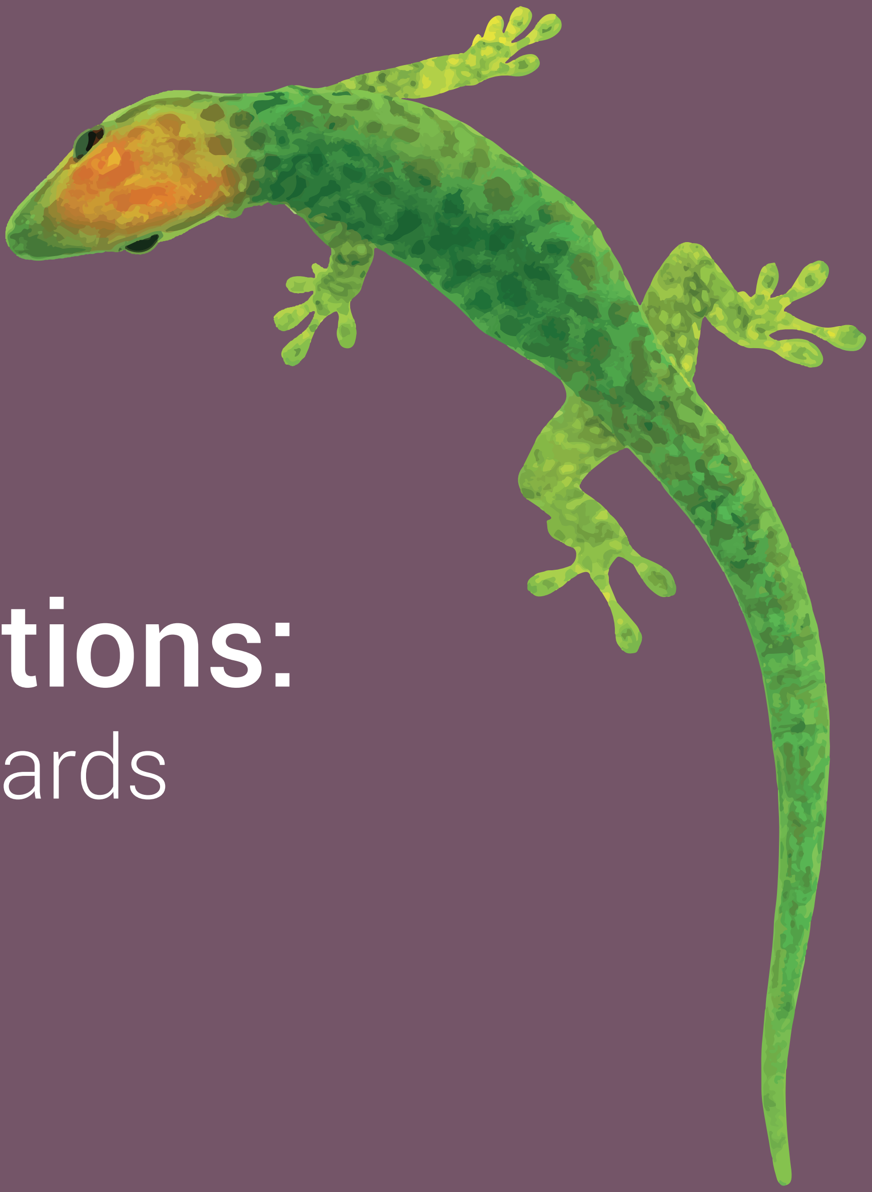


Functional Testing Options:

Oracle JD Edwards EnterpriseOne

A DWS Publication ©2017

Contents

1. Why Testing is Important

It may sound obvious, but software does not function like a physical process. This is most apparent when it doesn't function, or rather when it fails. Physical processes typically have a simple input and output relationship – you put a combination of elements in one end and something predictable comes out the other end. If what comes out the other end is not what was expected, there are typically few variables to be considered.

Software doesn't work that way. When it fails it tends to do so in unexpected and "mysterious" ways. The more complex the software design, the more difficult it can be to discover the reason for the error.

According to a paper by Carnegie Mellon University:

> **"Software Testing is the process of executing a program or system with the intent of finding errors. Or, it involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results."**

In the same paper, the authors go on to state that software testing is a key component of quality assurance and risk management; and that it can account for between 30% and 50% of a project budget. Feedback from JD Edwards EnterpriseOne customers indicates that software testing can account for up to 65% of the total upgrade effort.

For some organisations, software testing is like an insurance policy. They evaluate the potential risks associated with a change event, decide how much "protection" they need and then agree a premium they are prepared to pay. The amount of effort invested in testing should be commensurate with the potential risk to the business and the complexity and frequency of any changes made to the software or operating environment.

For example, software used to control military equipment, aircraft or pharmaceutical products will have potentially disastrous consequences should the software fail, so requires more rigorous testing than a simple non-critical business application.

The diagram below is based on a Microsoft Training paper (The Total Solution Life Cycle) and illustrates the associated and incremental costs of fault finding and fixing as a software project progresses.

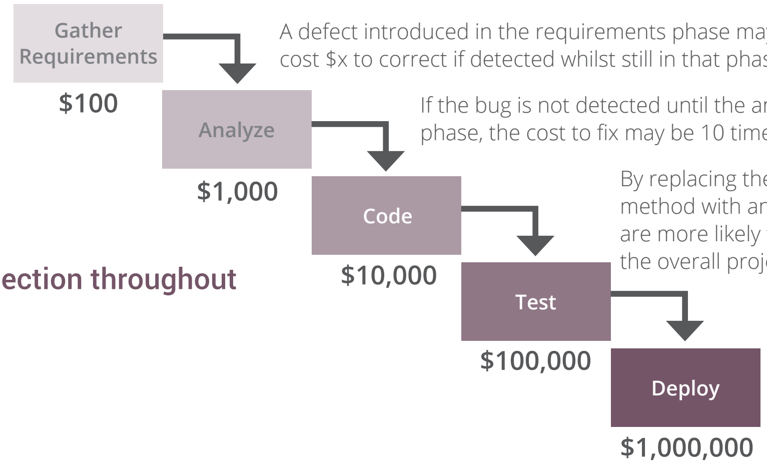
Reducing the burden of testing, and doing a better job of testing, is more important now than ever. Oracle JD Edwards have announced that Release 9.2 will be supported at least until 2028.

They have also made it clear that they intend to continuously deliver improvements from Release 9.2 and have no plans for a Release 9.3.

This means that companies need to think about their own continuous improvement strategies and the frequency with which they will run projects once they get to Release 9.2.

Every project can be made smaller, faster and smarter by improving the way companies plan, manage and execute their testing.

Traditional Waterfall Method



A defect introduced in the requirements phase may cost $x to correct if detected whilst still in that phase.

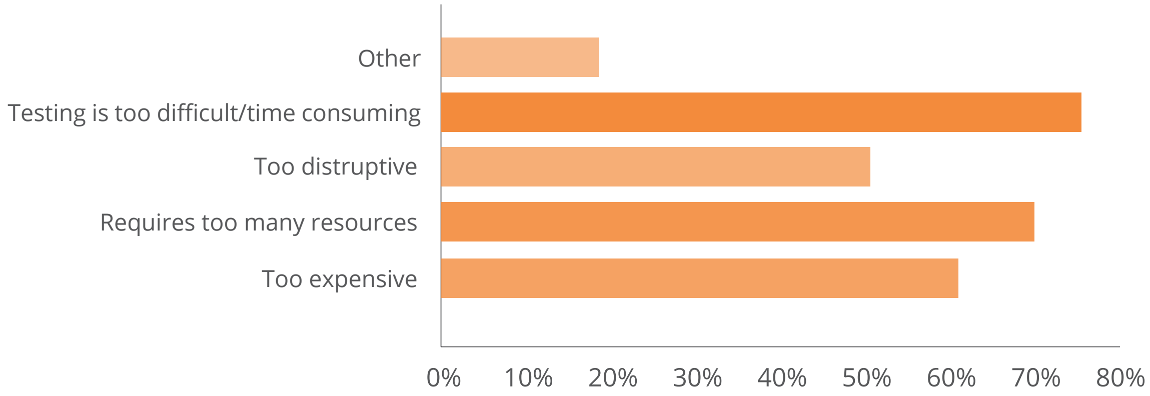
If the bug is not detected until the analysis and design phase, the cost to fix may be 10 times as much.

By replacing the 'waterfall' process method with an 'iterative' model, defects are more likely to be detected earlier and the overall project time and cost is reduced.

Incremental cost of bug detection throughout the lifecycle of a project.

2. User Experience of Test Automation

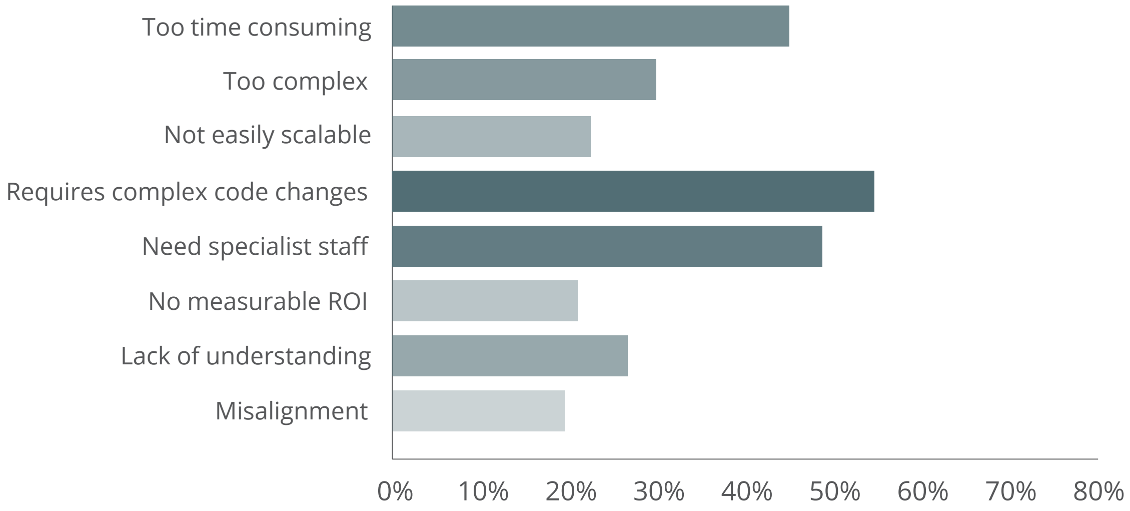
The Oracle JD Edwards keynote presentation at Collaborate 2017 revealed that just one in four JDE EnterpriseOne customers was on the latest release. In recent DWS research among more than 100 JDE EnterpriseOne customers, we asked what issues were stopping the adoption of a code-current strategy. 76% of respondents stated that "Testing was too difficult and time consuming"



What is preventing you from adopting a code-current strategy?

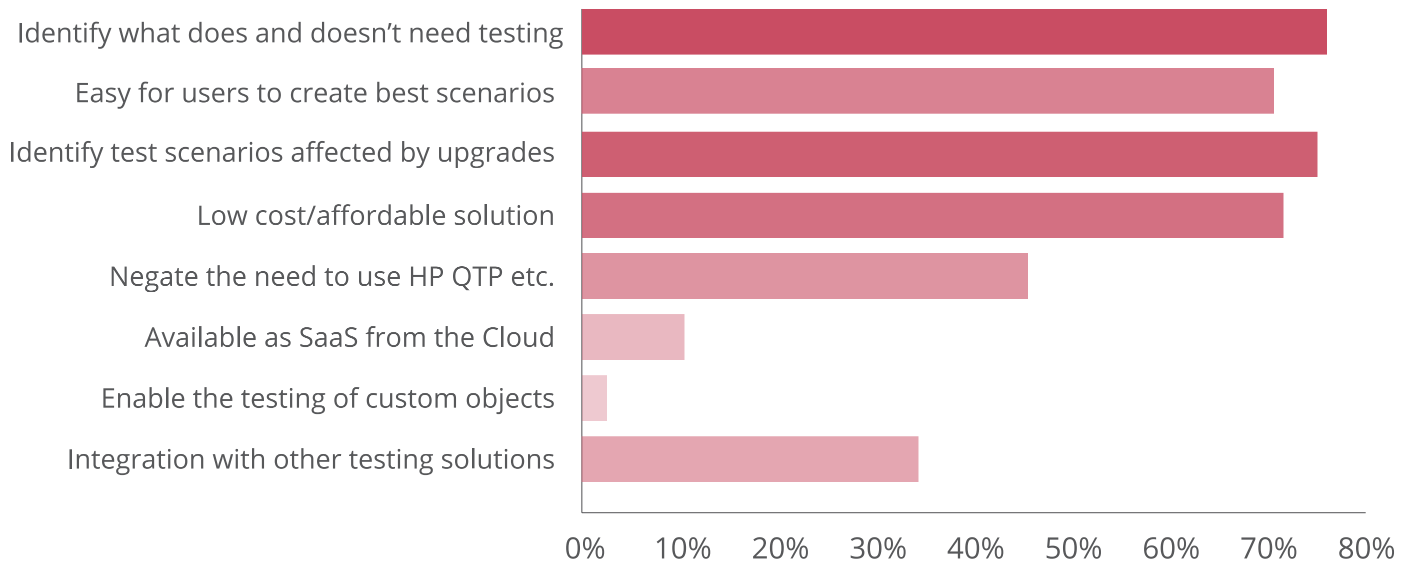
Code-current is the process of upgrading to the latest major release of JDE EnterpriseOne, then performing smaller, more frequent projects to keep pace with the vendor-released software updates.

Among those respondents that had previously used software test automation solutions, we were interested to know a little more about their individual experiences. In particular, we wanted to know about the challenges they had faced using the automation tools.



What problems have you experienced with test automation in the past?

Some of the negative experiences relating to testing solutions were surprising, given that test automation is designed to save time and make things simpler. So, we asked the same users what they would look for in the ideal test automation solution.



What do you look for in a test automation solution?

3. The Testing Challenge

When it comes to testing JDE EnterpriseOne implementations, modifications or upgrades, large or small, there are a number of key considerations and challenges, such as:

3.1. Test Planning and Management

- How to identify exactly what does and doesn't need testing
- How to establish how thoroughly each module or object affected by the change event should be tested
- How to allocate testing resource and monitor project progress

3.2. Test Execution

- How to adopt a more agile approach to testing than the traditional waterfall method
- Assessing the pros and cons of the different approaches to testing, such as record and play, scripting and scriptless

In addition, project managers need to understand what tools are available to help with effective test planning and management, and to reduce the overall burden of test execution.

4. Test Planning and Management

4.1. How do I know what needs testing?

The biggest dilemma facing your Quality Assurance team is how to identify precisely what needs testing. To do this you need to know exactly what has changed in the ESUs (Electronic Software Updates) you are installing and which objects are affected by those ESUs or other objects you may have modified. In addition, you will need to be able to identify all the dependencies within your EnterpriseOne system to truly understand the impact of those changes.

4.2. How thoroughly do I need to test?

Once you have identified the full impact of those changes, you need to assess by how much any object has been affected so you can determine how thoroughly you need to test each individual object.

For example, is it just a yes / no dialog box that has been repositioned or are there major changes to a whole business process?

Equally important, you need to be clear regarding what does not need testing, so you don't waste time, effort and money testing things that have not been impacted by the change event.

4.3 How do I allocate test resources and monitor project progress?

Any plan needs to consider the resources required. Every project is different, so the functional areas impacted, or the business processes affected, need to be considered.

When planning a project, the project manager needs to assign testing resources to ensure (i) that when the project is completed any changes are fully tested, and (ii) that the changes have not created problems elsewhere in the system.

Understanding the impact of a change event project (i.e. the scope) helps to ensure that you come up with a good plan. Then, over the course of the project, you need to ensure that the resources are being employed efficiently and making the progress expected.

5. Test Execution

5.1. How can I adopt a more agile approach?

The cost and effort required to identify bugs or other issues resulting from an upgrade or other change event typically escalate throughout the lifecycle of the project. However, there are several best-practices that can be adopted to facilitate a more agile testing process. A detailed analysis of your custom footprint will allow the project manager to create a test plan that is sequenced to fit with other requirements within the business.

This plan should be structured to allow for project "bundles" of development work to be created with full integrity. The bundles should be self-sufficient; allowing for them to be tested without having dependencies at a later stage in the project.

5.2. What are the pros and cons of the different approaches to testing?

Manual Testing

Research carried out DWS amongst EnterpriseOne customers indicated that the overwhelming majority still rely on manual testing. Understandably, a degree of manual testing will be required when testing the more obscure or complex workflows – including things a user might do out of curiosity or by mistake.

However, reliance upon manual testing has a number of drawbacks:

- Manual testing, by its nature, is very time-consuming
- Repeat testing requires the same amount of effort each time
- It is prone to human error, particularly when checking the results

With each iteration of testing, when bugs have been sent back to development to fix and then returned to the users to test, users tend to become less diligent. As a result, bugs will creep into production, where they are more expensive to fix.

Record and Playback

The Record and Playback approach uses tools that record the actions of a tester in a manual test, and then allows tests to be run unattended for many hours each day, greatly increasing test productivity and eliminating the tedious repetition of manual testing.

However, even small changes to the software under test require that the test be recorded manually again. Record and playback was the method used in test automation several decades ago. It has proven over the years to be neither efficient nor scalable and Return on Investment is rarely realised.

Scripting

Scripting, a form of programming in computer languages used in software test automation, alleviates many issues with the Record and Playback method. Often this is done by capturing functions/use-cases and then storing them as reusable test cases to be executed by a calling script.

However, the developers of these scripts must be highly technical and specialized programmers who work in isolation from those testers actual¬ly performing the tests.

Scripting is best suited for GUI testing and doesn't lend itself easily to embedded batch or other forms of systems.

As changes to the software under test require complex changes to the associated automation scripts, maintenance of ever-larger libraries of automation scripts becomes an overwhelming challenge. The Return on Investment using this approach is also difficult to achieve.

Scriptless

A scriptless approach to testing does away with the need for technical specialists to write or code scripts. Instead, a tool is used to enable non-technical end-users to generate test scenarios without the need for any programming knowledge.

This makes it far easier for end-users to become more involved in the testing of their business processes and far less time-consuming to rerun tests whenever necessary. With both the scripting and scriptless methods, test scenarios can be run as often as needed with minimal human intervention and accuracy can be maintained at a consistent and high level.

A scriptless testing solution, though, provides the most economic testing approach and the one that provides the best Return on Investment, as it is the easiest way to set up test scenarios without the need for expensive specialist resources.

A good scriptless testing solution should also make it very quick and easy for end-users to modify and edit test scenarios, without the need for any programming skills or knowledge.

6. What tools are available?

6.1. Test Planning and Management

Until January 2016 there was little available to help customers understand the impact Electronic Software Updates (ESUs) or software modifications would have on their EnterpriseOne system.

Of course, Oracle provides documentation to indicate which objects they have modified in ESUs and major version releases, and customers can set the modified flag when they make changes to the standard code, but this flag is often set incorrectly or, worse still, not set at all.

However, knowing which objects have been modified is only the starting point when planning your testing. You need to understand all the interconnections between these modified objects and other objects in the system if you are to focus your testing effort in the right areas and avoid unnecessary testing of code that is not affected by the changes.

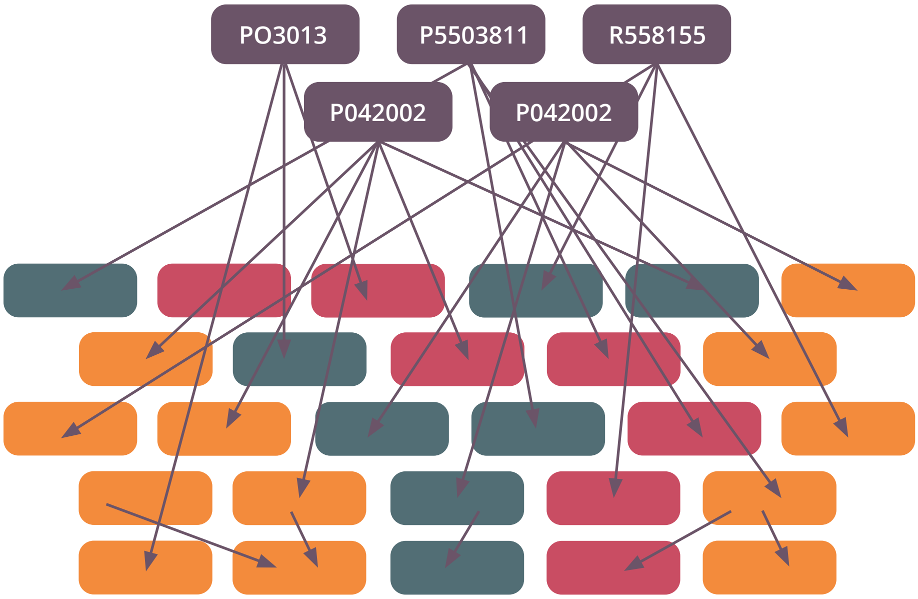
There is only one product commercially available at the current time that addresses the issues of test planning:

Dimension Focus™

The DWS Dimension Focus™ application performs three key functions.

- It identifies what needs to be tested
- It identifies how thoroughly objects need testing
- It identifies what does not need to be tested

It does so by investigating the programs that you use and how they would be affected by the ESUs; analysing the lower-level objects that are used by these programs, such as tables, business views and business functions. The interdependencies between these programs, tables, views and functions can be incredibly complex.



Dimension Focus™ also knows about the interdependencies (at the Event level) between these objects and can build up a multi-level hierarchy of how they are all interlinked. This cross-reference tree extends both up multiple levels above the ESU-affected objects and down to lower-level objects that may be called by several 'parent' objects and events.

Using this sophisticated analysis, Dimension Focus™ identifies those objects that would be directly changed as a result of installing the ESU or your own modified code into the EnterpriseOne system.

The number of objects affected by the change event could be as little as 5% of the total number in your system, leaving 95% unchanged, so the potential for saving test effort is significant.

Dimension Focus™ will analyse the true impact of the ESU or your own modified code and exclude from testing any objects that haven't changed or been affected. Having identified all the interdependencies in your system, Dimension Focus™ can also identify the top-level programs that are affected by changes in the lower level objects.

The result of all this is to potentially reduce the number of programs in your EnterpriseOne system that need to be retested by giving you the reassurance that they have not been impacted at the code level by any changes made either by Oracle or your own developers.

Regardless of which test management and execution method you use, the valuable information provided by Dimension Focus™ will help minimize the testing effort required and maximise your Return on Investment.

Advantages

- No need to install the ESU to prepare your test plan
- Analyze the impact of ESUs and modifications down to the event or function level
- Drill up and down through your EnterpriseOne system to identify all direct and indirect impacts from any ESUs or modifications
- Identify what needs testing and what does not
- Utilize a heat map and an impact score so you know where to focus your testing effort
- Significantly reduce the amount of testing effort required

6.2. Test Automation

OATS

Oracle Application Testing Suite is a comprehensive testing solution that ensures the quality, scalability and availability of your Web applications, Web Services, packaged Oracle Applications and Oracle databases.

This integrated, full-lifecycle solution enables you to define and manage your application testing process, validate application functionality and ensure that your applications will perform under load. With Application Testing Suite, you can deploy your Web applications and Web Services in less time while maximizing the efficiency of your testing team.

Application Testing Suite includes the following tightly integrated products:

- Oracle Load Testing for scalability, performance and load testing
- Oracle Functional Testing for automated functional and regression testing
- Oracle Test Manager for test process management, including test requirements management, test management, test execution and defect tracking

Oracle Application Testing Suite also provides a series of integrated testing accelerators for testing Oracle packaged applications and SOA (Service-Oriented Architecture) applications. These accelerators enable enhanced scripting capabilities for more efficient and optimized testing.

Advantages

- Reduced development time for script creation
- Improved security of data during testing with built-in encryption
- Integrated scripting platform for automated functional testing and load testing
- Powerful Java-based code view built on Eclipse IDE to extend scripts
- Enables automatic generation of load test scripts from Real User Experience Insight
- The testing accelerators for JD Edwards EnterpriseOne provide support for functional test automation of EnterpriseOne and the JD Edwards Data Grid component

Disadvantages

- The tool supports only Web-based applications

HP Unified Functional Testing (UFT) – formerly known as Quick Test Pro

Advantages

- UFT is a simple tool for even a non-programmer to understand and start adding test cases
- The tool has an excellent Object Identification process / mechanism
- Supports different add-ins like Java, Oracle, SAP, .NET, Web Forms, Siebel, PeopleSoft, Web services, Mainframe (Terminal Emulator) etc
- Well integrated with test management tools like QC
- The tool has many inbuilt functionalities by default
- Supports both Windows and web-based applications

Disadvantages

- High licensing and add-on costs makes it prohibitive for a large percentage of the JD Edwards user community
- The tool runs only in Windows environments
- Cannot test with all browser types and versions
- Slow in execution when compared with open source tools

Selenium IDE & Selenium Grid

Advantages

- Selenium is a Free Open Source tool and is good for Web GUI Testing
- The Selenium IDE is a Firefox plug-in which is used to record and replay tests in the Firefox browser
- The tool uses an HTML sort of language called Selenese. Though other languages (Java, C+, PHP etc.) cannot be used with Selenium IDE, Selenium IDE lets you convert tests in these languages so that they can be used with Selenium 1.0 or Selenium 2.0
- Supports Windows PC/MAC/UNIX Platforms
- Selenium Grid allows you to run your tests in parallel, that is, different tests can be run at the same time on different remote machines

Disadvantages

- The tool supports only browser based application, not the windows application
- Selenium IDE does not provide iteration or conditional statements for test scripts
- Does not support file uploads from local machines
- Selenium IDE can be used only with Firefox browser
- Offers only partial support for dialog boxes
- Selenium has no official technical support
- The tool does not provide test reporting and error recovery mechanisms

Oracle User Productivity Kit (UPK)

Oracle User Productivity Kit (UPK) and Oracle User Productivity Kit (UPK) Professional are collaborative content development platforms that support all phases of enterprise application implementation from planning through to testing, go-live, maintenance and support.

Traditionally UPK has been used as a documentation and user training tool. In 2016 Oracle introduced the "Test It!" mode to extend its usefulness to the testing arena. Users can now use UPK features for test case creation and execution.

Advantages

- UPK provides a consistent process for gathering test plans, execution and results analysis
- Provides in-application test support for non-professional testers performing user acceptance testing
- Publishes manual test plans for Oracle Application Testing Suite, HP Quality Center and IBM Rational
- Reduces the man time required to create test cases

Disadvantages

- UPK is a record only tool. Once a test case is recorded it cannot be edited. If anything requires changing the whole sequence must be re-recorded in its entirety
- There is no playback or automation of testing. All testing is still a manual process, so is time-consuming; especially if there are several cycles of testing and bug fixing. The user also must record whether a test passed or failed

DWS Dimension SwifTest™

DWS Dimension SwifTest is the only testing solution designed specifically for, and integrated with, JD Edwards EnterpriseOne. It has been designed with the non-technical end user in mind to make the creation, editing, scheduling and execution of testing as simple and efficient as possible.

Advantages

- Native integration with JD Edwards EnterpriseOne to simplify test scenario building and editing
- Easy for non-technical end-users to build and edit test scenarios
- No specialist or technical knowledge required
- No programming or script-writing required
- Uses unique scanning methodology to automate the generation of test scripts
- Can be used on custom code as well as ESUs
- No need for laborious record and playback test set-up
- As test requirements or the application changes, editing the test scenarios is quick and easy
- Multiple test scripts can be submitted into a queue for automated testing
- Pinpoints all test failures with user-friendly information regarding the reasons for failure
- Pay-as-you-save subscription model delivers a rapid and measurable ROI
- Integrates with the DWS Dimension Focus™ planning tool for maximum ROI

Disadvantages

- Only works with JD Edwards EnterpriseOne

7. Oracle Validated Integration

Oracle Validated Integration, available through the Oracle Partner Network (OPN), gives customers confidence that the integration of complementary partner software products with Oracle Applications and specific Oracle Fusion Middleware solutions have been validated, and the products work together as designed.

This can help customers reduce risk, improve system implementation cycles, and provide for smoother upgrades and simpler maintenance. Oracle Validated Integration applies a rigorous technical process to review partner integrations.

The DWS Dimension Focus™ and Dimension SwifTest™ applications are part of the DWS Dimension Hub™ which in April 2016 achieved Oracle Validated Integration of Dimension Hub™ 1.0 with Oracle's JD Edwards EnterpriseOne 9.2.

8. ROI Illustrations

The Return on Investment estimates below should be used as a guide only, as actual savings will depend on many factors; such as the number of ESUs installed, the frequency of ESU installations, the number and frequency of custom code promotions to the production environment and the number of objects actually impacted by ESUs or custom code modifications.

Other important factors include the number of days per year IT staff or end-users spend testing, their internal daily charge rate and whether you outsource your testing to one of a number of Oracle partners who can offer testing as a service. The price paid for your preferred test management and automation solution will also play an important factor.

The following ROI estimates are based on real customer profiles. Their usage scenarios and testing requirements are based on either a major upgrade or code current event taking place each year with an element of user customization also taking place.

The ROI calculations are based on using the DWS Dimension Focus™ application for test planning and DWS Dimension SwifTest™ for test management and execution. Other testing solutions will yield different ROIs.

8.1. ROI – Customer Profile 1

Number of JD Edwards EnterpriseOne users	220
Number of testers	3

	Year 1	Year 2	Year 3	Total
Scenario	Major Upgrade	Code Current	Code Current	
ROI (Savings minus cost of software subscriptions)	$23,310	$14,700	$14,700	$52,710
ROI %	212%	171%	171%	

8.2. ROI – Customer Profile 2

Number of JD Edwards EnterpriseOne users	1,100
Number of testers	20

	Year 1	Year 2	Year 3	Total
Scenario	Major Upgrade	Code Current	Code Current	
ROI (Savings minus cost of software subscriptions)	$361,000	$172,840	$43,240	$577,080
ROI %	352%	221%	130%	

8.3. ROI – Customer Profile 3

Number of JD Edwards EnterpriseOne users	2,500
Number of testers	44 (including 5 contractors)

	Year 1	Year 2	Year 3	Total
Scenario	Major Upgrade	Code Current	Code Current	
ROI (Savings minus cost of software subscriptions)	$1,100,000	$650,000	$90,000	$1,840,000
ROI %	490%	327%	130%	

9. Conclusion

Earlier generations of testing solutions required a reliance on expensive and scarce specialist technical resources. These early tools rarely yielded a measurable Return on Investment.

Recent research indicates that a significant majority of JD Edwards EnterpriseOne customers rely on manual testing. Yet, the latest tools and technologies available from Oracle and DWS would suggest that customers could make substantial savings in both time and money by investing in suitable test planning and test automation applications.

Many JD Edwards customers will be planning to upgrade to the latest major release of EnterpriseOne. The latest testing tools from DWS overcome the objections many customers had towards implementing a code current policy and they have been validated by Oracle to integrate with JD Edwards EnterpriseOne.

So, now would seem to be an excellent time to investigate the testing options available, as they could yield rapid and quantifiable ROI, whether you are planning a major upgrade, planning a simpler code current event or just making your own modifications and code customizations to your EnterpriseOne system.

> **“Using Dimension SwifTest™ allowed us to significantly reduce the volume of back-breaking testing we would have had to do. We also saw improvements during user acceptance”**
>
> CRM Applications Manager, Acal plc

About DWS

DWS Consultants is a leading provider of Oracle JD Edwards EnterpriseOne software services.

Since 1998, we have been providing specialist analytics, development and testing services to organisations looking to customise, upgrade or support their JD Edwards deployment.

DWS supports a global client base with bespoke module development, CNC, testing and software upgrade services; using our proprietary Dimension™ Suite of tools.

For further information please visit our website, or contact us:

UK: +44 (0) 1494 896 600 US: +1 888 769 3248 ANZ: +64 9427 9956
sales@dws-global.com **www.dws-global.com**

ORACLE | Partner