

Advanced Support Tactics for Optimizing Oracle Systems

ORACLE WHITE PAPER | NOVEMBER 2015



Table of Contents

Executive Overview	2
Oracle Advanced Customer Support	3
Support in Relation to ITIL	3
Advanced Support Tactics	4
Oracle Support and Systems Optimization	4
Support Tactics for Systems Strategy	4
Proactive Support Planning	4
Support Tactics for Systems Design	5
Planning for Supportability	5
Planning for Scalability and Capacity	6
Planning for Maintainability	7
Hardware Support Considerations	7
Support Tactics for Systems Transition	8
Design for Performance	8
High Availability	9
Production Go-Live	10
Support Tactics for Systems Delivery (Operations)	10
Event and Incident Management	10
Service Requests	11
After Action Reviews	11
Backup and System Restore	12
Patching and Upgrade Strategies	12
Support Tactics for Continuous Systems Improvement	13
Third-Party Best Practice Reviews	13
Technical Account Management	13
Service Delivery Planning	13
Remote Support	14
Benefits of Building Remote Delivery into the Support Plan	14
Remote Support Requirements and Planning	14
Oracle Priority Support and Oracle Priority Support Advantage	14
Conclusion	15



Executive Overview

Achieving an optimized, stable, reliable, available production system requires consideration of a wide range of topics. This white paper discusses a number of the core factors that must be considered in order to arrive at the desired optimized system in steady-state production. There are commentary and recommendations on tactics relating to faster service request resolution, supportability, maintainability, performance, scalability, availability, backup and restore, patching, and upgrade. The paper recommends specific actions that the customer can take to get maximum value out of Oracle Premier Support, and also touches at a high level on services that can be engaged from the Oracle Advanced Customer Support (Oracle ACS) organization.

Oracle ACS engineers provide proactive and preventive support using advanced diagnostic tools to help you increase system availability, reduce risk, and accelerate ROI across the Oracle stack—applications, middleware, database, servers, and storage systems. Customers can choose from a portfolio of mission-critical support services that are tailored to their IT and business needs.

Oracle Advanced Customer Support

Seamless data availability, optimal applications performance, and reduced IT risk are critical to business success. Oracle Advanced Customer Support (Oracle ACS) delivers tailored mission-critical support services to help customers maintain and maximize performance of all Oracle mission-critical systems. The Oracle ACS partnership with Oracle Support and Oracle's engineering teams, combined with the unique approach to building a collaborative long-term relationship with customer IT teams, provides a highly integrated approach to helping companies meet their complex IT requirements.

Oracle ACS engineers provide proactive and preventive support using advanced diagnostic tools to help you increase system availability, reduce risk, and accelerate ROI across the Oracle stack—applications, middleware, database, servers, and storage systems. Customers can choose from a portfolio of mission-critical support services that are tailored to their IT and business needs.

Support in Relation to ITIL

The intention of this paper is to outline tactics that are helpful in achieving an optimized and high-performance system in operations. As a structured approach, ITIL (formerly known as the Information Technology Infrastructure Library) provides an excellent framework for comprehensive discussion around systems optimization in operations. Any system (or IT service, in ITIL terms) that is brought into operations must first go through two prior stages: service design and service transition. The underlying service strategy drives the direction in general for the organization's IT infrastructure and approach. The following ITIL diagram illustrates five discrete sets of activities that should be considered for fully optimizing an enterprise system.



Figure 1: Core ITIL Service Concepts

The subsequent sections of this paper will refer back to these five ITIL stages and act as a point of reference. A service as defined by ITIL relates to a combination of people, processes, and technology that make up a complex system providing a function or "service" to the company (typically through hardware, networking, software, and so on). However, throughout this paper there will also be references to services that relate to support activities that a partner (or Oracle) can deliver to a company, in designing, transitioning, or delivering the company's systems. Because the term service can relate to several concepts, within this paper it will be limited to referencing services delivered by a partner to a company that is running an enterprise system. To address the



five core ITIL concepts, references here to the IT services will simply be cited as systems. Thus, the high-level ITIL stages within this paper will be cited as follows for clarity:

1. Systems strategy
2. Systems design
3. Systems transition (aka systems implementation)
4. Systems delivery (aka systems operations, or production)
5. Continual systems improvement

This modified terminology will allow for any subsequent discussion of “services” to relate more specifically to partner (or Oracle) supplied services that would typically be contracted as assistance for any of the above five ITIL stages during which systems optimization can be facilitated.

Advanced Support Tactics

The savvy IT manager or director will begin investing time early in the lifecycle of their systems to ensure that the production or operational system is optimized from many different perspectives. To drive the most-efficient support processes, some advanced tactics can be employed that make it easier to ensure that fewer surprises arise at the point of go-live and once in production.

These tactics can be described at a high level as proactivity, planning, and unbiased best practice review. By continuously looking forward toward the desired state in production, the objective of a sustainable, supportable solution can be achieved.

However, this can only be accomplished when these three tactics are employed very early in the systems implementation process.

Tactic #1: An optimized production system is best achieved by proactive planning very early in the implementation process.

Oracle Support and Systems Optimization

Oracle Premier Support for Software provides global, enterprise-class support and continuous access to Oracle innovation. Through this single offering, customers can receive consistent, proactive support across all of their Oracle software, including databases, middleware, and applications. Product updates and new releases enable ongoing business success, and Oracle’s unique Lifetime Support Policy means you can upgrade as it suits your organization.

Oracle Premier Support for Systems provides the industry’s most complete server and storage system support. Customers receive one-stop, 24/7 coverage for all Oracle system components, including, as applicable, hardware, operating system, virtualization and management software, and other integrated system software. Oracle Premier Support also includes ongoing updates and new releases for all covered software and firmware.

Support Tactics for Systems Strategy

Proactive Support Planning

Companies implementing Oracle systems—whether hardware, middleware, software, database, storage, or any combination thereof—typically focus during their systems implementation on functionality. The systems



implementation partner is typically given extensive business requirement details for the desired functions the system should provide.

What is often lacking is adequate consideration of subtle technical requirements of the system. In cases where the technical details are left to the discretion of the implementation partner, there is risk that the system may fail to meet the actual needs of the company. This is true even though the system may deliver the desired functionality. For instance, a nightly load of data that serves to integrate disparate systems may *technically* deliver the desired high-level result of synchronizing the data. However, if the window of time required for the data synchronization is eight hours and the actual time allowed for the data transfer between shifts is only 30 minutes, there are likely fundamental flaws in the design of the solution. Omitting the timeframe available for the data transfer can lead to significant rework costs prior to going live with the new system. It is this sort of performance requirement omission that leads to delays in projects. It is also a core reason for project cost overruns.

Thus, it is far more cost effective to proactively plan both the business requirements and the technical requirements up front, prior to beginning the implementation cycles. This leads to a more efficient implementation with fewer iterations of rework.

In a similar way, systems support activities and processes should be proactively planned. By looking at the big picture associated with the system, practices that may be inefficient, costly, or difficult to maintain can be avoided. Support can be driven to efficiency by planning for it in advance. Optimization of the support process must begin during implementation of any new systems. Furthermore, upgrades or technical refreshes of systems offer an ideal opportunity to reconsider and greatly improve how any (Oracle) system is configured and customized. The production or operations support period can be much easier or even near trouble free. It is also important to note that the support period for most systems will run much longer than the implementation cycle. Thus, focusing on how the system runs and is supported during the implementation or transition period alone is generally inadequate. The real effort to consider and optimize is the steady state that needs to be optimized on the operational, production system.

Support Tactics for Systems Design

Planning for Supportability

Any software in circulation will encounter issues. For example, new security issues arise that demand proactive adjustments to applications and software. The resulting patches and fixes for a complex application or the database will become bundles of fixes that must be applied. Based on customer requests for enhancements, as well as the ongoing efforts of the vendor to improve the software's functionality, upgrades to the base software are released that combine fixes as well as desirable new functionality. New fixes and new functionality are normal, of course, for all software.

Due to product evolution, over time the status of support for Oracle products moves from Oracle Premier Support to Oracle Extended Support and eventually into Oracle Sustaining Support. The costs and benefits of each level of support can be found by reviewing the Lifetime Support Policy (<http://www.oracle.com/us/support/lifetime-support/index.html>) for your specific Oracle product. There are many benefits of being on the newest version of any Oracle product. The customer benefits from all of the cumulative fixes and patches and also has access to the widest range of functionalities by upgrading to the latest versions. Furthermore, the level of support derived from Oracle Premier Support, which is available for the newest versions, is optimal.

In addition to the version of each product being run, there are considerations around supportability. It is possible to invalidate your support by running your Oracle product on an unsupported platform or by getting too ambitious with customizations.

For each product, you must follow the guidelines for proper use and configuration. In cases where customizations or advanced configuration changes are taking place, it is equally important to ensure that the configuration changes being made are within the guidelines of the support policies. For instance, certain products might not be supported when running in a virtualized environment. This does not mean that no support will be provided, but it does mean that the problem or issue would need to be reproduced on a supported platform before further actions are taken on the service request (SR). Thus, both the customer and Oracle Support benefit when the customer is aware of the support guidelines and operates within them.

Tactic #2: *Engage Oracle Advanced Customer Support services to ensure that the customizations and configuration being employed stay within best practices to ensure full support.*

The guidelines for support for specific Oracle products can be found in their documentation. Additionally, by engaging Oracle ACS onsite or through remote assistance, troubles related to support guidelines and limitations can be avoided.

Planning for Scalability and Capacity

The system must not only run under current loading, but must also be able to handle the scale of the system as it grows. Thus, the capacity of the system must be scaled according to the eventual needs, or there must at least be ability for the system to be improved over time to address the eventual scale of the system demand. The audit trail example cited later in the performance section of this paper shows a negative outcome in relation to performance.

Another consideration is the space requirements of the data. Imagine that every time a user opens a record and makes a tiny change to any data, another record or two shows up in the audit trail tables. Over time, those audit trail tables will mushroom into huge amounts of data that may dwarf the size of all other tables in the database. Will the system operate to the same performance standard over many weeks and months? Or will it become slower and slower due to the configuration and customizations that have been put in place? These are some of the questions that must be addressed in relation to scalability.

Furthermore, if the system is running great during the load testing and even after go-live, will that continue when the number of users is doubled next quarter? The system's demands should be weighed against future user levels, and there should be a path toward managing the eventual scale. In many cases, additional resources might be added to the system to aid in scaling to the increased demand, such as adding CPUs or application servers. Adoption of Oracle's engineered systems represents an optimal choice to address these types of scalability requirements.

When implementing the system, you should ensure that there is a clear path toward maintaining the desired performance without drastically increasing administrative overhead or the need for reimplementing to address the increased scale.

Tactic #3: *Scale the system for the eventual user levels, or at least have a clear technical path planned and budget to allow the system to grow as the scale of the system usage grows.*

Planning for Maintainability

Achieving the desired functionality is a key driver in how systems are implemented. However, when the system goes live, a longer period of steady-state production support begins. The largest costs for a system are incurred during the production maintenance phase of the system. During the production support phase, the system needs to be maintained and optimized. A variety of production support tasks need to be performed to maintain the system. Patches need to be applied as needed; data and log files eventually need to be purged. And eventually, upgrades will be made to one or more layers of the technology stack.

Important considerations include the following:

1. What are the routine administrative tasks that need to be performed, and can they be automated?
2. What is the likely impact of a patch or an upgrade? Will that process be smooth, or are there likely to be complications due to the way the system is configured?

One example of issues that can be encountered relates to customizations. If your application allows for scripting or custom code for enhanced functionality beyond out-of-the-box features, this scripting poses a potential risk to maintainability during upgrades. Every line of custom code has the potential of breaking during an upgrade. If the upgrade includes new scripting abilities, or if there are changes to how scripting is interpreted or instantiated, the old scripting can be broken. For this reason, it is imperative to customize systems cautiously. In most cases where upgrades are difficult, the largest problems are encountered where heavy customization has taken place. The systems integrator must be held accountable to the maintainability of their customizations, giving it the same priority as the immediate functionality requirements.

Tactic #4: Document all customizations in detail to ensure maintainability.

At a minimum, all customizations must be clearly documented. Then, during inevitable future upgrades, custom code can be updated as needed. In general, the less custom code there is, the less risk to maintainability.

Hardware Support Considerations

Hardware support needs are affected by the physical distance from the customer's data center to the nearest Oracle Hardware Warehouse. There are different service levels in hardware support if the customer is located within 25 miles (40 km), 50 miles (80 km), or more from an Oracle Service Location (<http://www.oracle.com/us/support/library/service-locations-073430.pdf>). One mechanism to drive up system availability is keeping onsite spare parts. The Oracle ACS onsite spares service is the way to keep support at the highest level and to ensure system availability.

Event and incident management can be enhanced with automated SR features. *Auto-SR logging* is an advanced tactic that customers using Oracle hardware can consider to improve support efficiency. Look to Oracle ACS for assistance with the implementation of auto-SR functionality.

When engineered systems are employed, there is a large boost in service levels obtained. Customers running engineered systems may be eligible for Platinum Support and can also make use of Oracle ACS including Advanced Monitoring and Resolution services, greatly enhancing the support infrastructure.

Support Tactics for Systems Transition

Design for Performance

When judging whether an end-user-facing application or system is meeting business needs, two factors are generally considered first: functionality and performance. The functionality can easily be proven through use case and unit testing. However, performance is a bit more difficult to judge. This is due to a host of challenges associated with performance testing:

1. Actual performance cannot usually be judged on a test system, which likely has a slight or even significant difference in capabilities from the production system.
2. Integration with external systems is typically very difficult to reproduce and often must be simulated with “stub” or “pseudo” applications that only emulate the actual systems. Those pseudo applications do not usually accurately represent the production systems from a performance standpoint.
3. The performance in a fully loaded environment—with potentially hundreds or even tens of thousands of users—will not be comparable to a test system that is only seeing activity from a handful of test users.
4. Performance in the production environment will depend on all layers of the stack employed. Performance bottlenecks can come from the database level, the middleware, any additional integration or workflows, the servers that are hosting the front-end application, and even the client machine that is running the front-end application.
5. Performance issues that are uncovered may require extensive rework of the configuration. Often the correct or most-appropriate fixes cannot be implemented due to time constraints.

Given all these challenges, it is common for production systems—once live—to fail to meet performance expectations.

The following is a simple example of where performance can be derailed and misunderstood. A business manager of a large international bank states as a security requirement that any change to any data on the mission-critical financial system should be logged using the native audit trail functionality of the end-user application. Because this is a standard function of the application, this would be supported and the manager feels this is a great way to take advantage of the native capabilities of the packaged application. On the test systems, the systems integrator easily achieves this requirement with configuration changes to the application. The resulting functionality is nicely demonstrated, and the business manager is satisfied that the desired functionality has been achieved. However, when the use case is later load-tested on the preproduction environment, the performance is shown to be horribly slow and the application is deemed to be a possible failure based on performance.

This example demonstrates an important concept that must be considered in relation to performance. Every configuration decision made to achieve functionality requirements has the potential to negatively affect performance.

In this case, the systems integrator followed the customer’s requirement to the letter, and also stayed within the parameters of the suitable use for the application. However, the performance failure is taking

Tactic #5: Every customization or configuration change must be weighed against the performance impact.

place because of a significant additional load that is created, nearly doubling the database activity due to logging

of every data change into audit tables. This dilemma was not seen in the unit tests because the number of users was very small and thus did not have a real impact on the system. However, once a real load was applied, the problem became apparent. This is why care must be taken with configuration changes and customizations. Planning for performance must be done throughout the implementation project in order to have a positive result in the production environment.

Tactic #6: Plan for performance testing and tuning all throughout the implementation process. Late detection of major issues cannot typically be addressed correctly without significant project delays.

As much focus needs to be on the performance costs of the customizations as on the functionality benefits. This is a reality that is sometimes lost when rigid requirements are delivered that drive onerous performance outcomes.

First, it may not be necessary to log every single change to all records in the mission-critical system. Upon further examination, it might be found that in reality the only

changes that must be audited are any changes to the customer account balances—the transactional data. Administrative changes such as address changes or telephone number changes might actually be omitted from the auditing. That simple adjustment may reduce the load being exerted on the database servers by a significant amount and largely address the performance issue. If, however, the requirement really is that every change to the system of every type must be audited, then another solution is to change the sizing of database servers, or to consider configuration changes on the database that make for faster processing of the audit trail activity. This underscores another key fact:

Multiple layers of the system may need to be tuned for performance depending on the particular requirement and the impact realized from the solution developed.

Another important consideration is the timing of performance testing. If performance testing is done late in the implementation cycle, the situation for fixing issues is severely altered. Although best practices are sought, time constraints can force the implementation team to tactically address performance issues,

because extensive best practice rework would lead to unacceptable project delays. This is a core rationale for conducting performance testing throughout the implementation process to ensure that performance bottlenecks are identified early wherever possible. This early identification of issues leads to more strategic and proper resolution of issues, rather than tactical workarounds.

Because performance management crosses the lines between potentially multiple different teams within the customer's landscape, it is highly advisable to have a dedicated cross-functional manager. That person can coordinate performance management whether the necessary changes will be made at the database level, on the operating system, or on the front-end application. Seek guidance or assistance from Oracle ACS if needed.

High Availability

In pursuit of maintaining high availability, companies typically invest significantly in redundant servers and infrastructure. These redundant databases, hardware, licenses, and so on are intended to ensure that, in the case of a system failure, the production system will continue to be available.

Tactic #7: *Expect that performance tuning will need to be done on multiple layers of the technology stack. Performance cannot be achieved by only examining the application or the database. All technical teams related to the entire technology stack may need to be involved in the performance tuning exercises.*



A common reality, however, is that the high-availability capabilities of the systems are not fully tested. When a demand arises for the redundant equipment to be put into service, the rollover to the redundant system is often a much more time-consuming task than business users expect. In extreme cases, the redundant servers can even fail to meet the system failover or restore needs.

Such systems are mission-critical, and availability must be maintained. After you make the investment to purchase redundant equipment, it is vitally important to plan and test the actual process for rollover to the backup system. In cases where the process is not fully designed, look to the systems integrator or to Oracle ACS for assistance with planning and testing to meet high-availability needs.

Production Go-Live

A point of great difficulty in the lifecycle of many implementation and upgrade projects takes place during the cutover to production. Extensive work is done to ensure that the desired functionality is achieved in testing, and even in the quality assurance environments or preproduction environments. However, once the cutover process takes place, problems are often encountered.

Unexpected issues occur during production go-live due to a host of issues that, in many cases, are difficult to avoid. The production system has real users doing tasks that may not have been covered in the test use cases. The configuration and loading on the production system can be significantly different in subtle ways than the test systems. The live integration with other real systems can also add layers of risk that are only fully appreciated once the system is live.

Due to the many issues and risks inherent in cutover to production, you must be prepared for the eventualities. There are a number of things that can and should be done to prepare for mission-critical go-live events. For one, in preparation for go-live, all available test use cases should be testing in the environment that most closely mirrors the production environment. If issues are found in this preproduction environment, they should be sorted out before attempting to cut over to production. Secondly, load testing should be done to ensure that the configuration and customizations are not going to surprise in production with unexpected performance issues. If significant performance issues are noted during the load testing, these issues must be sorted out prior to the cutover.

Additionally, the people who are closest to the configuration decisions made should be available at the time of go-live. It is not advisable to cut over to a new production configuration if the technical people who understand the context around the decision-making are unavailable if issues arise. Furthermore, having the system DBA and administration support on hand can hasten adjustments if needed.

A further step that is strongly recommended is to have Oracle Support expertise onsite during the preparation for go-live and during the time period immediately following the go-live. In the case of technical difficulties, the Oracle ACS organization is the customer's best source of fast solutions to keep the go-live on track.

Support Tactics for Systems Delivery (Operations)

Event and Incident Management

Part of the service provided by Oracle Premier Support is assistance in managing incidents. When a problem is encountered, an SR should be raised on My Oracle Support (support.oracle.com). The Oracle support team will then begin a process to aid in problem resolution, and identification of relevant patches if appropriate. The customer can optimize the support provided with some simple tactics.

Service Requests

When logging an SR, be prepared to answer specific questions about the problem. The support organization has a better chance of resolving the problem quickly if it is fully explained and understood. In cases where a bug exists or a patch might be needed, this can only be determined once the problem can be reproduced or at a minimum fully understood.

Typically, the Oracle Support organization will want to review system logs. These logs, from the relevant database or application, will reveal additional information that will allow them to understand the situation.

Therefore, it is important for the party logging the SR to know where the system-generated log files are located on the system. Being able to quickly upload the relevant log files to the SR will result in a faster resolution time.

Take screen shots of any error messages you encounter. The codes and text associated with the application or system-generated messages will often reveal extra details that Oracle Support can use in addressing issues.

If an SR is related to a *mission-critical system*, meaning that the system affected is a live production system and the business is being adversely affected, inform the support person right away. Issues on production systems can be given a higher severity upon request. If the issue is negatively affecting a production system and a "Severity 1" status is given to the SR, Oracle Support will work around the clock to resolve the issue. If a Severity 1 SR is raised, Oracle's commitment to work on it 24/7 should be accompanied by a similar commitment from the customer. Only this ensures reduced time waiting on the other party. The point of contact at the customer site must remain available to answer questions, or the efforts on the Severity 1 SR will stall. Thus, when a really important issue is affecting a production system, a commitment from both the customer and Oracle Support can lead to a faster resolution in many cases.

When such a mission-critical situation exists, request a higher severity for the SR. This request can be made through the My Oracle Support portal or by speaking directly with a customer support representative on the phone. Also, provide a phone number and request a "call back" from support for direct discussion and a faster resolution time.

After Action Reviews

Typically, when any kind of support need arises, customers will log the needed SR and seek to address the event at hand. Once the issue is resolved, the issue will be closed with a sigh of relief. However, from an incident management standpoint there is value in assessing how well the process was managed from all sides. If there is a particularly good or bad example encountered where resolution time was either very fast or very slow, it may be useful to spend a little time dissecting what took place in order to either develop internal best practice standards or to work with Oracle to improve performance around future issues as they arise.

Customers can request a formal *After Action Review*, which will be supported in many cases. This is easiest when there is a technical account manager or customer manager available that can facilitate that review with the

Tactic #8: *When logging an SR, be prepared to provide system logs. Know where they are found and even upload them to the SR before being asked. These logs will provide additional information to the Oracle Support organization, which will lead to a faster issue resolution.*

Tactic #9: *To achieve a faster resolution time on SRs, provide a phone number and ask for a "call back." A direct communication early in the process between the SR owner and the customer can lead to a significant reduction in time to get the SR resolved.*



customer. In cases where the customer is relying on Oracle Premier Support alone, the After Action Review might need to be facilitated by the customer. Typically, such a review would involve key participants looking at the timeline associated with a particular event and assessing whether the right actions were taken, at the right times. Upon identifying weaknesses in the process, and making adjustments as needed, a general improvement in issue resolutions can be obtained. This is even more valuable than improving the resolution time on a single SR, because the results of an After Action Review can mean an across-the-board improvement in resolution times on many or all support needs.

Backup and System Restore

As with high availability, companies typically invest heavily in systems and mechanisms for restoring data in the case of a system failure. These systems are excellent at helping IT managers avoid sleepless nights, knowing that the data is undergoing regular backup. However, failing to periodically test properly the restore from the backup data can lead to catastrophic results when an actual restore is attempted and fails. Thus, it is mission critical to build into the implementation and production process defined intervals for testing the backup and restore system to ensure that a restore can actually be performed. Furthermore, these testing cycles also allow for analysis of the timing, so that a proper expectation can be set about how much downtime a restore operation will take. If the experienced restore time is not within the acceptable threshold, steps can be taken to proactively adjust the equipment or process.

As with all of the other mentioned technical themes, you can achieve support for your backup and restore planning and testing by engaging the Oracle ACS organization.

Patching and Upgrade Strategies

As issues arise and are resolved for every product, inevitably patches are developed. In addition, patches are built continuously to address any security threats that arise. The customer must plan for patching exercises as part of the ongoing steady-state production maintenance cycles.

Typically, patching will be fairly pain free when the patch application instructions are followed. However, in cases where there is a customization that might conflict with the patch or where the customer's specific configuration choices are somehow in conflict, there is always the chance that a patch may need to be reversed. This is also typically a straightforward task for the administrator.

By doing adequate unit and use-case testing on an up-to-date test system, issues can usually be identified before patches create any conflicts in production. Best practice demands that all patches must be tested in a test system before application to any production.

Tactic #10: Maintain a collection of test use cases that represent the core tasks performed on the production system.

For this reason, it is also best practice to maintain a collection of use cases that represent the core tasks performed on the production system. These can routinely be rerun in the test environment as a preparation for any patch application in production.

Upgrades generally are a larger task. However, they are also a great opportunity to correct configuration or customization mistakes from previous implementation iterations. It is highly advisable to engage Oracle ACS when planning an upgrade. Oracle ACS can provide assistance in identifying functionality improvements that apply. Furthermore, they can assist the customer with an impact analysis to ensure that risks are mitigated and the full value of the upgrade can be efficiently achieved.

Support Tactics for Continuous Systems Improvement

Third-Party Best Practice Reviews

Unbiased reviews of the decisions being made are essential, especially in relation to customization and configuration. Most complex systems will provide multiple mechanisms for achieving complex requirements. An integration partner will make decisions that will have a serious impact on supportability, maintainability, and performance.

Unbiased third-party reviews provide a checking mechanism to ensure that the fundamental topics are not being overlooked in pursuit of purely achieving functional goals. There are times when some requirements may actually create a dilemma for the other objectives. The role of a third-party review is to aid in good decision-making, which is a balance between best practice and the business objectives.

Oracle ACS can provide a wide range of third-party best practice reviews.

Technical Account Management

The task of guiding systems optimization is a complex one. Most IT systems today consist of layers of complexity. Thus, when a business system is in production, it typically will consist of hardware, networking, multiple databases, middleware and/or external systems connections and integration, plus the front-end application(s). Because not all customers standardize on a full stack from one vendor, this typically means that any problems or challenges will need to be researched and technical solutions coordinated across multiple teams and vendors. Because problems or failures often involve one or more layers in the stack, determining who has responsibility to fix or address issues can be extremely challenging.

To address this technical challenge, the business customer needs an unbiased ally who can help guide problem resolutions and systems optimization. This is precisely the role of an Oracle technical account manager. The technical account manager works to help the customer optimize systems in several ways:

1. Helps guide proper technical decision-making during the implementation phase
2. Provides guidance on how to attain optimal performance
3. Acts as a checking mechanism to ensure that best practices are being employed in the implementation
4. Provides proactive guidance to the customer and the systems implementation partner on supportability, maintainability, performance, security, patching, and upgrade planning and practices

Oracle ACS is a great source of support for the customer that wants to maintain a technical account manager to bridge the gap across systems teams, projects, and implementations. Once an implementation project is complete, external teammates will disengage over a short period of time and the role of production support will typically transition to the customer. The people closest to the decision-making related to the implementation may not be available at crucial moments. This is precisely why documentation around the customizations and configuration is mission critical. The cross-project gap that exists can also be spanned quite nicely by maintaining a technical account manager who knows the project history and decision-making. The technical account manager can aid in quickly understanding the technical context around the systems and their integration points and configuration.

Service Delivery Planning

Planning for support tasks such as patching, upgrades, and reviews can be described as *support planning*. It is highly beneficial to work with the Oracle technical account manager in developing a mutually agreed-upon



Service Delivery Plan that spells out support activities to be performed throughout the year. This is best practice when utilizing an Oracle technical account manager.

Remote Support

In cases where Oracle assistance is needed quickly to address issues, an Oracle ACS engineer can come to the customer site to look at the system, and work with the customer to address and resolve any issues or technical objectives. In many cases, issue resolution can also be expedited by enabling some services to be provided remotely.

Benefits of Building Remote Delivery into the Support Plan

A key benefit of remote support is speed. Oracle ACS has thousands of qualified experts covering nearly all Oracle technologies. However, the expert who might be needed to address a specific product or issue in a particular geography might need to travel to get onsite. By enabling remote access, travel costs and—more importantly—delays can be avoided. Getting the right expert involved at the fastest possible pace is best enabled with the option of remote delivery for services.

Remote Support Requirements and Planning

When emergency support is needed for the mission-critical system, there is typically very little time for making preparations for remote service delivery. The most effective route to remote support is through an Advanced Support Gateway. Setting up such a gateway allows for 100 percent secure access to the system logs and configuration data so that problems can be resolved quickly and efficiently with support from Oracle ACS.

Setup of the Advanced Support Gateway is a proactive task that should be done prior to any issues so that remote support is possible when the need arises. A host of beneficial services can be enabled once a gateway is established.

Planning for a gateway serves a dual purpose of enabling efficient remote support when needed, and also allowing Oracle ACS to offer a number of exclusive services that are only available through the gateway such as Advanced Monitoring and Resolution.

Tactic #11: A host of beneficial support services can be enabled once an Advanced Support Gateway is established.

Oracle Priority Support and Oracle Priority Support Advantage

Oracle Priority Support/Priority Support Advantage is one Oracle ACS offering that provides a great entry point into the services Oracle ACS can provide throughout the lifecycle. These services offer Oracle customers a highly integrated service relationship with Oracle that takes support to the next level.

With Oracle Priority Support/Priority Support Advantage, you are assigned an Oracle technical account manager who is there to manage problem resolution and escalations. Your technical account manager provides tailored assistance when SRs have multiple related incidents, or when the standard escalation process on an individual issue does not address your specific business needs. When incidents put your Oracle implementation, upgrade, or other project deadlines at risk, and significant business impact is likely, the technical account manager will be there to manage the escalation process.

Oracle technical account manager activities may include the following:

- Develop personalized knowledge of the account and product environment as the designated technical account manager for the organization

- Monitor the virtual team activity and escalate critical SRs within Oracle
- Organize communication between team members as necessary
- Communicate the status of critical SRs to the customer contact(s) and senior management

The Oracle technical account manager also leads formal reviews to assess performance against best practice standards, discuss business and technical plans, anticipate and avoid escalations, and offer insights on how to ensure that the customer is receiving the most value from your Oracle relationship. At each service review, the technical account manager compares Oracle's performance against the defined Priority Support Level Standard response targets. In addition, customers receive detailed support metrics that give deeper insight into the value of the support investment. The customer can access all of this collective experience through an exclusive, customized Oracle Priority Support/Priority Support Advantage portal (accessible 24/7), a single source for up-to-date support information and communications. The portal gives rapid access to product webinars, live chats with Oracle experts, customer-specific problem diagnosis and resolution metrics, account contacts, and past reviews.

Conclusion

Optimizing Oracle systems for operations is a complex set of tasks that requires proactivity, planning, and unbiased best practice review. Customers can employ many tactics to speed issue resolution and to drive optimal systems in production.

Oracle Premier Support provides enterprise-class support and continuous access to Oracle innovation. However, beyond reactive support and patches, you also must ensure that your systems are supportable, maintainable, scalable, and performing to the standards required. Proper system planning with a support mindset is necessary during the system implementation and into production. Engaging Oracle Advanced Customer Support (Oracle ACS) directly meets this need.

For smaller to average-size customers, Oracle Priority Support/Priority Support Advantage provide an excellent starting point for applying advanced support tactics. For the larger, more complicated customer situations, a wide range of additional advanced support services are available that can be tailored to the customer's specific needs.

While much of this white paper focuses on proactive planning and implementation that will lead to optimized systems in production, an additional class of services is also very relevant and important: numerous activities must be performed periodically during the production phase. After the customer achieves steady-state production, the system must be kept patched, upgraded, available, reliable, backed up, stable, and so on. Oracle ACS provides an extensive range of production support options to assist customers in maintaining and continuously improving their systems utilizing best practices. These best practices have developed over decades of Oracle ACS production support engagements with thousands of customers globally.



Oracle Corporation, World Headquarters

500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries

Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

-  blogs.oracle.com/oracle
-  facebook.com/oracle
-  twitter.com/oracle
-  oracle.com

Integrated Cloud Applications & Platform Services

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. This document is provided *for* information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0615

Advanced Support Tactics for Optimizing Oracle Systems
November 2015
Author: Ken Coman
Contributing Authors: Christian Kraus, Gerold Leeb, Birgit Kreuz, Sandro Tassoni

 Oracle is committed to developing practices and products that help protect the environment