



Case study

Global transport systems company finds ALM strategy to be a cornerstone of implementation

A global transport systems company embarked on a multiple-phase, multiple-region implementation of Dynamics 365 Sales, Customer Service, and Field Service with an aggressive schedule for achieving a minimum viable product (MVP) go live.

To give business stakeholders a taste of the solution before the official project kickoff, the customer asked their Dynamics 365 implementation partner to lead a rapid prototyping phase, which the customer referred to as “Phase 0.” The Phase 0 team included technical salespeople, as well as consultants who would leave when the project kicked off.

Considering Phase 0 to be a rapid prototype, the implementation team didn’t see a need to focus on governance and quality. To meet the deadlines, the team made several quick decisions without assessing the impact on the overall application management:

- Staffing Phase 0 with resources who wouldn’t remain part of the project team.
- Developing the prototype using separate, unmanaged solutions with different publishers for Sales, Customer Service, and Field Service.
- Pivoting to managed solutions for test and production environments too late in the design phase, after unmanaged solutions were already built and deployed to these environments during Phase 0.

- Merging the Sales, Customer Service, and Field Service solutions into a single “workstream” solution during the build phase of the implementation lifecycle.
- Resolving errors too quickly during solution deployments because of pressure from project leadership, instead of taking the time to understand each problem’s root cause. (Such pressure led to developers customizing directly within the test environment.)

As the project team prepared for the go live, the seemingly independent decisions made during the initial phases resulted in deployment issues that eventually stalled the go live. Investigations by Microsoft Support confirmed that there was no ALM strategy in place, and identified the key issues:

- Unmanaged solutions caused solution-layering issues for testing and production, affecting the sanctity and integrity of the environments
- A prototype solution employed for production purposes introduced quality issues
- Failure to use ALM practices such as DevOps caused traceability issues and prompted developers to build functionality that wasn’t aligned with customer requirements
- Suboptimal code was implemented because tools such as solution checker weren’t used to enforce code quality
- Testing without traceability was insufficient, and buggy code was deployed to other environments

As the old saying goes, “By failing to prepare, you are preparing to fail.” The MVP go-live date was delayed by 12 weeks and Microsoft worked alongside the project team to determine the root cause of each issue. The team eventually acknowledged that a series of seemingly unrelated decisions affected the MVP go live, and they sent every team member working in a technical role to an ALM refresher training. The project team also confirmed plans that should have been solidified at the beginning of the project, including an environment strategy and a solution management and ALM approach.

After the refresher training, the project team started with a “crawl to walk” approach. During the “crawl” stage, they implemented mandatory ALM practices with these governance elements:

- Cleaning up their production and test environments, and moving from unmanaged to managed solutions
- Implementing a build process to deploy builds from development to test to production
- Establishing a governance process to restrict developer access to production and test environments
- Adding a bug-triaging process that allowed the development team to troubleshoot and fix issues in development environments and use the build process to deploy fixes in higher-tier environments
- Mandating the generation of solution checker reports

Once these practices were in place, the implementation team moved toward a partially automated ALM process that included setting up DevOps pipelines to automate build deployments and auto-generate solution checker reports. As this case study illustrates, an ALM strategy is a cornerstone of the entire implementation—whether it's a fresh implementation or a rapid prototype elevated to a production-use solution.