AVO® Automation

How No-code Process Discovery and Test Automation Simplify Oracle Testing

Index

1. What is Oracle testing?.....03

2. Why is automation a necessity for testing Oracle applications?.....03

   a. The Inefficiencies of Manual Testing.....04

   b. What are the specific challenges of testing Oracle Applications.....05

      1) Frequent updates and patches 05

      2) Customizations.....05

      3) Integrations.....05

      4) Migration.....05

3. Why are current automation solutions losing steam?.....06

4. What should a ideal test automation solution offer?.....07

5. How process discovery simplifies test automation?.....08

6. How does Avo offer unique benefits?.....09

What is Oracle Testing?

From offering ERPs and databases to various other core business process solutions, Oracle continues to be the most prominent and widely-used enterprise software vendor in the world. Like every other mission-critical application, ensuring your Oracle applications function seamlessly is crucial to keeping your business running.

Today, at least 90% of how you interact with a business is driven by the experience. It doesn't stop at one extremely gratifying experience since even a single instance of dissatisfaction could drift your customer away. A Forbes post states that – customer loyalty drops by as much as 20% when issues aren't addressed at the right time.

Software testing plays a pivotal role in assuring your applications function as intended and avoid any operational and financial consequences.

Oracle applications are usually a part of a broader technology stack. Testing them end-to-end is seldom easy. This ebook breaks down the complexities of Oracle test automation and will help you hop on the automation journey swiftly.

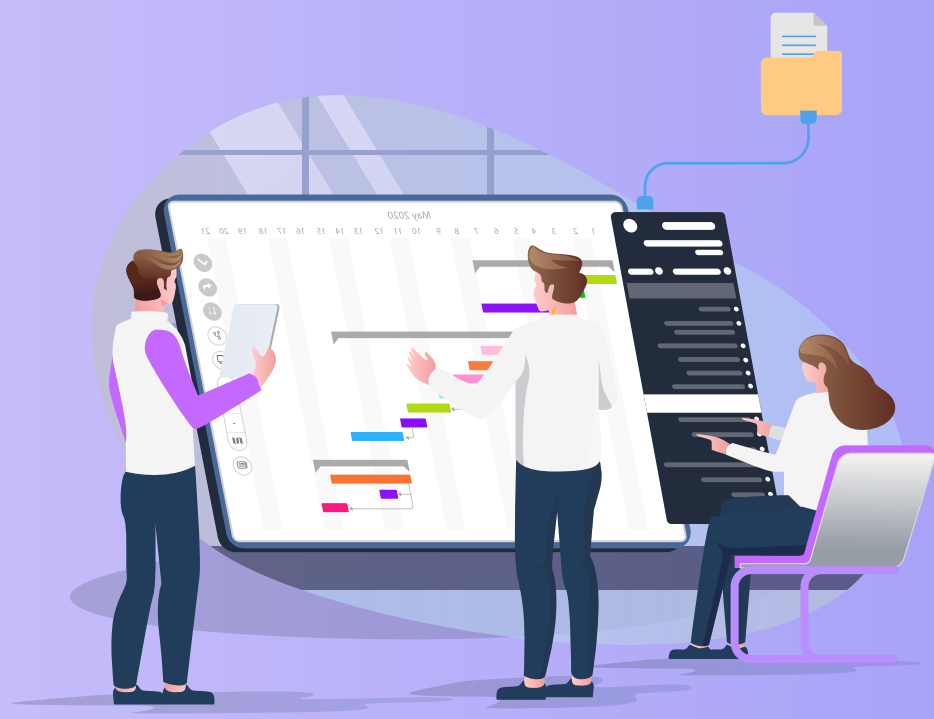
Why is Automation a Necessity for Testing Oracle Applications?

Manual testing was the norm until a few years ago. As the width and depth of business applications increase, the necessity to test them efficiently becomes prominent. Manual testing can't keep up with this growing demand

Manual testing is slow, error-prone, tedious, and counter-productive. The results – poor quality, slow and inconsistent results, and lower team productivity. A CISQ report[1] shows that the cost of poor quality software in the US in 2020 was approximately $2.8 trillion. Getting one level deeper – if fixing a bug in the development phase costs $100, the same bug costs $10,000 to be fixed in production.

[1] https://www.it-cisq.org/cost-of-poor-software-quality-in-the-us/index.htm#:~:text=The%20report%20states%20that%20the,from%20%24635%20billion%20in%202018)

The Inefficiencies of Manual Testing



Agile teams often consider manual testing as the primary bottleneck to faster delivery due to the challenges it poses:

1. Incomplete test coverage

With applications growing at the light's speed, it gets tricky to test every functionality manually. This results in untested features and, in turn, insufficient test coverage, which lowers the team's confidence. Not to mention the strain it puts on testing teams to squeeze their QA efforts into a 3–4 week sprint.

2. Technical debt

Each sprint has some residual issues from the previous sprint that impact the current timelines. If your testing methodology is manual, this technical debt exists, affecting the overall delivery timeline.

3. Delayed releases

From trying to achieve end-to-end testing to accommodating regression tests in agile sprints, manual testing usually causes some delay in releases.

Test automation helps you overcome these challenges. It promises higher quality, faster delivery, lowers technical debt to near-zero, and proves cost-effective if implemented strategically.

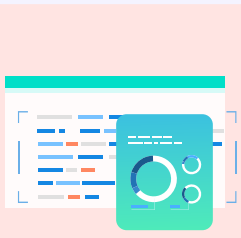
What are the Specific Challenges of Testing Oracle Applications

Frequent updates and patches

Oracle applications are synonymous with regular updates. These updates are a result of market demands or business strategy changes. Oracle apps should be tested frequently and faster to ensure these updates are up and running in no time. Manual testing can't catch up on speed, making test automation necessary.

Customizations

Oracle applications are customizable to meet the unique demands of each business. Thus, the more customizations, the greater the testing effort involved since the risk of breakage is high. Only automation can help teams test every possible scenario without bogging down your team's productivity.



Numerous integrations

Since Oracle applications rarely exist in siloes and involve diverse integrations, testing all the crucial integration points is tricky without test automation. When done manually, it is nearly impossible to test the entire breadth of the business process in lesser time and limited resources.

Migration from EBS to Cloud

While this topic deserves an elaborate explanation of its own, it doesn't hurt to consider it as one of the primary reasons why Oracle apps need test automation. Testing Oracle migration entails more than just one successful User Acceptance Testing cycle. Without continuous testing, teams can't mitigate risks, avoid roll-backs, or give faster feedback to dev teams.

Why are Current Automation Solutions Losing Steam?

A survey[2] by TechWell indicates that the widespread adoption of test automation still has a long way to go, as 75% of the respondents automate less than 50% of their testing. The same survey also reveals that 47% of the respondents use a combination of open source and commercial tools. 36% use open-source tools exclusively, and a mere 8% use commercial tools exclusively.

These numbers suggest that open-source tools still play a significant role in testing teams. Tools like Selenium transformed the QA automation space in the last few years. It is free, open-source, and has a large community of users. Fast-forward a few years, the emergence of low-code automation tools empowered teams with customizations, improved support, and more.

However, in both cases, the following challenges failed to keep the true promise of automation:

Tedious coding effort

Selenium or your commercial low-code test automation solution requires writing scripts to set up and maintain tests. If you still 'write code' to automate it, are you really ' automating?' In the long run, the complexity increases as your Oracle applications grow. This makes testing a tedious and error-prone task.

Test only browser-based applications

Most low-code and open source testing tools offer automation exclusively for web-based applications. That leaves non-web apps based on mobile, Oracle, SAP, Mainframe, and more for manual testing. Achieving end-to-end test automation is far-fetched, and executing one complete regression cycle is a tester's nightmare. This increased maintenance complexity rarely keeps teams from reaping the benefits of automation.

Poor reporting capabilities

Today, data is everything, and if your test automation solution doesn't offer it, it affects your teams' efficiency. Selenium fails to provide reporting, causing delays in troubleshooting the errors. For time-sensitive Oracle applications, a slight delay means consequent business losses. Also, since it is an open-source tool, your community is your support, and it may or may not be available 24/7.

[2] https://www.tricentis.com/wp-content/uploads/2018/05/Tricentis-Report-The-Evolution-of-Test-Automation-2018.pdf

What Should a Test Automation Solution Ideally Offer?

With various test automation solutions in the market, it could get tricky to choose the one that suits your requirement best. Here are a few capabilities to look out for.

No-code

A no-code test automation solution empowers you to test applications with a few clicks of the buttons or without having to write a single line of code. It saves time and effort in scripting and helps expedite testing. Additionally, it shortens the learning curve and speeds up the solution's adoption.

Most importantly, it lets a broader group, including software developers, QA specialists, and business analysts, participate in testing. It bridges the prevailing gap between IT and business, helping them collaborate better. This translates to faster feedback between teams, better software, and exceptional business outcomes.

Heterogeneous

When a solution is heterogeneous, your test automation isn't limited to web applications. A cross-compatible solution allows you to test web, Oracle, SAP, Salesforce, mobile, mainframes, and more. This way, you can test an entire business process from start to finish using one single solution, improving automation coverage. Added perks – you save the effort and time involved in downloading, installing, and maintaining individual plugins.

Intelligent reporting

The importance of well-interpreted data cannot be emphasized enough. The same goes for comprehensive and intuitive reporting. If a report helps you pinpoint a bug without looking more in-depth, it saves you time and helps fix the issue on the fly without interrupting the business.

Smart Scheduling

Your solution should allow you to execute multiple scenarios in a single VM independently or in parallel. Especially in the case of regression testing, this provides multitasking and saves time.

Integration with CI/CD Pipeline

Rapid development, testing, and deployment of new software contribute to a significant part of continuous integration/continuous delivery (CI/CD). Test automation solutions should be an integral part of this CI/CD pipeline to ensure the quality of applications isn't compromised for speed.

24/7 Support

Be it helping you get started with the solution or cushioning you through major setbacks while testing, a support team's availability round the clock is crucial.

How Does No-Code Process Discovery Simplify Test Automation?

Optimizing business processes before testing them is one of the primary objectives of enterprises. It improves cost-effectiveness and efficiency. The fundamental requirement is to know and understand a process to optimize it. Process documentation, when done manually, is cumbersome and takes weeks or even months to document one single enterprise business process.

A Gartner report claims that more than 80% of companies still depend on manual process mapping and spend 65% of their time re-documenting steps rather than optimizing the process.

A no-code process discovery solution saves time and effort. What took weeks or months now takes a few hours or days. With all your documenting efforts reduced to a few buttons clicks, your team now has more time and capacity to optimize the processes.

When you test these processes with a no-code test automation solution, the outcome is higher-quality, efficient applications.

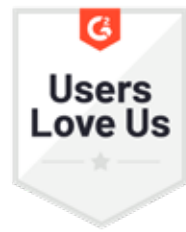
How Does Avo Offer Unique Benefits?

Avo's no-code process discovery and test automation solutions empower the largest Fortune 500 companies to reimagine quality assurance.

With Avo Discover, you can rapidly capture, document, and modify end-to-end business processes, enabling the team to optimize them. With its intelligent, no-code, and heterogeneous capabilities, Avo Assure helps teams achieve end-to-end test automation. The best thing – both Avo Discover and Avo Assure work in unison or individually to help you get the most out of your applications.

With our no-code automation suite, enterprises have achieved the following milestones:

- Expand test automation coverage to >90% by executing end-to-end tests for diverse applications, including Oracle
- Reduce defect injection rate in production to less than 2%
- Deliver high-quality applications at least 3x faster
- Achieve more than 40% cost savings
- Increase employee capacity for innovation by saving 85% of the manual testing effort
- Be 2x more productive
- Leverage over 1500 keywords and reduce testing time
- Save over 45% of your regression testing effort



Want to know how Avo can help you ride the automation wave?

Reach out to us, and we will schedule a demo for you with our experts.

Schedule a demo

You could also opt for a free 14-day trial to test a scenario of your choice. Click here.