I2R System Stabilization Tiger Team
C3 Performance Track
Summary Report

Author(s):	Melissa Liu
Contributors:	Rajiv Wani, Jyoti Sarin
Status:	Final Version: December Interim Summary
Date:	March 25, 2014

Table of Contents

1. Document Introduction

An analysis of P1 and P2 Alliance cases for I2R systems was conducted in the spring of 2006. C3 Performance issues were identified as a critical area contributing to P1 and P2 cases, and as a result, work commenced to tackle C3 performance issues. This document describes the work and results of that team.

2. Executive Summary

A systematic analysis of P1 and P2 Alliance cases for CACO-managed I2R systems was conducted in April/May 2006. The case data covered the period from August 2005 through March 2006. The case analysis identified key root cause factors behind the level of P1/P2 cases, the upward trend, and the volatility of the case load.

Due to schedule availability of key stakeholders, the P1/P2 case analysis readout was not conducted until the end of May. A "Tiger Team" to address critical stability issues was defined in June. Getting teams in place took a significant period of time; the initial Change Control and Case Management teams were staffed in late July; the C3 Performance team lacked critical resources until late August. At the time of this report, the Performance team is still in place and is planned to continue through delivery of the March release in 2007.

Initial Situation

In the case analysis, two root cause areas were related to performance: high utilization due to lack of proactive capacity planning; and sluggish response caused by data growth without sufficient proactive data and log file management practices (e.g., gaps in data archive/purge/retention policies and processes). Together, these categories accounted for 16% of the case load. An analysis conducted by Oracle confirmed these conclusions, and provided additional recommendations to improve C3 performance. (Lack of proactive maintenance policies were also noted in the case analysis as a possible reason behind the high rate of hardware failures for I2R systems; the other possible contributing factor mentioned was the sheer number of possible points of failure for I2R systems, given the large number of applications used by I2R.)

The bulk of the performance issues caused by data and log file growth were on C3; most of the capacity planning/utilization cases were on non-C3 systems. The analysis noted that C3 did not have a data retention policy; a data retention policy would indicate inactive data that could be archived and non-critical data that could be purged, as well as the schedules for each. The analysis also noted that the C3 performance issues were trending up. With Synergy on the way, it was noted that the data size and user base would only increase and the performance issues would likely continue to trend upwards – which was indeed the case.

Work Completed and Results

As of the December release, the Performance team completed 35 out of 47 planned specific improvements. A large proportion of those were delivered in the December release (21 fixes).

At the time of the original analysis, there were 1.2 P1/P2 cases per month related to C3 performance; by the July/August timeframe, this number had grown to 1.6 cases per **week**. Until the performance team was staffed in late August, cases continued to trend up while little work was being done to address them. A small set of fixes were included in the September release, and every release since has included performance fixes as well. The number of C3 performance-related cases is now beginning to trend down – back to 1 per week from mid-September through the end of November. This team is just now at the beginning of the downward trend, however, particularly given that the majority of the fixes to-date were just delivered.

Some fixes have resulted in performance improvements to specific business transactions or queries; some have driven improvements to overall health or performance of C3. Of the fixes with improvements to specific functions, the average improvement was **81%**. Complete results are listed in Section 4.

Remaining Work

Twelve planned improvements are pending for the January through March C3 releases. This estimate may decrease if intended fixes do not result in the projected improvements; it may increase if newly identified issues are indicated as critical by DLT and are therefore targeted for Performance team inclusion.

Recommendations for Next Steps

The upward trend in C3 performance issues that resulted in this track were due to process and organizational gaps. This team recommends that the following actions are taken to address these issues. Some of these actions have overlap or similarity in scope to tracks within the Critical Systems Resiliency initiative, as noted below.

- Institute proactive performance tuning practices for the C3 system – with clear responsibility for and priority on *implementation* of improvements, not just identification of issues – I2R OpEx
- Improve proactive monitoring, trending, and capacity planning across technology tiers (network, storage, authentication, web services, database, application, etc.) – CSR & OpEx
- Improve impact analysis and test coverage (addressed by “1.0” Change Control process); audit results of these new processes to ensure effective implementation – I2R OpEx
- Audit existing C3 test scenarios, test cases, and test data sets and revise to reflect current code-set and data distribution (small, medium, large) – CA-QA
- Consider doing fewer changes to allow adequate time and budget for comprehensive analysis of impacts and test requirements – CA RMO
- Address known performance environment issues – SIS and Infrastructure (CSR Test Track)

3. Overview

The I2R System Stabilization program (also called the "I2R Get Well Plan") was defined in July 2006. Some initial steps were performed prior to that, including a system change "lock-down" communicated June 21, 2006. The C3 Performance effort was a sub-team within that overall program. The background of both the program and the sub-team are described in the sections below.

3.1. Introduction to the Stabilization Tiger Team

In late 2005 and early 2006, system outages for I2R systems were at a high level (e.g. above 10 during a number of weeks), increasing, and volatile. At the time, not much was known about key causes, areas of greatest problems, and improvements needed. Although root cause analysis (RCA) was and is still performed for each case, these conclusions tended to be focused on each specific case at hand, rather than looking across cases to identify key themes and trends.

An analysis of P1 and P2 Alliance cases was conducted in April 2006 to identify the causes of the level, trend, and volatility of these cases and they key actions needed to address them. The analysis used case data from August 2005 through March 2006. Due to scheduling constraints, the actual review of the results did not take place until late May.

3.1.1. Results of P1/P2 Case Analysis and Oracle Analysis

In the spring 2006 case analysis, several major issues came to light:

- Change management issues caused 35% of the overall case load
- Lack of proactive maintenance policies and processes led to 16%, and probably contributed to another 8%
- Recurring application bugs accounted for 12% of the case load

In the case analysis, two root cause areas related to performance: high utilization due to lack of proactive capacity planning and lack of proactive data and log file management practices (e.g., significant gaps in data archive/purge/retention policies and processes). Both of these categories eventually lead to users experiencing poor performance. Together, these categories accounted for 16% of the case load, including both C3 and non-C3 systems. In addition, lack of proactive maintenance policies were noted as a possible reason behind the high rate of hardware failures for I2R systems, due to hardware not being proactively replaced as it ages (the other possible contributing factor mentioned was the sheer number of possible failure nodes for I2R systems, given the large number of applications used by I2R[1]).

[1] The count of I2R systems can range widely – the true count is not known. The I2R application repository, which lists all applications managed by CACO, has 158 I2R systems listed (or 136, if all C3 modules are considered a single application). However, this repository does not include TACSUNS systems or other I2R systems not managed by CACO (e.g., the PSIRT tool managed by I2R-IT); it also does not include "shadow IT" systems created and supported by the business.

The bulk of the performance issues caused by data and log file growth were on C3; most of the capacity planning/utilization cases were on non-C3 systems. The analysis noted that C3 did not have a data retention policy; a data retention policy would indicate inactive data that could be archived and non-critical data that could be purged, as well as schedules for each. The analysis also noted that the C3 issues were trending up. With Synergy on the way, it was noted that the data size and user base would only increase and the performance issues would likely continue to trend upwards.

One performance root cause issue that did not surface in the original analysis was that, although poor-performing SQLs were proactively identified, the fixes for those SQLs did not get prioritized into releases. For a complete discussion of performance management process-related issues that came to light during the conduct of the team, see Section 7.1.

At the time of the original analysis, there were 1.2 P1/P2 cases per month related to C3 performance. (By August, this number had trended up significantly to 1.6 per *week*.)

An analysis performed by an Oracle consultant[2] in June provided similar recommendations with respect to data and log file growth, and also provided a considerable number of Oracle best-practice recommendations to improve the performance of the C3 system. These recommendations, along with past P1/P2 cases and known poor-performing SQL statements, were the starting point for the C3 Performance track.

3.1.2. Objectives

The objective of the I2R System Stabilization program was to improve the stability of I2R systems by addressing key causes of outages.

3.1.3. Program Tracks

Two program tracks were initially defined: Change Control, and C3 Performance. A third track was quickly added based on very visible failures in Case Management (for example, two highly visible P1/P2 case outages were delayed around 24 hours before even being routed to the right IT organization to address them).

Track leadership and team membership changed many times over the course of the program, extending the program’s planned duration. Final team leadership was as follows:

- Change Control – Iain Campbell (SBS) and Danny Liu (I2R-IT)
    - Formerly led by Logan Wilkins, Ed Freeman, Ryan Schmierer
- Case Management – Chris Thomas with Igal Zadkovsky (both CACO)
- C3 Performance – Rajiv Wani (SIS) with Jyoti Sarin (I2R-IT)
    - Formerly led by Ed Freeman
    - This team was not staffed until late August

[2] Sumit Sarin, Oracle Consulting

- Program – Stephen Liem (I2R-IT). Supported through September by Melissa Liu, then by Rodney Rowell.

3.2. Performance Track

Background on the Performance track is provided below.

3.2.1. Team Objectives and Scope

The original case analysis noted that multiple I2R applications exhibited stability problems under high load, and that proactive performance management and capacity planning were a significant process gap. However, C3 performance issues were trending up significantly, and it was known at the time that the Synergy project was coming up and would increase the load on the already-taxed C3 system[3]. Therefore, this team focused specifically on C3 performance.

The objective of the Performance team, as agreed at the August 17 Sponsors meeting was to:

> *Improve performance of the C3 Oracle 11i instance*

The scope boundaries defined at the time were:

- *C3 11i production database (CTSPRD) and applications*
- *May include bolt-ons, portals, interfaces, etc. that affect C3 (11i) modules or the CTSPRD database, if those dependencies are identified to have performance issues*
- *The team will focus on the most critical C3 performance issues*

The focus on C3 performance versus all I2R performance came up for discussion multiple times over the course of the program, largely because the other tracks focused on all I2R systems. However, all key stakeholders agreed that C3 performance was the critical focus area, and the scope never did get broadened beyond C3.

In addition to the above scope, an outcome of the team must be to define the transition to a sustaining organization for performance monitoring and tuning – including defining roles and responsibilities for ongoing performance monitoring, tuning, trending, and capacity planning.

3.2.2. Team Focus Areas

The team used the following as sources to identify potential C3 performance fixes:

- Past P1/P2 Alliance cases
- Oracle recommendations from an analysis conducted in June 2006 by Sumit Sarin from Oracle
- Previously identified poor-performing or resource-hogging SQL statements

[3] Later sections note that, as SSM functionality was brought into C3, the number of C3 Performance P1/P2 cases did increase significantly.

- Noted process issues, such as those related to gathering performance statistics (the “Analyze” process) or to purging data in specific tables

3.2.3. Guiding Factors

From the above list of candidate improvements, several guiding factors drove the prioritization of which specific performance fixes would be addressed within the Tiger Team time frame, as well as the timing of each fix:

- Minor releases would be used to address smaller (“tactical”) issues, such as fixes to address specific P1/P2 cases or known poor-performing / resource-intensive SQLs. Localized Oracle recommendations and analyze process improvements could be addressed within Minor releases as well
- Major releases would be used to address larger-scale improvements, or fixes with the potential for broad impact. This approach provided extra lead time for development and testing, and made efficient use of regression testing already planned for those releases
- Because significant patching was already planned for the major releases (September, December, March) during the Tiger Team time frame, performance patches were prioritized by the following criteria:

  <u>Prioritization Factors:</u>

  - Oracle recommended performance patch list – patch is a readily available fix
  - Applicability in C3 environment – patch applies to C3 functions
  - Association with P1/P2 Cases – patch will resolve outstanding P1/P2 cases
  - Severity of the performance problem without the patch

  <u>Exclusion Factors:</u>

  - Patches were required to be “one-off” patches to be included as a Tiger Team improvement, requiring no other patches to be applied before or in conjunction with the candidate patch (i.e., any prerequisites must already be met in production or in the planned release)
  - Patches were required to not have cross-functional impact outside the business function(s) directly impacted by the performance issue, due to the risk to other system components and testing that would have been required
- In order of importance, the prioritization factors considered for specific program issues (in order of importance) are listed below. The nature of the fix and **testing requirements** were considered when scheduling specific fix into a tactical vs. CFP release:

  1. Specific program issues leading to **system level stability issues** (i.e., system outage), with number of occurrences

2. Specific program issues leading to **system level performance issues** (i.e., program leading to possible slowness for the system but not causing a hard outage), with number of occurrences
3. Specific program issues leading to **program level stability issue** (i.e., program itself not completing but not impacting the system as a whole), with number of occurrences
4. Specific program issues leading to **program level performance issue** (i.e., program completing with slow response), with number of occurrences
5. Improvements sought to improve response time without specific SLA or system stability issues

3.2.4. Team

The Performance team was the last Tiger Team track to be staffed with all the necessary roles. The team was largely in place by the end of August 2006. From late July until late August, the team had only part-time participation (including the lead role), and did not have the membership and momentum to make progress on team deliverables. Addition of two key members in late August, Rajiv Wani (SIS) and Jyoti Sarin (I2R-IT) enabled the critical work of the team to begin.

Unlike the other two tracks, the Performance team is not scheduled to disband until after the March 2007 C3 release.

The Performance team consists of the following members:

Team Member	Role
Rajiv Wani (SIS)	Team Lead
Jyoti Sarin (I2R IT)	Patch analysis and implementation life-cycle, infrastructure recommendations analysis, P1/P2 case analysis, initial issue prioritization, release assignments
Michael Hollar (CA QA)	CA QA – Performance Testing
Dilip Agarwal / Moe Jabri (CA QA)	CA QA – Functional Testing
Jaykay Selvaraj (Infra)	Coordinating point with DBA organization
Sudhir Jacob (CACO)	CACO – Client Support team representative, case technical analysis data
Shekhar Arcot (CACO)	CACO – Release Management, tactical release co-ordination, post-production results measurement
Ken Sharpe (SBS)	SBS representative
Belinda Jeffrey (SBS)	Business representative for SSC

Team Member	Role
Manjit Tiwari (I2R-IT)	Performance Analysis / Engineer for OM, Returns, Planning areas
FNU Shilpa (CACO)	Selectica IT track representative
Andrew Bulatowicz (CACO)	CACO representative
Randy Wolfgram (SIS)	Capacity Planning (amps hardware upgrade)
Chong Kim (11i-Perf)	Performance Analysis
Ajay Jose (I2R-IT)	Solution Engineering SRM developer
Melissa Liu (PRTM)	Interim (December) Summary Report and initial P1/P2 Case Analysis
Sumit Sarin (Oracle)	Oracle C3 Analysis and Recommendations

4. Interim Progress and Remaining Improvements

4.1. Schedule for Improvements

All improvements in this document are as of December 2006. The team will continue its efforts through the March release.

As of December, 35 fixes had been implemented, out of 47 fixes planned at the time. This progress represents 74% of planned fixes. The schedule for improvements and progress to-date are shown below.

	Sept - Major	Oct – Minor	Nov – Minor	**Dec - Major**	Jan – minor	Feb – minor	**Mar - major**	**Total count**
P1/P2 case based fixes	**4**	3	4	**16**	4		**3**	34
Oracle recommendations					1	2	**2**	5
Process fixes		1	1					2
Oracle apps performance patches				**5**				5
Infrastructure upgrade			1					1
Current Count	**4**	8	14	**35**	40	42	**47**	47
Projected Cumulative %	**9%**	17%	30%	**74%**	85%	89%	**100%**	

Progress as of December 2006

The original schedule had a higher number of planned total improvements. The baseline schedule included estimates of problems that would emerge later and would require newly-identified fixes (specifically for the January and February releases). In addition, some planned fixes were de-scoped due to inclusion in later work, due to realization that the planned fix would actually not achieve the planned effect, or due to lack of readiness of vendor-supplied improvements. New problems and fixes could be identified between now and the March release, which would affect the planned number of total improvements and the percentage of improvements completed.

4.2. Summary of Results To-Date

C3 Performance issues peaked during the July-August period and are now trending down. The original P1/P2 case analysis showed **1.2** C3 performance cases per month, and noted that the cases were trending up. The Performance Tiger Team track did not get staffed with critical resources until late July, delaying the start of much-needed performance improvements; in the meantime, the upward trend in performance cases accelerated and peaked at **1.6** cases per *week* during the July-August period.

The Performance team began to implement a small number of changes with the September release, and the team has been building momentum since. C3 performance cases were reduced to **1.38** cases per week in Q107, trending down to 1 per week in the last portion of the period.

The read of the team at this time is that the cases are beginning a downward trend; as the work of the team continues, the performance should continue to improve. *However, significant new functionality and major patch efforts within the C3 major releases over the last year seem to create new recurring cases with each release* (e.g., contract summary with the June release, workflow with the September release). These changes are driven by other teams, not the C3 Performance team, but the results affect the metrics used to gauge the success of the Performance team. Significant new functionality carries risks that should be expected; however, *there may be improvements now needed in the release processes and associated quality processes* that were not as significant at the time of the original analysis. With the expectation that additional functionality will be brought into C3 as precursors to and components of Synergy, it would be prudent to identify and address these issues to prevent new recurring issues subsequent to each of the 2007 major releases.

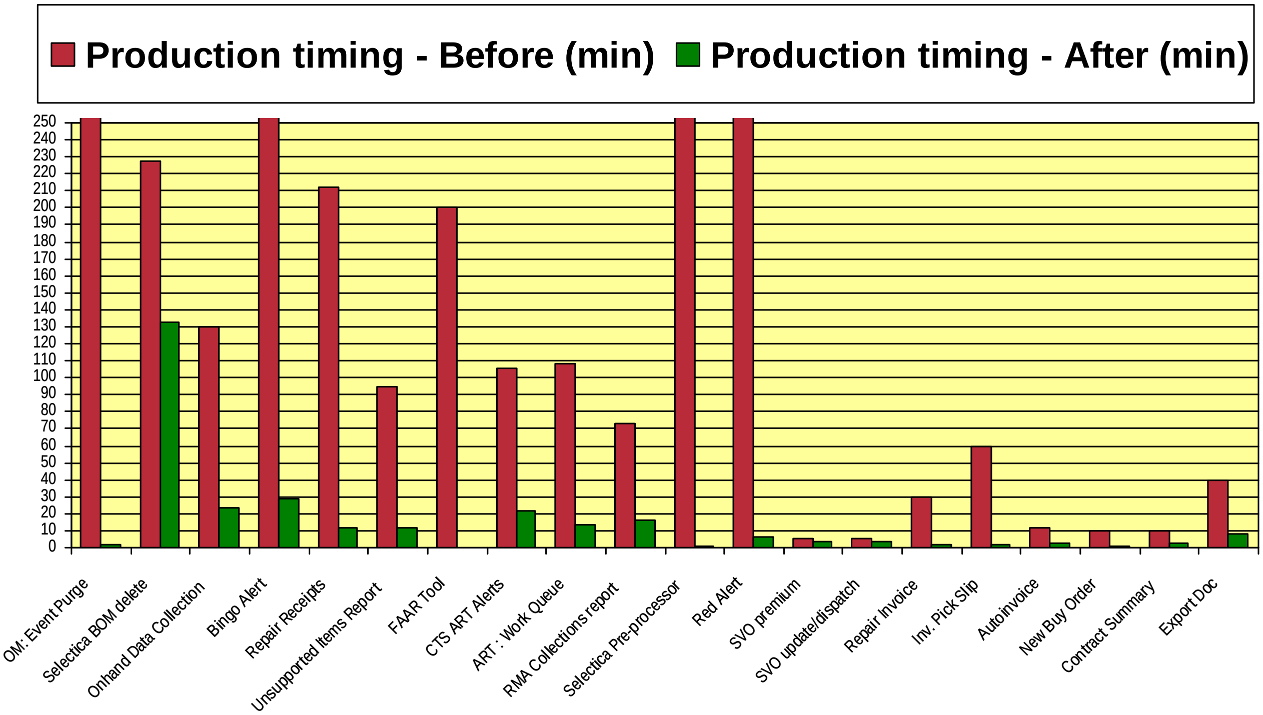
Of the changes that decreased the time needed to complete specific functions (as opposed to improving stability only), the elapsed time was reduced **81%** on average.

4.3. Performance Improvement Detailed Results

Specific fixes can have a variety of impacts on performance and stability, alone or in combination:

- The fix may improve the elapsed time to complete a specific function or query
- The fix may re-enable functionality not used (for example, because an function previously took too long or could not complete, requiring a manual workaround that avoided using the functionality)
- The fix may prevent errors that resulted from performance issues
- The fix may address load issues, hung threads, latches, or accumulation of records; in these cases the affect may be less to reduce the completion time for a specific operation than to improve the overall health and response time of the C3 system

Before and after performance is diagrammed below for the fixes that include an improvement in the duration of a specific operation. December changes (21 total) are shown at the left of the graph; September-November changes are shown at the right.



A detailed table with complete results, including stability and system health impacts, follows:

Issue Fixed	Release	Pre-fix condition	Post-fix status
Selectica BOM delete program	December	• Periodic rollback segment error, causing DBA monitoring and killing the program at times • 223 min Execution time	• Eliminated rollback error, significant reduction of rollback usage • Execution timing reduction to 133 min (41% reduction)
OM Event purge program error	December	• 10-20 hrs execution time	• 1-2 min execution time (99%+ reduction)
Onhand data collection interface program	December	• Execution time of 130 min • Program runs every 4 hours, sometime a single run would run into next run, breaking SLA • Period program failures with rollback error	• Execution time reduced to 24 min (80% reduction) • No overruns, no rollback-related failures
COT Admin – library contention from query without bind variable	December	• High usage and fragmentation of shared pool	• Significant reduction of the shared pool usage
SRM – send bingo alert performance issue	December	• Execution time of 271 min, high load SQL and high I/O	• Execution time reduced to 29 min, reduced load and I/O (reduction of 89%)
SRM – Send red alert performance issue	December	• Program placed on hold in production due to very long execution time (24 to 48 hours) • Infinite loop bug in a specific scenario	• Execution time reduced to 6 min (99%+ reduction), scheduled execution restarted • Fixed infinite loop bug
Repair Receipts program performance issue	December	• 212 min execution time	• 12 min execution time (94%+ reduction)
SVO update and dispatch tuning	December	• Execution timing over 5 min at times, leading to intermittent timeout • High load from SQL query	• Execution time reduced to 3-4 min, timeouts eliminated • High load SQL tuned
Unsupported item report	December	• 95 min execution time • Daily program execution	• Execution time reduced to 12 min (87% reduction) • Weekly program execution deemed sufficient
SVO premium submit (with labor line) tuning	December	• Execution time over 5 min, leading to intermittent session timeout	• Execution timing reduced to 3-4 min, timeout eliminated
FAAR tool tuning	December	• 200 min execution time • High load SQLs, high temp space usage	• Execution time of less than 1 min (99%+ reduction) • Reduction in load and temp space usage

Issue Fixed	Release	Pre-fix condition	Post-fix status
Alerts (pre-Alert and Auto Cancel) tuning	December	• 106 min execution time	• 22 min execution time (79% reduction)
Work queue generation tuning	December	• 108 min execution time • High load SQL	• 13 min execution time (87% reduction) • High load SQL tuned
RMA collections report tuning	December	• 73 min execution time	• 16 min execution time (78% reduction)
Selectica pre-processor combo tuning	December	• 344 min execution time • High load SQL	• Execution timing reduced to 1.25 min (99%+ reduction)
Export Doc generation (api) performance issue	November	• **30-40 min.** to generate export doc leading to **SLA issues** with partners (25-30 seconds per line)	• **5-8 min**. to generate export doc (~5 seconds per line)
Contract Summary –installed base	November	• **400-600 min.** execution time daily • Contract summary build 6 Hr. SLA broken for large data set	• **90-180 min.** execution time • SLA reinstated • Reduced I/O and CPU usage on ohms
Contract summary backup tables purge rollback issue	November	• **Rollback error** in purge of backup tables program • Long execution time and large records count (75M+ records)	• Disabled backup program, truncated tables • Relieved rollback, CPUs, I/O, storage usage
Excessive logging in P2RS interface	November	• Accumulation of millions of records (purged periodically)	• Stopped accumulation of logging records • Reduced I/O inserts/deletes
PRD (AMPS) node hardware upgrade	November	• Load range 1-2.5, **peak load of 3** • **Peak memory usage ~95%** • **Load above 2 with real deterioration of response time**	• Load range 0.1-1, **peak load of 1.4** • **Peak Memory usage ~75%**
Auto-Invoice Program	October	• Program error with rollback issue after **12 hr+ execution** • **$U placed on hold** for US and BV operating units • Manual GL entries by business	• Successful completion in less than **3 hrs.** without rollback issues • **$U enabled** for US and BV operating units • Manual entries process ended
New Buy Order import	October	• 50M+ PO interface error records • Performance degradation in receiving transaction processor	• No new errors for the specific scenario • Timing improved from **10 min. to 1 min.** for comparable data in receiving transaction processor

Issue Fixed	Release	Pre-fix condition	Post-fix status
Process change – Weekly analyze by DBAs to use specific PCT sample values	October	• Weekly SQL execution plan issues leading to P2 cases • High load for various SQL's	• No repeat SQL execution plan issues stemming from analyze
Report to identify purge qualified tables status	October	• Accumulation of large records count (~50M) in custom interface tables • Intermittent failures in purge programs	• Report helping CACO with early identification of focus areas for purge
SVO Status Tool	September	• Front-end tool **timeout after 5** min. • Continued high load and latching from query for up to 30 min after front-end timeout	• **No** front-end tool **timeouts**, response in **under 5 min** • Reduced load from tuned queries
Repair Invoice Form	September	• **30 min**. or longer execution time with high temp space usage	• **~2 min.** execution time, reduced temp space usage
Contract Summarization program	September	• Inserting large number of records • **Rollback error** in contract summary program for specific data condition	• Removal of insertion of excessive (duplicate) data • Elimination of rollback error
Inventory Pick Slip report	September	• **60 min**. or longer execution time from high load queries	• **~2 min.** execution time, significant load reduction

4.4. January to March Roadmap

The following items the identified candidate changes for January through March releases as of December 18. Items may be added or deleted as further analysis is done.

Release	Improvement	Type	Performance / Stability Impact
Jan	S2C LP Contracts Stage interface	Application	Improve performance of this program. Reduce overall load
Jan	SQL used by DBAs for tracking inactive sessions	Infrastructure	Improve the stability of system by enabling proactive action
Jan	Reclaim unused space, defragmentation and chaining	Infrastructure	Improve performance of system by making efficient data retrieval for queries. Improve stability by decreasing the overall system load
Jan	Update Standard Costs	Application	Improve the performance and stability of this program. Improve stability of the system by reducing the amount of rollback used.
Jan	Cost Import Process (problem at quarter end)	Application	Improve the performance and stability of this program. Improve stability of the system by reducing the amount of rollback used.

Release	Improvement	Type	Performance / Stability Impact
Feb	Use FND_STATS to optimize gathering of object statistics based on size	Infrastructure	Reduce the amount of time taken by analyze process. May improve overall performance of application programs
Mar	Performance issue with pick release program	Application	Improve the performance of this program
Mar	9i fast response optimization by using FIRST_ROWS_N	Application	Improve the performance of front-end queries run over WAN. Further analysis needed – may not be applicable to C3 implementation
Mar	Shared Pool Optimization	Infrastructure	Improve the overall performance of the system
Mar	Enter Requisition Form performance	Application	Improve performance of this program. This is currently running below SLA requirements
Mar	JTF Notes query from SR Form and Task Form	Application	Increase system stability due to reduction in the amount of temp segment usage. Currently, infrastructure team is mitigating the risk by managing the sessions that consume high temp segment space. This is a contained P1 issue
Mar	Inventory Pack Slip Report	Application	Current performance of this program is below SLA level. This tuning effort will help the program meet SLA and improve the performance of this business critical program

5. Roles and Responsibilities

The table below shows the current roles and responsibilities with respect to performance management for I2R systems. There are gaps in proactive monitoring, trending, tuning, and capacity planning – across technology tiers and IT organizations – that are not defined in current roles and responsibilities. Section 7.1, "Findings", describes these gaps.

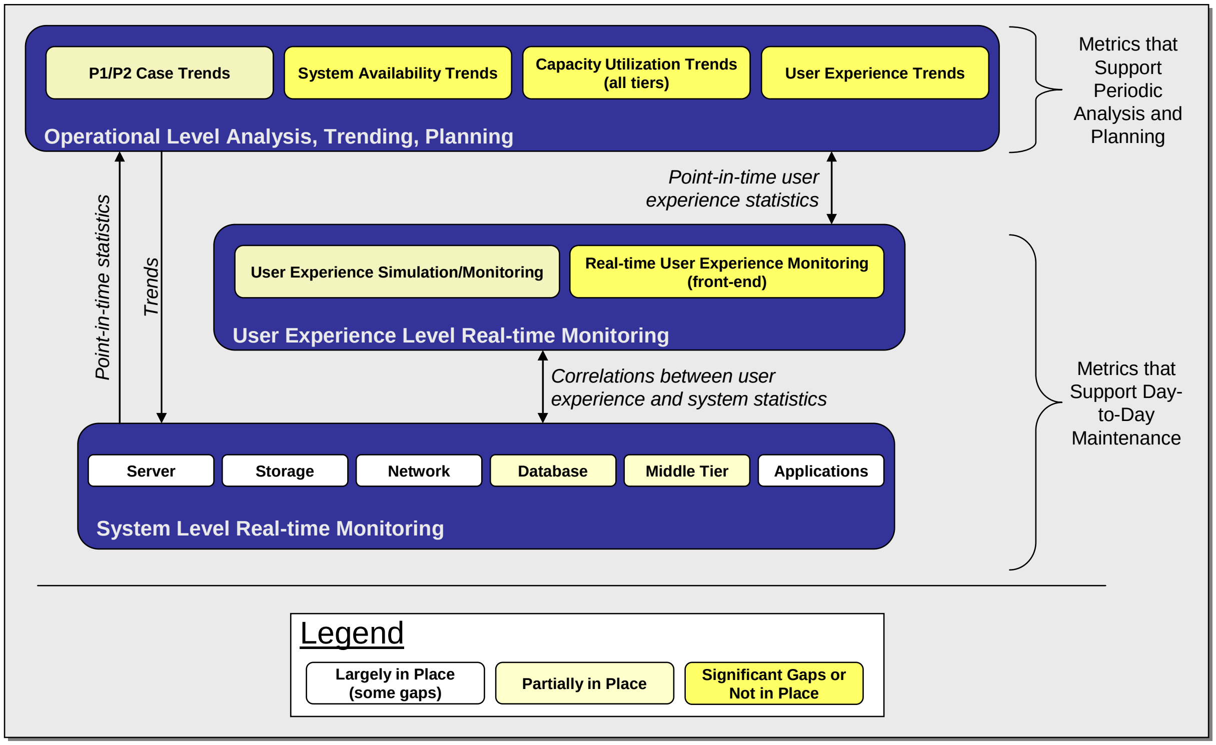
Role Name	Responsibilities
CACO	• Within Tiger Team: Performance issues technical analysis, solution development and testing for tactical release fixes • Production support for performance issues
DBA	• Database monitoring, production issue resolution participation
System Administrators	• Monitor and maintain hardware, operating systems, and storage for business systems servers
I2R-IT	• Solution development, code fixes, testing and deployment for performance issues (mostly in CFP releases)
Infrastructure – Network	• Provide network support for <company>
Infrastructure – AFS	• Monitoring and maintenance of middle-tier services for application servers (web services, authentication, etc.)
SIS Performance Team	• Solution development for application performance issues; provide tuning expertise to tracks and CACO for complex issues
11i Performance Team	• "Top SQL" execution (most resource-intensive SQL queries) to identify poor-performing SQL statements
CA QA	• Performance load testing / benchmarking

6. Proposed Metrics

Ongoing performance monitoring and metrics gathering is essential to achieving proactive application management and a sustainable model to maintain and manage system stability.

6.1. Background

Metrics gathering for monitoring, predicting, and ensuring systems health can be done on three levels, as shown in the picture below:


P1/P2 Case Trends
System Availability Trends
Capacity Utilization Trends (all tiers)
User Experience Trends
Operational Level Analysis, Trending, Planning
Metrics that Support Periodic Analysis and Planning
Point-in-time user experience statistics
Point-in-time statistics
Trends
User Experience Simulation/Monitoring
Real-time User Experience Monitoring (front-end)
User Experience Level Real-time Monitoring
Correlations between user experience and system statistics
Metrics that Support Day-to-Day Maintenance
Server
Storage
Network
Database
Middle Tier
Applications
System Level Real-time Monitoring
Legend
Largely in Place (some gaps)
Partially in Place
Significant Gaps or Not in Place

6.1.1. Operational Level Metrics

Operations-level metrics analysis uses statistics gathered at systems and user experience layers and results in analysis and trending across all *applications, all systems, and all tiers*. These analyses empower decision makers to take proactive measures to ensure continued delivery to performance requirements and to define specific improvement focus areas.

Some of the key measures include the following:

Component	Definition and Usage
P1/P2 Case Trends	• Measures number of P1/P2s events and their recurrence, associated outage durations, business impacts (by application and by the breadth of affected user base), and root cause categories. Another dimension to measuring P1/P2 outages can be to estimate business dollars lost due to a P1/P2 event.

Component	Definition and Usage
System/Application Availability Trends	• Measures system unplanned outage downtime by application/functional area/tool. This metric when trended can predict which applications are more volatile and hence need more focus
User Experience Transaction Trends	• Helps identify user-facing transactions that are falling behind on SLAs or trending down toward SLA minimum levels. Part of Monitoring track for Critical Systems Resiliency (CSR) Program
Capacity Planning and Trending	• Provides information for periodic capacity planning across all systems and all tiers to help ensure that capacity is added when needed or can be utilized elsewhere when capacity exceeds projected needs • Requires input from both infrastructure and application teams to create an effective projection of required capacity. This exercise, in general, benefits from the trending information based on system metrics and projection information based on application/user base growth trends and business direction

6.1.2. User Experience Level Metrics

The business perceives availability and performance based on user experience, which may not be adequately reflected in even the most complete systems-level metrics. User experience monitoring seeks to fill this gap by interacting with applications as a user would and tracking various user-experience-level parameters. This information may support early detection and root cause analysis for issues before they become business impacting, and can enable monitoring of and delivery to SLA commitments. User Experience monitoring can include the following:

Component	Definition and Usage
User Experience Simulation	• Emulates key business transactions in a controlled manner, going through the predefined transaction scenarios to create performance baselines that can be benchmarked against current and historical trends • Provides useful user experience information even before users perform similar transactions on the system • In the EMAN tool, transactions could be used simulate user experience by broadening the set of transactions used and the sites from which they are sent
Real-time End-User Transaction Monitoring	• Captures key user performance metrics, such as round-trip time, availability and application errors, from all users, at all locations, at all times • Helps identify business impact of an event by drilling down to precise number of users affected

6.1.3. Systems Level Metrics

System-level statistics are crucial to avoiding and troubleshooting potential and actual performance bottlenecks. They also support trending analysis for capacity predictions, definitions of normal/abnormal behavior, and early detection of developing issues.

System-level metrics need to be gathered across tiers and infrastructure components, including: database, application, storage, middle-tier, servers, hubs, routers, NAS, SAN and other components. Listed below are examples of metrics in this category:

Component	**Examples**[4]
Database-tier Metrics	• Top 5 wait events, v$sysstats, v$session_stats • RAC related metrics • SGA (Shared Global Area) Metrics – Cache hit ratios, Shared Pool, DB Buffers, Latch waits etc. • Key Database Processes – DB writer , Redolog and Archivelog speed etc. • OS Statistics – CPU utilization, memory utilization, run queue, load etc. • Errors/Exceptions
Middle-tier Metrics	• CPU and Memory Utilization • Load characteristics • JVM Utilization • Errors/Exceptions • Java garbage collection
Network LAN/WAN	• Latency • Throughput • Errors/Exceptions
Storage Capacity and performance	• Space utilization • Hotspots • I/O performance • Errors/Exceptions
Application Metrics	• Top-SQL • Workload Metrics – OLTP and Batch- concurrent manager queue performance, batch job times and distribution, OLTP load characteristics • Batch Process timing metrics – Concurrent job execution timing-leading to collection of data for trending. This metric can assist in issue predictions, capacity planning and workload management • Errors/Exceptions

6.2. Recommendations

Short and long-term recommendations for metrics are documented below:

[4] These lists are not exhaustive but do include some key metrics for each area

Tier	Immediate	Long Term
Operational Level Analysis, Trending, Planning	• Continue and enhance **P1/P2 case analysis and trending** • Institute **system availability trending**	• Broaden and enhance **capacity planning across all tiers** • Institute **user experience analysis** and **trending**
User Experience Level Real-time Monitoring	• Investigate use of **EMAN** capabilities for **broader transaction monitoring** that better simulates user experience (more transactions, site-specific transactions)	• **Evaluate better tools** for user experience monitoring. (Available products include Mercury, Quest, BMC and Oracle.)
System Level Real-time Monitoring	• Identify gaps and **enhance metrics** to include all tiers, all key metrics, and metrics pertaining to specialized setup such as RAC	• Build **metrics repository** to capture point-in-time statistics for use in trending, tuning, and capacity planning • Provide information to **correlate real-time systems statistics with user-experience**

7. Next Steps

This section describes findings (new issues or outstanding issues not addressed by the team), recommendations, next steps, and lessons learned over the course of the project.

7.1. Findings

The P1/P2 case analysis identified that C3 performance issues were trending up. Some root causes were identified based on the P1/P2 case data; however, complete performance management process decomposition and reengineering was not the focus or scope of the analysis. The work of the Performance team brought together IT organizations that were not working particularly well together before, and as a result, a fuller picture of the situation has emerged. This section describes process issues that contributed to increasing performance problems. Also included are findings from process issues encountered during implementation of the Performance team improvements, noted under the “SDLC Process Gaps” section.

Real-time Performance Monitoring Gaps

- **Failures to Address Issues Identified by Proactive Monitoring.** The 11i-Performance team proactively identified the most resource-intensive SQL statements needing improvement. However, their implementation focus was mostly for non-C3 Oracle environments elsewhere in <company>. The SIS team had a Performance Management team that reviewed this information and created I2Rs and TDs to correct the issues and notified the CACO organization; however, the fixes were not were prioritized into releases and implemented. The consistent I2R behavior to prioritize enhancements over fixes was noted in the case analysis as contributing to the large number of recurrences and growing LTF backlog – but it also contributed to the significant increase in performance issues over time.
- **Inadequate Database Server Alarming to Prevent P1/P2 Case Creation.** The Infrastructure and DBA organizations did have alarms set against specific database internal metrics and overall health of the database (load, temp space, memory, database contention, latching, etc.). However, alarm levels were not always suitable to support to proactive resolution. By the time an alarm was triggered, a case may already have been created – or, would be created by the time the DBA team would respond to the alarm. Some alarm settings were adjusted over the course of the program as P1/P2 case outages were experienced.
- **Inadequate Monitoring across Other Technology Tiers.** Proactive monitoring of other tiers was absent, inadequate, or unclear. Other tiers causing outages due to load, capacity, or performance failures included: authentication services (Siteminder, LDAP); application servers (Oracle, custom CCX/CCI); network services, and storage services. The proactive monitoring of specific services such as network and storage is unclear and

does not have direct integration with other support groups such as C3 support.

Performance Trending Process Gaps

- **Unclear Ownership for Trending and Proactive Response.** Ownership for trending of performance, utilization, and resource consumption is not clear – for *any* of the technology tiers. For C3 performance, EMAN monitoring data could have been used to predict future issues. However, no one actively monitored trends in EMAN-created P1/P2 cases or other EMAN performance data. As a matter of fact, P1/P2 cases created by EMAN by delayed EMAN monitor response tend to be ignored as "non business impacting," if the application is actually up and running. Increasing numbers of these delayed responses could be a predictor of an issue somewhere between the request and the response – due to trends in network capacity, application server load, database server load, or elsewhere the technology stack – regardless of source, the trend could be used to spot a growing problem that requires attention and diagnosis. It is unclear which IT organizations – if any – are monitoring trends for their technology areas.

Capacity Planning Process Gaps

- **Lack of Proactive Planning and Evaluation.** Server capacity for the C3 database server CPU, memory, etc. is planned and managed by the SIS Performance team and Infrastructure teams periodically, typically before each release. This evaluation takes into account both the changes planned for the release as well as the recent load history. However, there is no active planning and evaluation of the other tiers (application servers, network, authentication services, etc.) that is done in a direct way with C3 involvement. It is unclear what infrastructure processes exist for monitoring and periodic capacity planning. There were issues experienced during the program with the C3 application server memory consumption as well as network capacity.

Performance Tuning and Fix Implementation Prioritization Process Gaps

- **Inadequate Prioritization and Resourcing for Performance Improvements.** Immediate resolutions to address P1 performance or stability issues were implemented using the emergency bug fix (EBF) process. However, long-term fixes to address the P1 issues did not get prioritized into releases. In addition, P2/P3 and identified poor-performing SQLs ("Top SQLs") were logged but not prioritized due to their non-P1 nature and perceived higher value for functional fixes and enhancements. CACO owned implementation of the LTFs, but the prioritization process did not result in resources allocated to implementing those fixes. At the time, CACO owned responsibility for enhancements as well as bug fixes; enhancement work was transition to IT teams over the course of this program, with only SRM remaining to be transitioned. However, the fix implementation is subject to business prioritization, which tends to value functional over performance fixes. Although any single performance issue may not seem critical, unaddressed performance issues collectively form a large liability.

- **Communication and Consolidation of Information.** The systems information (metrics and issues) collected by various IT groups (CACO, I2R-IT, SIS, DBA, etc.) was not fully communicated among various cross-functional groups, and there was no effort to consolidate issues into a single meaningful, prioritized action list.
- **Inadequate Inclusion of Performance Improvements in Releases.** I2R release work remains overwhelming biased toward implementation of new functionality. Currently, only issues that show up in the functional performance testing are being addressed. All other application performance issues and other tier issues are not actively addressed by releases. I2R/TS organizations spend significantly more energy and resources toward implementation of the new versus maintenance of the old.
- **Fix Implementation Lead Time Too Slow to Prevent Recurrences.** New recurring issues have surfaced over the course of the program. Even when prioritized for implementation, the RCA-fix identification-prioritization-development-test-release lead time is sufficiently long that new issues may encounter recurrences before fix implementation. Examples during the program: contract summary, workflow, SR-RAT.

<u>**SDLC Process Gaps**</u>

- **Inadequate Impact Analysis and Test Coverage**. Analysis of the impact of fixes to other programs or to the overall configuration (e.g., size of JVM memory needed to support a new/enhanced application) is inadequate. The RCAs for many issues still point to inadequate testing resulting from unidentified impact, as was true at the outset of the program. For example, a performance fix in November broke another index in the same program. Clear ownership and methodology for impact assessment is needed. There were many contentious meetings where the responsibility for impact analysis and implications for testing were debated. The Performance team embedded IT track experts into the QA processes to mitigate this gap. The organization lacks the knowledge to do proper impact assessment. Specific gaps include:
    - Inadequate identification of impact to customized Oracle code when Oracle patches are applied
    - Impact analysis of database object changes (such as creation of indexes) needs to be both more rigorous and sufficiently translated into testing scripts
    - Inadequate code coverage in testing; while not all production configuration scenarios are possible to test, experience of the team shows that required scenarios are not always included in test scope

- **Outdated Test Scenarios and Inadequate Test Coverage across Data Scenarios.** The data set used is not complete enough to represent the many different types and sizes of data encountered in the C3 production system (big contract, small contract, etc.). Test scenarios and test cases need significant updates based on knowledge of recent production environment and data. Some test scenarios appear to be from the original set at C3 go-live four years ago.
- **Insufficient Test Environments for Performance Testing**. Performance environments do not adequately represent Production. The CTSPRF environment is a single node cluster, and the scale, number of CPUs, configuration, etc. are not comparable. Other differences between Production and Performance environments include items such as: storage type, speed, database server size (CPU, memory), database size (memory), interconnect network, availability of boundaries. In addition, there is a single performance testing environment; however, multiple releases are developed in parallel, creating contention for the environment (as was the case for the December and March releases).

7.2. Recommendations

The following recommendations are in priority order.

1. **Recommendation:** *Institute proactive performance tuning practices.*

   Priority: High

   Owner: Operational Excellence

   Key Organizations and Roles: DLT, SIS, CACO

   Currently, poor-performing SQLs are identified but not addressed (outside of the Tiger Team efforts). Some real-time monitoring is done, but trending is lacking. Clear ownership for identification and resolution of performance issues and ongoing performance tuning is needed. Suggested metrics for trending are included in Section 6.

2. **Recommendation:** *Improve proactive monitoring, trending, and capacity planning across technology tiers.*

   Priority: High

   Owner: Operational Excellence, in concert with other IT organizations

   Key Organizations and Roles: Infrastructure, SIS, I2R-IT

   Proactive monitoring across technology tiers is needed. Many issues resulted from load/capacity issues across tiers. Periodic capacity planning *is* in place for the C3 database server, but not the application server. Existence of and ownership for periodic capacity planning for other servers and across technology tiers is unclear. Site-specific performance monitoring, trending, and capacity planning is also needed; there is potential to use existing EMAN functionality for site-specific monitoring and trending. Suggested monitoring and trending metrics to support proactive planning are included in Section 6.

3. **Recommendation:** *Improve impact analysis and test coverage; audit results to ensure effective implementation.*

   Priority: Medium-High

   Owner: Operational Excellence

   Key Organizations and Roles: DLT, RMO, QA

   The new Change Control process includes a framework for risk and impact assessment that is linked to test strategy and planning. However, this is a new process and it will take time for the organization to become effective at it. To fully achieve the desired outcomes, the impact analysis and test documentation outputs should be audited to ensure the quality of the analyses. Otherwise, participants may produce the deliverables, but without conducting the analysis at a sufficient depth.

4. **Recommendation:** *Audit existing test scenarios, test cases, and test data sets and revise to reflect current code-set and data distribution (small, medium, large)*

   Priority: Medium-High

   Owner: QA

   Key Organizations and Roles: Subject matter experts in business, IT, SIS

   Outdated functional and data scenarios need revision.

5. **Recommendation:** *Consider doing fewer changes to ensure adequate time and budget is allocated for very comprehensive analysis of impacts and test requirements*

   Priority: Medium-High

   Owner: RMO

   Key Organizations and Roles: DLT, QA, Risk Analyst

   Repeated discussions over the course of the project highlighted the amount of work the teams felt would be required to properly assess the impact of changes, and the fact that they “did not have time” to do it – which probably contributed to a nearly uniform desire to have it be someone else’s job. Unless adequate time is planned into the work effort, the analysis may be performed too quickly and at too superficial a level. This possibility is a risk factor that could prevent I2R from achieving the desired results out of the risk assessment process.

6. **Recommendation:** *Address known C3 performance testing environment issues*

   Priority: Medium

   Owner: Critical Systems Resiliency, Test Track

   Key Organizations and Roles: SIS, Infrastructure

   The goal of this action is to address gaps in performance testing environments to address key dissimilarities between production and performance environments and to ensure adequate environments are available to support releases developed in parallel.

7.3. Next Steps for the Performance Team

This section documents other activities planned to occur after the Tiger Team transition point.

Action	Owner	Contributors	Timing
Complete tiger team work	Rajiv Wani	Tiger Team	Through March 2007
Address performance management process issues	OpEx with Infra-IT	TBD	TBD

7.4. Lessons Learned

Lessons learned from the conduct of this effort are listed below:

- **Resource focus and commitment.** The efforts of the overall program suffered repeatedly due to unplanned change in focus of the committed resources due to P1 issues, particularly for CACO resources. The performance tiger team had to re-plan the resources for technical analysis to use IT track resources to supplement the CACO ownership. This change also impacted the internal milestone dates.
- **Clear SDLC/PLC ownership.** This program struggled due to a lack of clear ownership of impact analysis stemming from Oracle patches and custom fixes. CA QA and the IT tracks each thought impact analysis was the responsibility of the other for the proposed fixes. The performance track mitigated this issue by enlisting help from business for functional impact analysis and by taking ownership of technical impact analysis within the Tiger Team with help from the IT tracks.
- **Staffing the Performance track.** The resource ramp-up for the Performance track was very close to the actual delivery dates thus increasing the risk to the first few releases. This also resulted in 20+ Performance Tiger Team fixes entering the December release in the middle of the second test cycle.
- **Streamlined decision processes**. Each resource issue the team faced was painful, bureaucratic, and time-consuming to address. Each decision needed required a large number of people, and in most cases multiple discussions, to identify a resolution. A fair amount of people-hours during the program was spent trying to get to resolutions for resource issues. The turnover in IT organization leadership (e.g., lack of both a director and a VP for specific periods) probably contributed in part to complexity of the discussions and decisions.

8. Appendix A. Glossary

Term	Definition
Alliance	Remedy incident management system used to manage outages and other issues in production systems
Analyze	Process that collects statistics about the database in order to optimize the performance. During and prior to Tiger Team execution, multiple P1/P2 cases were generated due to problems with the Analyze process
Capacity Planning	Process by which current load, historical load, and expectations for growth (e.g., in memory usage, CPU utilization, storage, network consumption, etc.) are used to identify resources required to support future needs. Capacity planning may be performed across technology tiers and infrastructure components
Contention	Situation in which two or more components are attempting to use a single resource in a conflicting way, for example, updating the same data object
Critical Systems Resiliency (CSR)	<company>-wide IT initiative to improve systems stability and improve stability-enhancing practices
CTSPDM	C3 database reporting instance used for non-real-time access
CTSPRD	C3 Production environment database
CTSPRF	C3 Performance testing environment
Data Archive / Purge / Retention	Policies and processes that define data that can be purged, data that must be archived and available as needed, and data that must be retained in the real-time system, as well as the applicable timing windows for each piece of data to be purged, archived, or retained
Emergency Bug Fix (EBF)	Immediate changes to address outstanding P1 or P2 cases (i.e., system stability issues) that are interfering with the operational status of a system
Top SQLs	Report managed by the 11i-Performance team to identify the most resource-intensive SQL statements in <company> Oracle 11i environments
Latching	A simple, low-level serialization mechanism to protect shared data structures in the System Global Area. Latching problems occur when a query obtains a latch and then does not give it up promptly upon query completion; alternately a hung query can cause a latching problem
Long-term Fix (LTF)	Change needed to address a the root issue for an experienced problem so that it does recur
Real Application Cluster (RAC)	An Oracle database feature that allows a database to be installed across multiple servers for scalability or redundancy
Root Cause Analysis (RCA)	Process used to identify the core issue that resulted in an Alliance case
Shared Global Area (SGA)	The shared segment containing temporary data caches used by the Oracle server during database operation. The SGA is initialized when the server is started and unavailable when the server is not running. Also known as System Global Area
Synergy	IT project to combine the I2R (C3) and SSM (Ciber) Oracle 11i systems

9. Appendix B. References

Reference	Link
Performance improvement line items tracking spreadsheet	http://workspace/Livelink/livelink.exe?func=ll&objId=16955781&objAction=Open
Oracle application patch analysis worksheet	http://workspace/Livelink/livelink.exe?func=ll&objId=17331422&objAction=Open
Specific fix details tracking	http://workspace/Livelink/livelink.exe?func=ll&objId=17055783&objAction=Open
Livelink folder for the Performance track	http://workspace/Livelink/livelink.exe?func=ll&objId=16959006&objAction=browse&sort=name
Application repository for CACO-managed systems	http://ework.<company>.com/Livelink/livelink.exe?func=ll&objId=12118697&objAction=Open
C3 Health Reports	http://ework.<company>.com/Livelink/livelink.exe?func=ll&objId=10398197&objAction=browse&sort=name
Electronic version of this document	http://workspace/Livelink/livelink.exe?func=ll&objId=18995315&objAction=browse&sort=name

10. Revision History

Date	Author	Version	Change Description
21Dec06	Melissa Liu	1.0	Initial Version