Tricentis

How to future-proof your Oracle testing strategy

INTRODUCTION

Cloud technologies are revolutionizing our working world – including Oracle Cloud. With a substantial list of legacy Oracle products sunsetting premier support over the coming years – including JD Edwards, Siebel, and PeopleSoft – Oracle customers should plan now for the migration and retirement of these applications. According to Oracle, moving to the new, cloud-based solutions will lower cost of ownership, increase agility, and improve productivity for Oracle customers.

But before you can enjoy these benefits, it's critical to develop a migration strategy that both minimizes risk during the migration and lays the foundation for long-term success. And the strategy must be broad: The migration will impact not only Oracle applications, but also customizations and connected technologies (UIs, APIs, mobile, databases, third-party applications, etc.) that collectively run your business. Without a well-planned migration, your business is at risk for cost overruns, system downtimes, security lapses, and other serious consequences.

One of the best ways to minimize these risks is a sound testing strategy that can ensure business processes function as they should in the new systems. But all too often, companies rely on legacy tools or manual methods that can't accommodate the speed and volume of testing required during an Oracle migration. The result: further extending the project timeline, increasing the IT budget, and derailing the integration of end-to-end processes. And that doesn't even take into consideration the testing required for quarterly updates for Oracle's cloud solutions or the customizations driven by your business.

This reality behind traditional methods makes testing stand out as one of the most significant barriers to innovation. With Oracle halting development on the Oracle Application Testing Suite (OATS), the need to develop a new testing strategy has become more pressing than ever. (After June 2022, Oracle has continued its investment in Oracle Application Testing Suite only for those customers needing to test E-Business Suite, according to a statement of direction published by the company.)

Regardless of your current approach, you will need to refine your testing processes to lay the groundwork for the future.

There are several testing approaches out there, each with their own pros and cons. Some include:

Open-source test automation

- Cost-effective and easily accessible
- Works well for cloud-native applications
- Doesn't integrate well with legacy applications, making testing during a migration difficult
- Often includes a disparate set of tools, which can create information silos and limit scale
- Test scripts require substantial maintenance, which can be a hidden cost

Manual testing

- No need to invest in a tool or adjust current processes for shops who are manual today
- No coding skills required means a broader team can contribute
- Time-consuming – meaning releases get delayed or move forward with inadequate test coverage
- The volume of testing can quickly become unsustainable with Oracle's quarterly release cycles

Legacy, script-based test automation tools

- Purpose-built automation tools improve timelines over manual testing
- Proprietary scripting language requires the highest-skilled resources, making test automation expensive to build and maintain
- Inadequate end-to-end test coverage

Let's discuss factors to consider while weighing your options.

DEVELOPING AN ORACLE TESTING STRATEGY: KEY CONSIDERATIONS

Software testing is essential for ensuring that business continues to run as usual before, during, and after an Oracle migration. Oracle is providing support to businesses migrating to Oracle Fusion Cloud (OFC) until 2032. You have plenty of time to prepare, but no time to waste. This migration is an enormous undertaking, so the sooner you create a sustainable testing strategy, the smoother your OFC migration will go.

You may be tempted to delay developing a test automation strategy if you are planning a major migration because your testing strategy will change. Instead, consider implementing a modern testing approach now that will improve your ability to keep up with on-premises updates and get your team familiar with the required process changes before you begin moving workloads to the cloud. If you start automating testing now, you can avoid becoming overwhelmed by the need for testing transformation in the midst of a major migration project. With that in mind, here is a list of key considerations for creating a modern testing strategy.

Complex systems landscapes

Your Oracle applications are part of a complex system landscape. A single business workflow might begin in Oracle, then touch dozens of these applications via microservices and APIs. To ensure Oracle business processes keep running, testers must replicate these complex, multi-application workflows and ensure none of those workflows are impacted when one of those applications is updated.

Cloud migration further complicates things. In the midst of migration, when only some workflows have moved to the cloud, many tests must function across the heterogenous architectures where business processes are occurring. That's why it's critical to ensure your testing strategy can support both legacy and modern applications, as well as accommodate end-to-end testing across complex workflows.

Faster release cycles

Oracle's mandatory quarterly release cycle for its Fusion Cloud environment deploys its pre-production two weeks before Oracle moves it into active production. You need to be ready to follow Oracle's pace.

In addition to accepting these required quarterly updates, you also need to plan for any custom Oracle development your organization plans to release. If you are working on an Agile timeline, you'll need to be able to complete the required testing as part of a two-week sprint. This makes test automation that enables "day one" testing on staging environments a must-have resource for cloud customers.

Business risk prioritization

Many Oracle test case portfolios suffer from a common problem — a large volume of tests but very little coverage of the application. Teams end up producing duplicate test cases and performing even more maintenance to cover the ever-growing suite of tests. This is an impractical and time-intensive process that can quickly become untenable as release cycles shorten. Adopting a risk-based approach can address this problem by prioritizing testing for the most business-critical functionality, ensuring that the highest-risk areas are covered and that your team doesn't waste time running unnecessary tests. To adopt a risk-based approach, teams should:

- Assign features, user stories, or requirements with high to low priority based on business decisions. Understand which functionality is most critical to the business, and focus on ensuring those areas are covered first.
- Set a risk coverage target, rather than a test coverage target. A common target is 90% coverage for the most business-critical functionality. Make your go/no-go decision confidently when you have achieved your target.
- Strive for the fewest possible tests needed to 1) reach your risk coverage targets and 2) ensure that when a test fails, you know exactly what application functionality to investigate.

Long-term scalability

Earlier in this white paper, we touched on the pros and cons of various testing strategies. By now, it should be clear that manual testing is untenable for the long-term scalability of a complex Oracle instance. If your team is equipped with a scalable test automation strategy, then there's nothing more to do than ensure you have the resources to maintain those scripts. But if you've made it this far, there's a good chance this isn't the case for your team.

But even if the team working on Oracle testing has achieved a high automation rate, many teams hit a roadblock when it comes to testing the end-to-end workflows across systems. This is because automating end-to-end tests is difficult to achieve with script-based tests – not to mention a nightmare to maintain.

This is an ideal opportunity to introduce codeless test automation, which will empower testers at all skill levels to meaningfully contribute to test automation, without the expertise required to build or maintain scripts. A codeless solution is also ideal for teams of manual testers who find themselves faced with an unsustainable testing timeline.

Regardless of where you are in your migration journey, the right test automation strategy can accelerate innovation while reducing workload, enabling you to transform testing into a strategic value-add rather than an expensive and time-consuming obstacle on the path to Oracle modernization.

Success story: Experian cuts Oracle testing timelines in half with codeless test automation

Experian recently embarked on a project to transform its Oracle EBS testing strategy to improve compliance with company policies, reduce costs, and improve quality and release confidence. For years, the team had relied on Micro Focus UFT, but with an increasing release cadence and a quickly growing test case library, the legacy way of working had become unsustainable.

Experian's regression suite consisted of 3,500 test scripts built in UFT, which relied on data, object properties, and custom libraries that were hard-coded into the test scripts. There wasn't enough time to update everything across all 3,500 test scripts before each release, so many of the test cases had become unstable and unreliable. Because many were not able to run without manual intervention, executing the suite took a team of four testers four weeks.

> Today, an efficient regression suite provides virtually instantaneous quality feedback on the risks of a given update, and a broader team of testers can contribute to new test automation that is needed for each release.

With the help of Tricentis Tosca's codeless technology, the team cut that timeline in half. They analyzed the current test suite in three months, migrated the 75% that was useful, and developed new test cases to address critical test gaps. Today, an efficient regression suite provides virtually instantaneous quality feedback on the risks of a given update, and a broader team of testers can contribute to new test automation that is needed for each release. Following its initial success, Experian has applied this test automation method to more projects since and is well positioned to tackle future application modernization initiatives.

DISCLAIMER: Note, the information provided in this statement should not be considered as legal advice. Readers are cautioned not to place undue reliance on these statements, and they should not be relied upon in making purchasing decisions or for achieving compliance to legal regulations.

ABOUT TRICENTIS

Tricentis is a global leader in enterprise continuous testing. The Tricentis AI-based, continuous testing portfolio of products provide a new and fundamentally different way to perform software testing. An approach that's totally automated, fully codeless, and intelligently driven by AI. It addresses both agile development and complex enterprise apps, enabling enterprises to accelerate their digital transformation by dramatically increasing software release speed, reducing costs, and improving software quality. Widely credited for reinventing software testing for DevOps, cloud, and enterprise applications, Tricentis has been recognized as a leader by all major industry analysts, including Forrester, Gartner, and IDC. Tricentis has more than 2,500 customers, including the largest brands in the world, such as McKesson, Accenture, Nationwide Insurance, Allianz, Telstra, Dolby, and Vodafone.

To learn more, visit www.tricentis.com or visit one of our locations, www.tricentis.com/locations.