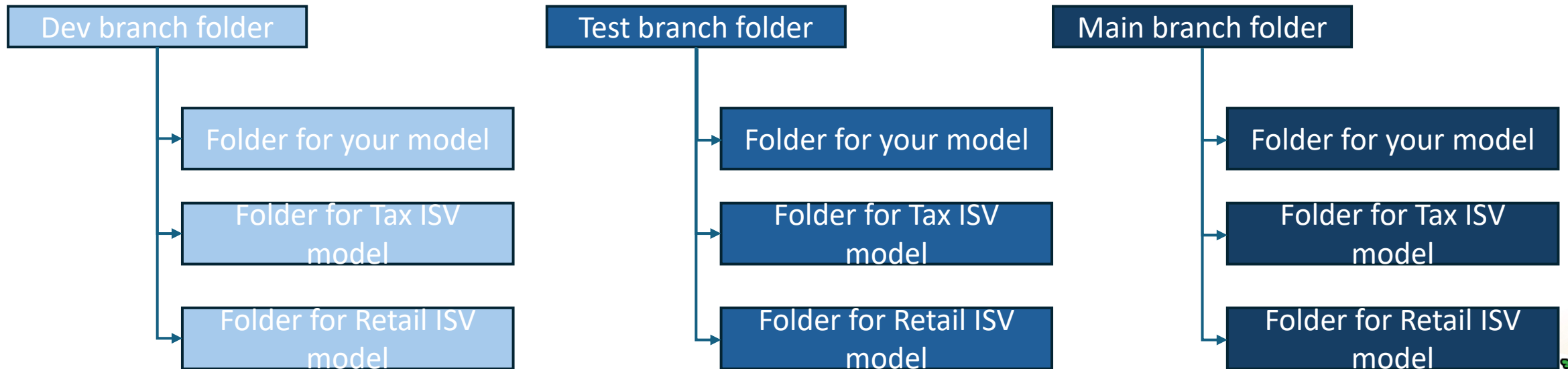
MS Model 2



Builds and Deployments

Branch - Simplest definition is "Folder" in your file system, it is a folder code files are stored.

Model – Simplest definition is "Subfolder", inside the branch folder where code files are stored and segregated.



Builds and Deployments

Code merge – Process to copy from one branch/folder to next, e.g. moving from Dev to Test.

Build – Process to compile code to ensure there are no errors.

Package – If build is successful, it generates a output package which is simply a zip file.

Deployment – Taking that package(Zip file) and applying it to next environment, Sandbox or Prod.



Builds and Deployments

Version control – Mandatory, choice between TFVC or Git. Either one does the same thing. Like SharePoint where you can see the history of a file edits, dates, users etc. and revert to a previous version if you need to.

Job of version control is to track code changes to files in those folders(branches)/subfolders (models) and merge them.

- Who made it
- When it was made
- What was changed
- Ability to merge multiple changes and rollback if needed



Builds and Deployments

Summary –



Builds and Deployments

Last two sections hopefully will help you these sample conversations in future

- Why is it working in Dev but not in Test?
- Can we go back to previous version?
- Can we do the code promotion tonight



Dataxxx

The 4 Concepts (Simplified)

Database

Where work happens

Dataverse

Smart business database

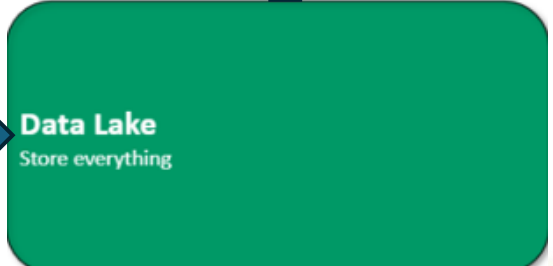
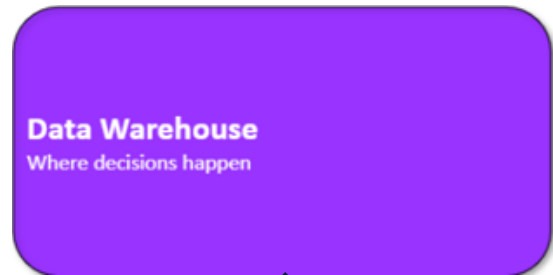
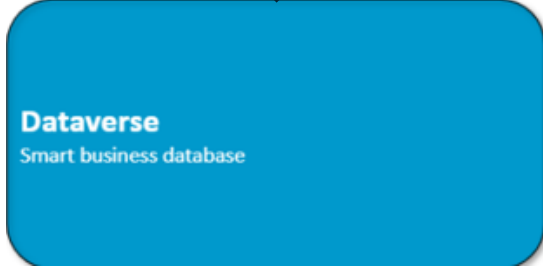
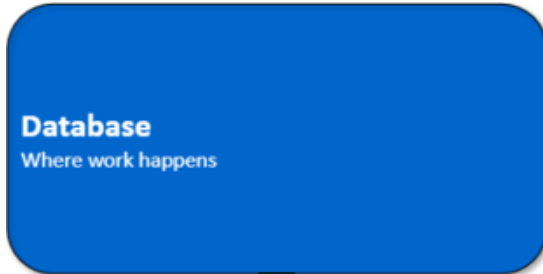
Data Warehouse

Where decisions happen

Data Lake

Store everything

- Database – Stores live ERP data. Someone creates a sales order it gets saved here. Each environment Dev/Test/QA has one SQL server database.
- Dataverse – External cloud-based database where ERP data can be picked and replicated and have tables of its own. Heavily used to build power apps and dual write integration with other D365 products like CRM, Field service, Project operations
- Data lake – Raw storage of data from multiple systems, data can be structured or unstructured





- **Stores all live transactional data**

Every action in D365FO (sales orders, invoices, inventory updates) is saved here in real time.

- **Powers day-to-day business operations**

This is the system working behind the scenes when users create, update, or post transactions.

- **Highly structured and organized**

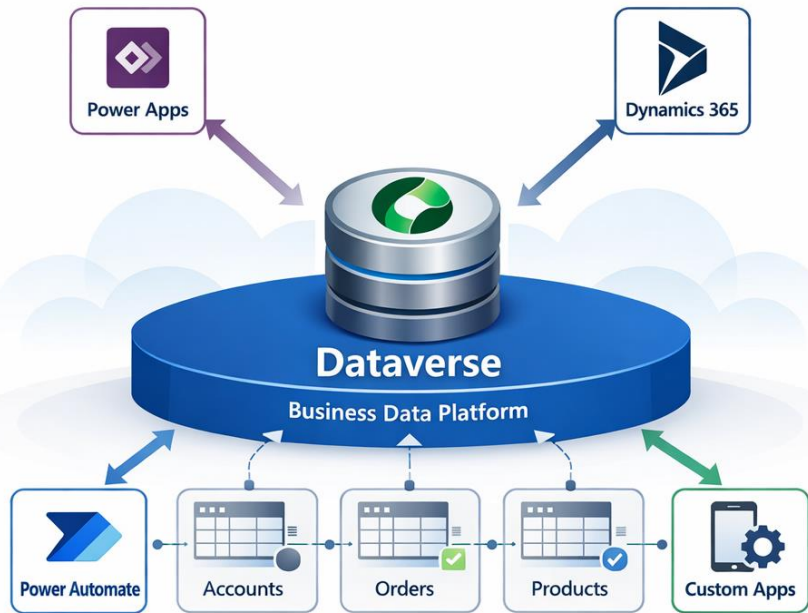
Data is stored in tables with defined relationships (customers, orders, products).

- **Optimized for speed and accuracy**

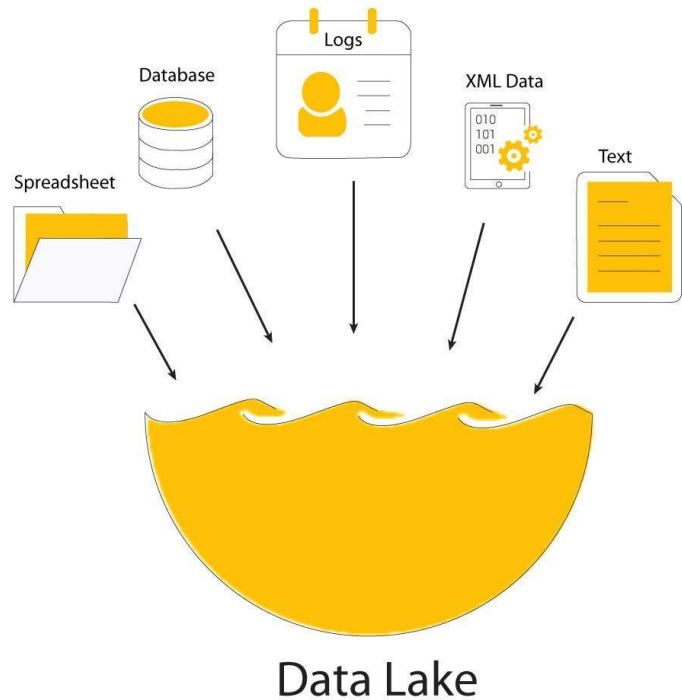
Designed to quickly process transactions without delays or inconsistencies.

- **Not meant for heavy reporting or analytics**

Running large reports directly on it can slow down the system—so data is usually moved elsewhere for analysis.



- External cloud-based database where ERP data can be replicated
- **Acts as a bridge between D365FO and Power Platform**
Enables apps and workflows to use D365FO data without directly touching the core system.
- **Simplifies integrations**
Instead of complex custom integrations, tools connect easily through Dataverse.
- Key component for dual write integration with other D365 products like CRM, Field service, Project operations



- Stores raw data from D365FO - Data from transactions, logs, and exports flows into a data lake without heavy processing.
- Handles large volumes easily - Ideal for storing years of finance, supply chain, and operational data at scale.
- Works with tools like Azure & Fabric - Integrates with platforms such as Microsoft Fabric for deeper insights.
- Acts as a staging area before reporting - Data is later cleaned and moved into a data warehouse for dashboards and reports.



- **Designed for reporting, not transactions** - Takes data from D365FO and prepares it for analysis without impacting system performance.
- **Combines data from multiple sources** - Brings together finance, supply chain, and even external data into one reporting model.
- **Stores historical, structured data** - Keeps clean, organized data over time (months/years) for trend analysis.
- **Optimized for dashboards and BI tools** - Works seamlessly with tools like Microsoft Power BI for fast reporting.
- **Transforms raw data into business insights** - Data is cleaned, aggregated, and modeled so users can answer business questions easily.



2026
DYNAMICS
CON



2026
DYNAMICS
CON



2026
DYNAMICS
CON



2026
DYNAMICS
CON



2026
DYNAMICS
CON









2026
DYNAMICS
CON

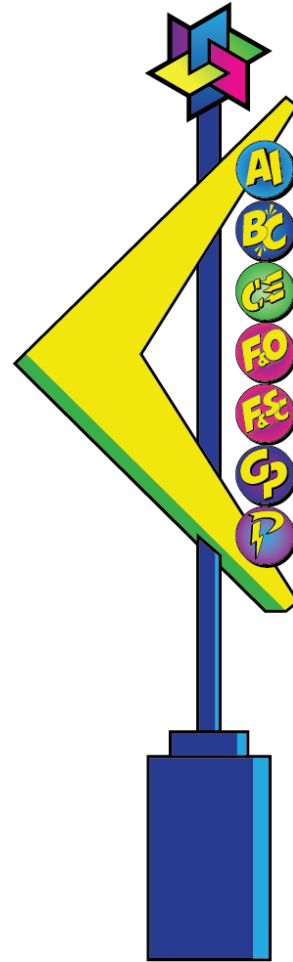




ASSETS



ASSETS



ASSETS

