

How to Manage an Oracle Optimization Project

Gary Goodman
NoCOUG – November 13, 2003

You are in the wrong room if you are looking for

- Technical tips and techniques (DBA junk food)
 - Check lists, Ratios, Parameters

You shouldn't be looking for this 'stuff' anyway!

- Technical detail on 10046 trace data
 - Collecting
 - Interpreting

Cary Millsap – Oracle Operational Timing Data (hotsos.com)

So why are we here? Because managing Oracle optimization is apparently very difficult.

- As a manager, you have to deal with
 - Technical experts argue over problem root causes
 - Experts claim excellent progress; users see no improvement
 - Hardware upgrades don't help or slow the system further
 - Performance remains unsatisfactory for months or years
 - Top resource consumer is WASTE

Persistent problems with traditional “Oracle tuning” methods

- Frequently no impact or negative impact
- Months to years without positively identifying root causes
- No objective way to measure end-user impact
- No way to predict outcome without trial and error
- Fixes only some types of problems, but not others
- Ambiguity requires immense intuition, experience, and luck
- Can't tell when we're “finished tuning”

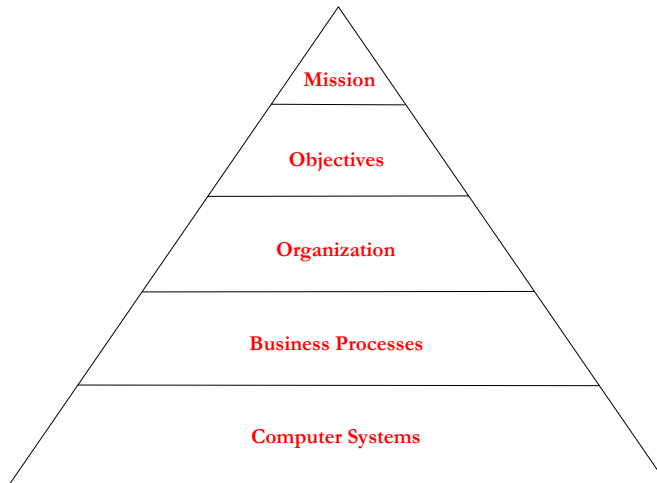
Optimization is really not all that hard

- You and your staff already know how
- Requires an understanding of the business you support
- Requires a technical shift in how the performance analyst collects data for issue diagnosis
 - User action based vs. System based
- 5 Guidelines for managing successful optimization projects

Managing Optimization – Rule #1

- Project goals must be aligned with the business process the system automates.

Business 101



Computer systems exist to automate and support the business processes of your company – period!

Define your goals with business, not system metrics

Good

- Increase orders processed per hour from X to Y
- Month end close processing completed in under 24 hours
- Never have more than \$X of WIP inventory waiting to be shipped

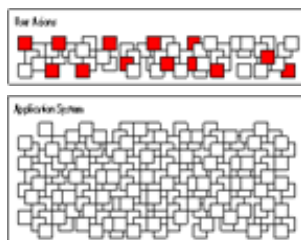
Wrong

- Reduce max CPU utilization to 95%
- Do anything with the BCHR (sorry – I'm shallow minded)
- Reduce extents to Y

Managing Optimization – Rule #2

- To maximize a system's economic value, you must prioritize on the product and customer centric functions.

Here's what we get to start with



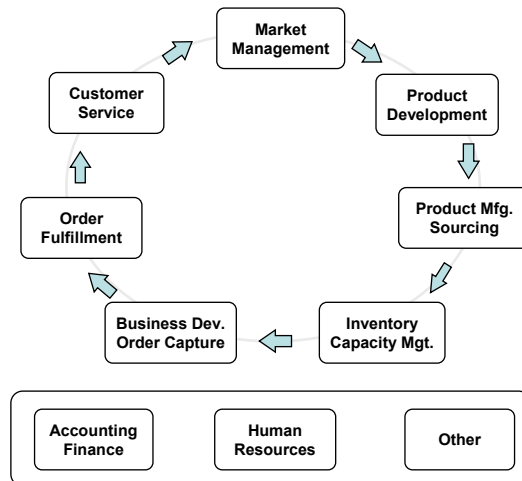
- Multiple users complaining of response time issues
- A complex application environment with hundreds of potential issues
- Your system is simply a collection of sessions
- Prioritize on the session with the biggest potential impact to the business bottom line

The Goal

To make money by increasing net profit, while simultaneously increasing return on investment, and simultaneously increasing cash flow.

—Eli Goldratt, *The Goal* (1984)

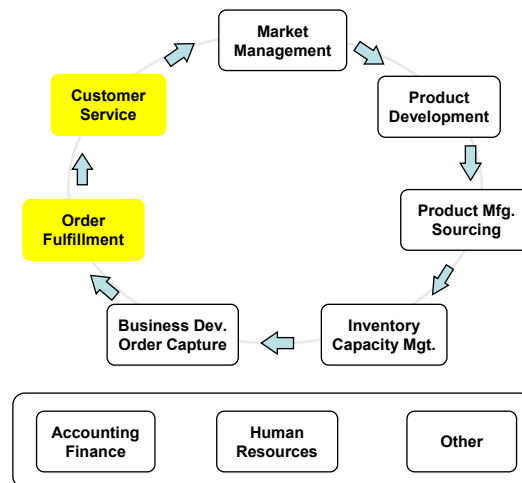
Business process segmentation



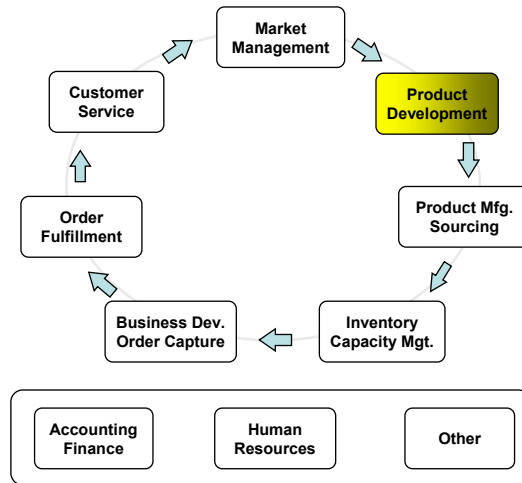
Segmentation

- Transformational:
 - How a company defines itself in the market
 - How a company defines its competitive advantage
- Core:
 - All other customer/product centric functions
- Foundational:
 - All other non customer/product centric functions

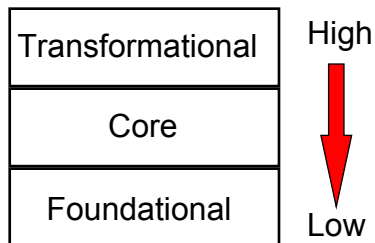
Lexus – Transformational focus



Oracle 1986 – Transformational Focus



Relevance to the “C-Level” officers



Prioritization of investment

- Must set minimum acceptable performance standards for all levels
- Focus on transformational
- Drop down to core only when
 - Transformational layer is optimized
 - Core is below minimum acceptable level
- Drop down to foundational only when
 - Transformational and core are optimized
 - Foundational is below minimum acceptable level

Business prioritization yields several important benefits.

- Benefits of business prioritization
 - The most important programs get fixed the soonest
 - Sacrifices always favor more important programs
 - Less important programs enjoy collateral benefits
- There's no downside
 - Even if the root cause is a less important program, this approach will find it

Managing Optimization – Rule #3

- The key to informed action is to collect the *wait interface* data for a specific user action.

Keys to ‘Informed Action’

- Performance problems are user based, not system based
- To fix a specific user’s problem you need to know what consumed that users response time
- Informed action requires knowing
 - What specifically is consuming the time
 - What will it cost to fix
 - What are the benefits of fixing

The only performance metric that your business really cares about is *session response time*.

- Event 10046 level-8 trace data has been available since 7.0.12!

Response Time Component	Duration	# Calls	Dur/Call
SQL*Net message from client	815.3s 90.3%	7,025	0.116057s
log file sync	43.0s 4.8%	3,084	0.013954s
unaccounted-for	27.4s 3.0%		
CPU service	14.6s 1.6%	3,652	0.004001s
latch free	1.9s 0.2%	123	0.015709s
buffer busy waits	0.1s 0.0%	65	0.000920s
SQL*Net more data to client	0.1s 0.0%	309	0.000212s
db file sequential read	0.0s 0.0%	1	0.001497s
SQL*Net message to client	0.0s 0.0%	7,025	0.000007s
Total	902.4s 100.0%		

Even when your “whole system is slow,” it is still critical to identify the most important programs to fix *first*.

“The performance characteristics of your slow program do not necessarily resemble the performance characteristics of your system-wide average workload”

- Case study: *Oracle Payroll – chasing latch contention*
- Reliable, deterministic analysis of system-wide performance statistics is virtually impossible
 - You can’t extrapolate detail from an average
 - You might get lucky, but usually you’ll rat-hole yourself

How can focusing on just a *few* programs fix your *whole* system?



- The per-program approach is valid when...
 - There is one root cause
 - Any single program will point you to “the” problem
 - There are many root causes
 - In fact, it’s the *only* method that works reliably in this case

Managing Optimization – Rule #4

- Track the ROI forecast and results to document gains and improve the predictability of your team’s efforts.

Predict the return for ROI

- Since we are working with business metrics you can assign \$ values to the incremental improvements
- With detail response time profiles, you can predict the impact of various actions

Response Time Component	Duration	Calls	Dur/Call
db file scattered read	399.92s 30.5%	164,776	0.002427s
buffer busy waits	397.41s 30.3%	216,267	0.001838s
CPU service	391.62s 29.9%	144,515	0.002710s
latch free	72.23s 5.5%	7,054	0.010240s
.			
Total	1,310.95s 100.0%		

Predict the investment of ROI

- 1.5 X faster CPUs will cost \$XX,000
- Improving the SQL will cost approximately 40 hours of labor
- Getting the repaired code from the vendor will cost patch application or a version upgrade
- Changing the schema will cost approximately 300 hours of development and testing

Track your results to improve predictability

- Baseline response time and run count
- Predicted investment and return
- Actual investment and return
- Delta predicted vs. actual
- New response time and run count
- Delta response time and run count

- Improved predictability over time

Managing Optimization – Rule #5

- Document project results with business metrics, not worthless system information.

Simply measure against the goals you defined

- Reduced WIP inventory waiting to be shipped by 37% resulting in a monthly benefit of \$150,000
- Investment for this project was \$200,000
- 100% payback in month 2!

- Benefits
 - Exact impact of project is known
 - Credibility with business leaders
 - Future project funding

Method recap

1. Project goals must be aligned with the business process the system automates.
2. To maximize a system's economic value, you must prioritize on the product and customer centric functions. Identify each session that you will diagnose
3. The key to informed action is to collect the *wait interface* data for a specific user action
4. Track the ROI forecast and results to document gains and improve the predictability of your team's efforts
5. Document project results with business metrics, not worthless system information

