



SEVEN BEST PRACTICES FOR AUTOMATED TESTING

In Dynamics 365

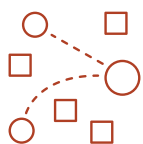
Automated testing is essential in modern software development, especially for complex systems like Microsoft Dynamics 365. It ensures that applications meet their functional requirements and maintain performance standards without excessive manual effort. This guide delves into seven best practices that can enhance the efficiency and effectiveness of your automated testing strategies for Dynamics 365.



1. Develop a Comprehensive Test Strategy

Creating a robust test strategy is foundational for successful automated testing. It should align with your business goals and cover all critical functionalities of your Dynamics 365 setup. Here are key elements to consider:

- **Define Testing Objectives:** Clearly define what you aim to achieve with your tests. Focus on risk areas and functionalities that are crucial for your business operations.
- **Select Appropriate Tools:** Choose tools that integrate seamlessly with Dynamics 365, considering factors like support and user community. A resource produced by Microsoft lists options for Dynamics 365 test solutions in this [Microsoft Learn article](#).
- **Plan Test Scenarios:** Design scenarios that cover every function, including edge cases. This comprehensive coverage helps ensure your application performs well under all expected conditions.



2. Establish a Dedicated Testing Environment

Testing in an environment that replicates your production setup as closely as possible is vital. This practice helps catch issues that could otherwise disrupt business processes:

- **Replicate Production Settings:** Mimic your production environment's configuration to ensure consistency between test results and actual performance.
- **Manage Test Data Effectively:** Utilize realistic and secure test data and maintain its integrity through tools and processes designed for data management.
- **Regular Environment Review:** Update the testing environment to reflect any changes in the production environment, which helps maintain the relevance and accuracy of your tests.



3. Prioritize Test Automation Areas

Prioritize areas based on their impact on your business and the complexity of testing:

- **Identify High-Risk Areas:** Focus on parts of the system that are critical to your operations or have had issues in the past.
- **Balance Cost and Benefits:** Evaluate the potential return on investment for automating different areas, focusing on those that offer the greatest value.
- **Continuously Evaluate Automation Impact:** Regularly review the effectiveness of your automation efforts to ensure they meet the evolving needs of your Dynamics 365 environment.



4. Implement Continuous Integration and Deployment

Integrating automated testing into your CI/CD pipeline can significantly enhance the quality and speed of your deployments:

- **Automate Early in the Development Cycle:** Incorporate automated tests early to catch issues before they progress through the pipeline.
- **Ensure Robust Configuration Management:** Use version control to manage changes in test scripts and dependencies, which helps maintain order and accountability.
- **Monitor and Report:** Utilize monitoring tools to oversee the testing process and use dashboards to provide insightful reports on test outcomes and system health.



5. Utilize Modular Test Scripts

Modular test scripts are easier to maintain and can be reused across different testing scenarios, making your testing process more efficient:

- **Develop Reusable Components:** Build test scripts that can be easily adapted or reused for various environments and multiple companies within your organization.
- **Test Multiple Security Roles:** Run the tests using multiple security roles to ensure that various users can successfully perform the tests or are not allowed to perform the tests, depending on the roles' security settings.
- **Leverage Script Libraries:** Maintain a central repository for scripts and other testing resources to facilitate easy access and reuse.



6. Conduct Regular Regression and Performance Testing

Regular testing is crucial to ensure new developments do not negatively affect existing functionalities:

- **Schedule Frequent Tests:** Automate regression and performance tests to run at regular intervals, ensuring continuous oversight.
- **Use Performance Baselines:** Set benchmarks for application performance and regularly compare ongoing performance against these standards.
- **Adapt Tests to Changing Conditions:** Update your tests according to the new changes and features added to Dynamics 365. Or work with a vendor that updates the test components for each version of Dynamics 365.



7. Engage Stakeholders with User Acceptance Testing

While automated tests handle many testing aspects, human insights are invaluable, especially for assessing user experience and application usability:

- **Automate UAT Setup and Cleanup:** Automating the setup for user acceptance testing simplifies the process, allowing more focus on the actual testing.
- **Facilitate Effective Communication:** Establish efficient communication channels for stakeholders to provide feedback easily.
- **Incorporate Feedback into Continuous Improvement:** Integrate user feedback to prioritize enhancements and fixes into your regular development and testing cycles.

Conclusion

By adopting these best practices, you can significantly improve the effectiveness of your automated testing regime for Microsoft Dynamics 365. Each step contributes to a more robust, reliable, and efficient system, ensuring your deployments are smoother and your systems are more resilient. As Dynamics 365 continues to evolve, so should your testing strategies to keep pace with new functionalities and industry demands.