

10 tips for successfully executing a JD Edwards upgrade

Lessons learned from delivering over 100 JDE upgrades

“I’ve been involved with more than 100 JD Edwards upgrades, so this experience gives me the opportunity to highlight some lessons that I’ve learned along the way. Some of these are super simple and some involve a little more planning and execution (and sometimes some external help).”

SHANNON MOIR, JDE INNOVATION SPECIALIST

By Shannon Moir Director - Digital Innovation | Fusion5

Shannon is Director Digital Innovation for Australia and New Zealand. He predominantly works across five main areas: Internet of Things (IoT), social media, cloud, enterprise mobility and big data.

Shannon is recognised as an industry leader in enterprise mobility, cloud and analytics for Enterprise Resource Planning (ERP). He’s also a JD Edwards expert, with over 20 years’ experience in building and designing JDE sites. He holds a certification in Amazon Web Services architecture, which underpins many of his modern design techniques, especially when it comes to integration and ERP.

Passionate about sharing knowledge, Shannon is an avid blogger. He enjoys troubleshooting problems and recording his findings on his blog. He strives towards standardised, cost-effective innovation to help businesses make IT work for them, not the other way around.



shannonscncjdeblog.blogspot.com



ESTABLISH A PROJECT SCOPE AND STICK TO IT!

There are generally a lot of choices for an upgrade, and they are only getting tougher. Do I change database, do I change platforms? Should I move to the cloud, do I need a managed service? Yes – lots of questions, but they really should not get in the way of the mechanics of what is being done. You'll still need to do the same amount of testing, and let's be honest, this is generally the critical path on any upgrade project.

Form a clear idea of what you want to achieve for the project. Scope the technical decisions early. Don't be afraid of a couple of architectural changes in addition to the upgrade itself - refer back to my observation on testing [critical path].



Figure 1: There are so many architectural options when considering an upgrade.

Use the testing and the project to achieve some strategic goals. Nearly every upgrade that I'm involved in has more than a single dimension. More often than not, we are performing upgrades at the same time as cloud migrations. As I alluded to, this is a good combination and a tight scope. If I was to take on an upgrade and cloud migration, I'd try and not change much else in that step.

Changing too much can introduce too much risk. For instance, changing your output management software [createform] can be very onerous and take a lot of time. If you think of all the time that you have spent on making your invoice print perfectly, every time... Any change in this area is going to take a lot of time! You can do that iteratively when you get an opportunity. Output management solutions can be run in parallel and give you a perfect fallback position.

Be tight on scope and agree early – what is in and what is out.



UNDERSTAND AND CONTINUOUSLY MEASURE YOUR TESTING

It's critical to understand the scope of testing. Whether you are doing test automation, test outsourcing or manual testing, you need to know the programs and versions that your users are currently using. This is going to allow for accurate test scenarios. Knowing your programs (both interactive and batch) will allow you to also choose candidates for automation (if that is what you are going to do).

A high level idea of your processes (whether documented or reverse engineered by clever e1 pages) will enable you to group your test cases and test resources. Quite often a logical grouping of your testing can ensure that end to end processes are tested and that the test data is going through a full cycle too.

Get a good list of programs and how often they are used. Get a good list of users from production and what programs they are running. Use these two project assets to then choose a test team and also ensure test case saturation in your test environments.

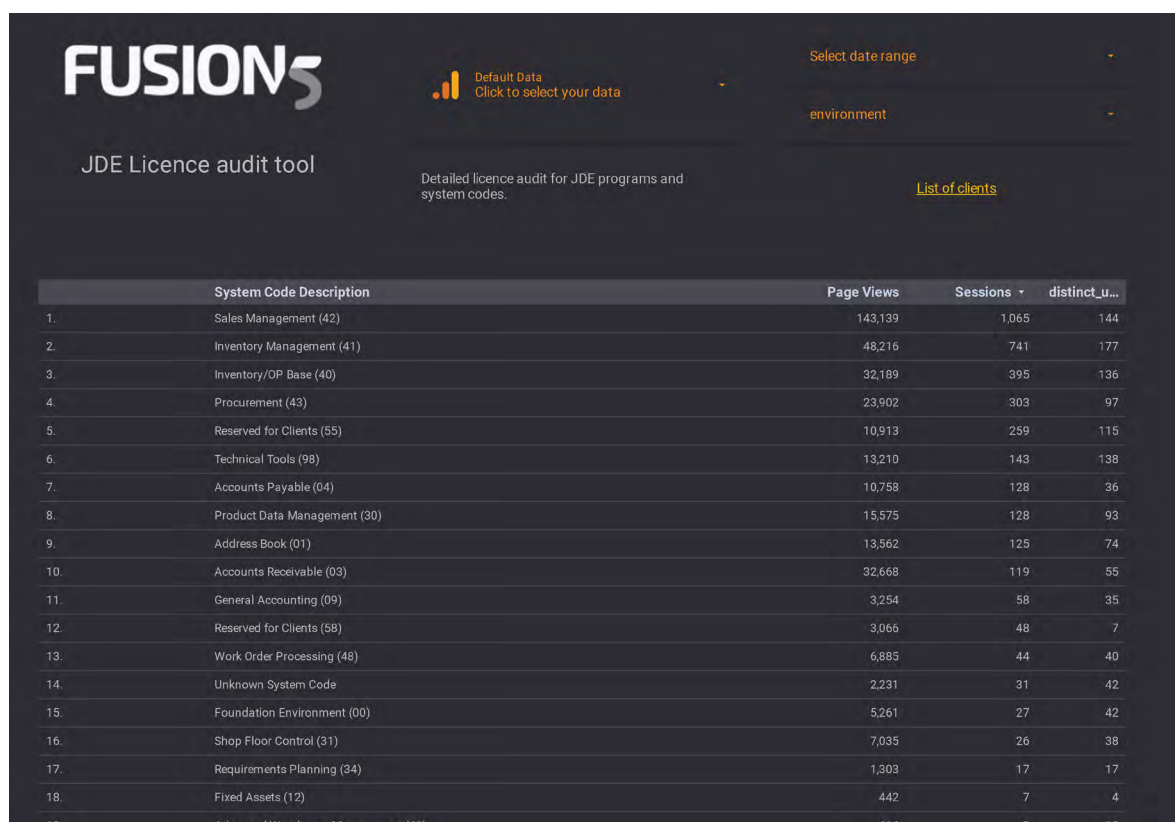


Figure 2: Using the Fusion5 ERP Analytics suite, you can quickly see programs and how much their usage is, using a date range and environment.

At Fusion5 we advocate the use of ERP Analytics to give you “easy to use” reporting on all user activity in JD Edwards. This software subscription can plug into your current production environment and your upgraded environment and provide constant feedback about test volumes and test case saturation – for both batch and interactive.



UNDERSTAND THE TOTAL IMPACTS AND COSTS OF CHANGE

Going from SQL Server to Oracle means that you will have case sensitive queries, when the users previously did not need to worry about cAsE. This is unavoidable and you'll need to educate the user community on this, so you'll need to follow change management processes and include lots of comms.

Going from Windows to Linux might mean changes to processing options for the location of files or interoperability. There are options for bulk Processing Option (PO) changes using the database. This is very important for finding a “\” for example and turning that into a “/” or vice-versa. This simple trick of interrogating the BLOB in the F983051 can change a very manual and error prone process to an exact science.

Media Object changes are critical and need to be understood as you upgrade JD Edwards. There are more storage options that you need to be aware of, some for good and some for your own peril. I say peril, the storage costs alone are similar (see below), but the database IOPS is something that you need to focus on VERY carefully for a cloud implementation – as this is what generally governs your service limits.

For example, let's look at this from a pure cloud cost perspective. In AWS 100GB of s3 is going to cost you 0.023 dollars per GB (\$2.50 a month). EFS (Elastic File System), a highly available storage format perfect for media objects, is going to cost you about 0.3 dollars per GB per Month (\$36 a month). Let's put that into your highly available database instance (multiple availability zone) and that is going to be 27.60 per month. Remember that you don't really backup or restore the EFS, so you are only paying for a single copy (you might snap it to S3).

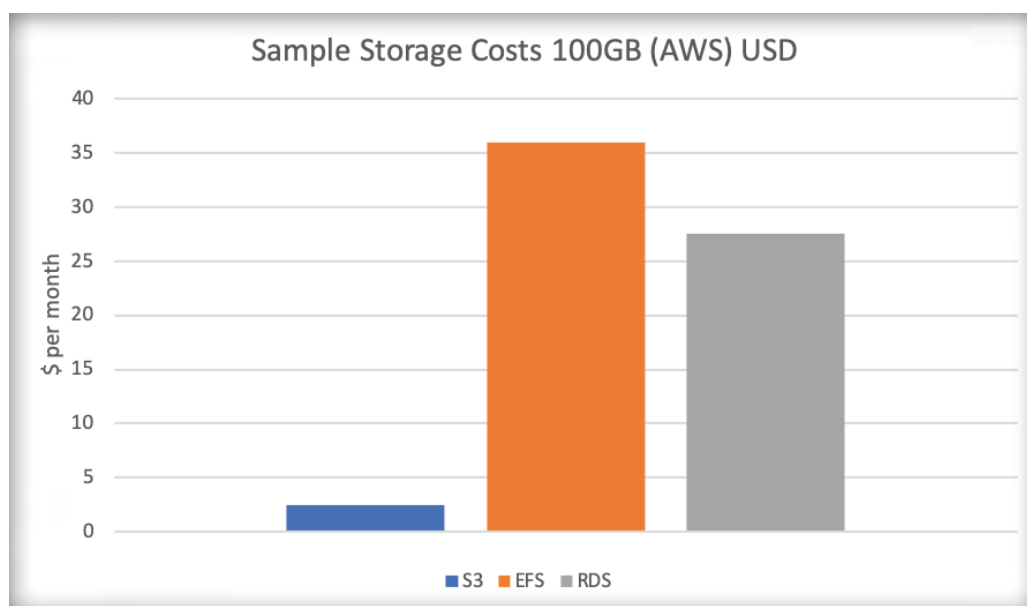


Figure 3: Comparative storage costs for media objects - imagine 100GB and the costs are monthly.

Look at your end game infrastructure and make sure that you understand both the change management impacts and the cost impacts of the new architecture.



GET YOUR BROWSER READY

Your browser is basically equivalent to your operating system in terms of compliance and importance to JD Edwards. You need to get it right. There are a number of important compatibility settings, security settings, proxy exceptions (and more) that you need to ensure are pushed out to the business as part of an implementation.

In general, URLs no longer change with an upgrade or migration of JD Edwards. A production URL (jde.fusion5.com.au) for example, is probably well known and has all sorts of favourites saved on all sorts of machines. Don't change it – it's painful. You'll have more calls into your helpdesk saying "JDE is broken" that you can poke a stick at. Please keep the URL the same and you'll have a better chance of everyone being able to login.

Ensure that you push out cache refreshes for any tools release change or any JDE change for that matter. It's critical to manage all of your browsers too, not just the ones that you think are being used. How do you know what browsers you need to cater for? Use ERP Analytics of course. This gives you detailed mapping of users and programs to browsers. It'll also allow you to ensure that you are using activeX (please don't keep relying on this) and what settings your CNC team need to put into the JAS.INI to ensure that all browsers are treated equally (well, as equally as possible). Supporting a broad base of browsers and technologies is always going to be best.

Browser performance constantly surprises me, as the two screen grabs below will attest to. We have two different clients that use JDE heavily, you'll see opposite results in terms of the performance. It really does make a HUGE difference though – 80% difference in site 1 and 30% difference in site 2 – purely based on browser choice.

Browser ?	Avg. Page Load Time (sec) ?	Pageviews ?
	1.14 Avg for View: 1.14 (0.00%)	406,396 % of Total: 100.00% (406,396)
1. Internet Explorer	1.05	322,909 (79.46%)
2. Chrome	1.85	47,050 (11.58%)
3. Firefox	1.06	28,102 (6.91%)
4. Edge	1.23	8,314 (2.05%)
5. Safari	0.00	21 (0.01%)

Figure 4: Internet Explorer is significantly faster than Chrome, and it is clearly the browser of choice.

Browser ?	Avg. Page Load Time (sec) ?	Pageviews ?
	1.17 Avg for View: 1.17 (0.00%)	558,257 % of Total: 100.00% (558,257)
1. Internet Explorer	1.28	312,968 (56.06%)
2. Chrome	0.99	206,768 (37.04%)
3. Edge	1.29	38,257 (6.85%)
4. Firefox	1.65	215 (0.04%)
5. Safari	1.55	49 (0.01%)

Figure 5: A different client sees Internet Explorer as the most popular, but 28% slower than Chrome.



LOAD TEST YOUR BATCH ACTIVITY

Batch activity is really a pure performance comparison which takes away a potential tier in your traditional 3 tier architecture using JD Edwards (Web, App & DB). The nice thing about this also is that you are really only testing your batch server and your database server.

JD Edwards (in my mind) submits two types of jobs

1. Type 1 runs a series of large SQL statements. These are generally not complex, as the batch engine's capacity to run complex statements (even simple aggregates) is not good. Therefore you are going to get large open selects, which will generally then perform subsequent actions based upon each row that is returned in the main loop. (eg. R09705 - Compare Account Balances to Transactions).
2. Punchy UBE that gets in with some tight data selection, generally runs a pile of BSFNs and then jumps out again. (R42565 - Invoice Print).

It's easy to categorise these jobs because of the amazing job Oracle did with "Execution Detail", specifically rows processed.

Host Name	JDEENTPT	Audit Report	R01BDWRD
Job Number	4236	Audit Version	SRM0002
Environment Name	JPD920PT		
Process ID	19855		
Start Date and Time	07/05/2019	08:59:51	Job Process Time
End Date and Time	07/05/2019	10:18:48	Rows Processed
			1:18:57
			1

Figure 6: View taken from "Execution Detail" row exit from Work with Submitted Jobs (WSJ).

You can actually databrowse this (V986114A) and see column Alias: PWPRCD, defined as "The number of rows processed by main driver section of the batch job". I use this in a lot of SQL around performance, as I can get rows per second for my UBEs - which is a great comparison device. If you see consistent low numbers here, probably a punchy UBE - lots of rows, probably category 1.

Make sure that you test UBEs in both of the categories that I have listed above. Some are going to test the database more, some are going to test the CPU on the batch server and some are going to test network I/O. Make sure that you "tweak" your TCP/IP too, as I have seen this make some impressive differences in batch performance. (search Doc ID 1633930.1 and tweak).

The Fusion5 UBE Analytics suite allows you to do this comparison immediately and gives you some impressive power to compare periods, servers and more.

FUSION5

BATCH_SERVER - ENVIRONMENT - Aug 11, 2019 - Aug 17, 2019 -

WATSACOWIE BREWING COMPANY

ROWS_PROCESSED - RUNTIME_SECONDS -

UBE_NAME	JCVERS	Reports run	% Δ	AVG ROWS	% Δ	AVG RUNTIME...	% Δ
R47022	XJDE0001	17,062	-22.6% ↓	1.68	8.5% ↑	3.21	0.6% ↓
R47042	XJDE0001	4,677	-20.4% ↓	5.45	23.2% ↑	29.49	212.7% ↑
R46473	RBAS0020	4,658	-23.0% ↓	0.99	81.2% ↑	0.81	-8.9% ↓
R46473	RBAS0030	4,658	-23.0% ↓	0.19	-17.7% ↓	0.83	-4.5% ↓
R46473	RBAS0040	4,658	-23.0% ↓	0.24	30.0% ↑	0.82	-6.8% ↓
R46L140	RBAS0010	4,658	-22.9% ↓	205.36	-32.1% ↓	1.08	-16.2% ↓
R564339A	RBES0010	4,658	-22.9% ↓	0	-	0.45	-10.4% ↓
R46L140	RBAS0020	4,658	-23.0% ↓	1,434.49	0.2% ↑	1.62	-4.4% ↓
R55ED107	RBVS0020	4,658	-22.9% ↓	0.54	25.9% ↑	8.55	-1.1% ↓
R564339A	RBAS0010	4,658	-22.9% ↓	0.55	59.3% ↑	0.49	-11.2% ↓
R564339A	RBVS0010	4,658	-23.0% ↓	0.04	-5.8% ↓	0.49	-1.2% ↓
R4277703	RBV0010	3,381	-25.3% ↓	10.32	-2.9% ↓	1.5	1.4% ↑
R42750	RBV0010	2,587	-22.8% ↓	13.08	0.0% ↑	5.88	-8.8% ↓
R42565	RBVS300	2,330	-22.9% ↓	0.36	100.1% ↑	7.34	-0.2% ↓
R42565	RBVS360	2,330	-22.9% ↓	0	-	4.2	-1.3% ↓
R4982	RBVS010	2,330	-22.9% ↓	479.91	0.1% ↑	16.95	14.2% ↑
R42565	RBSE400	2,328	-23.0% ↓	2.73	23.7% ↑	7.71	1.0% ↑
R42565	RBVS305	2,328	-23.0% ↓	2.17	4.2% ↑	9.12	0.6% ↓
R42565	RBVS350	2,328	-22.9% ↓	0.16	-4.5% ↓	4.11	-2.6% ↓
R56B65Z1	TUS0001	2,328	-23.0% ↓	0	-	0.63	-3.7% ↓
R42565	RBVS340	2,325	-23.0% ↓	0.15	15.5% ↑	4.13	0.0% ↓
R47011	RBEI220	1,691	-24.4% ↓	1	0.0% ↑	27.23	2.1% ↑
R46171	RBA0010	1,591	-22.8% ↓	0.85	32.5% ↑	1.18	6.7% ↑

Figure 7: UBE Analytics summary screen - week on week performance comparison.

We can choose a date range for compare and let the system do the rest. You can see that we can tell for each UBE and version combination that has been run for this fictional client in the date range specified, if you compare it with the previous period – performance has slowed down in the top 12 rows. I'd be looking at what has changed!

The UBE Analytics data is not stored in JD Edwards, so you never lose your history.



LOAD TEST YOUR INTERACTIVE ACTIVITY

You've got my full attention in this section, I really enjoy load testing. Whether this is using OATS or other software, it's a bit of a passion of mine.

Here is my recipe for success for interactive load testing. You have your batch results above, so you are pretty confident with database size and hopefully application (logic layer). We are now going to test the interactive performance of JD Edwards and how the user is going to experience things.

The first question you need to be honest about, is the peak capacity of users that you are going to test. If Server Manager tells you 150 users are on, how many people would you load test with? I can tell you – A LOT LESS! I would test 40 in that scenario with a wait time of 5 – 8 seconds.

Let me show you why: ...

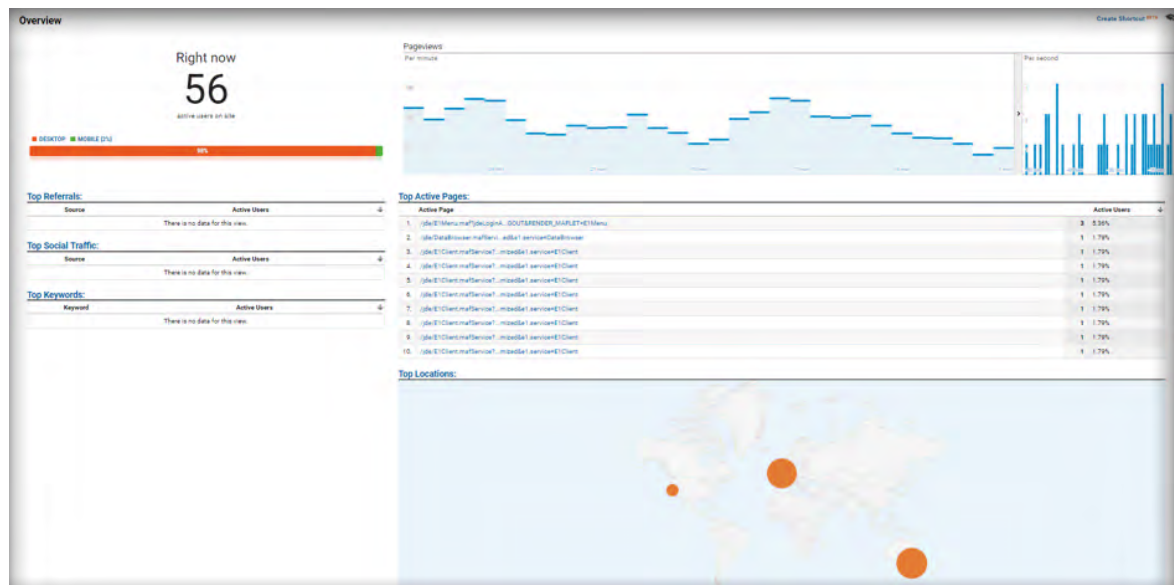


Figure 8: Standard ERP Analytics screen showing current activity, both throughput and location.

My interactive report says there are 56 users logged into JDE and active in the last 5 minutes. This is an interactive dashboard that Fusion5 ERP Analytics customers have access to. You can also see the pages per minute and pages per second. We are peaking at about 150 a minute in that snapshot, but I can find the peaks over the last 2 months if needed.

State	User Activity
Undetermined	None.
Running	0
Undetermined	None.
Running	1
Running	113
Running	139
Running	14
Running	11
Running	5
Running	5

Figure 9: Server Manager's view of the connected world is generally artificially high.

Yet Server Manager is trying to tell me that I have 288 users. Even with my classic double up the AIS users – we have 144 logged in, but only 58 active in the last minute.

What I'm trying to say here is don't stress your system with too many users. Tuning for 3 x the worst scenario possible is actually going to slow you down.

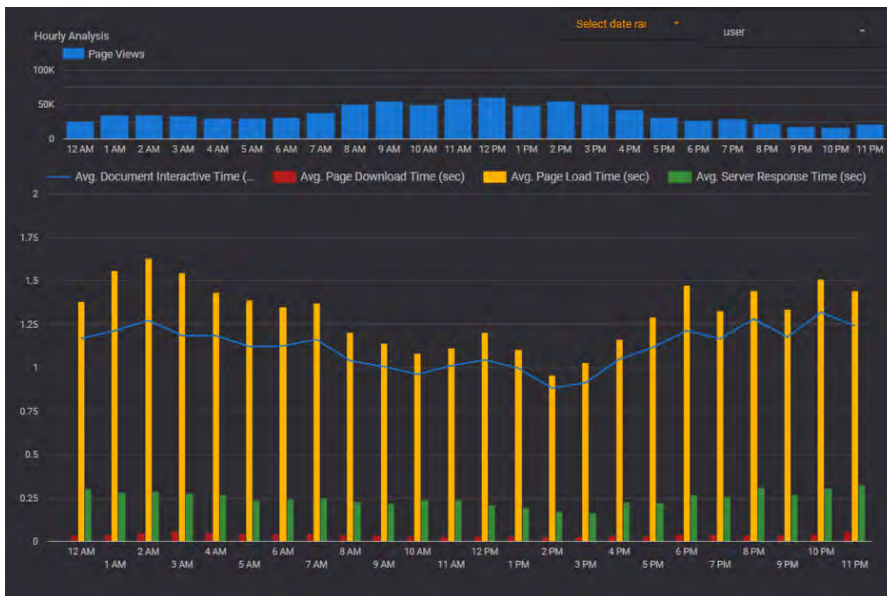


Figure 10: ERP Analytics screen showing time of day, page views and performance.
Reiterating the fact that a busy server is a fast server!

The graph is unequivocal in showing that performance is better when pages are busy. Do not have too many idle web servers because you have catered for 3x the users – your users are actually going to experience worse performance. This is MORE dramatic at the users drop off. I see around 20% performance improvement when a JAS server is loaded and cached up nicely.

Now that you can determine the number of users you need for load testing, you can execute this with the software or services that you have access to. At Fusion5 we use OATS and can assist with any load testing you need. We also validate and continually measure interactive performance using ERP Analytics, which can produce all of the graphs that you see above.

Anecdotally, good performance from JD Edwards is when pages load in about 1.1 seconds.



Figure 11: Another view of performance over time, but separating download time and server response time. The page load is generally in direct correlation to server response time.

We measure and record exactly what the end user experiences. We can also report on the network traverse time and the server response time. These are all critical values when determining what you need to fix. We can run this reporting on different users or geographies too, so you can compare performance in a single city or around the world.



TEST EVERYTHING WHEN YOU ARE DOING “MOCK” GO-LIVES

As I said, if I'm at a go-live, it's not my first rodeo. Sure, I'll always have a nervous demeanour and perhaps a bit of sick feeling in my stomach, but I do love it. I like seeing all of the data and statistics around me that somewhat affirm the planning and efforts that have gone into the project. It's simple, when things go to plan.

Of course when a user does an open find over the F4211 and F42119 using a single non-indexed field and then wants to go to the end of the grid... with probably 20 million rows to be displayed... I might not have tested that (nor catered for it in the sizing). Oh, and when it doesn't return and they do it another 10 times (to be sure), that also was not in our test plan. Nonetheless – there will always be challenges and things unexpected – your job is to reduce the number of them.

Mock go-lives are critical. They do the following important tasks:

- Assign responsibilities to all tasks, both prior to, during and after the upgrade. Ensure that these are on the run sheet and have all been tested before.
- Version your run sheet, make sure all line items are filled out and ensure that there are accurate timings. You will not get good timings on the first conversion, and perhaps not the second. Subsequent to that you should be building out exactly how long the conversion is going to take so that you can determine if you need to “look outside the square” when it comes to outage windows.
- Make sure that people run integrity reports and check the results every time. I've been involved in go-lives where an integrity did not match on the go-live weekend – but guess what? It never balanced in the last 5 mock go-lives – it was not compared. Getting everyone to run every step is a big lesson.
- I only really care about rowcounts, but I know that the business will want integrity reports – so you might want a few. Summing amount columns or hashing is another way to make technical people really happy.
- Ensure that you move some WSJ history too. Nothing worse than a user logging in and not seeing the report they ran on the Friday before go-live weekend. Anything you can do to reduce the call volume on the Monday after go-live – you should do it!
- Timing, if things are too fast you probably have a problem. If things are too slow, you probably have a problem. Make sure that things are predictable.
- Sleep is important, people do not make good decisions under lots of pressure and with a lack of sleep. Go-lives are tough and should be, but not at the expense of the team. Don't let the team drive if they've worked 20 hours, get a local hotel and an UBER. Plenty of food and drinks for the project team too.

Get a runsheet, live by the runsheet and succeed with the runsheet.
Regular comms are critical – good luck!



CHECK YOUR SECURITY

Security sometimes takes a back seat, but please, please – don't let it. Without exaggeration it's about 1,000,000 times easier to make security changes before a go-live than after.

Simple things need to be completed and tested before go-live:

- Complete production security model in place, including row security.
- All table auditing enabled (if you do this).
- Complex JDE password for database and for JDE.
- Do not use the JDE account for the system account for your users. Please do not do this, create "jdeuser" or something much better that will not get locked out.
- Check that the password expiry policy for Oracle is not default, or else your system accounts will start locking out.
- Change your default node configuration, do NOT leave this standard. This is one of the largest security holes in the suite.
- LDAP or SSO is critical. Controlling user access is easiest if someone else is doing it (I find). So if the desktop team is decommissioning users (oh and changing passwords) this is a big bonus and will save time and money. The Fusion5 SSO offering is really cool too, especially if you want to use Azure to MFA people under certain criteria – all done by someone else!
- Make sure that your data is encrypted at rest and in transit.
- Get your security groups tight and your firewalls enabled.
- Default access should be no access.
- Adopt the most stringent security posture your business can afford.

Here is an interesting tip, quite often row security can be good for performance.

Why? Because it ensures that there is a where clause that is generally an indexed field. If you are row securing MCU or CO, then the where clause is enforcing less IO and hopefully a quicker result!



MONITOR RELENTLESSLY

Tip 9 and 10 are closely related, but this tip is all about feedback. If you know things are going well it's great. If you know things are going poorly, that is great too – because you can fix it. The worst case scenario is that things are going pear-shaped in the background and you only hear about it when a user raises a case. You need to be KPI'd on finding errors before your users – period.

How can you find errors before your users? Here are a couple of tricks that Fusion5 implements for our clients:

ERP Analytics Sometimes referred to as the black box for your ERP, we use this to monitor performance and usage of JD Edwards. It records every page load, by every user – every minute of every day. This information is incredibly powerful for benchmarking and comparing ANY change in your entire enterprise.

UBE Analytics Having access to the runtime, rows processed, user and server information for every batch job allows us to continually monitor the critical two tiers of JD Edwards. Reading this with ERP Analytics gives more information on where performance problems might be and another point of data to compare with.

Log monitoring Fusion5 has a very advanced cloud formation in AWS which utilises cloudwatch to monitor all log files, UBEs and JD Edwards connections. This enables us to graph and monitor user connections, concurrent UBEs and search ANY logfile for ANY error – ever. This is a single console for all logs across JD Edwards. This approach and consistency can be used with many different installation types, not just limited to AWS.

DB growth monitoring Keeping an eye on table by table database growth is critical for understanding if a process has gone rogue. It's also critical for maintaining consistent database performance. Regular rowcount reporting and size reporting will ensure that you can deliver a service level to your users that is acceptable. Maintenance of your data size is important for costs and restoration times.

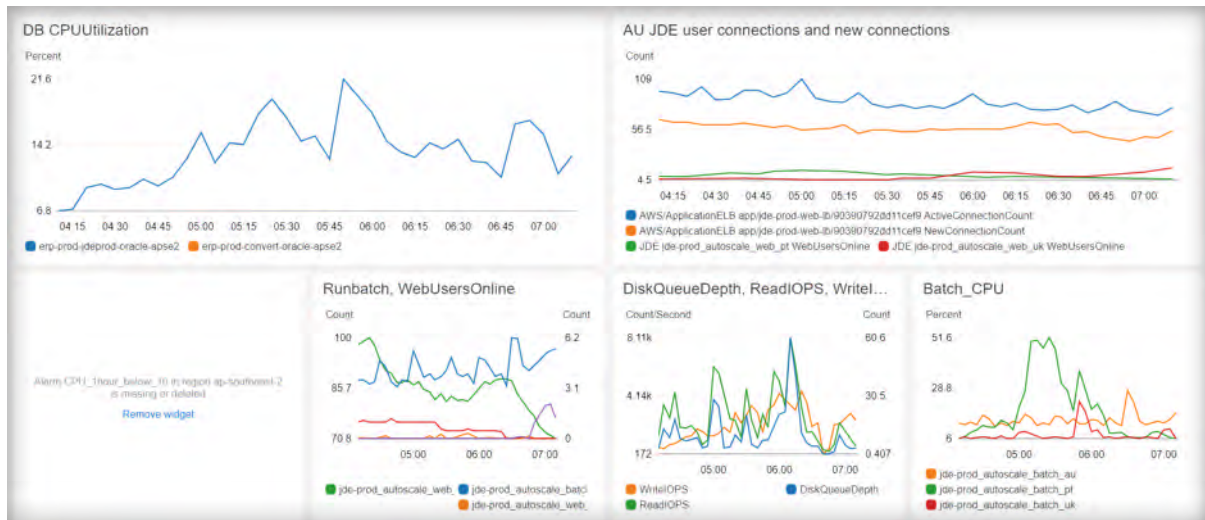


Figure 12: Sample custom dashboard showing metrics that are 100% relevant for JD Edwards.

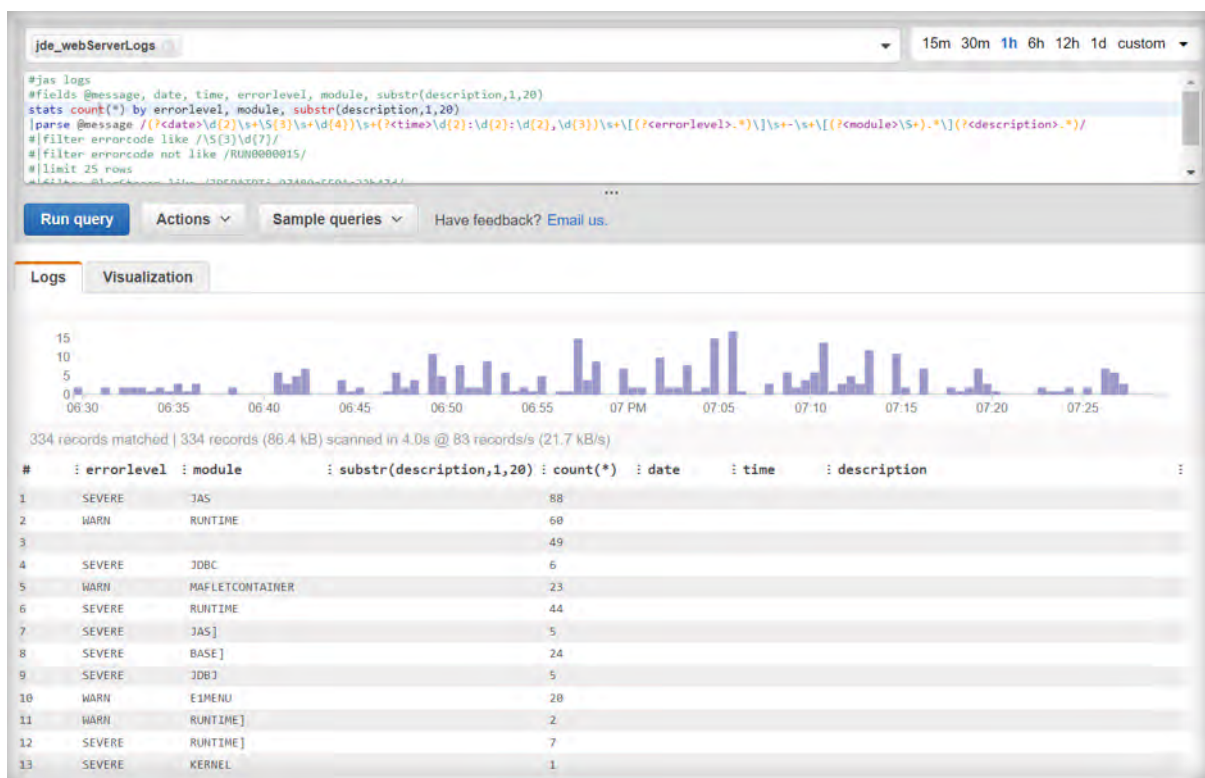


Figure 13: AWS log insights provides intelligence that would previously be impossible to find. This shows a graphical representation of errors and type of errors over 8 separate web servers.



ADOPT A CONTINUOUS IMPROVEMENT ATTITUDE

It's a theme that is not going to go away. If you have implemented all of the shiny new toys from JD Edwards, then you need to show ROI.

Even the way Oracle is releasing JD Edwards functionality follows the ideals of continuous delivery by committing to release 9.2 until 2030. We are getting improvements continuously, not in major releases.

I think there are 3 ways that you can make a difference to your business in relation to adopting continuous improvements with JD Edwards.



FINDING AND IMPLEMENTING RPA (ROBOTIC PROCESS AUTOMATION) OPPORTUNITIES

There is so much opportunity here, and all of the tools are at your fingertips. You can use ERP Analytics to find processes (applications / sequences) that are run frequently. Use this data to go back to the business and observe what the end users are doing. For instance, if you see that P42101 is being run 2,000 times a day – look for opportunities to improve this process. This could be EDI, this could be spreadsheet macros that call an orchestration. What's an orchestration I hear you ask?

Orchestration is the ability to turn any piece of JD Edwards functionality into an API. An API that is easy to call and can be authenticated with the user's username and password. So, exposing functionality to an Excel Macro – would be very easy. You could write an orchestration to enter a sales order (in my example) and then a smart macro to call the orchestration with the data on the spreadsheet. It could prompt for a username and password. If your users are being sent orders in spreadsheets – you may have just increased their productivity and reduced a significant amount of human errors.

RPA implementation can be for simple or complex processes. Look for repetition and eliminate it, as an ex-programmer – if you see repetition in code – there are inefficiencies in that code. ERP Analytics will then allow you to measure the success of your RPA, as the usage of the applications should go down with good RPA implementation.

Orchestration is free to implement and can make a huge difference to mundane tasks.

B CONTINUALLY OPTIMISE YOUR ARCHITECTURE

This may be more relevant for public cloud implementations, but let's be honest – most are public cloud implementations. You must continually drive for reduced hosting costs for all of the JD Edwards assets. Quite often this is difficult, unless you have architected your solution for the cloud, turned the monolithic JD Edwards into an elastic cloud tenant. This can be done.

Fusion5 has created what we think is a world first elastic JD Edwards cloud formation for AWS. This architecture has the ability to expand and contract with load and demand. We are able to define the rules to create new web servers and new batch servers and then retire them when they are not needed. This allows our clients to have a very efficient architecture and if they feel that they are paying too much, we can reduce the size and number of machines accordingly.

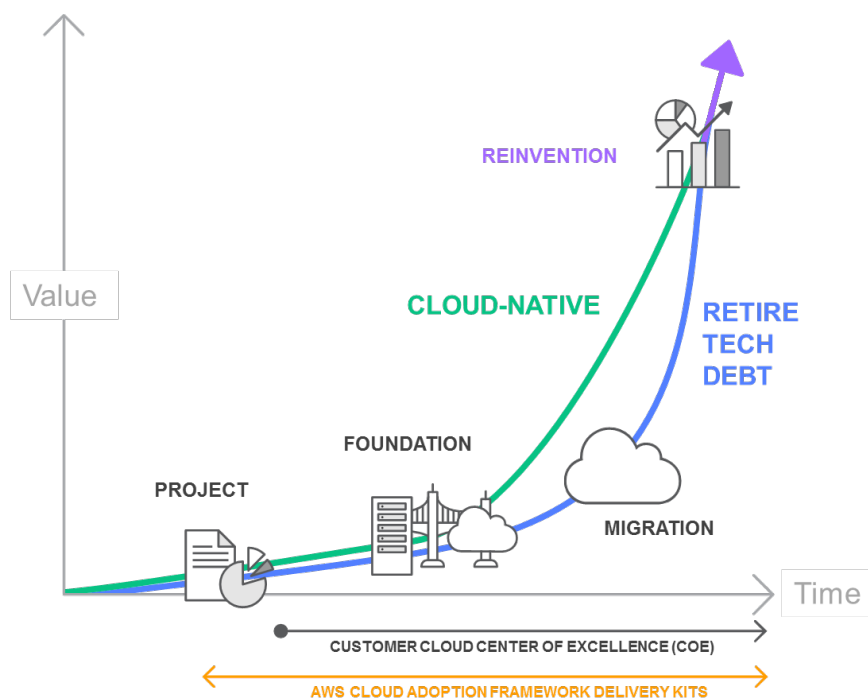


Figure 14: Choosing between re-platforming and rehosting can be difficult, a re-platform is going to provide payback over time.

A good illustration of the options you have available to you when you migrate is above, a lift and shift [rehost] is a simple project – but will not allow you to get true cloud benefits from native constructs (cheaper storage, elasticity or additional security). If you do a re-platform (as I recommend) you can reshape JD Edwards to be a much more flexible cloud tenant.

If you did a rehost, I'd guess you might implement about 8 cloud constructs (EC2, EBS, ALB, multiple AZ, EFS (if you are lucky), whereas if you were re-platforming, you might use (RDS, EC2, EFS, ALB, ASG, CloudWatch, step functions, route53, S3, Launch Templates, Target Groups and more!)

It is much easier to get savings out of a re-platformed architecture.

At a number of sites I've seen savings of more than 50% month on month when we work hard at cloud cost reduction.



CONTINUE TO UPDATE JD EDWARDS

Patches for JD Edwards are now continuous, so your adoption should also be continuous.

I recommend making a plan, with a schedule of when you are going to take patches, when you are going to test patches and when you are going to put them into prod. Start simple, choose twice a year and then work backwards for how long you are going to test, how long for retrofit etc.

If you've been lucky enough to re-platform (as above) then you are going to have some distinct advantages when it comes to deployment. That is that changes can be deployed and tested much more rapidly and actually, continuously. If you have a flexible cloud implementation you could build and deploy an alternate package for production and ease this out into the user community. Our AWS cloud formation allows us to deploy packages without outages, we can do this on a schedule and therefore allow environments to consume change at their own pace. If there is an issue, we can back it out immediately and fix it.

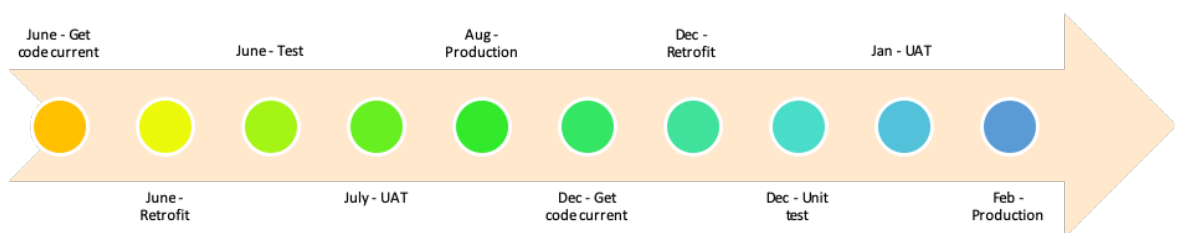


Figure 15: Sample continuous deployment schedule, simplicity is important.

A flexible architecture allows you to be more aggressive with your consumption of change and keep more “up to date” with the latest code from Oracle.

By Shannon Moir

<https://shannonscncjdeblob.blogspot.com/>

Our story

About Fusion5

We deliver business solutions that make a difference

Fusion5 offers a full range of digital innovation solutions, applications, consulting services, development, cloud and managed services as well as first class support services for your business. Our expertise covers all your key functional areas, including Enterprise Resource Planning, HR / Payroll, Customer Experience, Customer Relationship Management, IT Service Management, and Infrastructure.

We are also known for the vertical specific solutions we've developed across a number of key industries.

Our strategy and implementation teams work with you to improve the way you do business. Use our thought leadership and best practice consultancy to drive your company value. Feel the confidence that comes from having the backup of our highly skilled support services. And we're rightly proud of our track record of delivering projects on time and to budget.

Right now, over 800 customers are using our services. Our success is recognised by our key partners including Oracle / NetSuite, Microsoft, Avolin, Ivanti, and AWS.