TEN WAYS TO DE-RISK YOUR NEXT UPGRADE PROJECT

Stagnation is the enemy of innovation. Without innovation, there can be no evolution, let alone revolution. Just ask Darwin or Franklin.

Anxiety can be a powerful motivator, but also a barrier to change; especially when it's concern about the

unknown. Businesses can find themselves paralysed into inaction if they are more worried about the risks associated with change than they are of complacency.

Here are ten ways to de-risk your next upgrade project



Be RealisticCompanies tend to talk about upgrades as either being technical or

transformational. Transformations are hard. A technical upgrade is comparatively easy if, of course, you manage it the right way.



tackle, so let's just leave it'. Change is change. It was certainly be easier to get current on IDE F1 th

Don't Be Intimidated

Don't get caught in the mindset of 'this is too big to

to implement something new. This will be true even if you're on a very old version of the software (OneWorld XE, EnterpriseOne 8.x, etc.).

If you don't have the resource internally, engage with an experienced Systems Integrator to manage the project. Where necessary, leverage the experience of technical retrofit specialists to handle the challenging aspects of your modified

Partner For

Success

specialists to handle the challenging aspects of your modified code. Look for support from proven sources.



Know What's

70%

jettisoned from the mix before you start down the retrofitting path. Why would you spend time and effort dealing with redundant objects?

Changed and What Hasn't

It's considerably faster and easier to perform retrofitting if the customized

footprint is properly analyzed and audited. Unused objects should be

Not All Customized
Objects Are Created Equal

more efficient than trying to re-code everything – determine which objects you actually need and which you do not.

Some objects will require considerably more effort than others when

your customizations against the target release objects is often the least efficient approach. Like the last point, contextual planning for each object is

retrofitting. Think about a "Best Upgrade Approach" for retrofitting. Re-coding



Stick To Your Plan

Know what's going to happen when and hold people

in delays, throwing your project timeline out of sync.

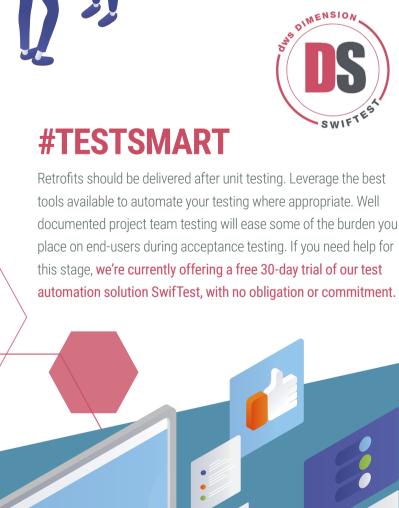
to avoid a code-freeze on the business during a project.

You should time your larger internal projects such that they avoid coinciding with an E1 upgrade or code-current event. Run each project when it best suits your business stakeholders. However, there is no **absolute** requirement

accountable. Retrofitted objects should be delivered in bundles that can be installed and tested immediately. Bad quality retrofitting will inevitably require reworks and result

Communication Is Key

You should build your project teams on a basis of trust and



st

(A) MONTHLY TEST EXECUTION HISTORY

respect. Encourage open dialogue and cross functional, cross

Embrace

Continuous Delivery

By embracing continuous delivery, your organisation

will avoid the irregular and large-scale upgrade pain points. Staying code current doesn't need to be difficult, especially if you appreciate that regular

"Don't put off until tomorrow

what you can do today!"

Get your free 30-day trial of our test automation solution SwifTest, with no obligation or commitment.



ORACLE