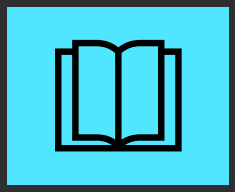

Case study

Corporate travel company learns how testing is critical to measure and optimize performance

A global corporate travel company was implementing Dynamics 365 Customer Service to drive a call center transformation. Several process flows required integration with legacy systems—many with high transaction volumes—and there was a business-mandated customization that the project team agreed couldn't be achieved using out-of-the-box functionality. This complex solution was projected to support 4,000 users at scale and needed careful planning to ensure optimal performance.

Despite the complex customizations and integrations, the customer had decided performance testing was out of scope due to budget constraints, assuming that because Dynamics 365 is a cloud service, its performance is owned by Microsoft.

While still in the design stage, the project team decided to use the best practices from the Success by Design framework and their own experiences to highlight the risks of leaving performance testing out of scope. The project team pointed out potential negative outcomes if performance wasn't considered:

- Users wouldn't adopt the system if its performance affected user productivity.
- Costs would be higher if reactive measures were required to address performance issues later.

The project architect worked with the company's stakeholders to explain that while Dynamics 365 is a cloud service, factors such as additional customizations, data volumes, and network latencies each play a role in driving the performance of the application. Their recommendations were summarized as follows:

- Review the application design and code for scalability.
- Establish baselines and review the areas that need to be optimized, including network configurations.
  - Formulate a baseline for form load times to understand performance and align a new solution.
  - Ensure performance with integrated systems meets the company's expectations.
  - Agree on realistic key performance indicators (KPIs) for the business.
  - Assess current end-to-end connectivity to determine network performance and latency.

Based on the project team's observations, the project steering committee was advised to approve adding solution performance testing to the test cycle, and agreed to do so.

The customer's architect worked with the implementation partner's architect to plan for performance testing by:

- Defining the key criteria for testing, including integration scenarios.
- Assessing the concurrent user loads and peak usage.
- Examining the data volumes.
- Establishing the tools for testing.

Performance testing helped the team identify several gaps before deploying the solution to production. In addition, the team:

- Optimized their network configuration.
- Identified sync calls that were blocking resources, and made the necessary design changes.

- Reinforced and incorporated basic patterns (such as retrieving only the required attributes) across their code components.

As mentioned earlier in this chapter, “Prevention is better than cure.” Performance testing ensured optimal performance and stakeholder alignment before the system was made available for end users. For overall success and higher levels of adoption, a performance plan is fundamental throughout an application’s lifecycle.