Risk-based Assessment of User Access Controls and Segregation of Duties for companies running Oracle Applications

Sarbanes-Oxley (SOX) has been a major burden for organizations and auditors since it was first enacted in 2002. There has been much confusion about how to comply with the act by both the external auditors and companies which must comply with its provisions. The primary focus of SOX compliance is the design and testing of controls that help companies prevent a material misstatement of their financial statements.

In our observation, the approach taken to comply with the provisions of SOX has been imperfect and inconsistent, at best. As we have reviewed various risk and controls libraries (RCL) used by companies to comply with SOX, the variation in both the detail and nature of the controls has been significant. In 2007 with the acceptance of Auditing Standard 5 (AS5) the rules have once again changed. No doubt the interpretation of AS5 will be as varied as the original interpretation of AS2 and the provisions of SOX.

One of the areas we have seen significant variation in the design of controls is the area of user access controls and segregation of duties (SOD). We have witnessed in many cases, controls have been designed to prevent material misstatement in response to SOX, but fail to address other risks such as fraud, access to sensitive functions, and access to sensitive data. In some cases, the design of controls has left companies with significant exposure to fraud, failure to comply with change management best practices, and overexposure of employees to sensitive data stored in their applications. In our opinion, some of these weaknesses could rise to the level of a significant deficiency or material weakness if detected by auditors.

As a result, while companies have addressed these areas from a SOX perspective, additional projects may be necessary to identify and address other areas of risk. Our expectation is that auditors will dig deeper into these risks as scope is reduced due to AS5 and that they will justify the scope change to include the testing of management's compliance with *all laws and regulations* and the reliance on general computer controls over systems that provide internal controls over financial reporting.

The purpose of this white paper is to discuss a comprehensive risk-based approach to review risks related to user access control, including segregation of duties risks. In doing so, we will examine this topic from a SOX perspective as well as address risks related to fraud, access to sensitive functions, and access to sensitive data. We believe this comprehensive approach is a better approach than addressing each compliance initiative individually and should be adopted by all companies starting their SOX compliance cycle. Companies already in the SOX compliance cycle should discuss this change in

strategy with management and their auditors to determine if the cost/benefit of such a project would be warranted. We believe in many cases such a project would be warranted and would have a significant ROI as it is likely to significantly reduce external and internal audit costs. You may also be able to convince your auditors to rely on this process and eliminate substantive testing on the controls related to this altogether because of AS5. (The author provides no guarantees and suggests discussing it with your external audit firm).

We believe this process will soon be recognized as a 'best practice' and should be adopted by all companies running Oracle applications regardless of whether they are required to comply with the US Sarbanes-Oxley act.

There are sufficient unmitigated risks for any company to benefit from such an approach. These risks derive from deficiencies in the design of the applications, deficiencies in the implementation practices of the software, and deficiencies in the design of controls related to processes using Oracle Applications. Such unmitigated risks could lead to fraud (i.e. theft of assets), operational risks (such as down time of your Oracle Applications environment), significant inefficiencies in the use of your costly investment into Oracle Applications, lawsuits related to failing to protect sensitive data, and fines from various regulatory agencies.

Therefore, the approach recommended in this white paper is prudent for any company looking to identify the risks they are facing and to develop a comprehensive strategy to address such risks.

A comprehensive SOD and user access controls matrix

Surprisingly, a comprehensive SOD conflict matrix has yet to emerge years after the passage of SOX. Auditors have been unwilling to share the rules they use to audit SOD and user access controls because of the perceived conflict of interest between the audit firm and the client. Although this has changed somewhat, there is still a lack of consensus on what conflicts have risk as it pertains to SOX, let alone other areas of risk, such as fraud. In most cases, this has left companies trying to comply with SOX with no public domain source for this information and has left them looking for answers. Some have chosen to develop rules internally. Some have relied on consulting firms to provide the rules as part of a consulting engagement. Others have relied on software companies offering SOD monitoring tools to provide that data as part of their seeded content.

In early 2007, we finalized the development our conflict matrix and related methodology. While some of our methodology and conflicts are applicable to all companies, many of the 'conflicts' we have identified are specific to companies running Oracle Applications because of unique risks posed by the design of these applications.

Types of risks

Several categories of risks as they relate to access controls exist and can be classified as:

- *Segregation of duties* – a user having two or more business processes that could result in compromise of the integrity of the process or allow that person to commit fraud
- *Access to sensitive functions* – a user having access to a high risk function that, in and of itself, has risk
- *Access to sensitive data* – a user having access to sensitive data such as employee identification number (US= SSN), home addresses, credit card, and/or bank account information.

Segregation of duties

SOD weaknesses most commonly come to mind when most people think about access controls. SOD risks can be system or non-system related. An example of an SOD risk that is system related in the payables area is assignment of both the entry of suppliers and AP invoices to a single user. A risk associated with a user having access to both functions would be the entry of a fictitious supplier and an associated invoice. Absent any mitigating controls, the invoice would likely be paid on a payment run and the employee would have been able to commit fraud against the company. The key in the development of good access controls as it relates to SOD is to identify the risk(s) associated with a single user having access to both, then addressing those risks in the design and implementation of controls. In our example, here are a few ways this risk could be mitigated:

- An audit of suppliers entered versus approved suppliers by tracing the data from the system back to a supplier setup form that has a proper approval signature.
- An independent review of a preliminary payment register or the checks before they are sent to the supplier.
- The budget to actual analysis may identify that the expenditure is fictitious if the reviewer doesn't recognize the nature of the expense and questions it. However, not all invoices are coded to expense accounts. Some are coded to balance sheet or non-expense accounts such as sales or sales returns where they may be difficult to identify due to the fact that small variances in larger accounts may not be questioned.
- Account reconciliations may cause an accountant to question the expenditure if it is coded to any account they are reconciling.

A proper risk assessment not only identifies a potential mitigating control, but also measures the ability of that control to mitigate the risk. In our mitigating control examples identified above, limitations are present in each of these controls. For example, an audit of suppliers entered versus those approved is limited by the strength of the approval process.

Limitations of the supplier audit process as a mitigating control

Anyone able to request a supplier be set up by filling out a supplier entry form effectively allowed them to set up a supplier in the system because, typically, the person performing the data entry function is merely entering the data based on an approved form. Therefore, as it relates to an SOD conflict between entering suppliers and entering invoices, any employee with the ability to fill out a new supplier form and enter invoice in the payables

module or generate a purchase order should be identified as having an SOD conflict. This illustrates the need to encompass both system and non system processes when identifying SOD risk

To provide oversight on the setting up of new suppliers, many companies have added an approval to the supplier setup form such as a purchasing employee to review and approve the new supplier before it is entered in the system. However, the limitation to this control is the volume of non-purchasing suppliers such as utilities, contract labor, rents, and supplies. The ability for a buyer to identify whether a supplier is fictitious depends on their incentive to question it. If the sourcing of the top of product or service is not in their realm, then their review will be less thorough. Likely, their 'approval' of those types of suppliers will be no more than a rubber stamp.

Another common 'mitigating' control for new suppliers is the receipt of a W9 which requires a supplier to identify their taxpayer identification number. However, the limitation to this is that the providing of a W9 doesn't guarantee that the taxpayer identification number is valid. A person that understands the W9 process who is trying to defraud the company knows that they merely need to enter Federal Tax ID number related to a corporation to prevent the company from sending them a 1099 at the end of the year (most companies don't issue 1099s to corporation because the IRS doesn't require it).

Limitations of a preliminary payment register review or check review as a mitigating control

In our example above, we identified the review of a payment register or the checks as a potential mitigating control for the risk in which an employee has both the ability to enter suppliers and enter checks. This mitigating control has some limitations at many companies because of the volume of the checks generated. Usually, this type of control has been designed to focus on significant checks to avoid a material misstatement. Therefore, the checks over a certain dollar amount (say $50,000) may be reviewed by having the supporting documentation pulled for review. In the cases in which goods are received, the receipt of goods (and related proof of delivery) helps to substantiate that there indeed was something received. However, as it relates to expense items, the person 'approving' the expense may be the same person that requested that the supplier be set up in the first place. So, it could be possible that someone requested a fictitious supplier to be set up, mailed an invoice to the AP department who sent the invoice to this person for approval. If the approval is within their limits, it likely would be entered and paid by Accounts Payable. You then are relying on the controls related to budget to actual analysis and account reconciliations to catch the fraud. What would happen if the invoice was coded to an inventory account with a subledger and GL control account that are out regularly of balance? Would such reconciliation catch the fictitious invoice? Would someone question the invoice being coded to that inventory account?

As you can see from this example, the risk assessment process related to this one control can be complicated and not as easy as it seems. Let's continue to look at how processes outside the system can impact a risk assessment process related to user access controls.

Processes outside the system with risk
An example of a process that happens outside the system impacting segregation of duties is the request to set up a new supplier. A common SOD violation is a conflict between the entry of suppliers and the entry of a purchase order. Tests related to this process typically focus on the access controls related to the entry of a supplier. However, rarely do tests extend to processes that happen outside the system. In this example, the question should be asked "how does a supplier come to be set up in the system", not just who has access to set up the suppliers in Oracle. For instance, if a company's process is to allow any employee, including those in purchasing, to fill out a new supplier form and submit the information to a clerk in Accounts Payable (AP) to enter the data, the SOD rule is likely violated. The effect of a purchasing agent requesting a supplier to be set up without any validation of the supplier or approval by an independent source(s) in effect allows that employee to establish a supplier in the system. The process the AP clerk performs in entering the supplier is merely a clerical function and scrutiny over such access only tests part of the risk in the process.

In this example, even if you are requiring someone independent of the purchasing agent to approve the new vendor request before it is sent to the AP clerk, there may still be risk. There is still the question about what the approver(s) is scrutinizing when reviewing the supplier request. Is that person judging the authenticity of the supplier? What about if the supplier is a related party or owned by the purchasing agent? Would it be identified as inappropriate? Therefore, just as important as reviewing the access to enter suppliers, the request and approval process is also key to addressing risk in the process to design both the business process and the proper controls.

If your risk analysis process hasn't identified such risks or if you don't ask the right questions when identifying and analyzing the mitigating controls, risk beyond what management is willing to assume may still exist.

Access to sensitive functions

Traditional SOD risk analysis focuses on the appropriateness of a user having access to two or more processes. What about the processes that have risk in and of themselves? Here are some examples:

- Bank accounts – changes to bank accounts could provide someone the ability to commit fraud.
- Security related forms such as responsibilities, menus or roles – changes to responsibilities, menus, or roles could allow a user to grant themselves access to sensitive data such as HR data or sensitive functions such as bank accounts, then change it back.
- Development related forms such as Alerts and Functions – access to such forms may allow a user the ability to manipulate data within the database by entering SQL statements that are 'run behind the scene' and without an audit trail. These SQL statements could be written to update sensitive data such as bank accounts and salaries unassociated with a user login to hide the accountability for the change.

Process example – Bank Accounts
Some companies have payments made from accounts payable via ACH (i.e. payments are not being made by check, but by a file sent to the bank that identifies the routing number, bank account, and amount of the payment). In this case, the access to the bank account data has risk in and of itself. For example, a person doing the bank account maintenance may know that a significant payment to a supplier is coming up. Having the ability change the bank account may allow them to change the routing and bank account number for that particular supplier just prior to the payment batch processing. Then, after the payment batch is run, they could change the routing and bank account back. What would cause this fraud to be detected? Likely, the first notice you would have that this is an issue would be a call from the supplier saying they haven't been paid. That call may not come for several days or potentially even several weeks after the payment was made. When the vendor calls to ask why they haven't been paid, the analysis on what happened to this payment will also likely take a couple of days.

This time lag may just give a fraudster the time they need to transfer the funds to another account and leave the country. The incentive of a significant payment may be enough to entice a person to commit fraud.

Access to sensitive data

Traditional SOD risk analysis is focused on access to update data, not necessarily access to sensitive data. Check fraud and identity fraud are big business for organized crime as well as desperate individuals. Gaining access to sensitive information necessary to commit such fraud is also big business. While more than thirty states have enacted laws to protect sensitive employee data, not enough has been done by companies to protect such data. One area that has largely gone unguarded is access to sensitive employee data found in Oracle's accounts payable module. Employees are often set up as Suppliers in order to pay expense reports. That can include entering employee address and bank information. We have also seen in some cases, the social security number entered as well. The nature of accounts payable data is that it needs to be seen by personnel doing budget to actual analysis and those responsible for processing accounts payable data. Oracle provides little help in protecting such data as there are no functions to represent such data that can be used to exclude access to employee accounts payable data from that data related to external suppliers. Therefore, the use of forms personalization or the custom.pll to restrict visibility to such data is necessary.

Companies trying to address access to sensitive data should follow the following process:

1. Identify the laws (federal, state, and industry specific – like HIPAA, GLBA, PCI) that apply to your organization.
2. Based on these laws, determine what your organization's obligation is under these laws. This analysis should include the identification of sensitive data, what companies must do to protect such data and their obligations if there is a breech.
3. Identify any other data that perhaps isn't required to protect, but morally, based on the sensitivity and nature of the data, the organization should protect.
4. Identify the applications and databases where this information may be stored.

5. Determine which people and/or roles should have access to the sensitive data.
6. Identify which people and/or roles actually have access to the data through the application by scrutinizing menus and responsibilities. Remediate, as necessary.
7. Identify which people and/or roles actually have access to the data through the application by scrutinizing request groups. Remediate, as necessary.
8. Identify which people and/or roles actually have access to the data through the database access. Remediate, as necessary.
9. Identify which people and/or roles actually have access to the data through the various reporting tools such as Discoverer. Remediate, as necessary.
10. Make changes to the Change Management process to require a review for access to sensitive data as changes are made to security.

A full white paper on the subject of access to sensitive data is planned for publication in the winter of 2007.

Risk Assessment Projects

In our experience, a completed risk assessment process exposes the following needs:

- An *SOD monitoring tool* (or one with a preventive workflow)
- A tool to develop a *detailed audit trail*
- Various *monitoring reports or processes* not provided by Oracle
- The need to *personalize forms* to support defined controls.
- *Custom workflows* to automate controls where Oracle's functionality is deficient
- *Process and/or controls changes*
- *Documentation and testing of non-key controls*
- *Access control changes*
- *Additional projects and research* that need to be done

SOD monitoring tool

Many companies have put the cart before the horse as it relates to purchasing third party software to monitor risks related to SOD. Often, companies have relied on the software provider or their auditor to define what risks should be monitored. The SOD conflicts provided by software vendors have varied greatly and have provided no certainty that they match up with the risks identified by their external auditors. Additionally, there has been little focus on fraud prevention, access to sensitive functions, or access to sensitive data. This has left buyers of such software frustrated with their projects. Some conflicts identified during these efforts have been 'false positives' in that the inherent risk(s) of such access is fully or substantially mitigated by controls in place. The result is a much prolonged remediation process. Additionally, the lack of a comprehensive conflict matrix that takes into account all types of risk (traditional SOD risk as well as fraud risk, access to sensitive functions, and access to sensitive data) has left companies exposed to unidentified risks.

A proper risk assessment process is the perfect compliment to the RFP process. The end result of a risk assessment process is the specs for such a tool including what rules need to be monitored while using the tool. These conflicts take into account where a user has

access to two or more functions at the same time as well as the risks of having access to a single function (examples of supplier and bank accounts discussed above).

With the dynamics of the consolidation in the GRC software space, a company may find it difficult to determine which tool meets your needs. However, a central part of the project should be to fully automate the user provisioning and SOD monitoring process via workflow. Even though many of the tools may not have a prebuilt workflow, a custom workflow can be built and should be considered an essential component. Most projects have audit cost reduction as a central part of the ROI calculation and a fully-automated control using workflow is critical to help that justification process.

Detailed audit trail

Another of the likely outcomes of a risk assessment process is the requirement to audit certain tables. For instance, because of the fraud risk associated with the entry of suppliers and bank accounts, the tables underlying these key setups are identified for audit so that someone independent of the data entry process can confirm that the additions and changes were authorized and the entry is accurate. Absent a detailed audit trail that is the result of trigger or log file technologies, the best you can hope to achieve with standard table data is the value as of the last update. See more information on this topic by requesting the white paper called Building an Audit Trail in an Oracle Applications Environment which can be requested at www.oubpb.com.

Monitoring not provided by standard Oracle

There are some challenges in Oracle that are difficult to overcome. One prime example is access to the Customers form. Depending on the modules that are being used, data in the Customers form may need to be maintained by two or more groups. The sales department may be responsible for adding customers. Another group may be responsible for maintaining price list information. The credit group would be responsible for maintaining the credit limit, check credit checkbox, and the payment terms. Typically you would prohibit the sales department from maintaining the credit related fields. This could be done via the use of forms personalization or, if the volume of credit changes is small, by monitoring changes to these fields via an alert that is sent to the credit department. Reporting could also be developed based on the detailed audit trail of all changes to these fields to review whether an unauthorized user updated them.

Forms personalization

As noted in the above paragraph on the Customers form, there are several groups that may have a business need to maintain information in the Customers form. In high-volume environments, a preventive control may be cost-justified. In these cases, the Customers form may be broken apart via the use of Forms Personalization to limit access to certain fields to maintain. In the above example, perhaps the salespersons have access to create customers, but are not allowed to update pricing or credit information. Perhaps sales management can maintain pricing information, but not set up a new customer or maintain credit information. Finally, the credit department could be given access to update credit information, but not enter a new customer.

Custom workflows

In some cases, significant risks and deficiencies in Oracle's application design might call for the development of a custom workflow. One example could be the development of a custom workflow in Order Management to route RMAs for approval. Typically, companies like to control returns and credit memos generated because of both operational and fraud risks. To date, Oracle has not provided a standard workflow to address this issue.

Process and/or control changes

Another likely outcome from a risk assessment process is the changes to processes and/or changes to the definition of controls. One example is the process related to the requesting of a new supplier to be established. Based on fraud risk and operational controls, we recommend that two signatures be accompanied with the request. One signature is by the requestor's manager to confirm the need for the goods or services and the authenticity of the supplier to meet those needs. The second signature is by the purchasing group reviewing from an operational effectiveness perspective. Purchasing should determine whether the company should establish a relationship with the new supplier or use an existing supplier. This recommendation may require a change to the process and a change in the documentation of your controls related to this process.

Documentation and testing of non-key controls

In some cases, companies have yet to even document non-key controls that may not rise to the level of material risk, but that are prudent to document from a fraud prevention perspective. The example stated above, related to a supplier approval process, may not be documented as a key control because of other key controls such as budget to actual analysis and month end financial statement flux analysis. However, the control is critical from a fraud prevention perspective and needs to be documented and tested to ensure the operating effectiveness of the control.

Access Control changes

The risk assessment process will also identify some changes needed to access controls even before a third party tool is purchased. For example, as you recognize the ability for an employee to commit fraud using a single function, you may want to further restrict those with access to such forms. You may also feel the place additional restrictions on access to the production database or operating system. This could include restricting off-shore developers where you may have restricted ability to bring suit against the individual if they were to commit fraud.

Additional projects and research

Other risks that will take more research and analysis should come out of the risk assessment process. Questions like:

- Which concurrent programs have the ability to update data or run interface jobs and who has access to such programs?
- Which concurrent programs and inquiry access (Discoverer, reports, database, etc) provide access to sensitive data and who has access to such data?

- How is sensitive data handled in non-production environments and how are access controls different in those environments?
- While profile options have control risk and how are they set up?

Some of these questions will open doors that will require additional research because they will challenge some key areas such as development practices, change management processes, and IT standards. The outcome of such questions could open Pandora's box and expose risk you never knew existed.

Conclusion

Companies looking to minimize risk associated with access controls are advised to take a risk-based approach. Only a risk assessment process that evaluates traditional SOD, access to sensitive functions, and access to sensitive data, taking into account a company's defined controls can adequately and holistically address the risks. Until a company invests in this process, it cannot be certain the risks have been properly mitigated.

What is sorely lacking is an effort in the public domain to address this process. Our intention is to be the leader in this area. We have dedicated much of the past year to developing a methodology and the content to perform such an analysis. If you are interested in participating in such a process or having us help you perform a risk analysis as discussed above, contact the author at jhare@erpseminars.com or sales@erpseminars.com.

About the Author

Jeffrey T. Hare, CPA CISA CIA
Jeffrey is the founder of ERP Seminars (www.erpseminars.com) and the Oracle User Best Practices Board (www.oubpb.com) and has written various white papers on Internal Controls and Security Best Practices in an Oracle Applications environment. He has presented white papers to various users groups throughout the country as well as at OAUG and Appsworld conferences. He is the author and presenter of the seminar "Internal Controls and Security Best Practices in an Oracle ApplicationsEnvironment." His background includes Big 4 experience, over six years experience in CFO/Controller roles, and in the Oracle Applications space since 1997. Jeff can be reached at jhare@erpseminars.com.

About ERP Seminars:
We recognize the need for companies to have continuing knowledge of industry Best Practices. We team with respected independent consultants and firms to provide quality, relevant seminars based on these Best Practices prepared and presented by well-rounded professionals with ERP expertise.

About Oracle Users Best Practices Board:
The mission of the OUBPB is the aggregation of willing writers and reviewers who will participate in a process to develop Best Practices for the Oracle community. The end result will be a repository of "best practice" white papers and other content for end users and consultants to reference in their projects and ongoing development.

Version Control

Date	Author	Version	Reference
25-Sep-07	Jeffrey Hare	1.0	Initial publication