WHITEPAPER

Comparing Oracle Forms Modernization Methodologies

pitss

For more than 20 years, Oracle Forms has been an early RAD (Rapid Application Development) tool for Oracle databases. Because of the ease and popularity of the technology, Oracle Forms has amassed a large customer base over the years.

For customers that have remained on the most recent, supported versions of Forms, Oracle continues to release new versions with critical patches and support. Despite incremental improvements, Oracle Forms still has limitations, especially when it comes to user experience.

Companies running Oracle Forms have invested years of development into their applications, and frequently Oracle Forms and Reports still satisfies their most critical business needs. However, customer expectations are rising in our increasingly connected and diverse digital marketplace. Customer demands for faster, better service means new business requirements are coming in to destabilize these legacy systems, and that mean organizations need new solutions to keep up.

So what happens next?

What are the Oracle Forms Modernization Options?

Commercial Off the Shelf Replacement

Migrate applications, business logic, and needs to a standard COTS solution.

Custom One-to-One Replacement

Semi-automatically or manually re-develop your application in a new architecture.

“Reface Don’t Replace” Method

A refacing tool will play back the Oracle Forms processes, then connect with third-party applications.

Strategic Modernization

Retire large sections of your system and modernize the important business processes that bring the most value to your business.

Will an off-the-shelf replacement work?

In some cases, yes, a COTS solution will work for your business. There is a time and a place for these options. For instance, if your business has an industry standard, routine process such as payroll, there are plenty of COTS offerings that are consistent and will work for your organization.

However, in most cases your Oracle Forms applications have been customized to your business needs, and COTS don't always fit. Legacy Forms apps have often been cut down to their most critical functions, highly customized and difficult to replace. There could be a COTS option that comes close, but nothing will 100% meet your requirements. 60% might meet out of the box, but what about that remaining 40%?

COTS solutions frequently offer flexibility and some amount of customization. But once you start to dive into the specific business needs and customizations, you can begin accruing costs and development time. This hidden cost to a COTS replacement is what concerns us the most at PITSS when considering it as an option. In addition to spending money on the new COTS solution, you'll find yourself spending money on development to account for the remaining 40% customizations that your business needs.

Remember: *In order to even begin a COTS replacement for Oracle Forms applications, you'll need solid documentation of your existing code and business processes, a concrete plan, and an off the shelf solution that's flexible enough to meet your requirements.*

Why is a one-to-one customization the wrong solution?

Like COTS, this can be a good fit, but the challenges are different. Transformation is difficult. If you focus on the one-to-one re-development of a complex Oracle Forms application in a new technology, you're not doing the thought work needed to see the true value in your application. This solution can bring you the cost, uptime, and talent pool benefits of new application technology, but not the value and new opportunities the application brings to the business. In other words, your new technology will not be giving its full potential.

With monolithic applications like Oracle Forms try to migrate to a new server side heavy app framework like ADF, you're already ruling out many of the benefits of scalability in the cloud unless you invest in services very specific to that framework.

Bespoke development projects can take longer than anticipated because although you're "saving time," you'll be disappointed because inevitably you'll hit difficulties that you didn't anticipate. These projects are ultimately wasteful because they bring much more code than necessary over into new architecture.

The planning steps for a bespoke development project are not impossible; after all, most Oracle Forms apps started as bespoke custom developments themselves. But these apps have been around in production for years, so they've carried with them years of technical debt. Completely re-developing these applications will be a complex, extremely challenging, risky, and costly endeavor.

In order to re-develop Oracle Forms applications in other languages that lead to more scalability and accessibility, you'll first need to establish documentation about the current system. You'll also need detailed business requirements for the target for your future-state system.

Remember: *Don't bring the past with you in ways that will hold you back. Why utilize outdated software practices and processes when you should be understanding new tools and architectures in a future state?*

Why is "Reface Not Replace" the wrong solution?

When you use a refacing tool, you'll have a new front-end user interface for your Oracle Forms applications. However, you'll still be tightly tied to the constraints of the original Forms.

Yes, a refacing tool has its use cases. For instance, if you only need a few screens to run on a mobile device, this can be an easy and quick option. If you have no plans to migrate, are under a time crunch, and are looking for a cheap option, this might work for you in the short term.

However, a "reface not replace" solution is like slapping a coat of paint on a ramshackle building. As your organization grows, your refacing tool will take on more and more of the work without fixing the issues that exist on the back-end. Adding more technical debt to the monolith and not addressing the underlying issues.

Refacing tools are quick and easy. But they're also easy to make a mistake. This isn't about redoing interface to make your end users happy. You're hiding what you should be fixing, and doing it fast. A higher level of thinking won't enter conversation if you take for granted that you can just take the rigid back-end code, replay it to html, and think that you've done your job.

These short-term benefits come at what cost. Are you willing to pay a cost that's going to set up for success? You'll pay lower upfront costs for inevitable failure down the road. Eventually this will end up costing your company money. After all, you get what you pay for.

Think further than next year. Leaders in the software industry are focused on infrastructure as innovation. Focusing on a tool that simply puts html in front of your app prevents you from thinking about the benefits of leading edge technical discussion.

Remember: *With this method, you're not going to be able to take advantage of future-state architecture. You're going to continue to pay the Oracle Forms licensing and infrastructure costs, but you're not doing necessary thought work to fix the problems.*

Why is strategic modernization the right solution?

Don't LIMIT yourself to the issues that you will continue inherit with legacy architecture. Apps designed around your business process needs are easier to train, much clearer to interact with and navigate, simpler to modify, more efficient, more measurable.

When you modernize your legacy systems and applications using a strategic modernization methodology, you can completely revolutionize your organization's flexibility, revenue, and market opportunity.

Strategic modernization can often be large, multi-year projects. Because these legacy systems can be critical to the enterprise operations, deploying a modernized system all at one time can be a big risk. As a result, strategic modernization projects are rolled out incrementally by each Oracle Forms application. The percentage of legacy code decreases as each piece is implemented, shoring up the organization's technical debt. Eventually, the legacy system is totally modernized.

***Remember:** There are dozens of tools out there. Many claim to automatically modernize your Oracle Forms code, but often the resulting code is far from what your business actually needs. Buyer beware when it comes to comparing your modernization options. You get what you pay for.*

"Typically there are 2 big challenges in any legacy Forms application modernization project. How to extract the business logic that is buried in the Forms application files and how to quickly expose the database tables, views, stored procedures as web services.

"PITSS methodology handles these challenges by combining the deep Forms application analysis capabilities that comes with the PITSS.CON tool with a powerful, front-end agnostic RESTful service layer. In other words, PITSS methodology can analyze the Forms application files, identify the business logic and extract it to the database. Then it can generate REST/JSON web services that can execute the business logic."

Serdar Yorgancigil, IT MRO Solutions at AAR CORP

About PITSS

PITSS modernizes, customizes, migrates, and extends the life of valuable legacy systems. Using data, proprietary software, and deep-dive analysis, we reduce the cost and scope of your digital transformation effort by focusing on the highest ROI processes first. Then, we pave the complete road between you and your new horizons with confidence and expertise. Our full-stack team of experienced UI/UX designers, developers, engineers, technical leads and project managers can guide you through each step of a project. No one knows Oracle Forms like we do—we can take you from planning, to pilot, to the cloud and beyond.

Meet our team to see how we can transform the systems that matter most to your business.

Contact us today.

🌐 **WWW.PITSS.COM**

☎ **1.248.740.0935**

✉ **US.SALES@PITSS.COM**