Oracle Scene

A UK Oracle User Group publication

BI Publisher

Seven Small Steps for a Consultant, One Giant Leap for Userkind

Building a Benefits-driven Business Case

PL/SQL Code: Auto-generation and Obfuscation

UK OUG
UK ORACLE USER GROUP
25th
25 years of serving the Oracle Community

Issue 34 | Summer 2008 | £10.00 | www.ukoug.org

AN INDEPENDENT PUBLICATION NOT AFFILIATED WITH ORACLE CORPORATION

Unaware of “Shadow Data” Systems?

To Do List
Business Intelligence Project Complete
Data Warehouse Project Complete
OLAP Cube Project Complete
I.T. DIRECTOR
© www.leebullen.com

You might think that you've completed your corporate reporting requirements and that all your reports are backed up and secure but is this actually true? You might not know about the informal management reporting system that is being built outside of corporate IT, which has:

- No security
- No backup
- No audit control
- No version control

If you know how to connect to the corporate ERP database then you are free to do what you like. You can even go out and buy a solution that helps you get data into spreadsheets and other databases. You can also:

- Email your confidential reports to people outside of the company by mistake
- Make one-sided entries to the numbers without going through the GL
- Have many different versions of the same report with different people's views on what the numbers should be

The Solution is:

an Accounting Intelligence system

To find out more go to **www.TheGLCompany.com**

Features

- **Runs directly from the secure ERP database**
- **No changes can be made to the actual double-entry records**
- **No technical skill required**
- **Ad hoc reports can be built in seconds**
- **Can be used by Executives, Finance and Business Managers**
- **All inquiries are available 24 hours a day, 7 days a week and 365 days a year**

NORTH AMERICA
Main Office: 303 872 6959 • **East Coast:** 646 673 8644 • **West Coast:** 714 371 4077 • **South:** 281 328 3222

WORLDWIDE
Europe: +44 (0)1494 582 039 • **Australasia:** +61 (0)2 8216 0810

The GL Company® AccountingIntelligence

ORACLE CERTIFIED PARTNER APPLICATIONS

www.TheGLCompany.com

Welcome

to Issue 34 of Oracle Scene!

Ophelia Dodds, Editor
editor@ukoug.org.uk

Thank you all for your positive feedback on the continued changes to Oracle Scene – as always, we are listening to your comments and suggestions and will continue to develop the magazine in line with our readers' requirements, (I should add that the new "Contents" page is one of my particular favourites).

What's in OS

This edition we have a great range of articles, as always.

I am really pleased to have an unusual, but key, B&M article to start with – "Building a Benefits-driven Business Case". This is an area of constant struggle in any IT organisation – how do you really define the true benefits in a business case, and how do you measure these at the conclusion of the project? Derek Hancock takes us through the details, giving an excellent framework for those embarking on business cases.

Regular contributor Satnam Brar then takes us through Training for ERP professionals in "The Training Game" – always a contentious topic for any manager, and talks through the shifts in attitudes to training, and what that might mean in 2008.

Simon Tomey made me smile with his BI Publisher article, with a scarily realistic storyline for our fictional BI hero! Another smile was generated by Tracey Bleakley's attention-grabbing "How Excel Can Kill your system in 5 years!", where she tackles the bane of every ERP professional's life.

For our Siebel users, a very helpful article, outlining "The Top Ten Problems of Siebel…and How to Avoid Them" by Duncan Scattergood offers some practical advice for new and existing users, whilst for everyone there is the second in a series of UPK articles from Larmer Brown, with "Considerations for Planning Content Development".

Moving into our Technical arena, long time contributor Lakshman Bulusu partners with Bulusu Rama to provide the interestingly titled "PL/SQL Code: Auto-generation and Obfuscation (the latter word providing much debate among the editorial team, resolved only by the Oxford English Dictionary), followed by our Top Tips from favourites Tim Onions and Jonathan Lewis, covering Performance and Data Cluster.

In the Community News area, we have an interview with Lawrence Clark (JD Edwards Director, Oracle), and UKOUG continues its celebration of 25 years serving the Oracle community.

Last, but by no means least, a wonderful "And Finally" from our Apps deputy editor, Gio, with special reference to me via Hamlet's upgrade to R12 dilemma (I have to say, I do notice a similarity between Hamlet and my users, particularly in his inability to get to the point!!).

Digital Oracle Scene

You will all know by now that the first digital edition of Oracle Scene was recently launched as a direct result of feedback from UKOUG members. The digital version is exactly the same as the print edition, but offers additional benefits by enabling instant access, being easy to read, navigate and share, containing active links to further information and being searchable.

The feedback received following the launch of the first edition has been overwhelmingly positive. We will be developing the digital edition further over time, so watch this space to be the first to find out about any enhancements to this new offering.

We hope you enjoy your digital edition of Oracle Scene. If you have any comments, please send them to brigit@ukoug.org

UKOUG Office

UK Oracle User Group, User Group House,
591-593 Kingston Road, Wimbledon,
London, SW20 8SA.

Tel: +44 (0) 20 8545 9670
Fax: +44 (0) 870 9000 335
Email: info@ukoug.org
Web: www.ukoug.org

Production

Designed and produced by Anne Kotan
Tel: +44 (0) 1273 833 923
Email: anne_kotan@hotmail.com

Printed by Woking Print & Publicity
Tel: +44 (0) 1483 884 884
Web: www.wokingprint.com

Contents

The views stated in Oracle Scene are the views of the author and not those of the UK Oracle User Group Ltd. We do not make any warranty for the accuracy of any published information and the UK Oracle User Group will assume no responsibility or liability regarding the use of such information.

All articles are published on the understanding that copyright remains with the individual authors. The UK Oracle User Group reserves the right, however, to reproduce and article, in whole or in part, in any other user group publication.

The reproduction of this publication by any third party, in whole or in part, is strictly prohibited without the express written consent of the UK Oracle User Group.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates, used under license. This publication is an independent publication not affiliated or otherwise associated with Oracle Corporation. The opinions, statements, positions and views stated herein are those of the author(s) or publisher and are not intended to be the opinions, statements, positions, or views of Oracle Corporation.

Building a Benefits-driven Business Case

by Derek Hancock, Whitehouse Consultants Ltd

These days it has become an accepted requirement that all new projects should be justified with a comprehensive business case. There are good reasons for this. Everyone has heard horror stories about IT project failures: massive overspend, systems way overdue, absence of benefits, poor user acceptance. It is not the purpose of this article to discuss in detail all the reasons why projects fail, but some of the reasons are connected to the subject matter and those will be examined.

This article argues that a comprehensive business case should be a detailed cost/benefit analysis showing the investment impact of the project and setting out the business benefits and how they are to be achieved.

In the past, as far as many people were concerned, producing a business case meant setting out the reasons for doing something, stating the potential effects if it was not done (sometimes these were the same) and itemising the costs involved.

Given that this approach is no longer sufficient (if it ever was), the purpose of this article is to outline what, in the current business environment, is needed in a comprehensive business case and to offer guidance on how to prepare and achieve the best results.

What is a business case?

According to the website of the Office of Government Commerce, "The business case documents information necessary to support a series of decisions. These decisions, over time, increasingly commit an organisation to the achievement of the outcomes or *benefits possible* as a result of a particular business change." (Italics are mine.)

This is interesting because it talks about a "series of decisions", that "over time, increasingly commit an organisation to the achievement of the outcomes… of a particular business change." In other words, the business case should be written not to seek a single all-or-nothing, yes/no decision to go ahead but to support a process of gradually increasing commitment over time. It follows that, at any time, the decision could be to discontinue. It emphasises not the action to be taken, but the "benefits possible" or results of the action. This is a sound approach and enables the company to take account of changing circumstances rather than fixing the case at a moment in time.

A good business case helps the organisation to determine exactly what it wants to achieve and how it will do so, rather than just being used to obtain funding. It should support sound investment decisions that are based on strategic fit, adequate options explored, affordability, improving efficiency, achievability and appropriate commercial arrangements that are value for money, informed by more reliable estimates.

Obviously the "series of decisions", the period of time and the size of the business case should be in proportion to the scale of the project in terms of cost, duration or impact on the business.

> "A good business case helps the organisation to determine exactly what it wants to achieve and how it will do so, rather than just being used to obtain funding."

Irrespective of size, any business case should contain information covering five key aspects:

- Strategic fit
- Options appraisal
- Commercial aspects
- Affordability
- Achievability

Again according to the OGC, a concise, well-written business case should allow organisations to:

- Make the right investment decision
- Deliver the changes
- Realise the benefits of those investments

Once again, the realisation of the benefits is a key point.

So, the purpose of a business case is:

- To document the justification for the undertaking of the project, assessed on the estimated costs of development and implementation and the risks versus the anticipated business benefits and savings to be gained.
- To gain Management commitment and approval for investment in business change, through rationale for the investment.
- To provide a framework for planning and management of the business change.
- To enable you to monitor and measure the realisation of the benefits of the project.

We can see from this summary that the business case is a touchstone throughout the life of the project. It documents the justification, it gains commitment, it provides a framework for project planning and it monitors the realisation of the benefits once the project is live and the system is in production.

Why produce a business case?

Justifying the investment – Once upon a time, IT investment was seen as "a good thing" per se, but in this highly competitive age, as with any change in the business, investment in IT has to be seen to offer benefits to the business: results such as reduced transaction costs, improved customer service or increased margins. IT investment must compete on an equal footing with any other business investment.

Reducing the risk of failure – Due to the high profile of many projects and the associated bad news whenever a failure occurs, management is much more aware of the potential risks of IT projects, and we all want to avoid being associated with a failed project. If a sound, logical, justified case can be put forward, showing benefits to the business, it is possible we can reduce the risk or at least put the project on a sound footing.

Increasing the likelihood of approval – Apart from the above, as IT professionals we want to increase the chance of a project being approved and thus preserving our livelihood, so it makes good sense to produce a business case anyway.

- "A project is 60 percent more likely to be approved with a cost justification and business case."
  Gartner

- "More than 82 percent of IT decisions now require a cost versus benefit analysis."
  Information Week

- "Return on investment is king, and projects with a quick and clear cost versus benefit are much more likely to get funding in today's uncertain business climate."
  The Industry Standard

In all the above statements, the idea of benefits comes through, either implicitly in a cost justification, or explicitly.

Another interesting statistic:

- In a survey of over 700 sites by CSC, 82% of respondents said that their ERP implementation was successful, but only 22% said that they had achieved the anticipated benefits.

There are three questions here.

1 If only 22% could say they achieved anticipated benefits how could the other 60% know that their project was successful?

2 How do you measure success if not by benefits to the business?

3 Why could only 22% say they achieved the anticipated benefits?

This is where reasons for failed projects relate to the subject of this article. There are many reasons why a project can fail.

> "Without a business case properly setting out the reasons for the change, the strategy to be adopted, the benefits to be achieved, and gaining management buy-in, a project is off to a shaky start."

Failure may be a case of poor project governance (by governance, I mean the whole project structure including staffing, scoping, planning, initiation, project management, risk management, quality), or it may be poor preparation for change within the organisation and therefore poor take-up of the system by the users. No matter how good the system, if the users can't or won't use it, it is a failure.

In many cases it is that the business case was not properly prepared. Perhaps it was too superficial, or not aligned to the business drivers, or it only considered costs rather than cost versus benefits. Equally, it may be that the case was well prepared but no monitoring of benefits was carried out. Sometimes, even though a project has seemed successful it is viewed as unsuccessful, perhaps because of poor perception by the users. In this case, documented results that prove success through comparison to the identified expected benefits can often change these perceptions. Without a business case properly setting out the reasons for the change, the strategy to be adopted, the benefits to be achieved, and gaining management buy-in, a project is off to a shaky start.

How should I prepare for the business case?

You will need to do a lot of preparatory work before you can produce your case. For example, you will need to understand the decision making process; who is involved and what motivates or interests them, who has the power and who has the influence, what decision-making criteria are there and what else is going on.

- Who expects what?
- What is the process for making the investment decision?
- What are the primary business drivers?
- Why do it now?
- What are the boundaries? What is the scope of the intended change?
- What are the intended outcomes and benefits?
- What options are there for making the intended changes? What products or services would be required?
- Who is the executive sponsor?
- Who is involved in the decision-making?
- Who cares about what?
- What are the decision criteria that will be used?
- Are the changes achievable and affordable?
- With what other possible projects are you competing?

What should be in a business case?

A business case typically contains the following sections:

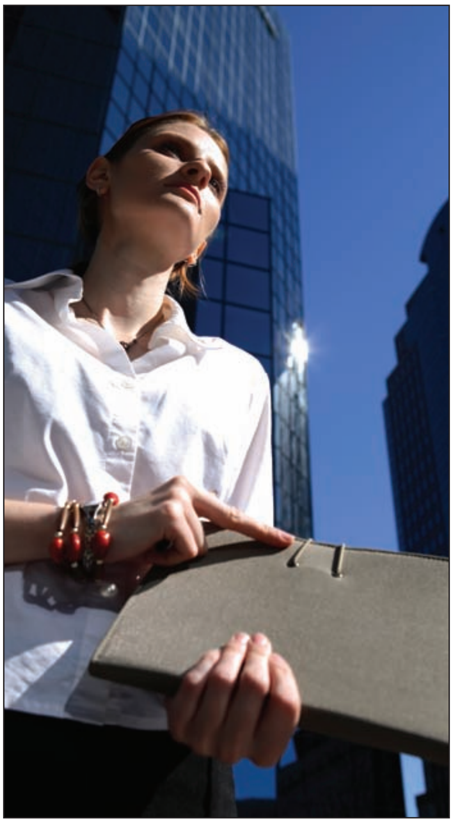
- Management summary
- Outline scope, reasons, strategic fit
- Options appraisal. Benefits expected and quantified. High level cost/benefit analysis of (ideally) at least three options for meeting the business need. Include analysis of 'soft' or intangible benefits that cannot be quantified in financial terms.
- Recommendation, preferred option with reasons for choice
- Costs & timescales. High level plan for achieving the desired outcome, with key milestones and major dependencies (e.g. interface with other projects).

- Investment appraisal, calculation detail, cash flow, payback, ROI
- Risks and mitigation. Contingency plans for non-delivery
- Assumptions
- Source information, proof, validation
- Benefits realisation plan

Here is the first mention of *how* to realise the benefits – a benefits realisation plan. For many this is a new concept but if you are going to justify a project based on savings, you better be clear how you are going to achieve them in reality. If done correctly, this should be a great help when the project goes live and you want to measure the results. It may also be the difference between a perception of success or failure.

Once you understand the playing field and have gathered your information; start putting your case together.

- Get it right!!! Be sure of your facts; check your calculations. Nothing undermines your presentation like an incorrect calculation spotted by a director – "Go away and come back when you have got it right".
- Make it relevant. Gear it to what you know about the decision makers and the company drivers.
- Focus on the top five costs, but do not overlook other costs. Directors will want the headline costs but not the minutiae. Aggregate the smaller costs. Beware of hidden costs.



> "The level of detail required at each stage depends on organisational standards and the scale or complexity of the project."

- Provide explanation, document well. It has to be a standalone document that readers can follow. Remember, you will have to come back to it later and understand what you did.
- Define formulae and variables. Show how you arrived at results. Do not leave gaps where questions or doubts can arise.
- Obtain metrics and calculate. Evaluate the expected benefits using data from the company. Be realistic. Use measures that are important to the company.
- Make metrics easily measurable, if possible. When it comes to assessing whether benefits have been achieved you will want to be able to do so without too much trouble.
- It's OK to estimate. In some cases you will have to, but be realistic and explain your logic.
- Include assumptions. Ensure people can understand how you arrived at a conclusion.
- Determine proof needed and provide evidence. Back up your assumptions and estimates with relevant evidence, using industry data if possible and anecdotal evidence as well. E.g. Company B increased their throughput by 15% when they implemented this system.

How big is a business case?

This may seem an odd question, but many people have not been exposed to a serious business case exercise so have no feel for appropriate size. It is no good trying to justify a multi-million pound project with one page of A4; conversely, 20 pages may be overkill for a new server, so the size is important.

The size of the business case should be proportionate to the size of the project and its impact on the business. (5 – 15 pages excluding appendices). The management summary should be about 10% of the document.

For small projects, you may only need to produce the business case in one stage, with confirmation of prices, funding availability and other relevant material.

For larger projects, you should consider a multi-stage approach:

- Preliminary business case (or Strategic outline case) to confirm strategic fit and business need, typically no more than one or two pages.
- Outline business case – indicative assumptions to support the preferred way forward (including procurement strategy, where applicable), variable length depending on the scale of the project.
- Full business case – validated assumptions to support the investment decision, variable length depending on the scale of the project. The level of detail required at each stage depends on organisational standards and the scale or complexity of the project.

The full business case should include a benefits realisation plan.

To give you an idea of the size of the exercise, it is recommended that you should spend up to 3% of the project's potential investment on building the business case. (So, on a half million pound project, that's £15000 or 15 – 30 days depending on your costing.)

Documenting the business processes that will be affected

One useful exercise that can be carried out for a relatively modest outlay is to map the business processes operating in the affected areas. This serves several purposes:

- It encapsulates the scope of the project for which you are developing the business case.
- It provides a graphical description of the scope for inclusion in the business case.
- It enables you to understand how the processes work.

- It enables you to capture the knowledge that normally resides in peoples' heads.

- It allows you to see processes that are "broken".

- It provides a sound basis for process improvement when the project gets under way.

- It forms the basis of your testing and training materials.

- It assists with benefits realisation by allowing you to monitor the improved processes.

> "With the right tool and consultants, you can carry out interactive workshops with the users and get their involvement and buy-in."

Find a specialised tool that enables you to maintain the process maps together with all the associated information and data on a server, so that you can easily maintain them as the processes change. Word and Powerpoint are frequently used for this sort of thing, but the results are static and tend to get filed away and forgotten.

Find a consultancy company who, equipped with and trained in the use of that tool, have the skills to help you carry out the initial exercise of identifying and documenting the processes.

Involve the users. With the right tool and consultants, you can carry out interactive workshops with the users and get their involvement and buy-in. This gives you an improved understanding of the processes, leading to a more thoroughly researched business case.

Producing the results

Before you can produce your business case, you need to gather and record all the information, carry out the calculations and analyse the results.

The key part of the business case is the investment analysis. This is the foundation for your conclusions and their justification. Different companies use different measures and some measures mean different things to different people, so make sure you know what are the preferred measures in your company. As with many things in life, it is not always simply what the numbers say, but a combination of factors that decide the best course of action; however, make sure the numbers back you up.

- Identify and describe the benefits that the proposed project should provide to the company.

- Record intangibles.

- Identify the metrics to be used – those measures that are relevant to the project and important to the business.

- Quantify the metrics in the current situation using company data where possible, or industry standard data if not.

- Document your reasoning, your assumptions, the costs, the metrics, the calculations, the evidence.

- Estimate the improvements based on what you know about the new system. E.g. reduces time to process a claim from three minutes to two. Try not to over- or under-estimate.

- Quantify the metrics in the new situation based on your estimated improvements.

- Calculate the benefits.

- Compute final results (IRR, NPV, ROI, payback period, whichever is used in your company).

When you have produced the numbers, you will then be able to derive your conclusions and recommendations.

- Evaluate each benefit in turn.

- Identify themes. Several minor, apparently unrelated benefits may add up to a particularly relevant one.

- Check for compelling benefits for different levels of the organisation. A percentage increase on the return on capital employed may not mean much to the Procurement Manager but it might to the CFO.

- Match results to acceptance criteria. Obvious really!

- Include charts & diagrams. Some of your audience may not have time to read the verbiage but may take in a graph at a glance. You know what a picture is worth.

- Determine recommendations. Analyse the numbers, decide which appears to be the best option(s) and make sure your decision is a result of reason, not emotion. By this I mean, don't work backwards to justify a pre-conceived conclusion.

- Use a worksheet layout to present the results, it imposes order.

- Keep it simple – one or two pages – any more and people will lose interest.

Presenting the business case

When you have pulled it all together, analysed the results and drawn your conclusions, ask yourself the following questions:

- Is the business need clearly stated?

- Have the benefits been clearly identified?

- Are the benefits consistent with the organisation's strategy?

- Is it clear how the benefits will be realised?

- Is it clear what will define a successful outcome?

- Where there is external procurement is the sourcing clear?

- Is it clear why this is the preferred sourcing option?

- Are the risks faced by the project explicitly stated?

- Are the plans for addressing those risks explicitly stated?

- Is it clear what the preferred option is and why it is preferred?

All of this sounds like it is a pretty labour intensive and manual exercise. This is true and I would suggest that for a large exercise you put a team together to share the load. Some people are better than others at writing, some are better at numbers and some like doing the investigation but not the analysis. If the exercise is going to involve 3% of project expenditure on a large project, it needs a leader and it needs to be planned. Its execution will also very likely have to tie in with board meetings.

One way of mitigating the effort is to find a supplier with a tool that is designed to assist with producing the investment analysis. Whilst you could do it yourself in Excel, a purpose built tool that allows you to record information about the factors or benefits, hold related metrics and calculate results, i.e. quantify the benefits in multiple scenarios, can save you a lot of time and effort.

With a purpose-built tool you can:–

- Explain each calculation
- Use it at any level
- Create as many scenarios as you want
- Be optimistic or conservative
- Include set-up costs and running costs
- Include intangibles
- Refer back to it
- Repeat the exercise with different values
- Show savings over a selected number of years
- Show IRR, NPV, ROI, cashflow & payback period
- Provide reports and charts
- Use it to monitor results

Conclusion

For IT projects of any size, a business case is a must. You must be clear about the business reasons for undertaking the project and about how it will benefit the business. The investment analysis is a key part of the case and should show quantified business benefits versus costs. The understanding of how the decisions will be made, who will make them and with what other requirements for funding you are competing is as important as the business case itself. When you put all this together to produce a benefits-driven business case, your project will stand a better chance of being approved and of being successful.

For more information about building a business case, refer to the web site of the Office of Government Commerce: http://www.ogc.gov.uk/delivery_lifecycle_business_case_management_.asp

With acknowledgement to the Office of Government Commerce.

About the Author

Derek Hancock has spent 23 years in the Applications Software business, joining MSA in 1985. Following seven years in sales and services with Marcam, he joined JD Edwards as a Client Manager in 1997, staying there until the completion of the Oracle take-over in March 2005. Whilst at JD Edwards he spent three years running a Customer Outreach program aimed at helping customers get more value from their software. He has worked in the JD Edwards marketplace ever since, spending time with Differentia and Cincom Systems. Now with Whitehouse Consultants, the only Oracle Certified Advantage Partner for JD Edwards in the UK, he still concentrates on working with the existing customer base to ensure they continue to get value from their software investment.

10g Upgrade Companion

The Oracle Upgrade Companion helps you upgrade your Oracle Database from Oracle9i Release 2 to Oracle Database 10g Release 2 (10.2). The guide is not an automated tool but instead, provides guidance for pre-upgrade, upgrade, and post-upgrade steps. This document is continually updated as new information becomes available. Please check back prior to your upgrade. The latest version of the 10g Upgrade Companion (v. 2.00) was released on February 13, 2008.

This current version includes information about:

Changes in the behaviour of Oracle Database 10g Release 2 that will affect customers who upgrade

This section documents important changes in the behaviour between Oracle Database Release 9.2 and Oracle Database 10g Release 2 (10.2). It also focuses on behaviour changes that require a DBA to make an informed decision in order to minimise the risks that may be introduced by the changes.

Best practices for pre-upgrade planning, upgrade operations, and post-upgrade analysis

The Best Practices section is not a replacement for the Oracle Database Upgrade Guide but rather a companion document – emphasising and elaborating on recommendations and requirements. Oracle technical staff derived the Best Practices which is an accumulation of real world knowledge and experience obtained whilst working with our customers.

Critical and recommended patches, by platform

This section will help you prepare for an upgrade by providing a list of patches that are recommended for your platform. This list of patches is based on the most common issues encountered by customers on version 10.2.0.3 of the database. Carefully review each patch and only apply the patches specific to the components that are currently installed on your database.

The document is completed through a reference list about relevant Oracle documentation. It is available in printable and interactive versions.

Please review MetaLink Note 466181.1 for details: https://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=466181.1 (MetaLink login required.).

The Training Game

By Satnam Brar, Maximus

The ERP sector and the IT industry as a whole have come a long way over the past twenty years in their attitudes to formal training. No longer content to rely on the command to 'RFM' or 'read the, ahem, manual,' organisations and individuals alike now spend millions of pounds every year on an increasingly wide range of training packages that cover everything from open 'boot camps' to the highly bespoke. But how effective is much of this training and what sort of return on investment does it provide to its purchasers? Does it really make companies and ERP specialists more productive and, if so, is there a 'best practice' method of delivering it?

Jason Wilcox, an Oracle trainer who focuses on finance, procurement and HR, argues that the best training comes from a combination of approaches. "I think you need something like 50% classroom based work, supported by 50% follow-up on specific issues with small groups and individuals." he says. "If you ask anyone who has gone through some form of ERP training they will always tell you that the second element was the most useful, but that's really only because they have already undergone the first part of the process. In my view it's vital to have a clear overview of how Oracle works before you get down to the nitty-gritty."

Another trainer, Tahir Hussain, who specialises in Oracle Financials, takes a similar view. "Let's face it, Oracle is a complicated product and its users will always need formal training to leverage maximum benefit from it. The classroom is the ideal place to deliver that formal training but, in my view, it's important to get the numbers of each teaching group right. Ideally you should be looking at a group of somewhere between eight and sixteen, but some organisations, particularly in the public sector, can't resist the temptation to cram more in to save time or budget. What formal training does very effectively, if you get it right, is to show people where they fit into the Oracle process and why their part of it is important. Done properly it addresses the 'hearts and minds' aspect that actually makes Oracle work."

According to George Harvey, who is both an organisational development trainer with Open Limits and a career management specialist at Amaze Associates, addressing this 'human element' is a benefit that often takes second place to pure technical training but is essential if IT specialists are going to accomplish the British Computer Society's stated aim of recognition alongside more established professions such as law, accountancy and medicine. "If IT is to deliver on its real promise then it needs to become more and more accessible and relevant to the needs of the businesses it serves," he says. "And this means that IT specialists need to develop and employ a much more sophisticated range of communication and inter-personal skills." However, because of the way that people in our industry develop in the early stages of their careers, focusing on the key technical aspects of the profession, teaching these skills is not necessarily easy. "Most IT specialists are brought up in an environment which is very process driven and totally logical – an environment where the first recourse is not to have a conversation. Consequently, the biggest challenge is often to get them out of their usual way of thinking, perhaps by running scenarios which focus on experiences away from the IT sector, such as recalling a particularly bad or good piece of customer service and then relating it back to the problem at hand."

> "...it's vital to have a clear overview of how Oracle works before you get down to the nitty-gritty."

Not surprisingly, therefore, the trainers seem to think that they are definitely adding value to the industry, but what of the organisations which actually have to pay for their work? Phil Wilson, consultancy director at inOApps, an Oracle implementation and training specialist, believes that training has a vital part to play in the development mix, and one which cannot be replaced with simple 'on the job' learning. "There is training being done out there by trainers who understand the subject on a theoretical basis, but who have no real life experience of using these skills in the workplace. Equally, there are highly technical and experienced individuals who struggle to convey the skills in a structured and understandable manner. You also see

a lot of on the job training which can often lead to people picking up bad habits and acquiring a very blinkered view of what the technology is capable of. If you really want to get it done properly, you have to be prepared to pay for someone who has the technical knowledge, hands-on experience and the ability to communicate this in an effective manner. Tailoring or customising the training to meet your current skill levels, working environment and requirements also improves the outcome and benefits of training and provides best value for money."

Wilson's view is largely shared by another purchaser of training, Paul Gillott of Oracle HRMS implementers, Symatrix. "Learning on the job will always be an invaluable part of developing as an ERP specialist but it has to be supplemented by formal training delivered by trainers who truly understand their subject and can draw on solid commercial experience. Without it you continuously run up against the 'we do it this way, because we've always done it this way' mentality so you never move forward, never really improve. Getting in professionals who have real command of their subject, are completely up-to-date and who know how to teach may look expensive initially, but it can be a very worthwhile investment."

Organisations are, of course, only one group of purchasers of training. Individual ERP specialists also dig deep into their pockets to join the growing number of open courses on the market. Ramnik Pattni, an experienced Oracle Applications Technical consultant, recently spent a significant amount of his own money on the five day course 'Extending and Building OA Framework Applications'. "Whilst the core development technology and toolsets have remained more or less the same over the past eight to ten years, in recent ERP versions there has been much more of a shift to Java based end-user self-service applications," he says. "The technology shift will become more visible in future releases of the Oracle ERP product and I was keen to keep ahead of developments rather than trailing along behind them. At first I thought I'd be able to do this on my own, but it's just not possible when you have the day-to-day pressures of work so I decided to get some formal training at one of the UK Oracle training centres. It wasn't cheap, but it was a very good investment of money and time and I learned a lot, both from the trainer and from other people on the course. After successfully completing the course I've been able to put the knowledge gained into action and develop self-service solutions for my current client. Developing real-world solutions gives you greater confidence in what you've learned and, I think, makes me immediately more marketable."

But does formal training really make you more attractive to potential employers? In over ten years as a recruiter in the ERP market, I can think of very few occasions when a client has specifically asked for a particular qualification – hands on experience has always been the key criterion in both permanent and contract roles.

> "Developing real-world solutions gives you greater confidence in what you've learned and, I think, makes me immediately more marketable."

However what training and qualifications do supply is a valuable 'icing on the cake' which can help individuals to stand out from the crowd. And they can also have less predictable but no less valuable benefits. "When it comes down to hiring, I'm less interested in the technical content of a training course, than I am in the fact that the individual has put themselves through it in the first place," says Paul Gillott. "It shows a degree of organisation, commitment and self-belief that makes a good consultant, but even with all these qualities, no course is going to take the place of good, old fashioned coalface experience."

Anecdotal evidence and industry surveys both suggest that the appetite for training is growing, but it will be interesting to see if it will be sustained if the much heralded economic downturn really does start to bite as 2008 progresses. As Jason Wilcox puts it, "Training is always one of the things that gets cuts first on over-runs," whilst Tahir Hussain admits that for many organisations, training is still regarded as a 'nice to have' rather than as an essential. However, George Harvey believes that we may have entered a new era where IT directors have begun to see training, not as a rather unfortunate and expensive necessity, but as a tool to boost their own professional credibility. "CIOs and CTOs are very keen these days to demonstrate that their departments are adding real value to the business in order to stop the trend towards outsourcing. And training is playing an increasingly important role in this as it helps IT professionals to plug directly into commercial reality. If organisations come to see IT specialists as valued advisors and implementers rather than 'back-room techies' then everyone will benefit, and formal training is one very effective way of making it happen."

About the Author

Satnam Brar is the founder and managing director of specialist Oracle recruitment consultancy and Oracle partner, Maximus. The firm works with individuals and organisations on permanent and contract assignments, both across UK and in key markets around the globe such as the Middle East and continental Europe.

He can be contacted at satnam@maximus-it.com

BI Publisher

Seven Small Steps for a Consultant, One Giant Leap for Userkind

By Simon Tomey, BeLife Ltd

Oracle BI (XML) Publisher is a rich new piece of functionality in E-Business Suite Financials and one of the easiest to implement. You can forget the text based reports you have been struggling with and delight your users with clean, attractively formatted documents and spreadsheets.

This is a story (fictional) about how your organisation could apply BI Publisher to good effect to its reports. The article gives an outline as to how you could do the same and concludes with where to get more information.

Delighting your users with BI Publisher...

"What's the point of paying all this money for a system, if we can't even get output we can send to our customers?", said Colin the credit controller.

I had implemented receivables some months ago and we used third party software to format the text based dunning letters from Oracle. When I designed the customer number, it seemed like a good idea to base it on an existing reference which gave a length of 12 characters. I had checked the length of the field in Oracle and it had worked very well with the rest of the team. Only now did I discover that the text based output only printed the first 11 characters.

"The director of marketing is going to make an issue of this one", continued Colin in his inimical way – it's KOLRC! (Kicking off left, right and centre).

I never really liked the idea of using third party software. We had to buy another print server to host it, the set up didn't really work and no one else knew how to configure it. I remembered a conversation with Oracle support where they mentioned "This thing we have called XML Publisher ." I wonder if I could get that to work. It's integrated and it's supported.

That lunchtime I went to the library to get some books on XML. "XML for idiots." Well, I felt like an idiot at that moment; what was I going to do about the customer number on the dunning letters? My wife Gilly called that evening; "the kids are desperate to see you".

"I have to stay a bit later tonight, something's come up," I replied, distracted by my thinking, "XML is all very interesting; a very flexible, self defining language with tags, but I just don't see how it fits with Oracle – the files don't even work without a schema".

Gilly interrupted my thoughts. "We need to decide about the extension, we can't have three children in one bedroom much longer, but the builder said it's going to be expensive".

"Yes, of course, we need to do something soon" I replied. Goodness if I can't sort this out, that 'something soon' might be going down to the Job Centre.

The next day I went to the finance meeting. Surprisingly for a finance meeting, I actually looked forward to these meetings. Jack, the head of finance, and Sarah were enthusiastic users of Oracle and they always appreciated what I could do, which made me feel better. I was trying to make sense of the XML user guide, so got to the meeting a little late.

> "The meeting was getting worse; it was not what I was looking forward to..."

The new girl, Janice, was in full flow... "When I worked for the Big Corporation, they had SuperFinanceSoftware and it was great. In my last company, you could download everything you wanted from the GL straight into Excel... I don't understand why we've got Oracle. The reports don't print properly and when you copy them into Excel they go all funny. While the RX reports are better, it is a hassle to run them and the description that we type into the GL gets cut off at only 15 characters so I can't get all the information." Clare chipped in "Yes, I've got a problem with reports too... Marketing need a list of payments by 'invoice paid', but I don't know how to get this". The meeting was getting worse; it was not what I was looking forward to, to cheer me up...

Later that afternoon, Jack took me to one side. Jack was very good with people and knew how to encourage others when they were struggling. Perhaps he had news on my promotion and I could tell Gilly to go ahead with the extension to the house.

"Michael, I've had a chat with John." Not only was John the Finance Director, but he had influence at Group Headquarters. "It seems a number of users have been going over my head and complaining about the issues with the Oracle reports... John has given me a deadline of the end of the month to get this sorted, or he will be looking to outsource the Oracle support. We need to deliver something for him, or I'm afraid we're going to have to put you on the re-deployment register," Jack continued. "I'm a bit worried about how hard you've been working recently. I'm supposed to be going to the UK Oracle User Group Financials SIG tomorrow, but perhaps you want to take tomorrow off to attend in my place".

The rest of the conversation disappeared into a mist as I tried to maintain my composure. Well at least a day out of the office will be a good chance to think. Perhaps I'll get some sympathy from others at the SIG – unlike my users, they understand the difficulty of the standard Oracle reports. Perhaps I'll get some leads on other jobs going.

"Financials"; "Special"; "Interest": they're words that don't go together easily..." I joked with the taxi driver as he dropped me off. My interest was piqued by one of the agenda items, "Transforming your reports with BI (XML) Publisher – possibly the richest and easiest to implement new piece of functionality. " I wondered, could this help me?

The presenter seemed to have great enthusiasm for the subject and opened by saying "Using BI (XML) Publisher isn't easy, but it's easier than it appears". He asked for a show of hands "Who is actually using BI (XML) Publisher?" (very few hands) "...And who thinks they'd like to implement it but have been put off because they don't know how to start?" (over 50 people put their hands up).

"What I hope you'll get today is enough knowledge to be able to get BI (XML) Publisher working on at least some of your GL reports, but I'll also be using 'Receivables customer open balance letters' as an example".

I jotted down notes from the presentation as follows:

The Seven Steps to Transforming your Reports with BI (XML) Publisher.

Step 1 – Choose a report and save the output

Choose your first report, run it and save the text based output somewhere. Make a note of the report code (see the log file). For example, customer open balance letter in receivables (ARCOBXL).

Step 2 – Change the output format to XML

Log in to systems administrator, concurrent programs, define and change the output format from "text" to "XML".

Step 3 – Re–run the report and save the XML file

Go back into the responsibility as before (e.g. Receivables Manager) and re-run the report. The report will now be produced as an XML file. Open the output and save it as an XML file somewhere where you can find it again (tip: put the report code in the file name to make it easier to find).

Step 4 – Create an RTF template

Use MS Word (or similar program that can create RTF) to create a template. It's easiest with the BI Publisher add-in, but you can do it "manually".

1. Set up your template so that it looks like the output you want. Include formatting and logos, watermarks, signatures, etc.

2. Put something in which will remind you where you want items to be filled with live data. E.g. put address line1 where you want address line1.

3. Open the XML output you saved in step 3 and put the tag names into the document where you want the live data to be. Surround the tag names by <??> e.g. <?address_line1?>.

4. Save the file as RTF (not *.doc!)

Step 5 – Link the Oracle output to a data definition

Login to XML Publisher administrator and create a data definition (home, data definitions, create). Fill out the boxes marked by asterisks and save the XML output file (saved earlier) as "preview data". Chose a sensible "name" (free text field), type the report code into "code" (this must be exactly as per the report), application is your module (e.g. receivables) and a start date of today or earlier.

Step 6 – Link the Oracle output, the data definition and your RTF template to a "XML template"

Stay in XML Publisher administrator and create a template (home, templates, create). Fill out the boxes marked by asterisks and save the RTF file (created earlier) as "file". Choose a sensible "name" (free text field), type the report code into "code" again (this must be exactly as per the report and as typed into the data template), application is your module (e.g. receivables) and a start

date of today or earlier, language is "English", territory is "United Kingdom" (assuming you are in the UK and communicate in English). Click "apply" and wait for the message saying that your template has been created. Preview what you have done by clicking the preview icon.

Step 7 – Run the report

Go back into the previous responsibility (e.g. receivables manager) and run the report. The output should now be produced as per your template (and as per the preview). If you get a message saying "unable to find the published output for this request" and that "no output exists for the request nnnnnn", don't panic – it's just that this report needs to be published via the two step process. (See my slides (number 87) or see Metalink Note ID: 364547.1).

I set aside the next day to see if I could replicate the presentation on my test environment. It took me most of the day to get the hang of it, but once I had done that I tried the same principles on dunning letters.

"These dunning letters are great," said Colin once I showed him the test output. "When can we implement to live?"
Not only were the letters able to display the full content of the fields, but the letters appeared much clearer and include logos, tables and signatures. I used the standard dunning letter output, so the text could be changed using the normal maintenance screen, and the formatting is easily modified via the RTF template. Best of all, they can be produced as PDF and emailed to a mailing house to print and send.

Encouraged by the success with the dunning letters, I set GL "account analysis report" to run with the Oracle pre-loaded template. Instead of having to copy paginated text into Excel and fiddle with it to clean it up, users could now paste the GL account analysis directly into Excel. Best of all, the full field content was displayed, so whatever was entered into the journal would appear directly in the Excel report.

The next day John called me into his office. "Michael, Head Office has heard what we have been doing with XML Publisher and is impressed. They are talking about setting up a team to migrate what you have been doing to all the other companies in the group. It seems that the money which we will save on maintaining our old Oracle reports will easily fund another two posts, not to mention the time we are saving in our division in not having to populate our MI reports. Our customers are impressed by the presentation of the documents we send out, the board members like the reports we give them and we're well ahead on converting our old reports for the planned R12 upgrade. Would you consider leading the team – it will mean more responsibility, but of course that will come with the appropriate rewards."

The End ... (or is it just the beginning?).

About the Author

Simon Tomey is a chartered accountant and Oracle Functional consultant. Simon has significant experience of Oracle BI publisher and is an enthusiastic advocate of its practical application. He would be delighted to hear from anyone with questions.

Simon.Tomey@BeLife.co.uk

Note: BI publisher used to be called XML publisher and is part of the functionality available from the whole Oracle BI Enterprise package. You do not need the whole BI package to transform your Oracle Financials reports, as the requisite parts are integrated into Oracle Financials (and other applications).

For more details on the seven steps to setting up your reports in XML publisher you can download a recent presentation from UKOUG's January E-Business Suite Financials SIG. See http://www.ukoug.org/calendar/show_presentation.jsp?id=8167 or http://www.belife.co.uk/OracleR12_XML_Publisher.html (username "OracleScene" password "Summertime").

How Excel Can Kill Your System in 5 Years!

By Tracey Bleakley, Edenbrook

Does your business rely on multiple spreadsheets to support your Oracle system? Is Oracle no longer the single source of information for the business? Do you have days of reconciliations and manual accruals at month-end? Is reporting an onerous manual Excel driven process? Does the business even trust the Oracle system when so much manual work needs to take place before any results can be published? How did we get here?

For those companies already in this situation, there is an easier way to fix this than to re-implement every time – a combination of change management, benefits realisation, and project planning are needed to rectify the systems problems. These three strands are all interconnected and the timings are critical. This article will suggest a methodology for getting back on track.

When spreadsheets are dangerous

Excel is a fantastic and invaluable tool for business. It is easy to use, universally understood, and makes calculations easy to define and repeat – saving lots of time and effort. This article does not suggest that Excel should be banned from an Oracle environment – far from it!

The distinction (and where spreadsheets become dangerous), is where they become de-facto databases, holding information outside of the system, replicated in different ways in different departments. Despite the well-known number of errors in the average spreadsheet, to the user these small databases become their data-master. By stealth, the pre-project state of multiple databases can become replicated or even worsened, and through a predictable series of steps, trust and use of the system can become so damaged that the only option can seem to be to re-implement. By understanding how this happens, we can assess if this is really inevitable, and hopefully avoid constant and costly reimplementation cycles.

> "...trust and use of the system can become so damaged that the only option can seem to be to re-implement..."

Can we get back on track, or do we need to re-implement? Well, this will depend on your individual set of circumstances. What is likely, though, is that by understanding how the situation happened, a more informed decision on the way forward can be reached.

How do the problems start?

The problems can start right from go-live. Users need to pick up a huge amount of information in a very short period of time (usually through training and UAT). Whilst common processes might be embedded, exception conditions and infrequently used processes may be quickly forgotten. Few companies have an ongoing training plan post go-live, expecting a stable system and user-base to transfer knowledge to new hires.

Post go-live there is pressure on the help-desk/support system and users are likely to prefer working things out for themselves rather than ask the 'How do I' questions, hence workarounds and sub-optimal processes begin to emerge simply because users are keen to get on with the job with the minimum of fuss!

When a project goes live it is very unusual for all requirements to be met and all bugs to be fixed. Usually there is a prioritisation process, with a gap and snag list left over. Without allocating budget at the start of the process to fix these issues, they are often forgotten. It is very hard to subsequently raise a business case for 'gap and snag' projects, as the projected benefits should have been realised by the previous project and the incremental differences are hard to define. It is sensible to set aside an allowance and timescale in the original budget to rectify these issues, without which the projected project benefits may never be realised.

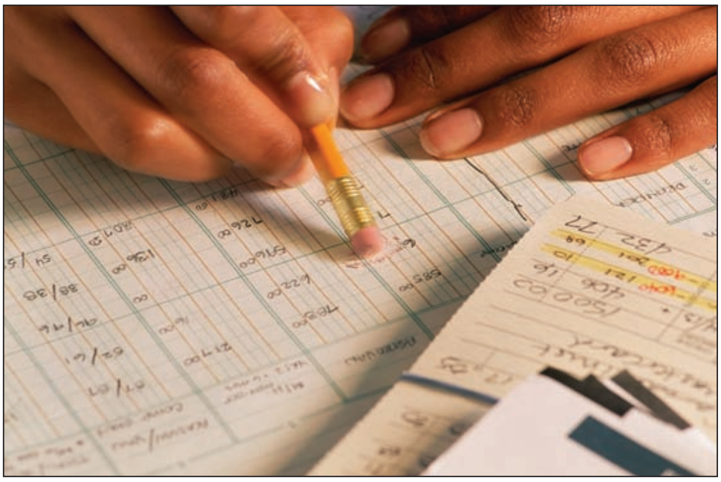
Also, the push to minimise customisations and take vanilla processes (whilst sensible), can lead to manual processes outside the system being required if applied too rigorously. Even standard functionality has some gaps (examples including prepayments and accruals) meaning that most companies need some manual processes from day one. Consideration should be given to the costs of maintaining these spreadsheet-based processes when evaluating the TCO (Total Cost of Ownership) of customisations, especially with SOA (Service Oriented Architecture) making customisations easier to support.

The problem with workarounds

Implementation projects rarely focus on the manual process workarounds needed where standard functionality is not a direct fit. It is unusual for manual processes to be defined and tested prior to go-live, as the focus is on the system. Therefore, Excel-based processes are often developed at speed, and different departments may generate different spreadsheets using different information for the same task. Already, there is no standardisation.

If it is the case that the manual workaround is replacing an existing process or spreadsheet, then this will often become the standard, with the new system needing to work around the existing Excel report. Immediately the fit will be worse than before, leading to dissatisfaction with the system.

Even if there is a genuine appetite for a snagging phase, the demand may disappear once the manual processes are in place. Remember, the users will be more comfortable with Excel than the new system (they have used it for longer!), and will be more



likely to trust it, despite the average number of errors being proven to be much higher. Users can at least see all the data and they know how it is being manipulated (using a new system often feels as though data is disappearing into a black hole!) Already this means that reconciliations between system calculations and results and those generated from Excel-based processes are needed, taking time and effort. Users are more likely to trust the results from Excel and rarely have the support or expertise to investigate any differences. In addition, if users do not trust system reports they are more likely to generate their own in a spreadsheet.

Changing business needs

At some point it is inevitable that the business will begin to change from its original state when the system was designed. Perhaps this happened before go-live, perhaps it was several years after implementation.

If the business changes are accompanied by the time and budget to change or update the system, then many changes can be accommodated. New reporting models can be built in using OFA (Oracle Financial Analyser) or Hyperion, for example. The Chart of Accounts can be extended if necessary by using spare segments (do not be afraid to use these – that's what they are there for!); additional modules can be used for new business areas; new industry specific systems can be integrated using SOA; multi-org can be used to divest and integrate new companies. Also, SOA can be used to integrate new business front-ends (such as employee portals), so that the big systems integration work can be accomplished in a more sensible timeline.

However, the danger is when the changes are undertaken without giving notice to IT or the systems users, or they happen subtly and incrementally, meaning that the users try to accommodate the changes using more and more manual processes and spreadsheets. By using ITIL processes (change management), a process can be put in place to assess business and systems changes and provide governance over how changes and additions should be managed. It is easy for a system to be perceived as inflexible, when it is more the case that due consideration has not been given to the options available, through lack of time, forethought or budget. Again, the business case should take into account the real costs of not making the systems changes in terms of the costs of implementing yet more manual processes.

Another issue is the 'Fat GL', where the business needs more information and the GL is expanded to hold this at the detailed level. By doing this, the system becomes more inflexible to subsequent changes. The GL was not intended to hold business specific detailed reporting data, and this is often best held in a subledger or data-warehouse.

System maintenance

System administration is an area that is often overlooked during a project. Whilst there may be some training on system administration it may focus on the more immediate tasks such as managing responsibilities, security, audit, printers etc. Often the processes necessary to keep system data clean and efficient are ignored, such as:

- Chart of accounts values maintenance
- Closing purchase orders
- Maintaining reports
- Accuracy of core data (such as suppliers, customers and employees)
- Allowing journals generated by subledgers to be updated in the General Ledger
- Allowing journals to be entered in the General Ledger using accounts used for systems generated processes (leading to increased reconciliations)
- Allowing journals to be entered in the General Ledger that should be processed through the subledgers

By controlling these areas from day-one, it is possible to keep data and the system in check and prevent the system errors and problems that lead to reconciliation differences on the systems side. An example is where accounts are created outside the parent, leading to reporting errors. Another is where two accounts are created in different ranges for the same item, leading to confusion and misleading results. A third is where multiple accounts are used for systems based processes (such as accruals), leading to manual processes to allocate amounts, lack of drilldown and a high reconciliation overhead.

Again, as more and more off-system reports are generated and more reconciliation processes are necessary, the data starts to become more and more fragmented, stored amongst a number of spreadsheets and departments. Data becomes duplicated, fragmented and prone to error, making accurate reporting (even across spreadsheets) difficult.

Of course, all these extra manual processes and reports also take time to prepare. This may result in more people being hired, or simply longer hours, low morale and increased stress amongst the user base, especially at month-end or during requests for ad-hoc management reports.

> "...Often the processes necessary to keep system data clean and efficient are ignored..."

Training – is it really necessary?

At the same time, the lack of ongoing training will be starting to bite. The new users will never have been trained in the new system and will possibly be entering data without knowing why, meaning they are even less likely to trust the system. The old users will remember little of the initial training and if user guides exist, it is unlikely that they will have been kept up to date. The business is now not in a position to effect any improvements to the system or processes through lack of knowledge, and the only option may be to bring in external help.

A well-trained and updated user base will be in a position to effect improvements and updates to the system and processes –

providing valuable assistance to the business. This is one of the key tenets of the 'shared services concept'. When evaluating training costs and the business case for training, these benefits should be taken into account.

For the wider user base, it is essential to keep up-to-date with how to use the reporting tools and self-service functionality employed. This not only reduces the load on the core users and IT, but also sweats the asset (i.e. the system) to its full capability, and the system then becomes a full business partner.

Also, consider internal super-user groups and informal networks. Training does not have to be external or costly. There are a range of options, and often using a combination works best. Consider online training content, especially for induction training and refreshers or high level introductions; consider classroom training for detailed topics, using external facilities to get people out of the office environment to increase concentration and generate new ideas; and use internal training and networks to bolster and embed training, knowledge transfer and new ideas.

Without all of this, the data will often be well and truly in silos. The Excel spreadsheets can almost become a rival system, with users claiming to prefer and trust their spreadsheets. However in reality they now represent a huge number of databases, often using the same data in different ways and with multiple different values. This is possibly worse than the pre-implementation situation where multiple legacy systems may have existed.
At this point, data integrity is completely compromised.

At this stage, performance will also be affected and no amount of hard work by the users at month-end and critical reporting times can mitigate this.
System performance can be compromised by badly managed data, proliferation of reports, and lack of training, such as users running reports without a start date or without using parameters to restrict data.

At the same time, the hardware may be struggling to cope with the growth in data volumes that may not have been addressed. This becomes cyclical as users become frustrated by poor performance.

On top of this, the manual processes and reconciliations can take days at month-end, and reporting can be a nightmare of multiple spreadsheets, calculations and reconciliations. Worse still, any adjustments may mean starting the whole cumbersome manual reporting process again.

So what can be done?

At this point, all is not lost. A combination of user training, data cleansing (treated as systems improvement projects), a hardware review and a reporting project (possibly to implement either new solutions, or new models and training within existing systems) may be appropriate. It is often easier to build a business case on the combination rather than the individual parts, as all of these elements combine to give the outcome – bringing back the initial business benefits of the system. As the system is likely to now be a corporate issue, due to the time taken to compile month end and ad-hoc reports, as well as user complaints, this is likely to be a timely solution.

Left unaddressed, the problems become even more severe. At some point the accuracy of reporting will become compromised – and this cannot be overcome by simply adding more manual processing time. The same problems will multiply, and if this is combined with business change, the issues can begin to look insurmountable.

However, there is still a solution. In addition to the changes advised earlier, it may be possible to effect a number of systems changes to bring the system in-line with the business. Again, it is recommended that the business case be built on the combination rather than the individual parts, as all of these elements combine to give the outcome. If this is the case, then it is recommended that the individual components be implemented as a programme so that each element is fully implemented at the right time, in order to maximise the benefit.

And if we do nothing?

If the problem is not addressed, then trust in the system can become irretrievably compromised. As the problems escalate, re-implementation may start to seem like the only option because trust in the system is gone, and the likelihood of signing off additional funds to resolve systems issues is low.

> "As the problems escalate, re-implementation may start to seem like the only option."

Things can only get worse if not addressed. When standards in data entry are forgotten, then reporting can completely break down. In Accounts Receivable this can mean an inability to produce an aged-debt, in General Ledger it can mean that reports need to be generated using child values as the parents are inaccurate – opening up the process to more errors, and so on.

Performance will continue to be impacted, as will user morale. The system will start to become a serious management issue.

So what now?

Recovery is still possible, but must be scoped as an overall programme. Small, tactical one-off projects are unlikely to succeed as the problem is spread across the system and users and management will need to see quick-wins and constant improvement.

The question of re-implementation versus improvement will need to be seriously considered. If the business has changed significantly, if the version of the application is old and there are serious gaps between that and newer versions, if the chart of accounts is no longer valid, if the General Ledger is full of detailed obsolete business specific data, if the shape of the company has changed (requiring a new operating structure), or if the hardware is obsolete, then re-implementation may be a valid option.

If the answer to all of the above is no, then improvements as part of a consolidated programme may be a cheaper and more

pragmatic option (especially if the newly trained user base can participate in the improvements).

So in summary, how did we get here?

By going live without all the requirements and bugs being addressed and by not revisiting these later, we deliver an incomplete system that cannot possibly realise the projected benefits. If the initial training is not adequate or not planned as an ongoing effort, then the users will not be empowered to maximise use of the system ongoing. Trained users, good system administration and governance procedures, and a focus (and budget) for constant improvement, underpinned by ITIL, are essential.

This may be expensive – but consider the costs and benefits as opposed to re-implementing every five years and the effect on the business as the system diverges in between!

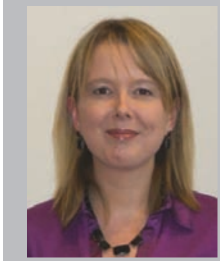
If you think your system needs a re-implementation rather than a series of improvements to get back on track, then consider the following:

- A Chart of Accounts and Calendar fit assessment
- Fit between current business processes and system processes
- Ability of the system to support management and statutory reporting requirements, including consolidations, group reporting and tax
- Future business flexibility required
- Potential merger or divestment activity, or legal/market changes
- The current level of customisations
- The gap between the current and the latest software release and functionality
- Organisational changes such as shared services

And if we do re-implement, what should we do differently this time?

- Identify target benefits upfront and compare to progress, even after go-live
- Budget and plan for ongoing training and system development
- Identify an internal systems governance/ administration role and processes
- Review latest best-practice thinking and revise business processes
- Take advantage of new functionality available (such as SOA) to increase flexibility
- Decouple data where possible: thin General Ledger, standard reporting lines, correct use of subledgers etc
- Archive old data where possible and implement an archiving strategy
- Take the opportunity to update support arrangements
- Spend time on conference room pilots, UAT and training
- Implement regular system reviews and processes – this is a living corporate asset!

About the Author



Tracey Bleakley is Head of Change Management at Edenbrook and a director of UKOUG. She has 12 years experience of designing, implementing and managing business transformation projects supported by Oracle solutions across a range of sectors including telco-media, financial services, retail and professional services.

Top Tips: Part One

By Tim Onions TOdC Limited

Before I dive into the meat of this issue's tip(s), I need to clear up something from last time and I also have one or two follow-ups to slip in. I am indebted to Des Browning who queried something I implied in my tip about getting the database server hostname:

> *I've never used a login.sql because when you switch between databases with logging it isn't re-executed to "refresh" the prompt. Has this changed? I didn't see a reference to it in your article.*

This is a brilliant point which brings up two further (small) tips. As Des correctly suggests, yes, the way SQL*Plus handles the LOGIN.SQL script has changed – LOGIN.SQL is re-run, from the 10g version if my memory does not fail me, whenever you use the CONNECT command.

However, where the 10g version of SQL*Plus is not used, what I have done for years (well, since Oracle7) is to create and use a script called CONNECT.SQL, place it somewhere within SQL_PATH and call this rather than use the CONNECT command directly (i.e. always use @CONNECT when changing users/sessions). My CONNECT.SQL simply issues a CONNECT command based on any command line parameter supplied and then re-runs LOGIN.SQL. So when in SQL*Plus you can use:

```
@CONNECT user/password@alias
```

and be confident that your prompt and variables are set appropriately for the database to which you are logged in. Granted, you need to remember to add the "@" symbol (or START command) otherwise things will be out-of-date. I have found, however, that after a very short while this gets engrained in your psyche and rather than forgetting to use it, you will find yourself forgetting NOT to use it when you are on a client (or database server) other than your normal means of running SQL*Plus. Note, it does also pay to just run @LOGIN manually if at any point you are unsure of your prompt.

A few months ago, a work colleague recommended to me a blog that had just been started up by Oracle's own "Optimizer Development Group". To my shame I only just very recently found time to take a look – but, when I did, I found the content there (and on the associated "Oracle Database Real-World Performance & Scalability blog") very interesting and valuable indeed. What particularly caught my eye was that at least two of the articles provide follow-on material for recent Oracle Scene articles: one covers V$SQL_PLAN/DBMS_XPLAN and hence relates to an article I wrote a little while back; another deals with ANSI syntax outer joins and hence builds on Tony Hasler's article in the last issue of Oracle Scene. So, if you have a technical, quizzical mind and want to hear direct from those in Oracle who should know (and from reading these blogs most definitely do know) then check-out:

http://structureddata.org/about/

and

http://optimizermagic.blogspot.com

Top Tips Part Two can be found on page 37.

The Top Ten Problems in Siebel Projects ...

... and How to Avoid Them

By Duncan Scattergood, Customer Systems plc

Organisations often spend huge sums on buying and implementing Siebel, and many see a fantastic return on their investment - increasing sales, improving customer service and reducing internal costs. However, others struggle and often become disillusioned with the whole process after years of effort. Why do Siebel projects go off track, and how can organisations ensure that theirs will be a success?

Siebel is an immensely powerful product with an extremely wide set of functional capabilities. However, it is the way in which the product is implemented that determines whether the anticipated benefits are delivered.

A project can fail for a variety of reasons. During our ten years working with Siebel we have been called on to rescue many projects that have gone off track, and time and time again the cause can be traced to a basic set of problems.

By bringing together the ten reasons why a Siebel project is most likely to fail, this article examines the factors that separate success from failure. We explore potential pitfalls and how these can be avoided, and consider how to bring problematic projects back on track.

The Top 10

1 Unrealistic short term expectations

One reason why a Siebel project may be perceived to fail is that there is an unrealistic expectation of how quickly the benefits will be seen.

Merely installing Siebel and making it available to users will not transform the effectiveness of an organisation. Instead, realising the full capabilities of the application takes a lot of hard graft from technical and business oriented teams. Time and effort are essential, and not all the benefits will be delivered at once.

For example, if a key benefit is more accurate forecasting of sales opportunities, that benefit will not be realised until the organisation's opportunities have been actively managed through the new process and there is consistency in the data.

To avoid setting unrealistic expectations, care must be taken to understand at the outset:

- The actual IT costs of the project
- The anticipated benefits and when they are expected to be delivered
- The prerequisite process steps, both technical and business, to obtaining the benefits
- The necessary commitment by the business of time and resources to the project

If adequate consideration is given to these points when developing a project plan, it is possible to celebrate milestones along the way to the ultimate goal, rather than becoming disillusioned by how much work has been done and yet how far there still seems to be to go.

> "Time and effort are essential, and not all the benefits will be delivered at once."

2 Insufficient Siebel knowledge in design team

Experts in the relevant business processes, or experienced business analysts who have worked on other applications, can play vital roles in the analysis and design team, but they must be complemented by real Siebel experts. A lack of Siebel expertise is likely to give rise to the following scenarios:

- Processes are designed based on how non-Siebel applications work, leading to many screens being required for something which could be achieved with one or two button clicks. This leads to significant and often unnecessary re-engineering of Siebel.
- Particular ways of working are imposed on the application, when Siebel already supports the exact requirement in a slightly different way. Recreating functionality that exists in the core product is unnecessary and expensive, both in terms of initial development, and of ongoing maintenance, enhancements and upgrades.
- The system fails to meet all the requirements because it is perceived that certain things cannot be achieved in Siebel when in fact, with appropriate skill, they are fairly small configuration tasks.

The importance of engaging real Siebel experts at the early stages of a project, with both functional and technical knowledge about what is and is not achievable – and how best to achieve it – cannot be overstated.

3 Separate technical team

In a traditional IT project, a business analyst speaks to users in order to gather requirements, before documenting them and passing them to a technical designer. The designer translates them into a technical design which is then implemented by a programmer before the system is tested and deployed. This approach results in a large gap between the users with the original business problem, and the technical team charged with solving it. In a more extreme version, the development work may even be carried out off-shore.

This model may still have a place in some situations but it is not appropriate in a Siebel project. Siebel is different because a skilled configuration expert can bend the core application to meet a particular requirement very quickly. This presents an opportunity to run projects in a way that greatly increases the probability of the delivered system meeting users' needs, while lowering project costs by reducing the project team size.

> "Siebel is different because a skilled configuration expert can bend the core application to meet a particular requirement very quickly."

The ideal model for a Siebel project is a small, expert team where each individual is responsible for all activities relating to a particular functional area, from requirements gathering, through design, development and testing, to deployment. In larger projects, there may well be people who specialise in design, build or test, or people who are involved in only one area, but the core team should remain throughout the whole project.

4 Insufficient user involvement

Ultimately, a Siebel implementation is a business project that must deliver business benefits. The project is for end users and their managers, so it is critical that they are fully engaged throughout the project to ensure that the final system delivers what they need.

Often, other responsibilities mean that users have only limited time to give to the project, but if they are only prepared to commit very small quantities of time – or even no time at all – then our experience is that a project will not be as successful as it should be.

Typically, the most important stages for significant user involvement are:

- In the design phase, including workshops and prototyping to verify that the user interface will support what they need to do. At this stage, it is also important to run through each of the processes that users will perform, to check that nothing is missing.
- In the user acceptance test phase, to check that what was finally built really does do the job.
- In the deployment phase, to assist in rolling the system and associated process changes out into the wider user community.

In addition, continual (if less frequent) involvement throughout the project, including build, is advisable. This involvement really does help users to have a strong sense of ownership of the project, as well as allowing frequent checks that the project is on track, avoiding surprises during user acceptance testing.

When planning user involvement, a key consideration is who to involve. This varies from organisation to organisation, and also depends on the task to be performed. Of all the user-centric tasks, design workshops are normally the most critical to a project's success. Senior management often do not have enough time to do the activity justice and may not actually understand the finer detail of the processes that users will perform on a day to day basis. On the other hand, running workshops with low level users can result in the collection of information that reflects the details of the current system rather than the changes that senior management are trying to implement. Therefore, the best candidates for these types of activities are often found at team leader or middle management level.

5 Poorly thought-out integration

Many Siebel implementations involve complex integration with other applications. Even if each individual application that supports a business process is well designed and robust, if the interfaces between those applications are flawed, then the resulting overall system will not be solid.

Again, expert Siebel knowledge is essential in this scenario. If the team handling the integration has little Siebel knowledge, the following often occur:

- Inappropriate split of tasks between applications: for example, something that could be easily achieved in Siebel is customised into another application because no one knew better.
- Unresolved inconsistencies between data models: for example, a contact can have many addresses in Siebel, but only one is supported in a particular legacy application.
- Incorrect interface type selection: the wrong decision is taken over which integration technology (EIM, eAI, Web Services etc) to use. Integration routines may also be poorly developed, since this is primarily a Siebel configuration job requiring expert Siebel skills.

Even when Siebel expertise is engaged in the design and build of integration, organisations often choose to subdivide the Siebel team along the line between configuration and integration. This is a mistake because decisions made at the design stage about how the user interface should work can affect the format, style and frequency of integration – and choices made in integration (often forced by legacy applications) can affect how information should be presented to users. Therefore, the two aspects are by no means independent, and as a minimum, the team building the core configuration should be expert in the issues surrounding integration and vice-versa.

Integration must be considered in the context of the overall application design, and not dealt with as an afterthought. Of course, individuals with high levels of expertise in the other applications concerned should also be involved, and collaboration is the key to successful integration projects.

6 The difference between sales and service projects

A Siebel application is usually intended to be used by one of the following groups:

- Users selling high value products to a relatively small number of customers.
- Users dealing with low value, high volume sales and/or service type transactions.

A particularly common mistake resulting from a lack of Siebel expertise is the failure to recognise the significant difference between these two types of project.

Low value, high volume sales and service environments are typically very transactional with emphasis on users gathering set pieces of information in a set sequence, and the style of user interface built needs to reflect this.

By contrast, the nature of transactions in a high value, low volume scenario means that processes are typically quite unstructured, with skeletal information about new opportunities being fleshed out throughout the sales process. It is therefore difficult to predict the sequence in which users will navigate through Siebel and enter information, and again the user interface must reflect this.

Furthermore, high value, low volume sales users can effectively boycott the application if they do not like it – organisations do not fire their top performing salespeople simply because they will not use Siebel.

Additionally in this situation, increased management control is often a key stated objective of the Siebel project. However, if this is the sole objective of the project then end users may feel that the system's only impact on their day-to-day activities will be a negative one. Consequently, user adoption will be poor and the project will fail.

> "The key is to encourage user adoption by offering genuine benefits for the sales force."

The key is to encourage user adoption by offering genuine benefits for the sales force, such as the provision of up to date competitor analysis, or automatic generation of presentations, proposals and quotations.

An application that is well designed for one type of scenario rarely suits the other. Misunderstanding the user types and consequently not designing the user interface correctly is an expensive mistake to rectify later, not only in terms of build cost but also in terms of lost user confidence when an inappropriate application is deployed.

7 Insufficient knowledge in development team

Configuring Siebel quickly and efficiently is not an impossible challenge but it is a specialist job. Whereas with many programming languages an expert might work two or even three times as fast as an average programmer, with Siebel development the factor is much bigger, maybe as large as 20.

A development team with an inadequate skill set risks delivering any of the following:

- An application that does not meet users' requirements because the people performing the configuration do not know what is possible or how to implement it.

- An application that was expensive to develop because it contains more configuration than necessary to achieve a given task, in particular extensive coding to implement something that could have been handled far more easily in Siebel Tools if the developer had had the relevant knowledge.

- Following on from the previous point, an application that is difficult and costly to support, enhance and upgrade.

- Unacceptable user response times due to inefficient configuration or poor user interface design.

As with many of the problems outlined in this article, the solution is to ensure an appropriate level of Siebel knowledge and skill is engaged at this stage of the project.

In the ideal project there would be no separation between the design team and the build team, and in the best projects the design team will simply be absorbed into the core of the build team.

8 Insufficient or inappropriate testing

It is not necessary to spend an excessive amount of time checking the standard features of the packaged Siebel application, which can usually be assumed to work correctly. Pre-deployment testing must instead focus on the specific configuration that has been carried out to meet the organisation's particular requirements. Ensuring that the application supports the business processes as intended, and that processes run correctly across multiple applications, is also vital.

The trick is to use the limited time period available for testing in the most effective way possible, concentrating on the areas of highest risk and/or those with the most impact if there is a problem. This usually means focusing on the following:

> "Ensuring that the application supports the business processes as intended is vital."

- Do the processes, as defined in the workshops, work as anticipated? If a good job has been done of involving users through the design and development process, this should not normally be a major area of concern.

- Are any exceptions to those processes supported?

- Do integrations work between applications?

- Is all the master data, e.g. list of values, correct? Do the initial data loads import all the data correctly?

- Will the application work correctly in the production environment? For example, the development environment might sit on a single LAN, but multiple firewalls might be involved in the production environment that require opening of ports and other infrastructure changes.

- Do any specialist modules, e.g. Siebel Remote, work as expected?

- Will the application work with production volumes? As it can be very difficult to replicate production usage scenarios, careful thought ought to be given to whether this style of testing is necessary and practical and, if so, what scenarios will be simulated.

Test plans are also advisable to ensure a degree of repeatability, although these should be as brief as possible to keep the emphasis on high quality testing, not on the maintenance of test plans.

9 Poor user adoption

It is not uncommon to come across Siebel implementations that are, in the main, technically sound and well aligned to user requirements, but are poorly adopted because the users do not understand how to take advantage of the facilities offered.

This lack of knowledge can and does occur at multiple levels:

- Not knowing how to perform basic navigation, search and data entry: for example, taking multiple actions to achieve something that can actually be done with a couple of button clicks.

- Not understanding how the application relates to the organisation's business process: for example, not knowing when to enter an opportunity in the system, or what stages it should go through. This can result in one user not completing all of the necessary steps in the system to allow the next user involved in the process to do their job.

- Incomplete or inappropriate data administration: for example, if an administrator fails to correctly set up the organisational structure within the application, a user could find that they are not presented with the subset of data they need to see in order to do their job.

> "...a key requirement is to extract data from the application and put it in a form that managers at all levels can use to make decisions."

One of the main causes of this is poor training. The training provided may simply be insufficient, or it may be inadequate in that it only covers the mechanics rather than the procedures specific to the organisation. Training that is not delivered very close to the point of system deployment, or training that does not reflect a person's prior technical experience, will also not yield the best results. For example, it would not be appropriate to use the same training material for a field sales person who has previously hardly used a computer, and a technically literate power user. However, both of these users still have training needs.

Surprisingly often, projects get as far as the end user training and deployment stage before it is discovered that the detailed procedures for using the system have not been fully agreed and documented. This lack of clarity is another reason why users do not use the system that has been developed. The normal cause of this is insufficient user involvement throughout the project and it is often one manifestation of more deep seated problems in an implementation.

To avoid delivering a system which users cannot use to its full potential, it is essential to thoroughly plan the way in which the application will be used, and to provide adequate training to users across the organisation.

10 Inadequate reporting

When designing and configuring a Siebel application, it is easy to focus entirely on the entry of data, and how that data should be transferred to other applications by integration.

However, in most implementations, a key requirement is to extract data from the application and put it in a form that managers at all levels can use to make decisions. In other words, the application must have the capability to report on and analyse information held in the database.

Oracle provides a number of reporting and analysis tools including Siebel Reports (Actuate), Oracle BI Publisher and Oracle Business Intelligence Enterprise Edition (Siebel Analytics). The most suitable tool depends on the situation, and it is often appropriate to meet multiple requirements with multiple technologies.

It is essential to consider reporting at the start of a project, to ensure that the information captured suits the reporting requirements, and to ensure that reporting and analysis are delivered either in parallel with, or very shortly after, the main Siebel releases. If any of these steps are omitted, an otherwise excellent Siebel implementation can elicit a reaction of "so what?" from the business community.

The common theme of product knowledge

The majority of the problems outlined here are underpinned by a common theme. If the project team does not understand Siebel well enough from a functional and/or technical perspective then it is very unlikely that the system will be well aligned to users' needs, of reasonable technical quality and cheap to create.

Product knowledge cannot be measured in years of experience. It is quite possible for an individual to have worked with Siebel for several years without actually having learnt a great deal about the product. In fact, true product knowledge encompasses three key things:

- A thorough understanding of what the core product does and how that can be applied to particular business situations.
- A deep comprehension of the principles behind how the product is built. It is not possible to hold the entirety of such a vast product inside one person's head, but it is perfectly possible to have a firm enough understanding of the underlying technologies to allow any new functional area to be learnt very quickly.
- An understanding of how to gently bend the core product to any given business situation and, in particular, which of the multitude of available configuration techniques is most applicable in which situation.

Projects may have effective team members who have either a strong functional or technical emphasis in their skill set, but a truly worthwhile team member will have knowledge about both dimensions.

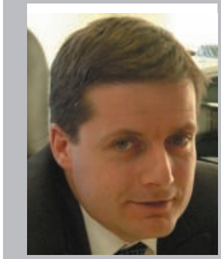
In our experience, the most successful Siebel projects have a small team of experts who can both speak to users about requirements and effectively configure Siebel to meet those requirements. The value of real product knowledge (as opposed to certificates or years of poor quality experience) cannot be overestimated.

In conclusion

A Siebel project has the potential to increase an organisation's sales, improve customer service and reduce internal costs. However, without adequate effort, planning and expertise, the project can run into a number of problems, and ultimately fail to deliver on expectations.

To achieve success, it is essential that your project team is equipped with the skills and resources, both internal and external, that they need to maximise the benefits of this powerful software suite.

About the Author



Duncan Scattergood is Operations Director and co-founder of Customer Systems plc, where he is responsible for all sales, marketing, project delivery and recruitment functions. Customer Systems is wholly focused on Siebel, Siebel Reports and Siebel Analytics and has accomplished hundreds of successful implementations. Based in the UK, they have delivered services in 29 countries across 4 continents. 19% of the customers listed in Siebel's last annual report have used Customer Systems.

Duncan can be contacted at dscattergood@customersystems.com

Oracle User Productivity Kit (UPK)

Considerations for Planning Content Development

By Peter McClintock, Larmer Brown Consulting

In the last article, we introduced Oracle's UPK as a valuable tool to help gain user acceptance and adoption of the business application being implemented. We need to start, however, with a word of caution. Just as the best software and technology are not sufficient to ensure application success, so the best training and support tool, on its own, will not miraculously ensure user acceptance. Because of its powerful capability and ease of use, UPK is often presented as an out-of-the-box solution that organisations can pick up, use easily and achieve success first time and every time.

When to start development?

This article will focus on the preparation and considerations necessary to use UPK effectively within a software implementation project. The first of these considerations is planning. Because UPK can contribute across the project lifecycle, it is important to begin the planning early in the project. As consultants, we find that the most common point that we are asked to plan UPK Content development is around the CRP-2 stage. I would suggest that this is too late to achieve many of the benefits that UPK can bring. For example, since UPK can generate test scripts for System Testing and User Acceptance Testing (UAT), it makes sense to integrate UPK recording with the activity of the Testing Team. This will reduce the likelihood of doubling up on the work of generating the test scripts, (see Figure 1).

So, when should we start UPK content development? I would suggest that if we are going to use UPK for test script generation, it should be as soon as there is a reasonably stable application environment to generate the test scripts against. If there are changes to the screens after recording, the UPK content can quickly be updated, using the new Guided Re-record capability that is in UPK V3.1. This allows the developer to run quickly through the recording, replacing just the screens that have changed. All of the other content, such as explanations and support references, remain intact.

The start of the UPK development, therefore, should coincide with the availability of the first stable application environment. That environment should have sufficient data to allow meaningful process scenarios for training purposes to be recorded. For example, if it is a financial application, there should be relevant suppliers and customers, set up with sufficient data, to allow 'search and select' to operate effectively. It is vital, when following this methodology, that there is a mechanism in place to inform the UPK development team of any changes to the application or processes that may affect the content.

How many developers will you need?

We now know the start of the development window, and we can also determine the time when the content needs to be ready. This will either be a week or so before UAT, or in time to commence the user training roll-out, depending on whether it is planned to use UPK to support testing. It would be usual to allocate two weeks between completion of UPK content and the start of training, to allow for the development and piloting of the user training courses. We will discuss the development of blended training with UPK content in a future article.

In most ERP implementations, we have found that the available development window will be about eight weeks, and based on an average scope of 400 topics, this means that a development rate of about ten topics per day must be achieved.

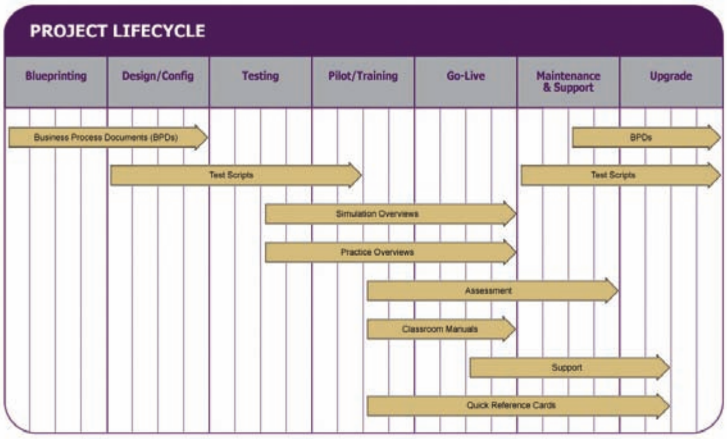


Figure 1: Project Lifecycle

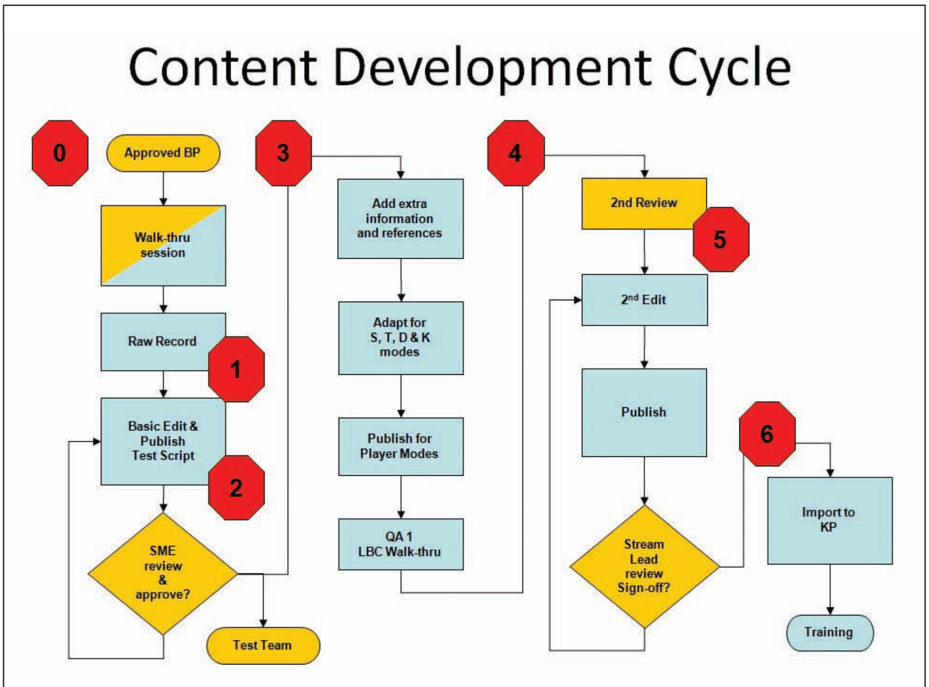


Figure 2: Content Development Cycle

It needs to be borne in mind that this is not just the recording of the topic, but includes taking it through all the review, editing, QA, Sign-off and Publishing stages needed to deliver the final output ready for deployment. We have found that an average UPK development team will be made up of a Lead Developer and five or six other developers. Clearly, this should be thought about at the PID development stage, so that the required number of UPK Developer Licences can be purchased and the UPK development team included in the project costs. It is surprising how many times an organisation finds itself facing development with only one or two developer licences and no budget to purchase more.

Creating the development team

When considering the UPK content development activity, it is important to think about the role of the Subject Matter Experts (SME) and the Business Process Owners (BPO). It is our view that those with a training background usually make the best UPK developers. They are most capable of developing content with the full awareness of how it will be used, and therefore have a natural ability to produce the most effective content. It is also valuable to separate the role of SME from that of UPK Developer. The SME will be steeped in the new processes and a fresh, independent user view from the UPK Developer will ensure that the content is developed in a way that the learner will pick up easily. The interaction of the SME and the UPK Developer must therefore be structured to ensure that the contribution of both is maximised, without generating overhead.

Managing the development factory

The content development must proceed within a structured development cycle that will contain the following elements, (see Figure 2).

- For each topic to be developed, there will be an initial walk-through to ensure that the developer understands the process and any critical requirements of the user.
- The Developer then makes a Raw Record of the topic. This activity can be combined with step one in certain cases.
- This Raw Recorded topic is then taken through a basic edit, eliminating any entry errors or other defects in the recording.
- The topic will be published onto the review server and the SME will perform a First Review, checking that the basic logic is correct.
- The Developer will take up the topic again and, having corrected any errors noted by the SME, will go through the first edit. This is where all the additional instructional text, explanations and references are added to the topic in order to transform it from being a simple step-by-step process into a rich piece of instructional and support material.
- It is normal to do a QA at this point, where the UPK Development Team Lead checks the topic against good practice and specifically the development standards for the project. This helps to ensure that all development has a consistent look and feel to the user.
- The topic will again be published onto the review server and the SME will perform a second review. This time checking the explanations and references, ensuring accuracy of instruction and completeness.
- Finally, the topic will get a second edit from the Developer and go onto the review server for final sign-off by the BPO.

Following a process such as this will help ensure the quality of all the content.

In a typical E-Business implementation, there will be about 400 UPK topics to develop. It will be obvious, therefore, that the requirement to track the progress of 400 topics through the eight stages outlined above is vital. It is important to have tracking software and methods to ensure that there are no bottlenecks across any of the eight development stages, and that it can be clearly presented to the Project Manager how development is proceeding against plan, (see Figure 3).

There will also need to be a system to manage the flow of topics between Developers and Reviewers, and a way to document the feedback of review comments. UPK development in an application implementation project is similar to a production line environment and, if efficiency is to be achieved, the interfaces between the production steps must be seamless.

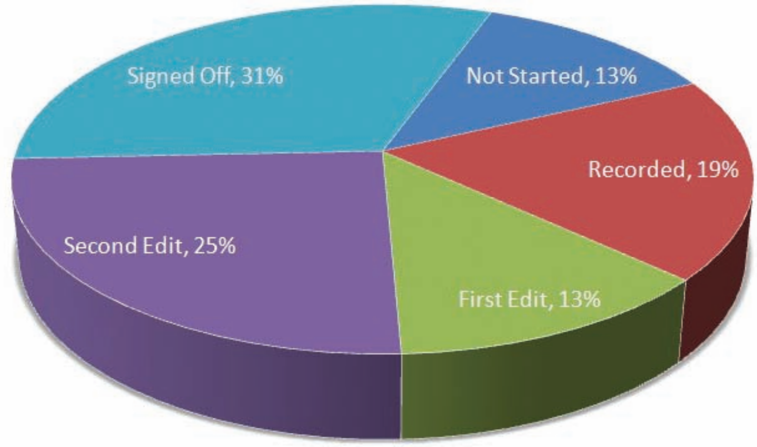


Figure 3: Progress Chart

UPK STANDARD
APPROVED

Setting the standards

Before any development work commences, it is critical to establish a comprehensive and detailed set of development standards. There are a great many variables in the way that the UPK Developer environment can be configured and, if advance thought is not put into the unambiguous definition of standards, the result will inevitably be that significant development time is wasted reworking topics when problems are encountered part way through development. We have seen many situations where development has proceeded without sufficient attention to the definition of standards, resulting in content that is full of inconsistencies that are off-putting and distracting to the user. The work required to correct this situation is substantial, expensive and generally will not fit within the project plan.

There are literally dozens of settings and development considerations that should be agreed before development commences. It is important that instructions to the user are given in a consistent style and language, otherwise the output will be confusing, distracting and lack the professionalism that is needed to get the positive adoption of UPK and the Application that is desired.

The following are examples of simpler and more obvious points that need to be addressed in the definition of development standards:

- Screen Resolution at which the recordings will be made
- Look and feel of the content including fonts, sizes, and colours
- Branding with the organisations logo and standards
- Frame delays in See-It mode
- Pass score for Know-it mode

Other more advanced development standards are no less significant in terms of their effect on the usability of the content.

Having developed a standards document and created sample topics, we recommend that all outputs are reviewed against these standards to get feedback from business process owners and potential users of the content. This will help establish if all the configuration choices and styles will indeed come together to create an end result that meets the needs of the business and the users.

Consideration should be given to how the UPK Content will be deployed. Will Know-it mode be used to assess users' competence, just to allow them to assure themselves that they know the processes, or not at all? Will UPK be available in Do-it mode and be linked into the application using Help Menu Integration (HEMI) or SmartHelp? Will Usage Tracking be implemented? We will be dealing with all these questions in subsequent articles; watch this space.

About the Author

Peter McClintock is Director of Services for Larmer Brown Consulting, UPK Training and Project Specialists.
He has worked in the Projects, Consulting and Training business for over 20 years.

He can be contacted on services@larmerbrown.com

PL/SQL Code: Auto-generation and Obfuscation

By Lakshman Bulusu, Genex Technologies Inc., USA & Bulusu Rama, Orissa Engineering College, Bhubaneswar, India.

This article discusses, from an application developer's perspective, two primary aspects of the PL/SQL language: code auto-generation and code obfuscation. The techniques introduced include auto-generating static and dynamically executable PL/SQL code and hiding the source code of PL/SQL programs, both statically and dynamically generated. The techniques described here pertain to versions 10g and 11g of Oracle. Wherever possible, code listings/segments are provided for implementing the concepts.

1. Introduction

Auto-generation and hiding of source code are two often-wanted features that automate the code generation process and hide the source-code so-generated from clients, external or unauthorised users, and third parties, respectively. PL/SQL through Oracle11g has a rich set of functionality that enables these two techniques to be implemented in the programming phase. The following sections address these two techniques in detail from an application development perspective, along with a brief explanation as to why such an implementation is required.

2. Auto-generating PL/SQL code using dynamic SQL and/or PL/SQL

This section makes use of native dynamic PL/SQL to demonstrate auto-code generation of static and dynamically executable PL/SQL. This is mostly required in situations where the code to be executed is to be centralised or decentralised based on dynamic parameters input at runtime or when the database objects referenced by the code are to be created dynamically based on these parameters.

One of the more common examples of auto-generating code is generating SQL scripts that perform functions such as granting privileges to or disabling all triggers on all the tables in a particular schema. This is achieved by concatenating the command text with the table name and spooling the output to a text file.

Here's an example of SELECT statement:

Figure 1

```
SQL> SELECT 'GRANT SELECT, INSERT, UPDATE ON
  '||table_name||' TO DVLP;'
 2 FROM  user_tables
 3 ORDER BY 1;
```

The output of the spooled file is as follows (see Figure 2):

Figure 2

```
GRANT SELECT, INSERT, UPDATE ON ODS_TAB1 TO DVLP;
GRANT SELECT, INSERT, UPDATE ON ODS_TAB2 TO DVLP;
GRANT SELECT, INSERT, UPDATE ON ODS_TAB3 TO DVLP;
GRANT SELECT, INSERT, UPDATE ON ODS_TAB4 TO DVLP;
GRANT SELECT, INSERT, UPDATE ON ODS_TAB5 TO DVLP;
```

This is just a simple example of how code can be auto-generated with the help of SQL. In fact, there is some primitive use of dynamic SQL in this because the table_name is specified as a non-hard-coded value. The table name can also be specified as a substitution variable or as a parameter at run-time. This dynamism can be extended to auto-generate code ranging from PL/SQL blocks and stored procedure definitions to SQL and PL/SQL that can be executed dynamically. This means that the EXECUTE IMMEDIATE statements can also be auto-generated. The code for enclosing the PL/SQL block or the subprogram can also be auto-generated.

The techniques for auto-generating static and dynamic PL/SQL code are discussed in the subsections that follow.

Consider an application that involves different regions of a country and contains a list of a multiple users, each having a set of tables pertaining to a particular region. The application also contains a global region table that has records of all the regions. The individual region tables have to be populated from the global region table. There are two scenarios here:

[*] A global user wants to populate individual region tables created in a centralised schema.

[*] Each of the individual users wants to populate his or her own region table from the global region table.

This section makes use of native dynamic PL/SQL to demonstrate auto-code generation of static and dynamic PL/SQL for implementation of each of these scenarios, although it can be accomplished with other methods.

Auto-Generating Static PL/SQL Code

The first scenario involves generating auto-code for creating as many single procedures as there are regions, but all in the centralized schema. This makes use of static PL/SQL code. Listing 1 on the left shows the code listing for this process, (see Figure 3).

Figure 3

```
CREATE OR REPLACE PROCEDURE auto_gen_static_plsql
              (ip_region_name VARCHAR2)
AUTHID CURRENT_USER
IS
 v_dyn_string VARCHAR2(32767);
BEGIN
 v_dyn_string :=
 'CREATE OR REPLACE PROCEDURE p_ppl_region_'||ip_region_name||'
  IS
   TYPE t_region_names IS TABLE OF
     global_region_tab.region_name%TYPE;
     global_region_names t_region_names;
  BEGIN
   SELECT region_name BULK COLLECT INTO
     global_region_names FROM global_region_tab;
   FORALL i global_region_names.FIRST..global_region_names.LaST
      INSERT INTO region_'||ip_region_name||'_tab
      SELECT * FROM global_region_tab
        WHERE region_name = global_region_names(i);
   COMMIT;
 EXCEPTION WHEN OTHERS THEN
   RAISE_APPLICATION_ERROR(-20001,
     ''Error in creating proc for region
     '||ip_region_name||': ''||SQLERRM);
 END p_ppl_region_'||ip_region_name||';'
 ;
--  EXECUTE IMMEDIATE v_dyn_string;
 DBMS_OUTPUT.PUT_LINE(v_dyn_string);
 DBMS_OUTPUT.NEW_LINE;
EXCEPTION WHEN OTHERS THEN
 RAISE_APPLICATION_ERROR(-20002,
   'Error in proc auto_gen_static_sql: '||SQLERRM);
END auto_gen_static_plsql;
/
```

Listing 1 Program to auto-generate static PL/SQL code

On the left is the output of executing the procedure in Listing 1, (see Figure 4).

Figure 4

```
SQL> exec auto_gen_static_plsql('EAST');

CREATE OR REPLACE PROCEDURE p_ppl_region_EAST
IS
 TYPE t_region_names IS TABLE OF
   global_region_tab.region_name%TYPE;

 global_region_names t_region_names;
BEGIN
 SELECT region_name BULK COLLECT INTO
   global_region_names FROM global_region_tab;

 FORALL i global_region_names.FIRST..
   global_region_names.LaST
  INSERT INTO region_EAST_tab
   SELECT * FROM global_region_tab
   WHERE region_name = global_region_names(i);
  COMMIT;
EXCEPTION WHEN OTHERS THEN
 RAISE_APPLICATION_ERROR(-20001, 'Error in
   creating proc for region EAST: '||SQLERRM);
END p_ppl_region_EAST;
```

Notice how the code for the procedure p_ppl_region_EAST is auto-generated and it contains static SQL to be executed. The procedure auto_gen_static_plsql calls an enclosing PL/SQL block that takes the region names from an outer implicit cursor loop. Here's the code to demonstrate this, (see Figure 5).

Figure 5

```
BEGIN
 FOR idx IN (SELECT region_name FROM global_region_tab) LOOP
  auto_gen_static_plsql(idx.region_name);
 END LOOP;
END;
/
```

Auto-Generating Dynamic PL/SQL Code

The second scenario involves creating one procedure for each individual user in his or her schema that uses his or her version of the region table. This involves creating these procedures with invoker rights. This example also uses native dynamic SQL to execute the INSERT for populating the region-specific table.

Now let us convert the auto_gen_static_plsql procedure to auto-generate PL/SQL that is dynamically executed. Listing 2 shows the code for this process, (see Figure 6).

Listing 2 Program to auto-generate dynamically executable PL/SQL code

Figure 6

```
CREATE OR REPLACE PROCEDURE auto_gen_dynamic_plsql
            (ip_region_name VARCHAR2)
AUTHID CURRENT_USER
IS
 v_dyn_string VARCHAR2(32767);
BEGIN
 v_dyn_string :=
 'CREATE OR REPLACE PROCEDURE p_ppl_region_'||ip_region_name||'
 AUTHID CURRENT_USER
  IS
   TYPE t_region_names IS TABLE OF
    global_region_tab.region_name%TYPE;
    global_region_names t_region_names;
  BEGIN
   SELECT region_name BULK COLLECT INTO
    global_region_names FROM global_region_tab;
   FORALL i global_region_names.FIRST..global_region_names.LaST
    EXECUTE IMMEDIATE
      "INSERT INTO region_'||ip_region_name||'_tab
        SELECT * FROM global_region_tab WHERE
          region_name = :1" USING global_region_names(i);
   COMMIT;
 EXCEPTION WHEN OTHERS THEN
   RAISE_APPLICATION_ERROR(-20001, "Error in creating
     proc for region '||ip_region_name||': "||SQLERRM);
 END p_ppl_region_'||ip_region_name||';'
 ;
--  EXECUTE IMMEDIATE v_dyn_string;
 DBMS_OUTPUT.PUT_LINE(v_dyn_string);
 DBMS_OUTPUT.NEW_LINE;
EXCEPTION WHEN OTHERS THEN
 RAISE_APPLICATION_ERROR(-20002, 'Error in proc
     auto_gen_dynamic_sql: '||SQLERRM);
END auto_gen_dynamic_plsql;
/
```

On the left is the output of executing the procedure shown in Listing 2, (see Figure 7).

Figure 7

```
SQL> exec auto_gen_dynamic_plsql('EAST');

CREATE OR REPLACE PROCEDURE p_ppl_region_EAST
AUTHID CURRENT_USER
IS
  TYPE t_region_names IS TABLE OF
    global_region_tab.region_name%TYPE;
global_region_names t_region_names;
BEGIN
  SELECT region_name BULK COLLECT INTO global_region_names
     FROM global_region_tab;
 FORALL i global_region_names.FIRST..global_region_names.LaST
    EXECUTE IMMEDIATE
      'INSERT INTO region_EAST_tab
         SELECT * FROM global_region_tab
           WHERE region_name = :1'
           USING global_region_names(i);
  COMMIT;
```

The following points are worth mentioning regarding the dynamic version of the auto-generated code:

[*] The dynamic PL/SQL code generated by using the EXECUTE IMMEDIATE statement made use of bind variables in the INSERT statement to pass the value of region name to it. However, the values for generating the table name and procedure name made use of simple concatenation.

[*] The auto-generated procedure in this case is defined as an invoker rights procedure.

These are very good practices in coding dynamic SQL statements using NDS.

Also, the parent procedures auto_gen_static_plsql and auto_gen_dynamic_plsql are both defined as invoker rights procedures. This, again, conforms to the two criteria for using NDS, namely, using bind variables for data and using invoker rights for the enclosing procedure.

```
 EXCEPTION WHEN OTHERS THEN
  RAISE_APPLICATION_ERROR(-20001, 'Error in creating
     proc for region EAST: '||SQLERRM);
END p_ppl_region_EAST;

PL/SQL procedure successfully completed.
```

Figure 7 continued

3. Hiding Statically Written and Dynamically Generated PL/SQL Code

This process deals with making the source code unavailable to clients, customers, or third parties. Up until Oracle 10g, there was only one way of hiding such code using a PL/SQL binary version of the source code. As of Oracle 10g, a new technique of obfuscating source code that is dynamically generated has been introduced. This section describes both of these techniques.

Wrapping the PL/SQL Source File to a PL/SQL Binary File

This process enables PL/SQL code in a source file to be wrapped into a binary version of the containing source code. The output produced is a file with .plb extension. This is done with the help of the WRAP utility. Consider the following example.

Assume the file Hide1.sql contains the following source code, (see Figure 8).

Figure 8

```
CREATE OR REPLACE PROCEDURE p_hide1
IS
 TYPE t_dept_arr IS TABLE OF dept%ROWTYPE;
 depts_arr t_dept_arr;
BEGIN
 SELECT * BULK COLLECT INTO depts_arr FROM dept;
EXCEPTION WHEN OTHERS THEN
  DBMS_OUTPUT.PUT_LINE(SQLERRM);
END p_hide1;
/
```

This file can be wrapped using the following command (executed from the OS prompt), (see Figure 9).

Figure 9

```
WRAP INAME=hide1.sql
```

It produces a corresponding hide1.plb file whose contents can be viewed as text and appear as on the left, (see Figure 10).

Figure 10

```
CREATE OR REPLACE PROCEDURE p_hide1 wrapped
a000000
b2
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
7
db f7
40vjfoxGcVEFFVjlZmIzro1zri4wgy7wLcsVfHSiWPiUh8ip+
   ybH6BtS3qKrW8bzfAH7vd/5
qhXN4dYBTxC7blh2kpfnTzMjJEEQH/Jbyv5fytcLp998hBbK
   1vDnIRXCaGVMH9DMu1HZ/JL7
6AMwnqFkRO97Nx20q3KPFuJ1FGskuJHs6t57NK3NN9NxSVlnZ
   agex5AqdDSQDIzFvpfMCpXI
uOrR81jB7RX6hEhepbQT1x/qMntM

/
```

As per above results, the source code is unreadable and hence unavailable. The hide1.plb file can then be run in a schema to create the p_hide1 procedure that can be called like a normal standalone procedure.

The text column in user_source view contains the wrapped binary version of the code instead of the original source code. On the left is a query to illustrate this, (see Figure 11).

Figure 11

```
SQL> select * from user_source where name = 'P_HIDE1';

NAME            TYPE       LINE
---------------------------- ------------ ----------
TEXT
------------------------------------------------------
P_HIDE1          PROCEDURE       1
PROCEDURE p_hide1 wrapped
a000000
b2
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
7
db f7
40vjfoxGcVEFFVjlZmIzro1zri4wgy7wLcsVfHSiWPiUh8ip+
  ybH6BtS3qKrW8bzfAH7vd/5
qhXN4dYBTxC7blh2kpfnTzMjJEEQH/Jbyv5fytcLp998hBbK
  1vDnIRXCaGVMH9DMu1HZ/JL7
6AMwnqFkRO97Nx20q3KPFuJlFGskuJHs6t57NK3NN9NxSVlnZ
  agex5AqdDSQDIzFvpfMCpXI
uOrR81jB7RX6hEhepbQT1x/qMntM
```

Now let's wrap a procedure that contains dynamically generated PL/SQL code.

Here's the source for this, (see Figure 12).

Figure 12

```
create or replace procedure p_hide2
authid current_user
is
v_str varchar2(2000);
begin
v_str := '
create or replace procedure p2
authid current_user
is
begin
execute immediate ''insert into dept (deptno) values (1)'';
commit;
end p2;';
execute immediate v_str;
end p_hide2;
/
```

Figure 13

```
WRAP INAME=hide2.sql
```

Figure 14

```
create or replace procedure p_hide2 wrapped
a000000
b2
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
7
107 144
ifX8NkiIAJuASiVOSm/qbccvxOgwg2zQfyBGfHQC2vjqMJl+
  IQvMW57MlmqzXqleeKK2CvD1
No7x5SSdvJ8tQeTk5DEfTjzlSGWe/jrZ2vFQWYahv7Tc1E0
  /brA1iFAd4QYyHGDWsdSgsNSF
5itcXDylUYcyKLUZ64KeS/enolPpX6CabwxTxtfx2feNRmx
  JOWvXfwlaxyIWrCSP1iIQcOdz
yNd2/h7y/5ofve4dKGzSqHl9Ea5UaPa08uSHSFjLVhwMqC
  fcBXlMEOU3rAsr1wCB4oPew01y
Fz0pQD2aJn7SpIfJP1z6ku4fmbCuGg==

/
```

Wrapping it using the command on the left, (see Figure 13), produces the binary code on the left, (see Figure 14).

Wrapped code is parsed by the PL/SQL compiler; however, syntax and semantic error checking is deferred until the .plb file is run. It is a good practice to leave package and object type specifications unwrapped so that the source code definitions contained in them are accessible to other PL/SQL programs (or external programs).

The wrapping of source code works in the following instances:

[*] The source code is in an OS file and is not stored in the data dictionary.

[*] The source code is not an anonymous PL/SQL block or trigger code.

[*] The project-specific requirements are met with regard to source-code availability given the fact that the translated binary version is not query-able using the data dictionary views DBA_SOURCE, ALL_SOURCE, and USER_SOURCE to get the source code. This means that the wrapped procedure code is stored in its binary version in these data dictionary views. An exception to this is dynamically generated source code.

To illustrate the second point, let's execute the file hide2.plb and query the USER_SOURCE view for the TEXT column for the P_HIDE2 and P2 procedures. Note that P_HIDE2 is created using the wrapped version of the code, whereas the source code for P2 is generated dynamically in the body of P_HIDE2, (see Figure 15).

As is evident, the source code for P_HIDE2 is stored as a binary version, whereas the source code for P2 is stored as the original source code.

To avoid this limitation, PL/SQL 10g introduced a new technique that enables storing of dynamically generated code in wrapped binary form. The next subsection discusses this technique.

Figure 15

```
SQL> @hide2.plb

Procedure created.

SQL> exec p_hide2

PL/SQL procedure successfully completed.

SQL> select text from user_source where name = 'P_HIDE2';

TEXT
----------------------------------------------------
procedure p_hide2 wrapped

a000000
b2
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
SQL> @hide2.plb

Procedure created.

SQL> exec p_hide2

PL/SQL procedure successfully completed.

SQL> select text from user_source where name = 'P_HIDE2';

TEXT
----------------------------------------------------
procedure p_hide2 wrapped

a000000
b2
abcd
abcd
abcd
abcd
abcd
abcd
abcd
7
107 144
ifX8NkiIAJuASiVOSm/qbccvxOgwg2zQfyBGfHQC2vj
   qMJl+IQvMW57MlmqzXqleeKK2CvD1
No7x5SSdvJ8tQeTk5DEfTjzlSGWe/jrZ2vFQWYahv7
   Tc1E0/brA1iFAd4QYyHGDWsdSgsNSF
5itcXDylUYcyKLUZ64KeS/enolPpX6CabwxTxtfx2fe
   NRmxJOWvXfwlaxyIWrCSP1iIQcOdz
yNd2/h7y/5ofve4dKGzSqHl9Ea5UaPa08uSHSFjLVhw
   MqCfcBXlMEOU3rAsr1wCB4oPew01y
Fz0pQD2aJn7SpIfJP1z6ku4fmbCuGg==

SQL> select text from user_source where name = 'P2';

TEXT
--------------------------------------------------
procedure p2
authid current_user
is
begin
 execute immediate 'insert into dept (deptno) values (1)';
 commit;
end p2;
7 rows selected.
```

Obfuscating Dynamically Generated PL/SQL Source Code

Hide dynamically generated PL/SQL source code using two packaged subprograms[md]DBMS_DDL.WRAP (a function) and DBMS_DDL.CREATE_WRAPPED. These are new in Oracle 10g.

First, assign the dynamically generated source code to a string variable.

Then use the DBMS_DDL.WRAP function to return wrapped binary PL/SQL code with the assigned variable as the actual parameter value. This step does not execute the so-generated binary code.

Listing 3 on the left shows the code, (see Figure 16).

Figure 16

```
create or replace procedure p_hide2_new
authid current_user
is
 v_str varchar2(2000);
begin
v_str := '
create or replace procedure p2_new
authid current_user
is
begin
 execute immediate ''insert into dept (deptno) values (1)'';
 commit;
end p2_new;';
dbms_output.put_line(SYS.DBMS_DDL.WRAP(v_str));
end p_hide2_new;
/

SQL> exec p_hide2_new

create or replace procedure p2_new wrapped

a000000
b2
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abc
d
abcd
7
80
c6
uUDiJQYgDRXOzqx8Krn/JQ5PP5cwg5TZLcusfI5gOME
   ilVWpzDzctPmz2qDIJWDE1NvLFOgv
JqWS
BsCbYhe47azLy6Cgf7BM9a+aVLee/6+SBGBXr4xTj5crL+
   77lHrj8DeLuWkqQyoQni6+
R9LyDGuSmnp
4oACamqAJEEjFQjTleEDtjkvsQN+jaEMpW2n7FBc=

PL/SQL procedure successfully completed.
```

Listing 3 Program to show the use of DBMS_DDL.WRAP

Note that the source code for P2_NEW that is being dynamically generated is wrapped but not executed. This can be verified as follows: (See Figure 17).

Figure 17

```
SQL> select text from user_source where name = 'P2_NEW';
no rows selected
```

Use the DBMS_DDL.CREATE_WRAPPED function to wrap the PL/SQL code being dynamically generated, and then execute it. This step stores the dynamically generated source code in wrapped form in the data dictionary.

Listing 4 shows the code for this process. (See Figure 18).

Figure 18

```
create or replace procedure p_hide2_new2
authid current_user
is
 v_str varchar2(2000);
begin
v_str := '
create or replace procedure p2_new
authid current_user
is
begin
 execute immediate "insert into dept (deptno) values (1)";
 commit;
end p2_new;';
SYS.DBMS_DDL.CREATE_WRAPPED(v_str);
end p_hide2_new2;
/

SQL> exec p_hide2_new2

PL/SQL procedure successfully completed.

SQL> select text from user_source where name = 'P_HIDE2_NEW2';

TEXT
-------------------------------------------------------
procedure p_hide2_new2
authid current_user
is
 v_str varchar2(2000);
begin
v_str := '
create or replace procedure p2_new
authid current_user
is
begin
 execute immediate "insert into dept (deptno) values (1)";
 commit;
end p2_new;';
SYS.DBMS_DDL.CREATE_WRAPPED(v_str);
end p_hide2_new2;

15 rows selected.

SQL> select text from user_source where name = 'P2_NEW';
TEXT
-------------------------------------------------------
procedure p2_new wrapped
a000000
b2
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
```

```
abcd
abcd
abcd
7
80 c6
uUDiJQYgDRXOzqx8Krn/JQ5PP5cwg5TZLcusfI5gOMEil
  VWpzDzctPmz2qDIJWDE1NvLFOgv
JqWSBsCbYhe47azLy6Cgf7BM9a+aVLee/6+SBGBXr4xT
  j5crL+77lHrj8DeLuWkqQyoQni6+
R9LyDGuSmnp4oACamqAJEEjFQjTleEDtjkvsQN+jaEMpW2n7FBc=
```

Figure 18 continued

Figure 19

```
create or replace procedure p_hide3_new
authid current_user
is
 v_str varchar2(2000);
begin
v_str := '
create or replace procedure p3_new
authid current_user
is
begin
 execute immediate "insert into dept (deptno) values (1)";
 commit;
end p3_new;';
SYS.DBMS_DDL.CREATE_WRAPPED(v_str);
end p_hide3_new;
/
This can be wrapped using the following command:
WRAP INAME=hide3_new.sql

to produce the binary hide3_new.plb containing the wrapped version

create or replace procedure p_hide3_new wrapped
a000000
b2
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
7
122 15c
AL37zYzF7kdr53Spxx6QhyJBWFcwg2LQNSDWfHQCv/9
  kpbJTw6Dtss1n9A1meXRsUCew40QK
bcPhnvphT531VZXC5Ir+Og0GO+YZmUbpEOz3M5cle
  NX4uVTHTSOoSynDlsxLo+EdVz4xvVeN
UXun4Jy5m3pynp/dwUU4v80IvdPopLJBcJ+PEb7
  L7HBSA+h/U7RViISl8+d7V4A8CkvxC8kB
LPV/36x9B/Xjb8nsESF4AxR/39FCfDxyfCUGqO1
  c2helKnEdO8hIAxQ+Bg1BsPrOrPektULp
btm/vu4H6iZXqjHD4dnLPZOJeW16cxIiYK9AhArgI2gKOXKVapYFjpMq
```

Listing 4 Program to show the use of DBMS_DDL.CREATE_WRAPPED

The following points are worth noting:

[*] Neither the DBMS_DDL.WRAP function nor the DBMS_DDL.CREATE_WRAPPED procedure wraps the source code of the enclosing procedure in which they are called. They wrap only the dynamically
generated code contained in the enclosing procedure. The DBMS_DDL.CREATE_WRAPPED procedure,
in addition to wrapping also executes the dynamically generated code, so that is it stored in wrapped binary form inside the data dictionary.

[*] The call to SYS.DBMS_DDL.WRAP function can be executed using native dynamic SQL (with the EXECUTE EIMMEDIATE statement) to simulate the functionality of DBMS_DDL.CREATE_WRAPPED.

Using the WRAP Utility in Conjunction with DBMS_DDL.CREATE_WRAPPED

The WRAP utility can be combined with the DBMS_DDL.CREATE_WRAPPED procedure to wrap the dynamically generated source code as well as its enclosing procedure source code together and store them in the same form in the data dictionary.

Consider the file hide3_new.sql containing the following source code, (See Figure 19).

```
SQL> select text from user_source where name = 'P3_NEW';

TEXT
-----------------------------------------------------
procedure p3_new wrapped
a000000
b2
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
abcd
7
80 c6
9nBpGubd7GdZ3LZ2Q0vCUDlPv8owg5TZLcusfI5gOMEilV
  Wps3yfFTPQGWsCid9FKhufyn8C
WgKxH5y3IK3WHpT1uA5j8K3NUZn51OzDuMs8xnwVvRyVRO
  phrOcoTY81fnQhvwflPmQpXhGX
Aejqo549tYu+a806+eiN/Mde+oZc7X+KY6k9IaCzBtGuJGkQFN4=
```

Figure 19 continued

This way, both the statically and dynamically generated code are stored in wrapped form in the data dictionary.

Summary

This article described techniques in PL/SQL relating to auto-generating code using dynamic SQL and PL/SQL, and hiding statically written and dynamically generated PL/SQL code. The concepts are illustrated with code segments to help developers implement them.

About the Authors

Lakshman Bulusu is the author of *Oracle PL/SQL: Expert Techniques for Developers and Databased Adminstrators* (from Charles River Media/ Course PTR), *Oracle9i PL/SQL: A Developer's Guide* (from Apress), *Oracle and Java Development* and *Oracle Developer Forms Techniques* from SAMS publishing. His book *Oracle and Java Development* has been translated into Japanese and Polish. He holds an Oracle Masters credential from Oracle Corporation, is an OCP-Certified Application Developer, and is a double Honors graduate in Computer Science & Engineering and Mathematics. He has more than fifteen years experience in technical management, mentoring, design, development and implementation of Oracle projects both in the client/server and Web environments. He has presented papers at numerous national and international conferences and also contributed to lead technical journals in the U.S. and U.K.

He can be reached at lbulusu@genex-info.com or balakshman@yahoo.com

Bulusu Rama is a Senior Faculty member having rich academic experience in mentoring, teaching and guiding students in Computer Science discipline with special reference to Database Systems, Computer Graphics and Data Structures.
He has participated/guided candidates in Oracle-related projects with special focus on PL/SQL Programming and implementation in client-server and Web environments. He holds a Masters of Technology in Computer Science, a Masters of Science in Mathematics and is also a double Honors graduate in Computer Science & Engineering and Mathematics.
He has presented papers at various national conferences and contributed previously to Oracle Scene and also to technical journals in India.

He can be reached at rama_bulusu@yahoo.com

Top Tips Part Two: Good Performance, Bad Performance and Statistics level ALL

By Tim Onions TOdC Limited

So, now to the main thrust of this article. Not so much a tip as a warning and not so much a hard-and-fast "do this and your life will be so much better" as some knowledge to have in the back of your mind should certain events ever occur.

From time to time, I read posts on the main Oracle discussion forums, the gist of which are as follows:

> *I set* STATISTICS_LEVEL=All *from its default setting of* TYPICAL *at the database level on my 9.2/10.0/... database. Shortly after making the change, CPU utilisation reached 100% and stayed there! I put the* STATISTICS_LEVEL *parameter back to* TYPICAL *and CPU returned to its normal level.*

There are not that many such posts I grant you but it mimics precisely my own experiences with the parameter. I did a bit of research to see if there was anything out there on the web that gave a hard and fast rule on the overhead of upping the statistics level from ALL to TYPICAL and came across a couple of articles from reliable sources that suggested the impact was, in general, pretty negligible. The articles included test cases and TKPROF output and were highly credible. None suggested any significant overhead (perhaps ~5%) – nothing that would account for a previously well tuned and well-behaved system going into CPU meltdown the moment the statistics level was changed.

Then I hit on a posting from Jonathon Lewis:

http://jonathanlewis.wordpress.com/2007/04/26/heisenberg/

Although the topic under discussion was not STATISTICS_LEVEL, the diagnosis of the problem at hand did involve it. A 300% degradation in performance of a certain query was witnessed when ALL was used (over that observed when TYPICAL was used). That, taken with my own observations (in some real world, very painful situations) plus the occasional posts from others leads me to the inevitable conclusion that something I would class (loosely) as "bad" is going on – albeit in particular situations – with the ALL setting of STATISTICS_LEVEL.

The most striking of incidents I have encountered with this came about shortly after migrating a system from 9.2.0.7 to 10.2.0.3. All the usual due diligence had been done and many hours of tedious and exhausting regression testing had been completed successfully. Yet, the moment the application was switched back on after the upgrade the CPU pegged at 100%. Statistics were up-to-date; workload was normal; data volumes clearly had not changed. Execution plans of the key SQL statements were identical. However, after closer inspection, it could be seen that those key SQL statements, although using the same execution paths as before, were using up to 300% more CPU on 10.2.0.3 as they had on 9.2.0.7! [total statement CPU consumption was obtained from the V$SQL view and averaged over the number of executions].

Take a look at the following stats – which show V$SQL data for the 9.2.0.7 system and corresponding data for the 10.2.0.3 system shortly after it went live; for the exact same statement using the same execution plan. Sure, you would expect some difference but not anything as dramatic as shown here (small differences in the rows retrieved are attributable to the fact that the system was live and users were changing data to some small degree). The same trend was seen over and over again on other SQL.

```
9.2.0.7:  Execs   CPU       CPU/Ex    ROWS    ROWS/Ex
          -----   -------   -------   -----   -------
          2       265,625   132,813   7,904   3,952

10.2.0.3: Execs   CPU       CPU/Ex    ROWS    ROWS/Ex
          -----   -------   -------   -----   -------
          2       843,421   421,711   7,882   3,941
```

Now see the revised stats – this shows V$SQL data for the exact same statement for the 10.2.0.3 system when STATISTICS_LEVEL was set back to TYPICAL (thankfully it is a dynamic parameter in 10.2.0.3 and can be changed via a simple ALTER SYSTEM command). The average CPU is now less than the original 9.2.0.7 version of the system even though more data is being retrieved (as a result of backlogs in the application resulting from the overall slowdown) and the execution plan remained the same:

```
10.2.0.3: Execs   CPU       CPU/Ex    ROWS    ROWS/Ex
          -----   -------   -------   -----   -------
          2       244,969   122,485   8,775   4,388
```

So what in heck can be going on here? Unfortunately, I do not have a hard and fast answer for you (my intention here is to bring it to your attention so that should your system's performance suddenly take a huge performance hit, with CPU going berserk, you can quickly check the setting of STATISTICS_LEVEL). However, I collected together a few ideas from the postings I have read:

- *The overhead may be more platform dependent than anything else*
- *The nature of the execution plan may make a big difference to the overhead*
- *You get additional execution statistics for all parts of the query resulting in a very large amount of internal "work"*
- *On a platform where the system timer call is expensive, the CPU increase can be significant*
- *The more work each SQL does, the greater the impact of ALL*

To sum up:– Dependent on the amount of internal work a particular SQL is doing – irrespective of how quickly it returns data – if it does a huge amount of joins, hashes, nested loops, filters etc to get the resulting nn rows each of those internal "steps" results in extra work internally to capture execution statistics and timings. In particular, if there are such SQL statements and they are run very frequently and concurrently this will exacerbate the impact (perhaps at application start-up as one poster suggested and one set of my own tests indicated).

Tim Onions is an independent database consultant with over 15 years' experience with Oracle databases. Tim specialises in the application and database design of high performance systems, as well as tuning and optimisation techniques and can be reached at Tim.Onions@TOdC.co.uk

Disclaimer: You must always check the hints, tips and scripts presented in this paper before using them and always try them out on a test database before running against a live system. Whilst every care has been taken to ensure the examples given function properly and are totally unobtrusive and benign (when used properly), neither the authors nor the UKOUG can take any responsibility or liability for what effect they have when you use them.

Top Tips: Data Cluster

By Jonathan Lewis, JL Computer Consultancy

To a very large extent, good performance is dependent on loading your data into the right place, indexing it properly, and then making sure that the optimiser can see what you've done by giving it some good statistics. This article is a discussion of the basic issues, triggered by a question about the unexpected performance of a particular query.

I received an email some time ago containing the following question:

*For packaged applications, like the Oracle EBS, some indexes tend to develop a high **Clustering Factor** over time, like the one shown below (Figure 1):*

Figure 1

INDEX NAME	BTREE LEVEL	LEAF BLOCKS	DISTINCT KEYS	CLUSTERING FACTOR	INDEX NUM ROWS	TABLE NUM BLKS	TABLE NUM ROWS
XXX_PK	3	778150	77,842,100	17,163,350	77,842,100	865,805	77,043,200

We have some queries that perform range scans on this index and run slower because they have to visit too many blocks from the table. How should one deal with this type of index?

This question raises a couple of important points that are worth reviewing.

As a starting detail: did you notice that the number of *"rows"* in the index was greater than the number of rows in the table. This can happen when you set a sample size less than 100% to gather table stats, and use the "cascade" option. Oracle often uses a larger sample on the index than the one specified for the table, with the type of results that you see above.

This isn't a big deal – unless the ***clustering_factor*** also exceeds the number of rows in the table, at which point you may find that queries that should "obviously" be using the primary key index start using a slightly less desirable non-unique index. This is a side-effect of the cost calculation which uses the expression ***selectivity[tb_sel] * clustering_factor*** when assessing the suitability of the index.

If your primary key ***clustering_factor*** is larger than the number of rows in the table then, for a single column index, it will also be greater than the number of distinct values for the column, and the expression ***selectivity[tb_sel] * clustering_factor*** will be greater than one for the predicate *primary_key_col = constant*. If this happens then another index may appear to have a lower cost than the primary key index.

Coming back to the original question though –- what do you do when you have an index where the ***clustering_factor*** keeps growing? This question prompts an even more important question: is the ***clustering_value*** telling the truth? The clustering factor is about the pattern of data scatter. There is the real scatter pattern, the pattern that the optimiser thinks exists, and there may even be a pattern that you want to fool the optimiser into seeing.

If your data is well clustered then the ***clustering_factor*** should be *"small"* and the optimiser has a higher probability of using the index. If the data is well clustered but, for some reason, the ***clustering_factor*** is *"large"* then the optimiser may well ignore the index when you know it would be sensible to use it.

If your data is very poorly clustered then the ***clustering_factor*** should be *"large"* and the optimser is likely to use the index only in special cases (such as unique scan, index-only, or bitmap conversion paths). If your data is very poorly clustered but the ***clustering_factor*** is *"small"* – possibly because you've hacked it – then the optimiser may well use the index when it shouldn't and perform an excessive amount of (logical) I/O.

In this example, the ***clustering_factor*** is quite large, and the user is complaining that queries using index range scans are running slowly because Oracle is visiting too many blocks from the table.

This means there probably isn't a problem with the ***clustering_factor*** – if you have to go to the table for the data, you have to go to the table for the data. Although the ***clustering factor*** may, in principle, be too small, the fact that we work harder and harder as time passes (i.e. as the data sizes grow) means the index has not been defined appropriately for the queries where it is being used. As a simple example to demonstrate this point, consider a query involving customer data. We always ask for data for a customer for the last 4 weeks. (Figure 2).

Figure 2

```
select    hst.*
from      customer_history hst
where     hst.customer_id = 12345
and       hst.tx_date > trunc(sysdate) - 28
order by
          hst.tx_date desc
;
```

If we have an index just on the *(customer_id)* column, then as time passes Oracle has to visit more and more table blocks, discarding data that is too old only after it has visited the table.

If we have an index on *(customer_id, tx_date)*, then the work done visiting the table will stay constant as time passes, because the volume of data we examine is independent of the size of the history we have generated.

Of course, if we can recreate the table in our example as an index-cluster on *(customer_id)*, we can minimise the number of excess blocks we visit (although we will still be visiting excessive numbers of rows) and this may be of some benefit. Alternatively, if we can partition the table by *(tx_date)*, our query will be able to use partition elimination to minimise the number of redundant table blocks we visit – provided the index on *(customer_id)* is a local index (although this may increase the number of index block visits to an undesirable level if we use a large number of partitions).

In conclusion: the ***clustering_factor*** can easily be too large and not reflect the true pattern of data scatter – this can make the optimiser ignore the index when it should be using it. But if the ***clustering_factor*** really does reflect the data scatter, and your queries get slower because they are visiting increasing numbers of blocks in the **table**, then you probably need extra columns in the index to allow you to eliminate some of the visits to the table.

Jonathan Lewis is a freelance consultant with nearly 20 years of experience with Oracle databases. He is one of the best-known speakers on the UK Oracle circuit, and is also popular on the international scene. He specialises in designing, optimising, and troubleshooting applications that make use of the Oracle database engine. Jonathan is the author of *'Cost Based Oracle – Fundamentals'* published by Apress, and *'Practical Oracle 8i – Designing Efficient Databases'* published by Addison-Wesley.

He edits two websites: jonathanlewis.wordpress.com, and www.jlcomp.demon.co.uk – which also hosts The Co-operative Oracle Users' FAQ for the Oracle-related Usenet newsgroups.

Content supplied by:
Global Oracle Contractors Network
ORACLE contractors.com

Attaining Approval for a Services Oriented Architecture Project

By **Mark Jermolenko**

The promise of Services Oriented Architecture and the associated benefits of web services integration and process automation using the BPEL standards has the potential to deliver a great deal for an organisation. However, the hardest part as we all well know is achieving any sort of backing for a project to deliver these benefits. To many of us this is a "no brainer", but to many executives including General Managers and CIO's, the term SOA is still a mysterious black hole and unknown to many.

As Oracle and other vendors propose, SOA is a journey. It is a journey that must be undertaken one step at a time. Oracle's SOA maturity model suggests that the first step is on quick win projects and then the organisation progresses onwards along the maturity path until it becomes a fully integrated part of the culture of an organisation. Further, the SOA implementation model also suggests a cyclical approach to implementation including the following steps:

- Model and Capture
- Integrate and Orchestrate
- Develop
- Deploy and Manage
- Govern
- Monitor

Each of the above process steps presents its own challenges for the organisation. Does the first project for quick wins really need to address all of the above steps?

Having recently undertaken a project to implement a SOA proof of concept, I'd like to share some of our learnings for others who may be contemplating a similar journey.

- Create a vision through the business case – spend time educating the Management team of the benefits that can be attributed from the implementation of a SOA based solution and the fact that it's a journey not just a single project.

- Identify a quick win pain point – select a process that is not too visible to the business but will prove all the elements of a SOA solution. We selected an automated form for application access to Oracle Financials with integration into Oracle, Active Directory and HP Openview

- Cover off the develop and deploy and manage parts first – focus on delivering a working solution rather than implementing the entire cycle.

- Position the first project as a proof of concept – allow funding for an R & D type of delivery for experimentation and to manage the expectations that more projects will need to follow.

- Don't make it too complex – a simple and effective process will provide the concept of web services, BPEL and ESB more easlily that something that is highly technical and will take more time to deliver.

- Architect the solution to suit the organisation – a combination of Microsoft, Oracle and IBM technologies is entirely possible and can be used effectively to their advantages.

Above everything else, no matter what people tell you, the implmentation of a SOA solution is not easy and requires lots of initial pain to change the mind set of a number of people including executives, managers and IT staff. However, the benefits will outweight the pain.

Mark Jermolenko has over 10 years of Project Management and consulting experience in the area of Oracle E-Business, Oracle Middleware including SOA and other related technologies spanning government, financial services, manufacturing and retail sectors.

“Happy birthday to you, happy ...”

By Simon Corbett, Jargon PR

This year UK Oracle User Group (UKOUG) celebrates its 25th anniversary having played a key role as a partner to Oracle since it began. Simon Corbett, from Jargon PR, looks back at the past twenty five years and explores the development of UKOUG, its relationship with Oracle and the future for one of the world’s highest profile user groups.

Introduction

When UK Oracle User Group was first conceived in the early 1980’s, a number of Oracle employees met in a boutique hotel in Mayfair on a wet and windy afternoon to discuss how existing users could talk to each other in order to get the most out of the Oracle software they were using.

Little did these individuals realise that the group they had formed would become one of the most successful user groups on the planet encompassing over 1,600 corporate members, producing over 120 events per year with over 30 special interest groups – supporting customers from the world’s largest enterprise software company.

> “In order to get closer to our users, we must genuinely understand their needs.”

The early years

“In order to get closer to our users, we must genuinely understand their needs” comments Ronan Miles, UKOUG chairman, as he describes the rationale behind the formation of UKOUG twenty five years ago.

At this time Oracle was primarily focused on the design and development of a relational database. However, within the organisation there was an understanding of the business potential that lay within enterprise software and an awareness of users and the importance of them meeting to discuss the common issues they faced.

“I wasn’t there at the time” comments Miles, “but I understand the user group was designed to be run informally with the feel of a local club. Expectations were low and the group small and informal, yet the benefits to both the users and Oracle seemed so obvious, it was a path worth exploring.”

With financial and management support from Oracle, the user group was set up as a division within the company, supported by secretarial and administrative support from a small internal team. One of the early managers of the group was Paul Drake who worked as chairman while working full time for BP.

Throughout the next four years, the user group began to grow in popularity. Membership reached more than 100 members after the fourth year and a number of special interest groups had developed, designed to address specific issues facing Oracle users. “At this time, there was an acknowledgement from Oracle that the users should manage the group”, comments Miles, and in 1988 the user group became an independent company, limited by guarantee, with a board of directors, and the organisation formally took the name of UK Oracle User Group.

New products – new direction

With membership numbers increasing, one of the first challenges facing UKOUG was the management of its members. “The group had grown from nothing to over 100 members very quickly”, comments Miles, and in early 1989 membership was remodelled into personal and corporate membership, so organisations and individual users could obtain maximum benefit from UKOUG membership.

However, as the membership of UKOUG changed, so did Oracle and the product set the user group supported. Having begun as a technology company with a limited range of supporting development tools, Oracle acquired the database company RdB, significantly expanding its software portfolio.

“It was at this time that UKOUG made one of the major decisions that have contributed to the success enjoyed today”, Miles recalls. As Oracle made its second acquisition, Express, UKOUG set a precedent that the group would outreach to existing user communities to welcome them into the group.

“Contrary to some opinions, Oracle did not asset strip but continued to support existing users. In response to this we decided to support all Oracle users. Our rationale was simple – we’re all in this together”, comments Miles.

As Oracle continued to make a series of small to midsize acquisitions, UKOUG rapidly expanded as it supported each new user community with interest groups; designed specifically for the product set Oracle had acquired. However, it was not until 2005 that UKOUG encountered its biggest challenge to date.

“I recall hearing the news that PeopleSoft had acquired JD Edwards” comments Miles, “and then I heard that Oracle was to acquire PeopleSoft!” With both JD Edwards and PeopleSoft having their own user groups, the two companies had begun discussions into how its forthcoming conference would be managed. “Having heard the news that Oracle was now effectively acquiring both PeopleSoft and JD Edwards, I knew I was going to have a busy few weeks” exclaims Miles.

> “Having heard the news that Oracle was now effectively acquiring both PeopleSoft and JD Edwards, I knew I was going to have a busy few weeks...”

After several weeks hard work both the JD Edwards and PeopleSoft user group conferences were regarded as a success. Reflecting on what happened, Miles comments, “all acquisitions, whether small, medium or large sized

… UK Oracle User Group celebrates its 25th anniversary

companies, have been warmly welcomed into the UKOUG community, JD Edwards and PeopleSoft, although a little bigger than most, was no different."

Professor Dudley recalls the meeting, "When the PeopleSoft acquisition was announced we were all facing a huge challenge. After a lot of debate we concluded that we must do what was in the best interest of our members and the Oracle users." One of the core ideals behind UKOUG is to make the existing software work better. "Based on the idea that two heads are better than one, we decided to embrace all acquisitions in order to share expertise, experience and enthusiasm", comments Dudley.

"One of the most important messages to communicate to any new user group are the reciprocal benefits of UKOUG membership", comments Professor Dudley, "not only does UKOUG support new user communities, assisting them with their own tailored activity, but we provide them with the opportunity to learn, network and develop with our existing members throughout other special interest groups."

The partnership continues

Throughout the past 25 years, a great deal has changed in both Oracle and UKOUG, as Professor Dudley explains, "the relationship between Oracle and UKOUG has always been a partnership. However, throughout the past few years UKOUG has increasingly grown in importance to Oracle due to the impartial feedback and consultancy it can provide, gained firsthand from its members."

> "UKOUG was recently described as 'a user group other countries would do very well to copy'. "

The value UKOUG creates has certainly not gone unnoticed in Redwood Shores as UKOUG was recently described as 'a user group other countries would do very well to copy'. As well as being an example to other user groups in terms of its membership, the quantity and quality of the events run and the focus of its special interest groups, UKOUG has increasingly begun to influence the behaviour of Oracle from a corporate level, as Miles explains.

"The membership survey conducted in November 2006 showed that a small number of customers were concerned about the communication between PeopleSoft, JD Edwards and Oracle. As a result of this feedback UKOUG helped Oracle introduce 'communicating with Oracle' days that have proven highly successful", comments Miles.

Following the success of 'communicating with Oracle' days, both UKOUG and Oracle are continuing to combine resources for the development of the special interest groups. Miles comments, "having listened to the survey results, it's always gratifying to see we have made a difference when the 2007 results demonstrated an improvement in communication resulting in higher satisfaction levels."

In addition to the communication between PeopleSoft, JD Edwards and Oracle, UKOUG has increasingly helped Oracle regarding the Oracle Fusion strategy. "When Oracle first announced its strategy surrounding Fusion, it was met with a mixed response" comments Miles. "In response to this, UKOUG mobilised a team of 30 users to consult on the planned Fusion strategy. Since examining each area of the proposed plan, UKOUG has played a huge part in refining the strategy and contributing at the very highest level."

The journey is only just beginning ...

As UKOUG celebrates its 25th anniversary, Miles comments on the future of the organisation, "the only certainty is that there are no certainties", he says with a smile. With Oracle showing no signs of deviating from its acquisition strategy, it is likely that UKOUG will continue to be very busy, actively welcoming new communities to UKOUG as it expands the special interest groups, the events and its staff from over 20 employees and 170+ volunteers.

Professor Dudley comments, "Information technology has gone from an organisational cost to being one of the biggest drivers of value for any commercial organisation. As technology increasingly provides added value to both businesses and people, it has helped transform the UK into a knowledge economy. With Oracle being one of the main drivers of the UK economy due to its size and customer reach, I have no doubt UKOUG will remain a major player within the Oracle story."

UKOUG: Key Milestones

- **UK OUG events per year**

1990– approximately 25 events per year
1995– approximately 40 events per year
2005– approximately 50 events per year
2008– UKOUG runs over 120 events per year

- **UK OUG Special Interest Groups**

1990– 10 special interest groups
2000– 15 special interest groups
2005– 22 special interest groups
2008– 32 special interest groups

- **UK OUG Conference days & streams**

2002– 3 days – 10 streams
2005– 3 days – 15 streams
2007– 4 days – 20 streams

25 years of serving the Oracle Community

Simon Corbett With over ten years experience in technology public relations, Simon Corbett is the owner and founder of Jargon PR (www.jargonpr.com). The agency specialises in business-to-business and business-to-consumer technology public relations campaigns that raise awareness, communicate company USP's and deliver business value through the media.

Interview with Lawrence Clark – JD Edwards Director, Oracle

The JD Edwards (JDE) UKOUG committee had a planning meeting with UKOUG, Oracle and JDE Partners to look at how we can best serve the JDE user community moving forward into 2008.

On the basis of the feedback we have received after the UKOUG annual conference in December 2007, we have taken on board the comments and made some changes for 2008. Our objective is to engage with the JDE community as a whole and ensure that all companies receive the benefit of being part of a working User Group. We will be providing two special events on 8th May & 10th July and an annual JDE Conference to be held in mid-October purely for the JDE community that will be open for European attendance. For further details please see www.oug.org/jdedwards

An important part of moving forward was to take the opportunity to interview Lawrence Clark the JDE Director at Oracle, to clear up any confusion amongst the JDE community.

Oracle's acquisition of JD Edwards in 2005, as part of the PeopleSoft deal, was initially met with some scepticism from JD Edwards customers, especially around Oracle's commitment to supporting the JD Edwards product set moving forward.

We met with Lawrence Clark, Director of JD Edwards, Oracle UK, Ireland and Israel, to get a better understanding of how the company is going about reassuring this important group of customers.

Interview

Q *There is a feeling amongst some JD Edwards users in the UK that development plans will not be followed through. How are Oracle proving their commitment to the JD Edwards product?*

A "As has been said on a number of occasions by executives from Oracle HQ, we have a dedicated management and product development team looking after JD Edwards EnterpriseOne and JD Edwards World, who are responsible for the release of two major versions of the product last year, the JD Edwards World release being the first significant update for 10 years.

Additionally I would point to **Applications Unlimited** and **Oracle's Lifetime Support Policy**, which are designed to offer customers reassurance about their software investments. If they haven't done so already I would strongly advise customers to go and check out the links to understand more."

Q *How can customers provide feedback on JD Edwards product functionality and how it might help them moving forward?*

A "We have a number of mechanisms to solicit input from our customers as to future product functionality; these include feedback via Customer Advisory Boards (CAB's), Special Interest Groups (SIGs) run by UKOUG, as well as working closely with UKOUG to engage with our customers at User Group events being held throughout the year. My advice to our customers would be to join these communities."

Q *What more is Oracle intending to do to communicate to customers in the future?*

A "We are committed to ensuring that you receive the best possible service, and with this in mind last year we established the JD Edwards Guardian Programme, assigning all our JD Edwards customers an Oracle Guardian to help guide them around the company and get a better understanding of the services we offer. We're going to be talking about this at the 'Communicating with Oracle' day on 8th May at our Thames Valley Park site, so I would encourage everyone to come along to find out more."

Q *One area people have asked about is support for JD Edwards products, can you tell us what you're doing there?*

A "Firstly I would like to say that we are committed to support the JD Edwards products with our Lifetime Support policy, therefore extending all the support of the existing products beyond the current agreements.

If customers are not getting the quality of support they believe they should be getting, then in the first instance I would advise them to contact their Customer Guardian. If satisfaction is not forthcoming please contact me via the details listed below."

Q *How do JD Edwards users benefit from being part of the Oracle 'Family'?*

A "As Charles Phillips has said on a number of occasions, our goal is to offer our customers a superior ownership experience. There are ways we are starting to do that for our JD Edwards customers, firstly by beginning to integrate JD Edwards with some of our other products such as Oracle Fusion Middleware and Oracle Demantra, our Demanding Planning solution.

Q *Why is Oracle not selling JD Edwards to new Oracle customers?*

A "We are. We made the decision in 2006 not to sell JD Edwards to net new customers, because we wanted to concentrate on improving the effectiveness of our engagement model with our existing customer base. Now that we have the right partners in place we are marketing and selling to new prospects. With the support of the Partner Channel this decision has now been reversed.

We have sold JD Edwards to 4 new customers in the last 6 months and have 5 marketing campaigns, being led with JD Edwards, currently executing."

Lawrence Clark can be contacted on 0118 924 7103, or by email at Lawrence.Clark@Oracle.com

Being part of the JDE community in UKOUG ensures that we collectively provide one voice to articulate any concerns and needs about the product to Oracle, which in turn is more powerful. Our events provide the opportunity to network with Oracle partners and representatives as well as with other Oracle customers with similar issues.

Working alongside Oracle and the partners, we are bringing the JD Edwards community their very own conference, which will take place in October of this year. Watch this space for more information on the Conference.

08 PeopleSoft Conference & Exhibition
Sharing knowledge within the PeopleSoft community
UKOUG PEOPLESOFT
4th & 5th December 2008
ICC Birmingham

Save the date

UKOUG PeopleSoft Conference 2008

Thursday 4th – Friday 5th December

After much discussion at UKOUG 2007 Conference & Exhibition, it was decided that the PeopleSoft community required a Conference & Exhibition of their own.

This event will cover Financials, HR and Technology streams and will include a wide range of topics, appealing to users at all levels. It's a fantastic opportunity to network with other PeopleSoft users whilst getting updated on Oracle's latest initiatives regarding PeopleSoft.

Call for papers

To submit an abstract to present on the day you just need provide us with a 200 word description giving a basic outline of your presentation.

Submission deadline:
Monday 9th June 2008

Submit your abstract at:
www.ukoug.org/peoplesoft

Exhibition

The PeopleSoft community will be having their own Exhibition during the PeopleSoft Conference. Exhibition space is limited, so don't miss out on the opportunity to showcase your offerings.

Sales will be opening shortly – contact opportunities@ukoug.org to register your interest in exhibiting.

Debra's diary

Each year at Christmas, I am one of those people who writes a letter about what I have been up to for the year. It always starts with a sentence along the lines of: "How fast time flies, and it has been as busy as ever" – well that sums up the three months since I wrote the first 'Debra's Diary'.

Ronan and I went to the International Oracle User Group Council Summit in January, where Oracle provided top executives to this very interactive session. We make a good team as he can attend the technical stream and I can cover applications. So what are the main topics for this year?

Applications Integration Architecture or AIA needs to be understood by all those who work with any of Oracle's Application families, as we were told "Understanding the process is as valuable to customers as the actual integrations". It was described it as the "Next generation process integration for business transformation"". I can't do it justice in the column but you can find lots of information at www.oracle.com/aia and if you are an AIA practitioner how about submitting an article for the next Oracle Scene?

Steve Miranda talked about Fusion Applications. In particular, he talked about this first release having a broad footprint covering Financials, Procurement, Project Costing, HR and Payroll. Oracle have done extensive research as to what was needed first and customer feedback has helped to determine this. There will be a new technical stack, with no Forms or Peopletools; it will be all Java and XML; and all flows will be governed by BPEL with no traditional workflow. The first release will be tightly controlled and monitored by Oracle, using customers who have a track record of early adoption.

At Open World they announced the first Fusion Modules as Social CRM and we heard that some mobile applications may be announced at Collaborate in April. These 'edge' modules will come with predefined AIA integration to your existing CRM and ERP.

Ronan gave a presentation to the other user groups about our annual survey as it had been held up as an example of best practice by Oracle, and, he is now leading a working party to assist them in doing similar surveys.

Collaborate is the big US User Conference where OAUG, IOUG and Quest International come together. This year it was being held in Denver and will be the subject of my next diary entry.

Every year we look for nominations for SIG chairs to attend these overseas conferences as a way of saying 'Thank You' to our volunteers. Perhaps we can get those attending this year to write an article around what they learned.

Closer to home we have launched a new Partner SIG. Partners are users of the Oracle Partner Network product and, like all users, want to make the most of their investment. As part of our role, we wanted to help facilitate this. There was an initial survey with Oracle invited to come and discuss the findings, and everyone who attended found the event very useful.

The Board has an annual strategy 'away day' which this year took place on 29th February. There were no marriage proposals but plenty of ideas about how to engage the ever expanding Oracle community and to ensure we continue to service the established members.

I couldn't finish without mentioning the excitement in the office as we move through our celebration year – 25 years of serving the Oracle community.

Debra Lilley is a Principal Business Consultant with Fujitsu Services. She is both an Oracle Certified Professional (Applications) and Oracle Master (IT Professional). Debra has been a UKOUG director since 2004 and is currently Deputy Chairman. She is also responsible for the Product Development Committees at both EMEA and International Oracle User Community.

UKOUG CALENDAR OF EVENTS 2008

JUNE

03 **App Server SIG Meeting**, London
03 **Siebel SIG Meeting**, Reading
04 **Document Management & Workflow Special Event**, Berkshire
05 **AGM UK Oracle User Group**, Birmingham
10 **Business Intelligence & Reporting Tools, SIG Meeting**, London
10 **Apps DBA for OEBS SIG Meeting**, Manchester
12 **Irish HCM SIG Meeting, Dublin**, Ireland
17 **Local Government CRM Customer Forum**, West Midlands
17 **Management & Infrastructure SIG Meeting**, London
17 **UKOUG – Hyperion User Forum**, Middlesex
19 **Northern Server Technology Day**, Newcastle upon Tyne
24 **OUG Scotland BI SIG Meeting**, Edinburgh
25 **Modelling Analysis & Design SIG Meeting**, Manchester
26 **Statutory Compliance Sub Group,** West Midlands

JULY

03 **Public Sector Combined SIG Event**, London
08 **DBMS SIG Meeting**, London
10 **Development Engineering SIG Meeting**, West Midlands
10 **JD Edwards Upgrade Day**, Reading
15 **Acquire to Retire Process SIG**, London
15 **Archive & Purge Special Event**, London

SEPTEMBER

09 **Oracle Financials SIG Meeting**, London
10 **PeopleSoft Combined SIG Event**, London
11 **UNIX SIG Meeting**, Slough
11 **Siebel SIG Meeting**, TBC
16 **Stellent SIG Meeting**, TBC
17 **Prospect to Order Process SIG Meeting**, Slough
17 **Oracle on Windows SIG Meeting**, Leeds
18 **Oracle Projects SIG Meeting**, Reading
18 **HCM SIG Meeting**, Midlands
23 **Oracle Spatial SIG Meeting**, Midlands
24 **Irish HCM SIG Meeting**, Dublin, Ireland
24 **Irish Technology SIG Meeting**, Dublin, Ireland
25 **Irish Applications SIG Meeting**, Dublin, Ireland
25 **Public Sector HCM Customer Forum**, West Midlands
30 **Modelling Analysis & Design SIG Meeting**, London

OCTOBER

01 **Oracle and .NET SIG Meeting**, London
01 **App Server SIG Meeting**, West Midlands
02 **RAC & HA SIG Meeting**, Midlands
02 **Supply Chain & Manufacturing SIG Meeting**, West Midlands
07 **Apps DBA for OEBS SIG Meeting**, Bristol
08 **OUG Scotland Conference & Exhibition 2008**, Glasgow
08 **Business Intelligence & Reporting Tools SIG Meeting**, London
08 **Local Authority Shared Services Customer Forum**, London
09 **Local Government CRM Customer Forum**, West Midlands
09 **OGUG SIG Meeting (Formally known as OFGUG)**, London
14 **Development Engineering SIG Meeting**, London
15 **JD Edwards Conference 2008 – date to be confirmed**
16 **Management & Infrastructure SIG Meeting**, London
23 **Statutory Compliance Sub Group**, West Midlands

NOVEMBER

04 **Education & Research SIG Meeting**, West Midlands
04 **Procure to Pay Process SIG Meeting**, London
05 **Local Government Applications SIG Meeting**, London
06 **Criminal Justice SIG Meeting**, London
06 **DBMS SIG Meeting**, Wolverhampton

DECEMBER

01 **UKOUG Conference & Exhibition 2008,** ICC, Birmingham
04 **UKOUG PeopleSoft Conference 2008,** ICC, Birmingham

All event dates are subject to change

UKOUG
UK ORACLE USER GROUP
25th

25 years of serving the Oracle Community

UKOUG Back to Basics: Database Administration Event – A Review

On February 28th, UKOUG held the first event aimed solely at beginners in the world of Oracle Database Administration.

'Back to Basics: Database Administration' was the result of feedback from delegates at various events, who felt that there was not enough material targeted at a beginner level audience. After such a successful event, watch this space, as there may be more coming…

As always, UKOUG took notice of the feedback from its members and decided to run a one-day special event. This proved to be a very popular event attracting over 130 delegates.

The day benefited from a comprehensive agenda and four well known speakers, each an expert in their own field, taking their material 'Back to Basics'.

The day began with Tom Kyte's presentation on 'Best Practices: Just Say No', in which he explained why the concept of following best practices is not always a good idea. His message was, that just because something is right some of the time, does not mean it will be right all of the time. Take care to ensure that the 'best practice' you are following is right for your particular circumstances and set-up.

> "Left me wanting more! Blew a lot of ingrained teaching out of the window!"

Pete Finnigan followed with 'Oracle Security Basics' in which he talked through some of the basic concepts and techniques for securing your Oracle database. Pete gave examples on how vulnerable databases can be to attack, both internally and externally, and gave some basic advice on how to avoid this.

> "Made me sit up and listen!"

Jonathan Lewis gave a presentation on 'The Beginner's Guide to Cost Based Optimisation', which gave the delegates an insight into the CBO and how it worked.

> "I received a lot of confidence with this presentation. Jonathan Lewis explained CBO exactly how it should be taught, excellent."

This was followed by Julian Dyke's presentation on 'Back to Basics: High Availability and Real Application Clusters' in which he introduced RAC and DataGuard concepts and terminology.

> "This was a good presentation. Pitched at just the right level."

The day was rounded off by a Panel Session during which all four speakers took questions from the audience on a whole range of topics.

Delegate feedback from the event confirmed the day was a great success and something that should be repeated in the future, with many people commenting that they had learned a lot and really appreciated the opportunity to attend a UKOUG event aimed purely at the beginner audience.

OUG Ireland 2008

April showers came early to Ireland and the UK last month, but that wasn't enough to deter the delegates who braved all to attend the 5th annual OUG Ireland Conference & Exhibition.

Despite the weather, over 250 Oracle customers, prospects and partners attended the event at Croke Park, Dublin, sponsored by Oracle, ImageNow, Quest Software, The GL Company, SolStonePlus and Tata Consultancy Services.

This premier networking event opened with keynote presentations from Paul O'Riordan, Managing Director, Oracle Ireland and Ian Smith, Regional Senior Vice President, Oracle UK, Ireland and Israel, which were then followed by dedicated sessions on Technology, Applications, BI & Reporting and complementary partner technology. Delegates also had the opportunity to quiz key Oracle staff during the "Ask Oracle" panel session before continuing on to the post event drinks.

For more information on future events in Ireland visit www.ukoug.org/calendar

and finally...

Giovanni D'Alessio
Deputy Editor – Applications
deputy_apps@ukoug.org.uk

In this article, I will speak of the 'Upgrade Dilemma' that each and every company has to address.

It could be the upgrade of your Application Version, the move to a new database, the installation of a more secure and stable Application Server or the procurement of more reliable hardware: as Oracle users you know that, at one point or another, you will need to deal with an upgrade.

From an Oracle E-Business perspective, a survey from UKOUG at the end of 2007 (Angela Eager, http://www.ukoug.com/news/show_news.jsp?id=11911) shows that:

- 65% of E-Business Suite respondents were using the latest version of 11i
- 21% were using the slightly older 11.5.9 release and
- Only 2% had taken up version 12, which was announced at the start of 2007.

In 2007, a lot of companies were still 'on the fence' and took a 'wait and see' approach to the upgrade to R12 and Fusion. In 2008, however, there are signs that an increasing number of companies are preparing for the upgrade and I have no doubt that, in the next several months, we will see more and more companies starting the upgrade process.

In this issue, we present a great article by Derek Hancock on the reasons for creating a business case to justify and support your IT projects. The concepts and principles described should also be applied to your upgrade project.

- **Clearly documenting the justification for the undertaking of the project.**

  Why are you upgrading? Technical reasons? Proven functional reasons? What is the overall cost? How did you quantify the benefits?

- **Gain management commitment and approval.**

  Who is supporting this project? Do they understand the effort involved? Are they aware of the risks? Have they made an informed decision?

- **Provide a framework for planning and management of the business change.**

  New features mean modified processes, new training requirements, potentially a change in the organisation to support the new process. Have you put in place a team focussing on the change management aspect of the upgrade?

- **To enable you to monitor and measure the realisation of the benefits of the project.**

  How do you ensure that all the desired benefits are realised? Was the effort worth while?

Now, if Hamlet were an IT professional instead of a Danish Prince, and if he had encountered an Oracle salesman instead of a ghost, the classic soliloquy, translated in Oracle language, could have read something like:

> **"To upgrade or not to upgrade to R12 that is
> the question
> Whether 'tis nobler in the mind to suffer the
> pain of an upgrade,
> with partially working scripts and
> unidentified bugs
> Or to accept the current version and
> problems and live with the decreasing level
> of Oracle support."**

One thing is certain: how much easier it would have been for all of us if Hamlet had written a clear business case! I can picture it in my mind:

– *(Hamlet)* I must kill my uncle. A ghost told me that he is my father's murderer. I can't let a murderer rule my country.

– Instead of pretending to be a mad man, he could have explained to Gertrude and others the reasons to justify Claudius' death; even better, he could have convinced her to do it in his place (who better than the wife anyway?)

– After gaining consensus on the necessity of Claudius's death, he could have written a series of instructions for all conspirators to carry out the plot; he could have planned a grand ball and banquet to celebrate the new king.

– *(Hamlet)* I took revenge on my uncle; I am now the King of Denmark; I can happily marry Ophelia or whomever I please...it's good to be the King!!!

To all of you: "Happy upgrade"!

OracleScene

From the UK Oracle User Group

A Hard Act to Follow

The Human Side of the ERP Market in 2008

Now available online in digital format at www.ukoug.org

Oracle User Productivity K

Oracle Data Integrator Enabling the Transition to SOA

UKOUG
UK ORACLE USER GROUP
25th

25 years of serving the Oracle Community

Issue 33 | Spring 2008 | £10.00 | www.ukoug.org

AN INDEPENDENT PUBLICATION NOT AFFILIATED WITH ORACLE CORPORATION www.pillardata.co.uk

Pillar Axiom enables you to provision storage from an application or databases' point of view. It automatically matches resources to multiple applications' requirements and delivers differentiated storage services based on application or database priority.

By collapsing multiple storage systems into a single storage grid architecture, Pillar is able to deliver the performance where it's needed, with customers still achieving 80% utilisation rates. This means you can buy less storage and get more out of it. Also as your Oracle grid environment scales, you can scale the storage in exactly the same way.

Pillar's intelligent use of storage resources and consolidated tiering makes the perfect companion for partitioned data warehouse environments and with its integration with ASM and Grid Control managing Oracle storage has never been easier. And when it comes to changing demands, the Axiom can dynamically reassign resources as new projects start, or existing projects temporarily demand more.

With the Axiom, you can drive down costs and increase efficiency, all while assuring the service levels your Oracle environment requires.

Call +44 (0) 1256 785789 or visit www.pillardata.co.uk

The First and Only True Application-Aware Storage™

pillar®
DATA SYSTEMS