

ORACLE®



Developing a Risk Based Testing Plan for Enterprise Applications Systems

Alex Collier
Senior Director, Services Engineering



The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



Synopsis

Developing a Risk Based Testing Plan for Enterprise Applications Systems

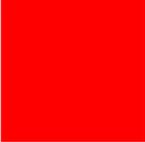
A simple fact of life is that it is neither economically nor technically feasible to completely test Enterprise Applications today. However, these same systems are most often "business critical".

Enterprise systems are being changed at a head-spinning rate – patches required for Sox or security compliance; new LOBs coming on line; upgrades; new components in the stack; new customizations and so on.

Given today's business mantra of "doing more with less" the challenge is how to determine what to test and more importantly what not to test.

The answer lies in using pragmatic "risk management" to tailor your test plan to the business priorities.

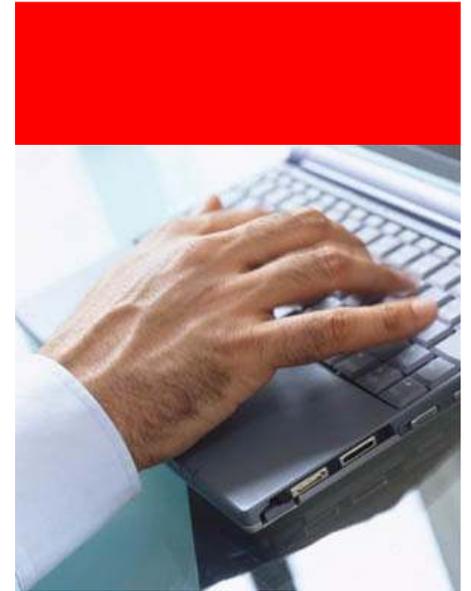
This session will present a scalable methodology for creating a broad test plan, assessing the risk and focusing the test plan to provide the maximum value and minimize business risk.



Disclaimer

THE FOLLOWING IS INTENDED TO OUTLINE OUR GENERAL PRODUCT DIRECTION. IT IS INTENDED FOR INFORMATION PURPOSES ONLY, AND MAY NOT BE INCORPORATED INTO ANY CONTRACT. IT IS NOT A COMMITMENT TO DELIVER ANY MATERIAL, CODE, OR FUNCTIONALITY, AND SHOULD NOT BE RELIED UPON IN MAKING PURCHASING DECISION. THE DEVELOPMENT, RELEASE, AND TIMING OF ANY FEATURES OR FUNCTIONALITY DESCRIBED FOR ORACLE'S PRODUCTS REMAINS AT THE SOLE DISCRETION OF ORACLE.

Agenda



- Establishing your business case
- Types of changes on the application (increasing risk...)
- Risk – the critical dimension
- What is testing?
- How to scope the testing problem
- Types of testing and test classification (decreasing risk...)
- How to test effectively at scale

About the Speaker

Alex Collier, Senior Director, Oracle Corporation



Alex Collier is the QA Strategist for Services Engineering at Oracle. Alex has been involved in software development and engineering for over 20 years. He joined Oracle in 2000 as a part of an acquisition of Carleton Corporation, where he was vice president of development.

Prior to Carleton, Alex started a company called Position Sensitive Robots, which was an early market leader in PC-based industrial automation. He spent his earlier career in the field of semiconductor automation.

Alex has an honors degree in electrical and electronic engineering from Hertford University, England, and an executive M.B.A. from Suffolk University, Boston MA.

Advanced Customer Services

Solve Your IT Challenges

System Availability



Support Account Management



Technology Adoption



System Performance



Change Management



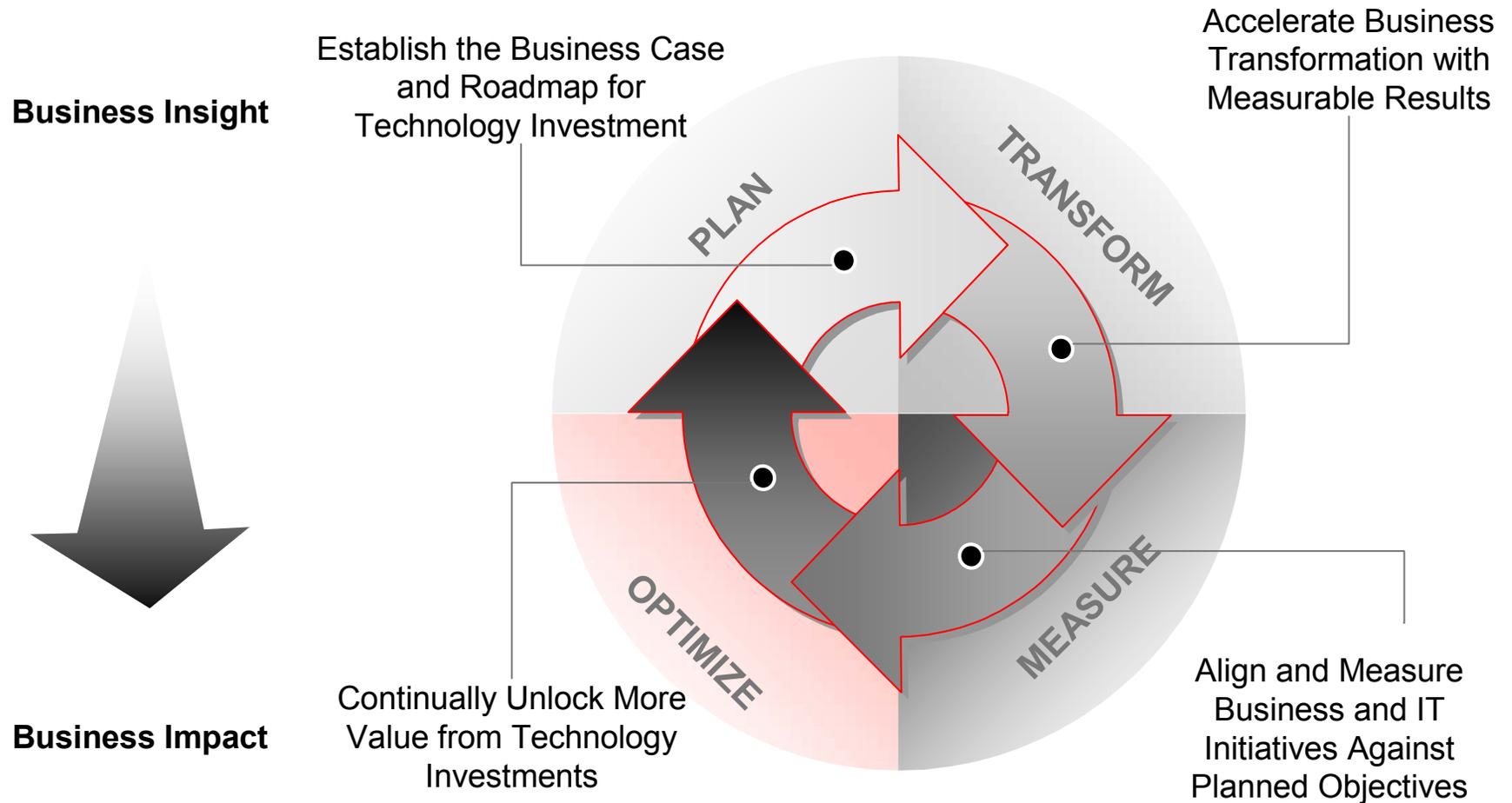
Faster Problem Resolution



ORACLE

Oracle Services Framework

Advanced Customer Services at Every Phase





Establish the Business Case...

- High upgrade costs hinders IT to set continuous upgrade policy
 - Over \$2,000 per user for Major Upgrade
 - Over \$1,000 per user for Minor Upgrade
- Upgrades and patches require enormous resources
 - Planning, Installation, Testing, Data Migration required even for minor releases
 - Most companies do several test cycles, each requiring support from users on the extended project team
- The Risk to the Business (and to the CIO's job) are significant
 - The complexities are overwhelming so problems will and do occur at go-live that can significantly impact the business
 - It is not economically feasible (or technically practical) to “test everything”



Sultan Aziz

PeopleSoft Systems Manager , PMI

Summary of the PMI Environment

- ⇒ FSCM 8.8 Tools 8.4.5
- ⇒ HCM 8.9 Tools 8.4.6
 - ⇒ 1,000 names users
- ⇒ Global user distribution
- ⇒ 3 month migration plan
- ⇒ First PeopleSoft customer in On Demand

PMI Business Value

What were the business drivers that justified to PMI that this testing should be done?

- New server Configuration
- Migrating to Linux Servers
- Converting to use Unicode





PMI Business Results

- *Testing engagement required little time from PMI*
 - ...just a few hours discussion with the testing team
 - ...and a few hours debugging
- *Performance and Capacity predicted for the On Demand instance by the testing engagement has been confirmed by actual production experience*



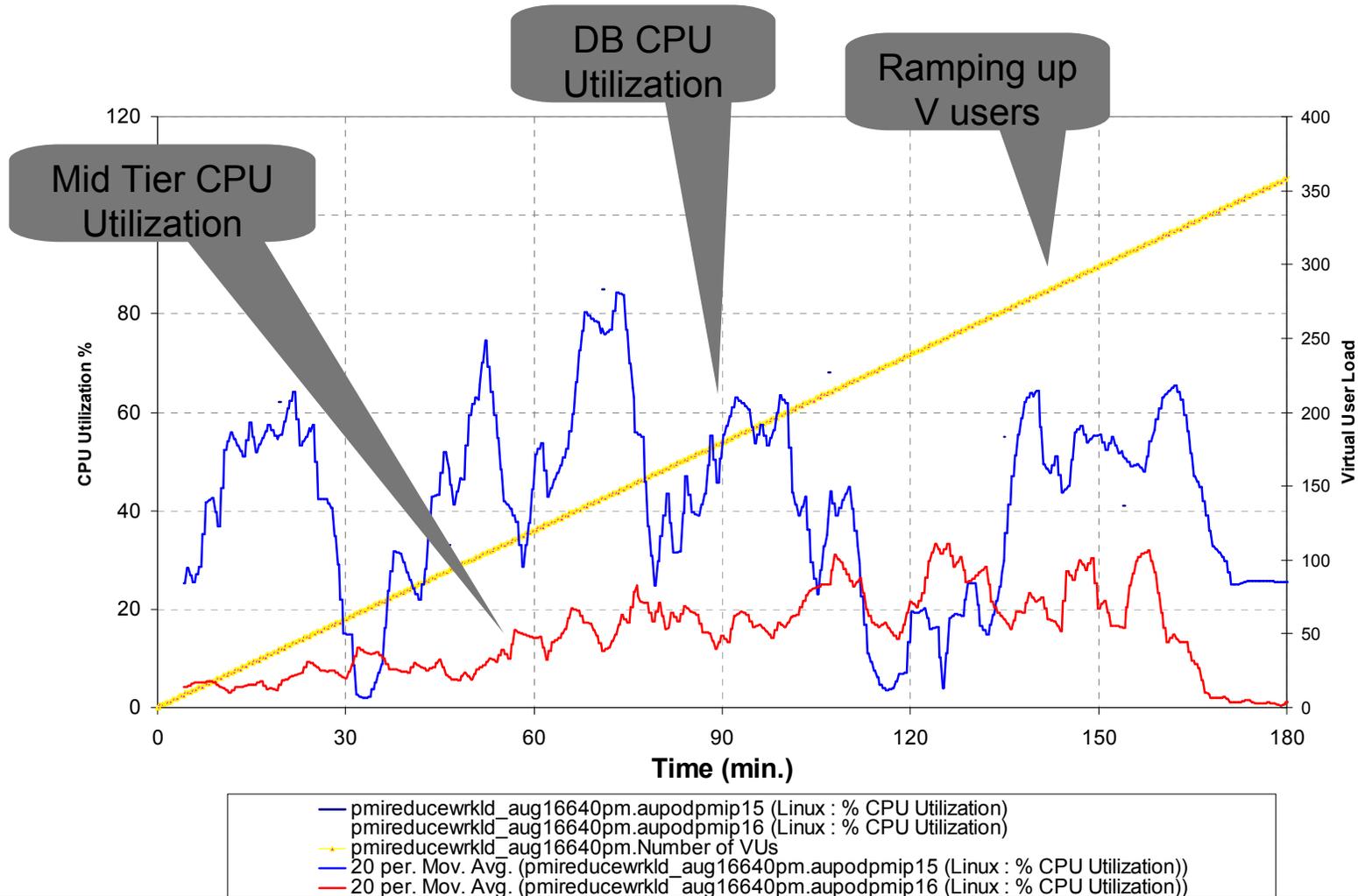
PMI Performance Test Results

The Executive Summary of the “Capacity Baseline” report stated:

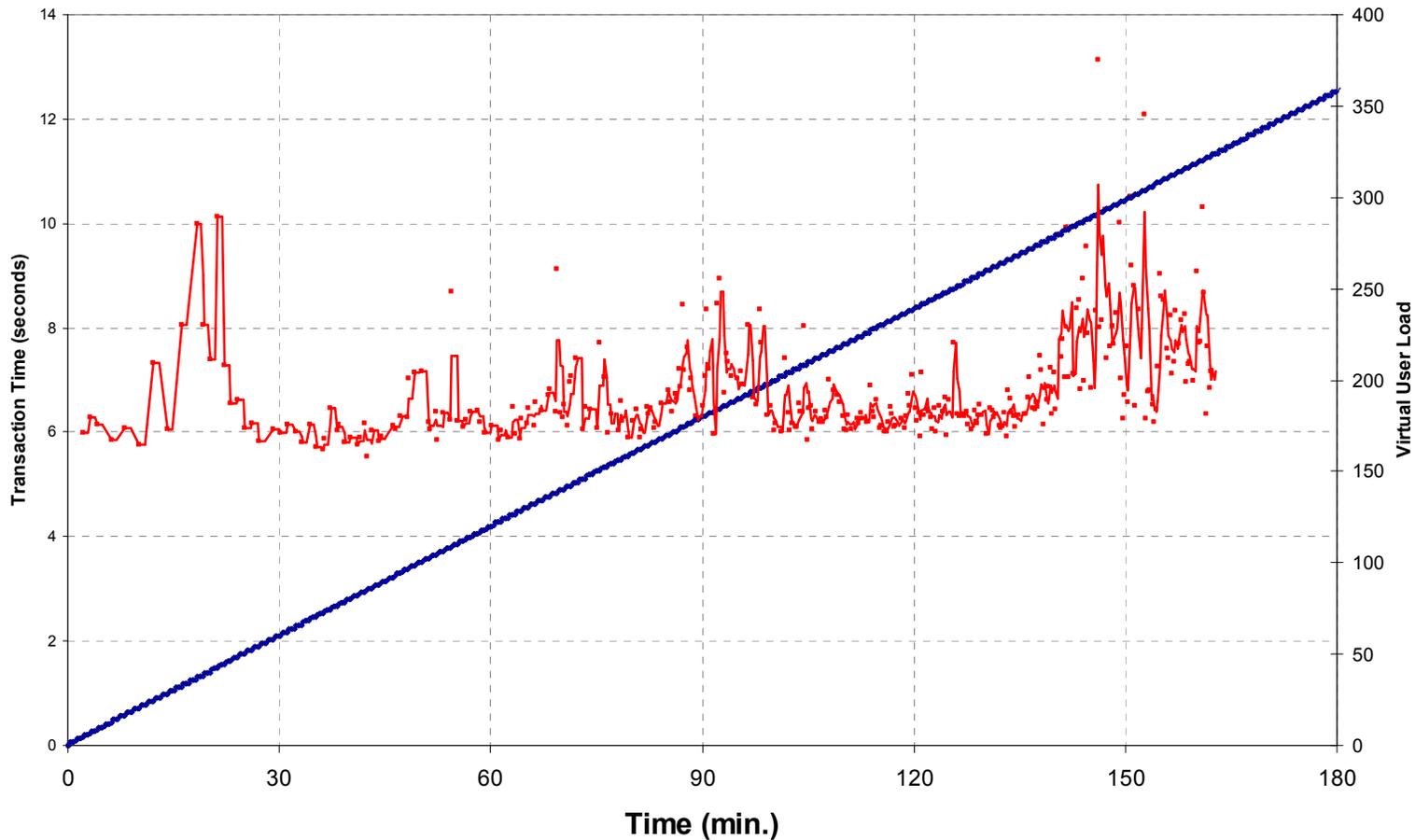
- 5 cycles of load testing were undertaken for Financial Pillar and 3 cycles for HR Pillar.
- The Financial Pillar handled 322 concurrent users at 4 times the transaction rate than required with the mid-tiers at 30% CPU and the DB peaking at 70% CPU Utilization average. 89 concurrent users required.
- The HR Pillar handled 140 concurrent users at 8 times the transaction rate than required with the mid-tiers at 9% average CPU Utilization (40% max) and database at and 6% average CPU Utilization (24% max).

The testing showed that the performance should exceed requirements...but what actually happened after “go-live”?...

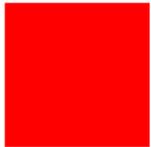
Financial Pillar – Test Cycle 5 CPU Utilization



Financial Pillar - Transaction Response Time Under increasing User Load

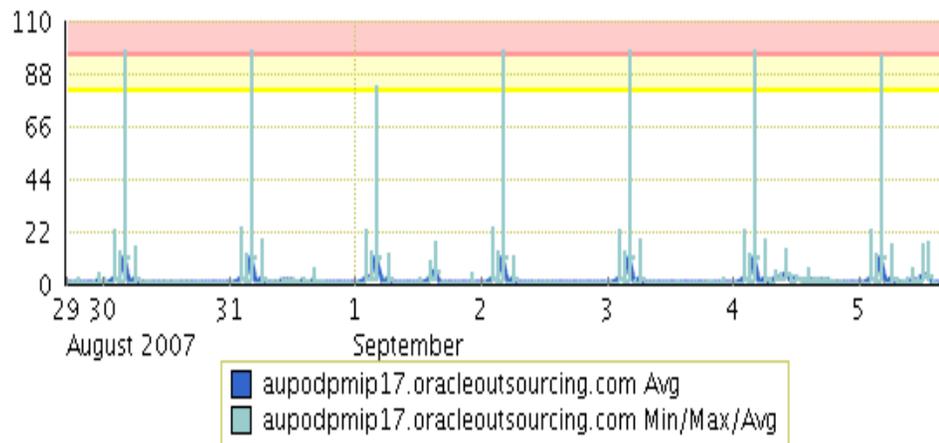
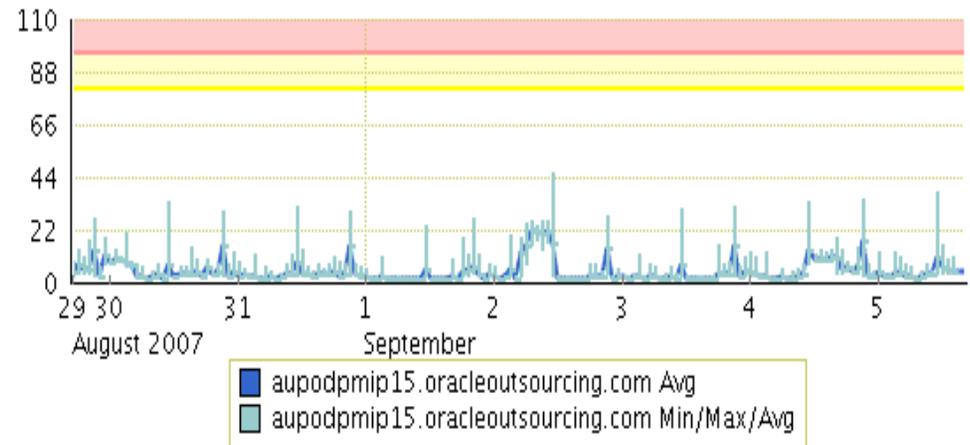
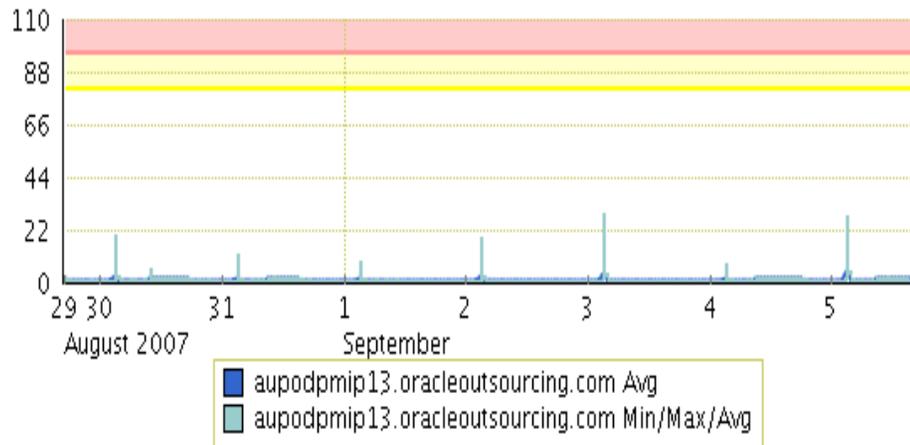


pmireducewrkId_aug16640pm.Online Journal Entry.Avg Server Time (sec)
pmireducewrkId_aug16640pm.Number of VUs



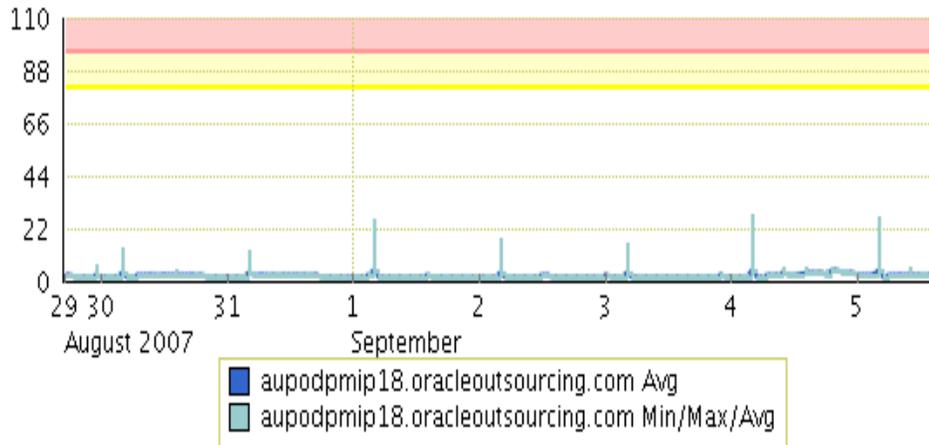
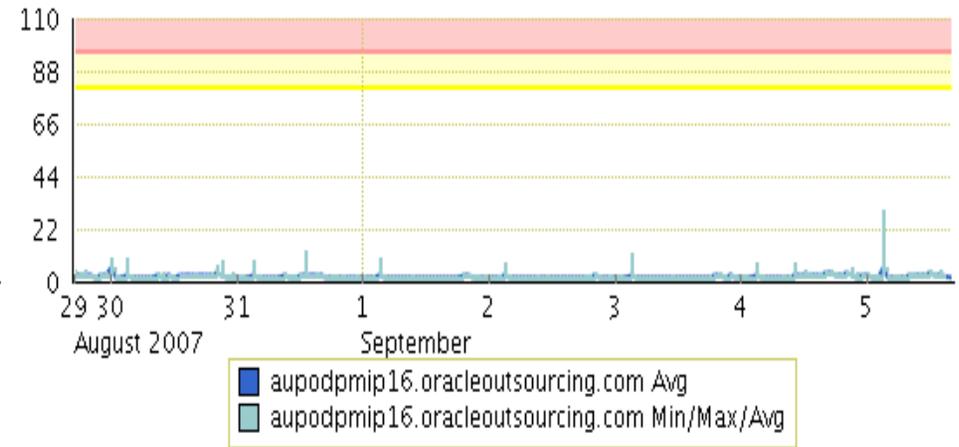
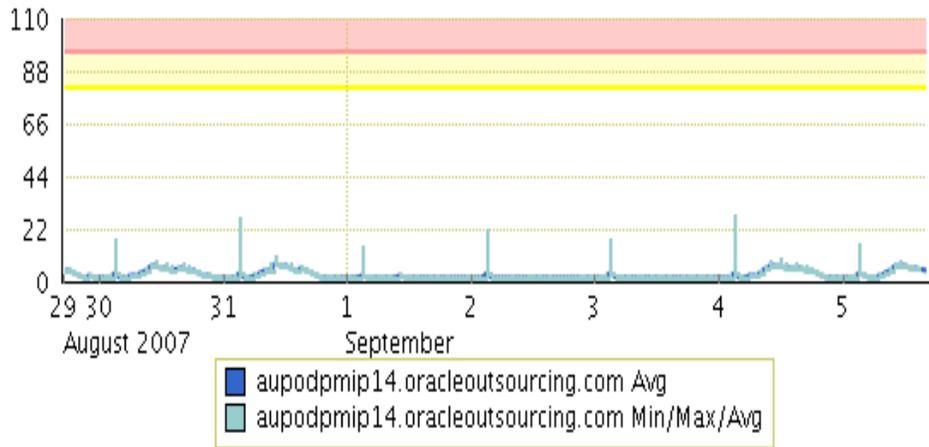
PMI Operational Performance Metrics

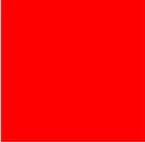
CPU Utilization on DB nodes



PMI Operational Performance Metrics

CPU Utilization on App Server Nodes





What does it take to implement enterprise software systems?

- Business requirements
- IT systems and infrastructure
- User communities
- Applications packages
- Middle tier, database and OS choices
- Customer configurations
- Customer data
- Customer customizations
- Security systems
- Tools

And all of this while...

“...doing more with less...”



Changes that have impact on an Instance (increasing risk...)

- Go live
- Instance consolidation
- New functionality / modules
- New customizations
- New user groups possibly in new regions with new languages
- Upgrades & patches
- Security patches
- Sox patches
- Hardware and/or infrastructure changes/upgrades



What is Testing?

“Testing” is defined as:

“The process of validating that an item conforms to its functional requirements over the complete life-cycle of that item”

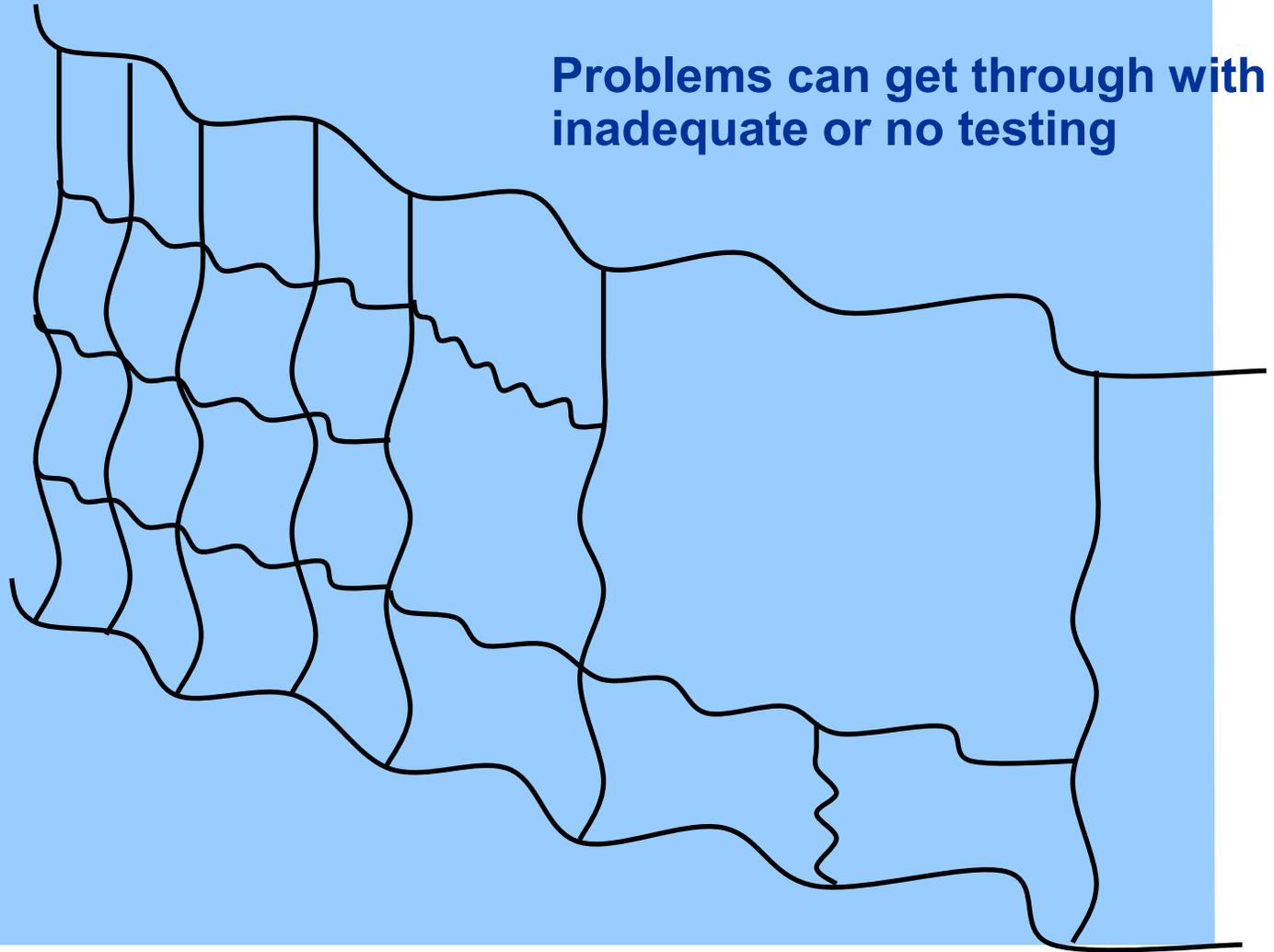
The “item” tested may be as diverse as a customer instance or an internally developed product or tool

The “functional requirements” may be represented by the number of explicit (specified) and implicit (expected but not necessarily specified) states that the item can achieve

The "QA Net"

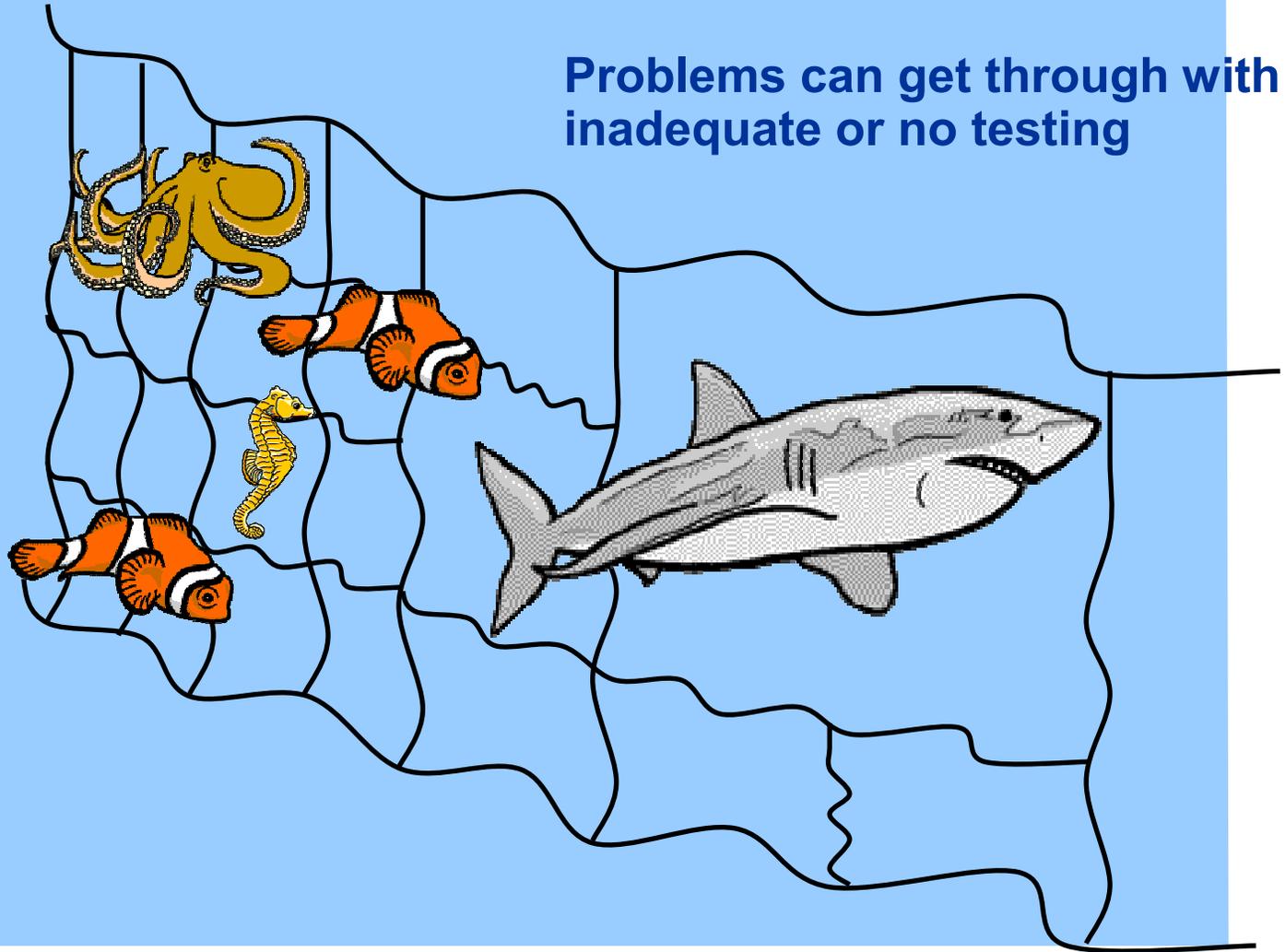
Good testing catches many issues

Problems can get through with inadequate or no testing



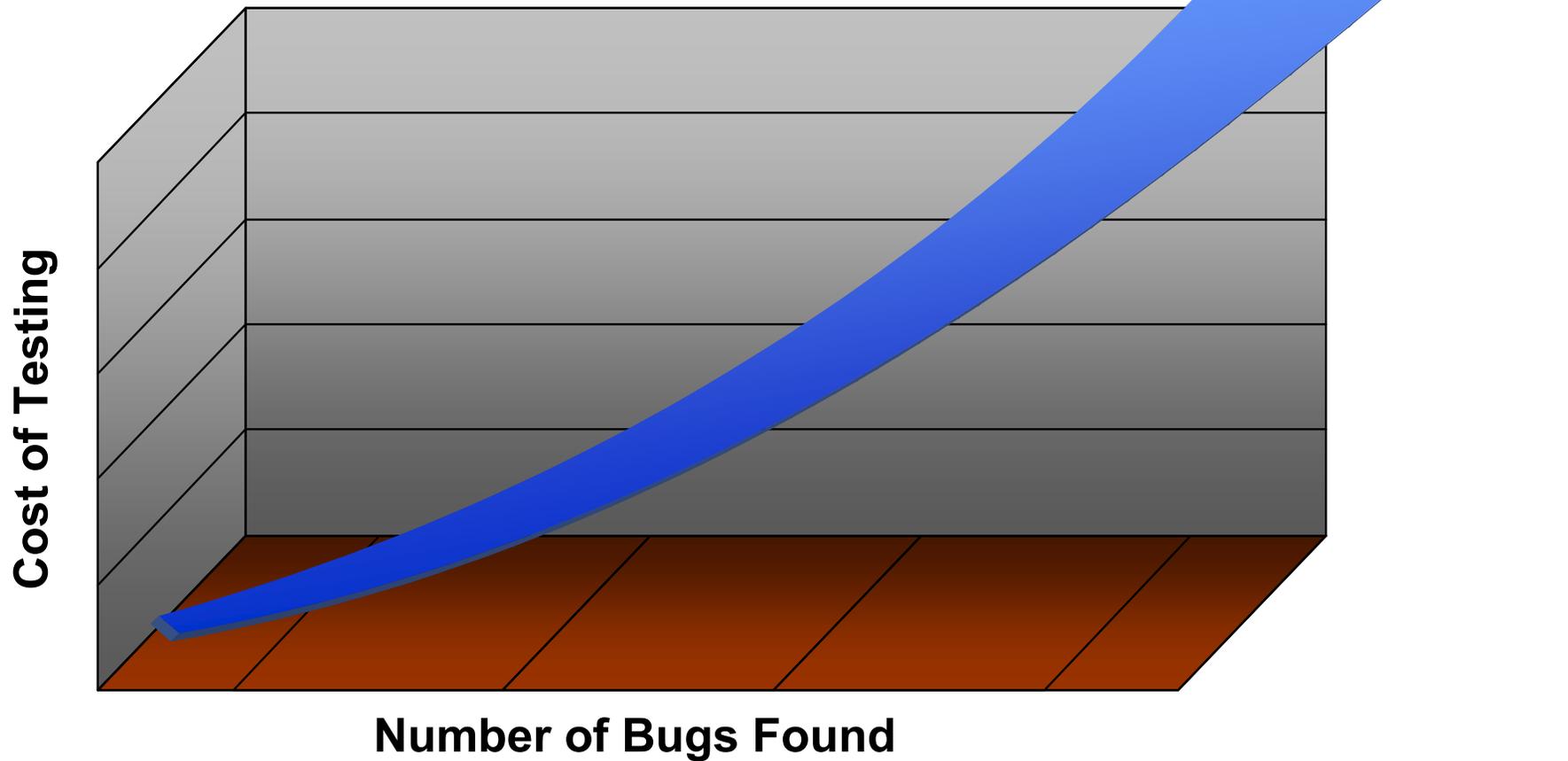
The “QA Net”

Good testing catches many issues



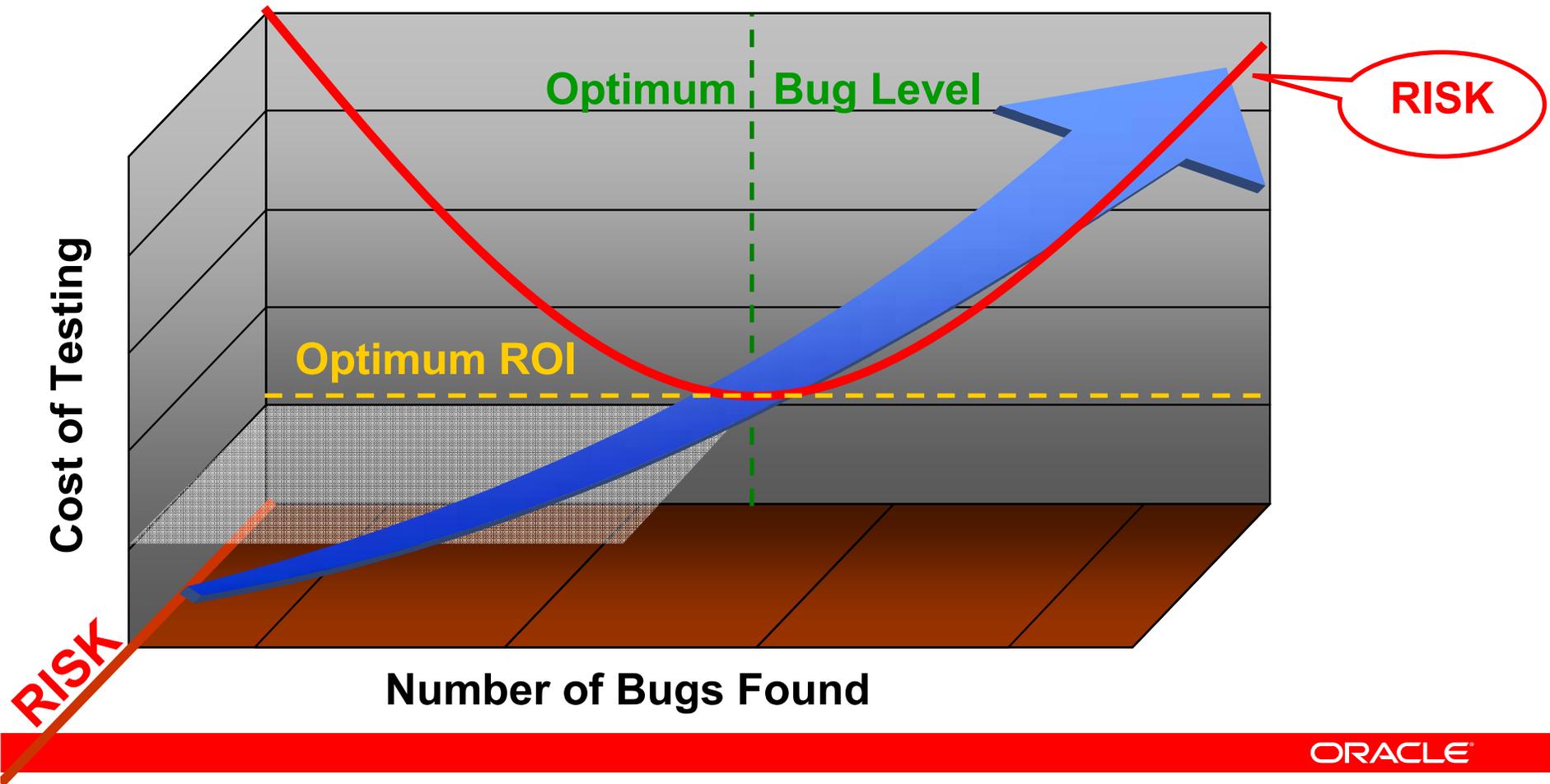
The QA paradox...

The more testing I do the more bugs I will find
but the higher the cost...so how much is enough?



The Paradox resolved...

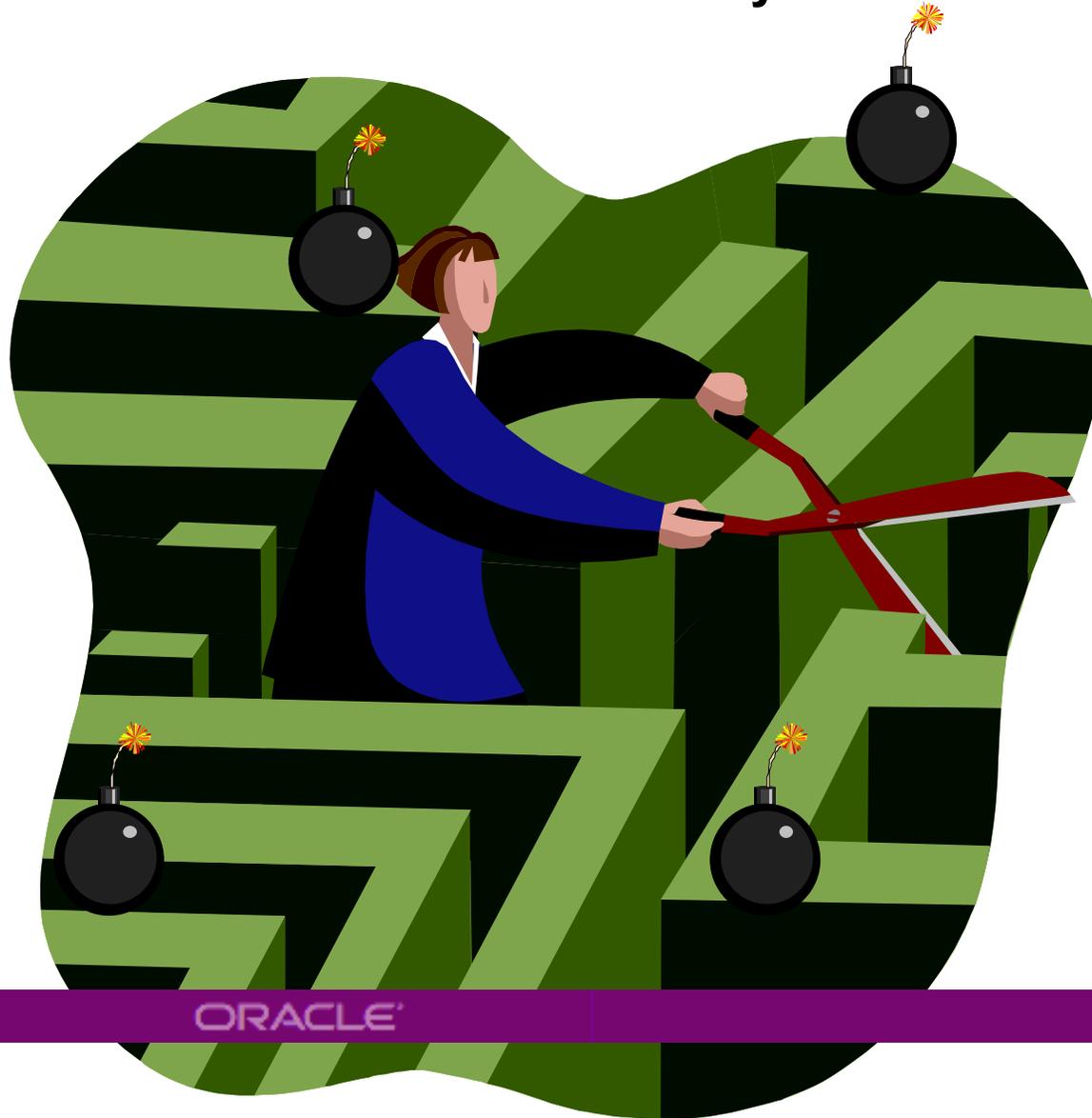
I should test for minimum overall risk to the business





The QA Problem Space

Do you know where the mines are in your mine field?

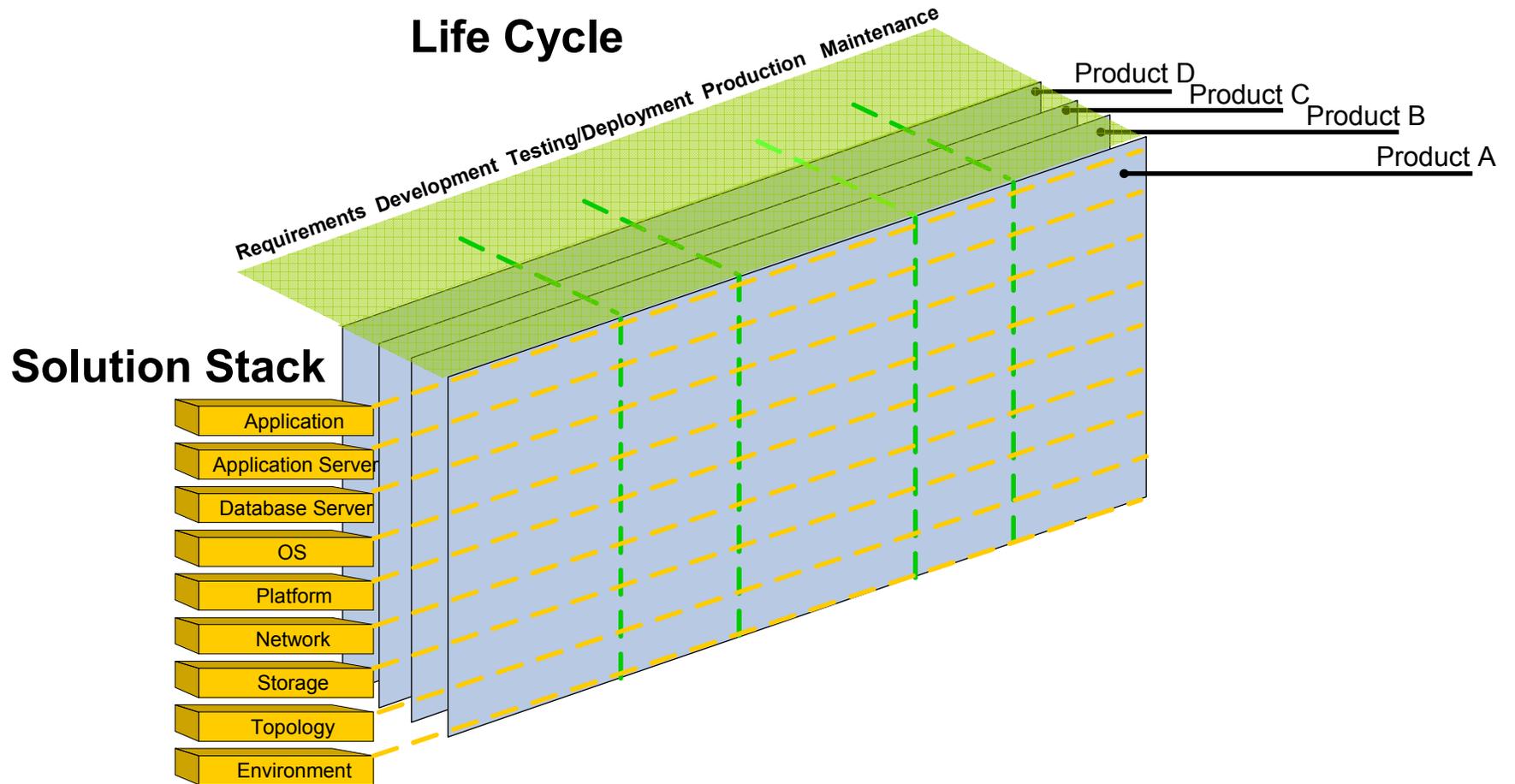


ORACLE®

ORACLE®

The QA Problem Space

A complex, multi-dimensional environment



Example of a QA Audit

A Risk Assessment Tool for the QA “Universe”

Environment	Stability	Functionality			Performance	Security	Usability
		Install/Deinstall	Function	Integration			
Component	Low Risk Area	Medium Risk Area	Medium Risk Area	High Risk Area	High Risk Area	Low Risk Area	High Risk Area
Product/Bundle	Low Risk Area	Low Risk Area	Medium Risk Area	Low Risk Area	High Risk Area	Low Risk Area	Medium Risk Area
Reference Configuration	Low Risk Area	Low Risk Area	Medium Risk Area	Low Risk Area	Low Risk Area	Medium Risk Area	Medium Risk Area
Customer Instance	High Risk Area	Low Risk Area	Medium Risk Area	High Risk Area	Medium Risk Area	High Risk Area	High Risk Area
Environment	High Risk Area	Medium Risk Area	Low Risk Area	n/a	High Risk Area	High Risk Area	High Risk Area

High Risk Area

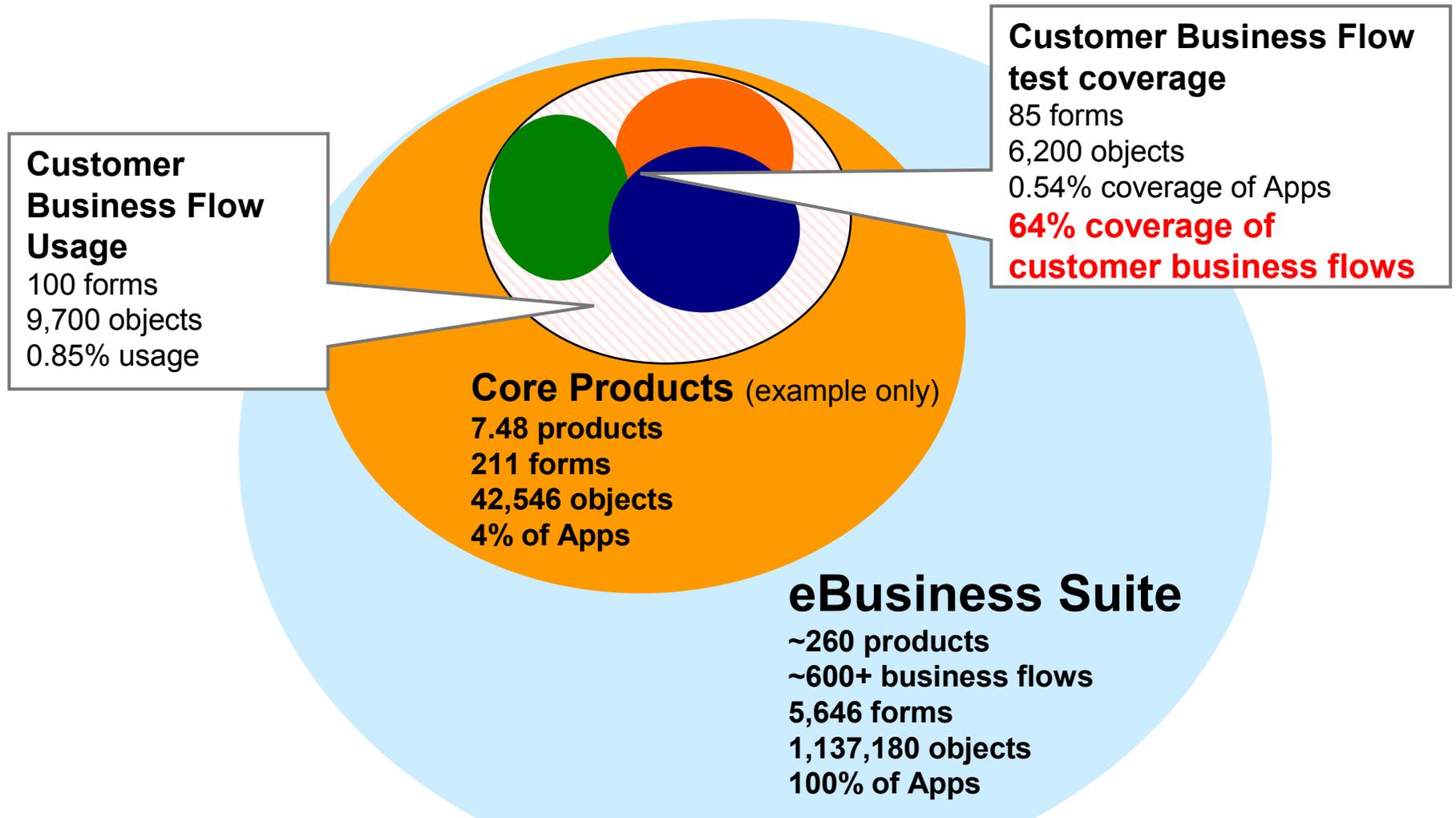
Medium Risk Area

Low Risk Area

A “QA Audit” helps assess the risk areas today and determine a plan for tomorrow

Why Customer Business Flows?

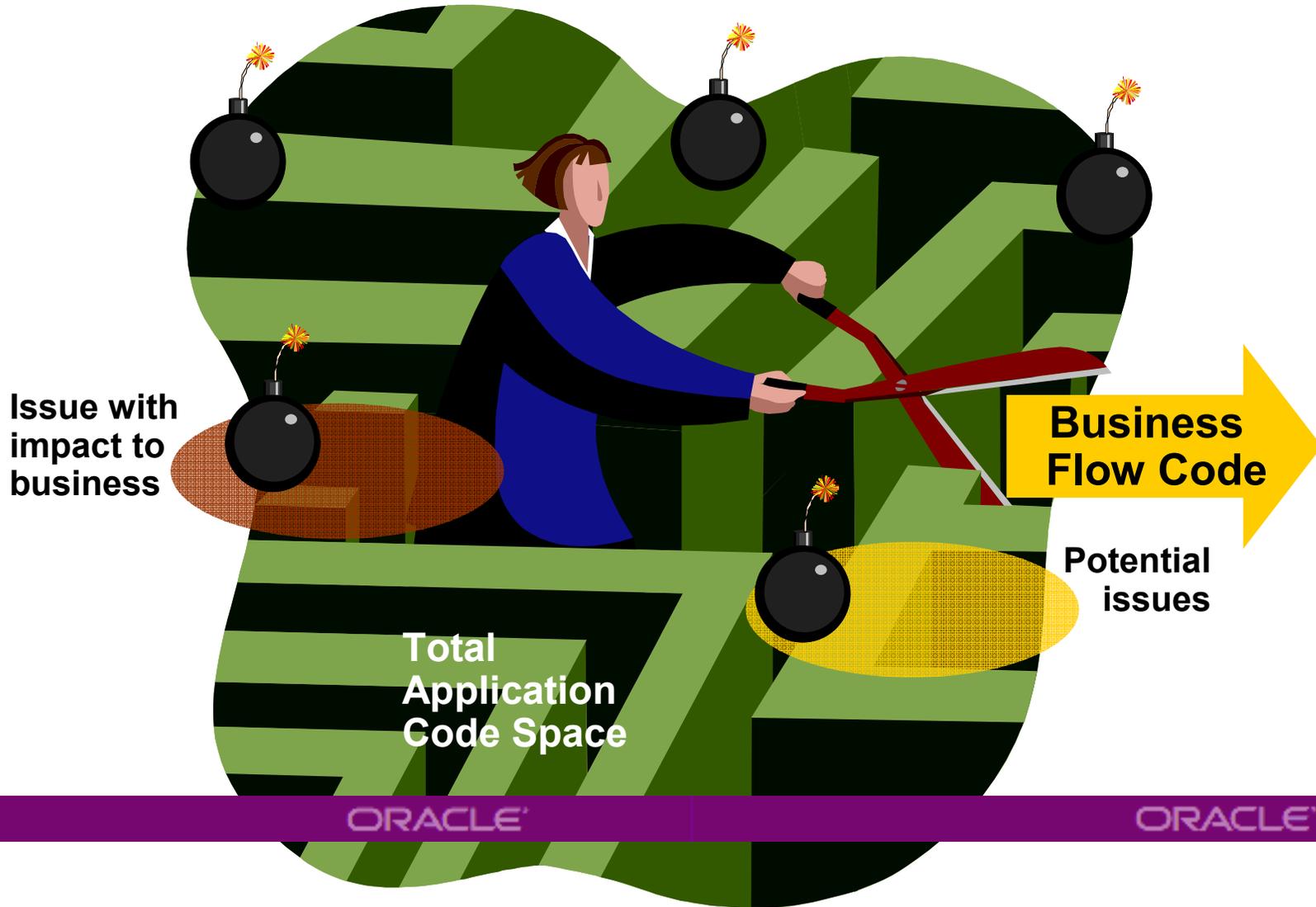
...it's all about "Scope"...

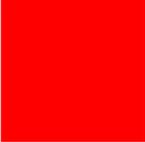




...narrowing the focus

Test what is important to the business





Classification by “Importance to the Business”

- Business critical – has immediate impact to customers, daily operations or the bottom line
- Important and urgent – will impact the business but not immediately. For example, closing the books at quarter end.
- Important but not urgent – impacts the operational effectiveness but has viable workaround until resolved
- Nice to have – non essential items that help the business run smoother



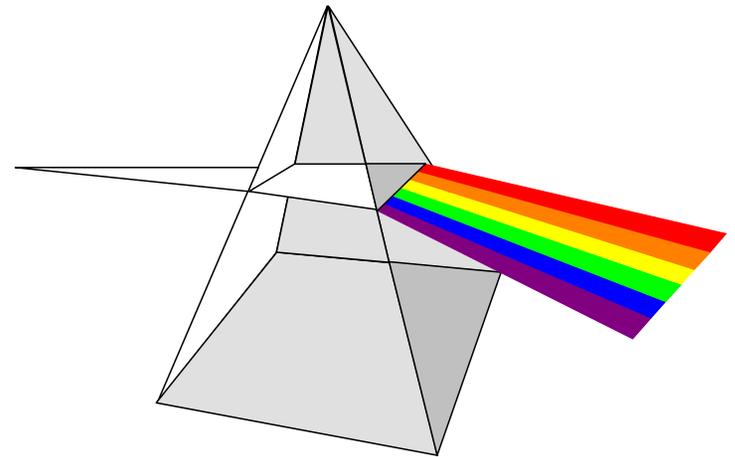
Testing Dimensions

- Consumers – customers, vendors, platforms, dev groups, critical accounts...
- Product - EBS, PSFT, Siebel, JD Edwards...
- Technology – RAC, distributed loads...
- Verticals – Financials, HR, CRM, manufacturing...
- Testing domains – functional, load, capacity, performance, security

Primary QA Metrics...

How do you define your success?

- # Issues detected before release to production
- Total cost of testing (manual & automatic)
- Testing yield (# issues caught / total time taken)
- # Issues detected in production (not all issues are bugs)
- Cost to the business of issues that get to production (not all issues are equal)

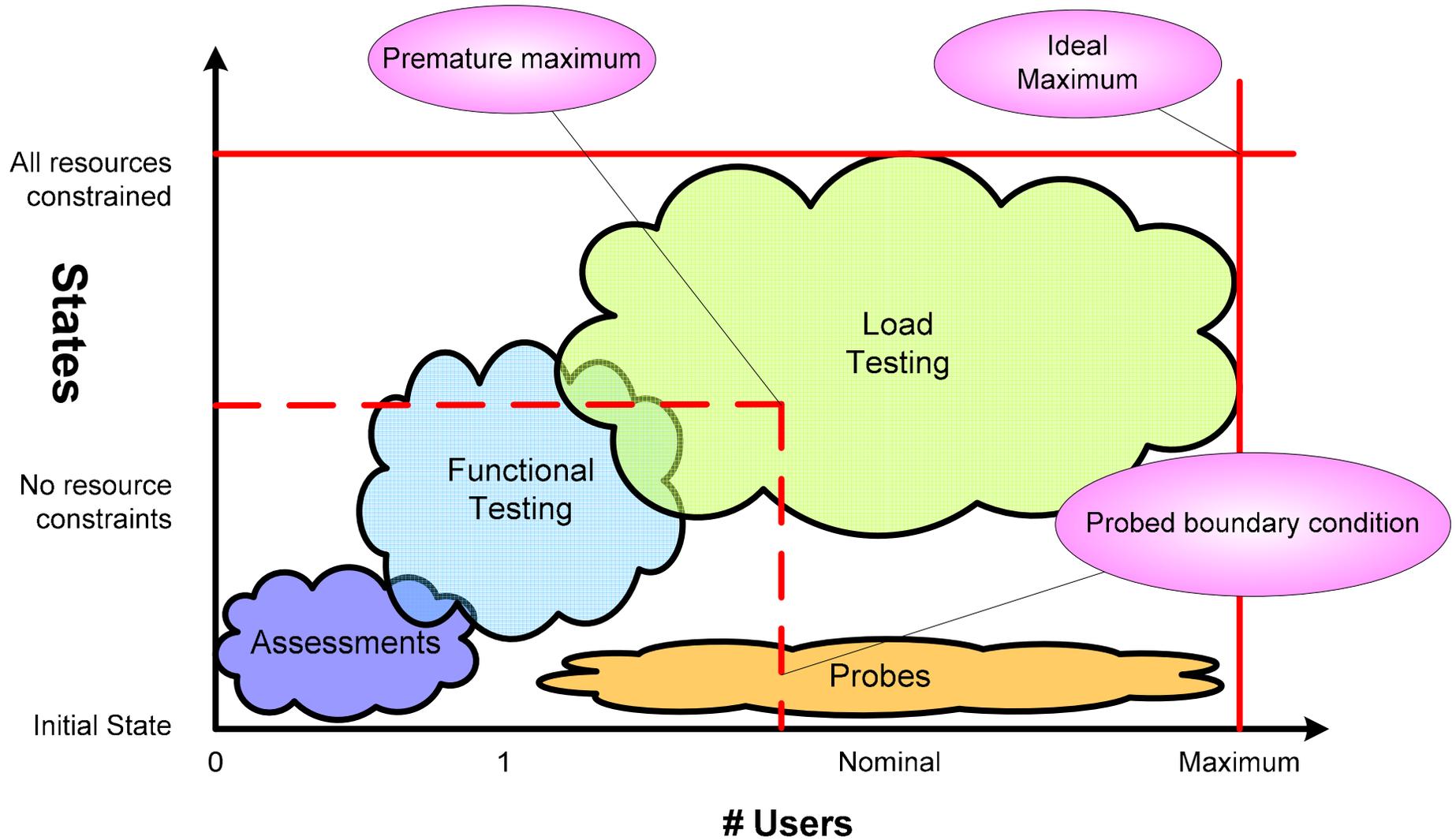




Types of testing to validate the changes

- Oracle Generic
 - Functional, performance
- Customer Generic
 - Upgrades, Early Implementer Program, support readiness, release notes, platform specific issues, sizing, feature testing e.g. RAC, Shared Appletop, certified configurations, parameter settings
- Customer Specific
 - Assessments, Health Checks, Oracle Diagnostics, RAC Environment Analysis, Functional Testing, Capacity Baseline, Instance Load Profile

“Testing” Validates State Coverage





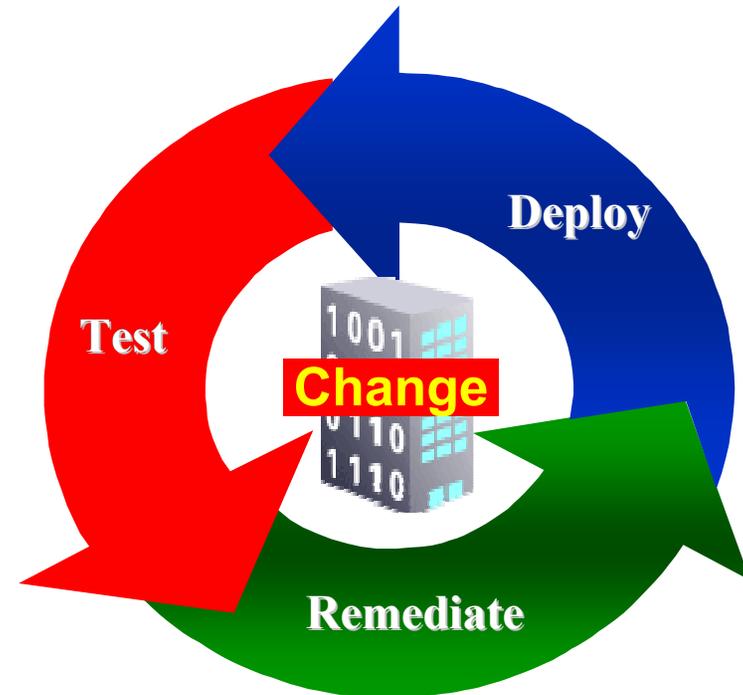
Classification of Testing

Typical QA Yield Distribution Jan 7 - Mar 9

- Assessments
 - Validate the initial state
- Functional testing
 - Validate those states without resource constraints (no boundary conditions)
- Load testing
 - Validates additional states with resource constraints (certain boundary conditions)
- Probe testing
 - Validates specific states under defined boundary conditions

Real Application Testing

- Value
 - Rapid technology adoption
 - Higher testing quality
- Business Benefit
 - Lower cost
 - Lower risk
- Features
 - Database Replay
 - SQL Performance Analyzer (SPA)



Solution for the Agile Business

Database Replay

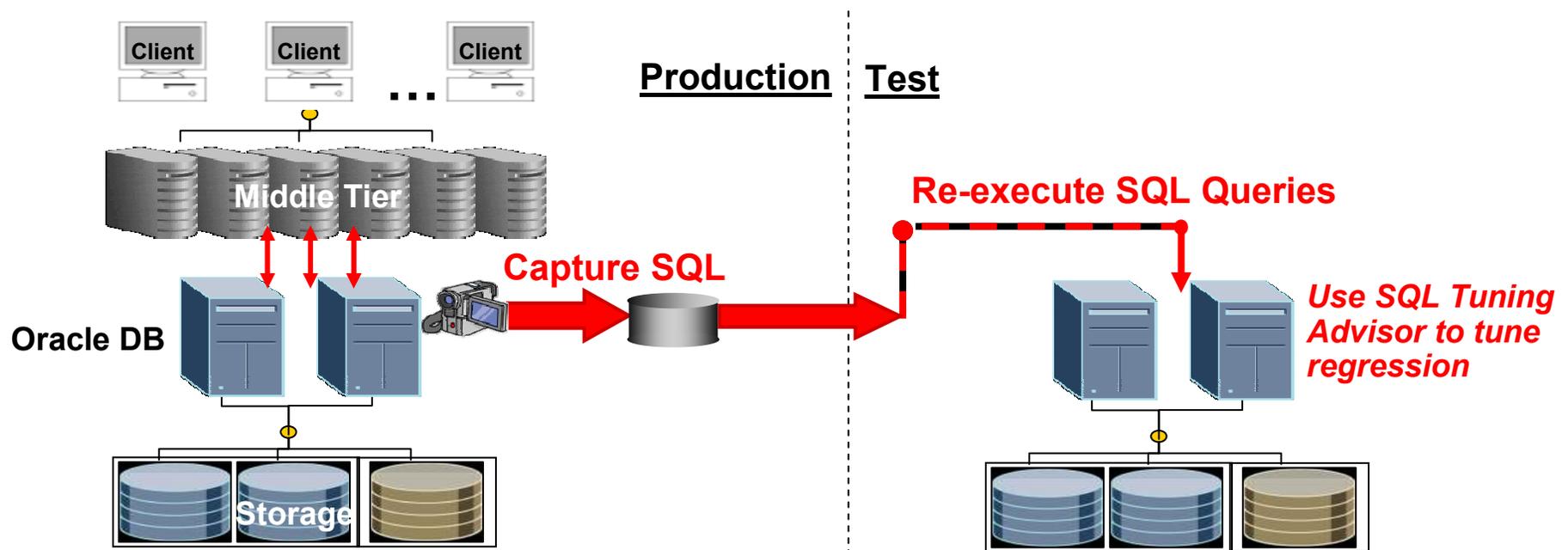
- Replay actual production database workload in test environment
- Identify, analyze and fix potential instabilities before making changes to production
- **Capture Workload in Production**
 - Capture full production workload with real load, timing & concurrency characteristics
 - Move the captured workload to test system
- **Replay Workload in Test**
 - Make the desired changes in test system
 - Replay workload with full production characteristics
 - Honor commit ordering
- **Analyze & Report**
 - Errors
 - Data divergence
 - Performance divergence



Analysis & Reporting

SQL Performance Analyzer (SPA)

- Test impact of change on SQL query performance
- Capture SQL workload in production including statistics & binds
- Re-execute SQL queries in test environment
- Analyze performance changes
 - Identifies all SQL regressions and improvements



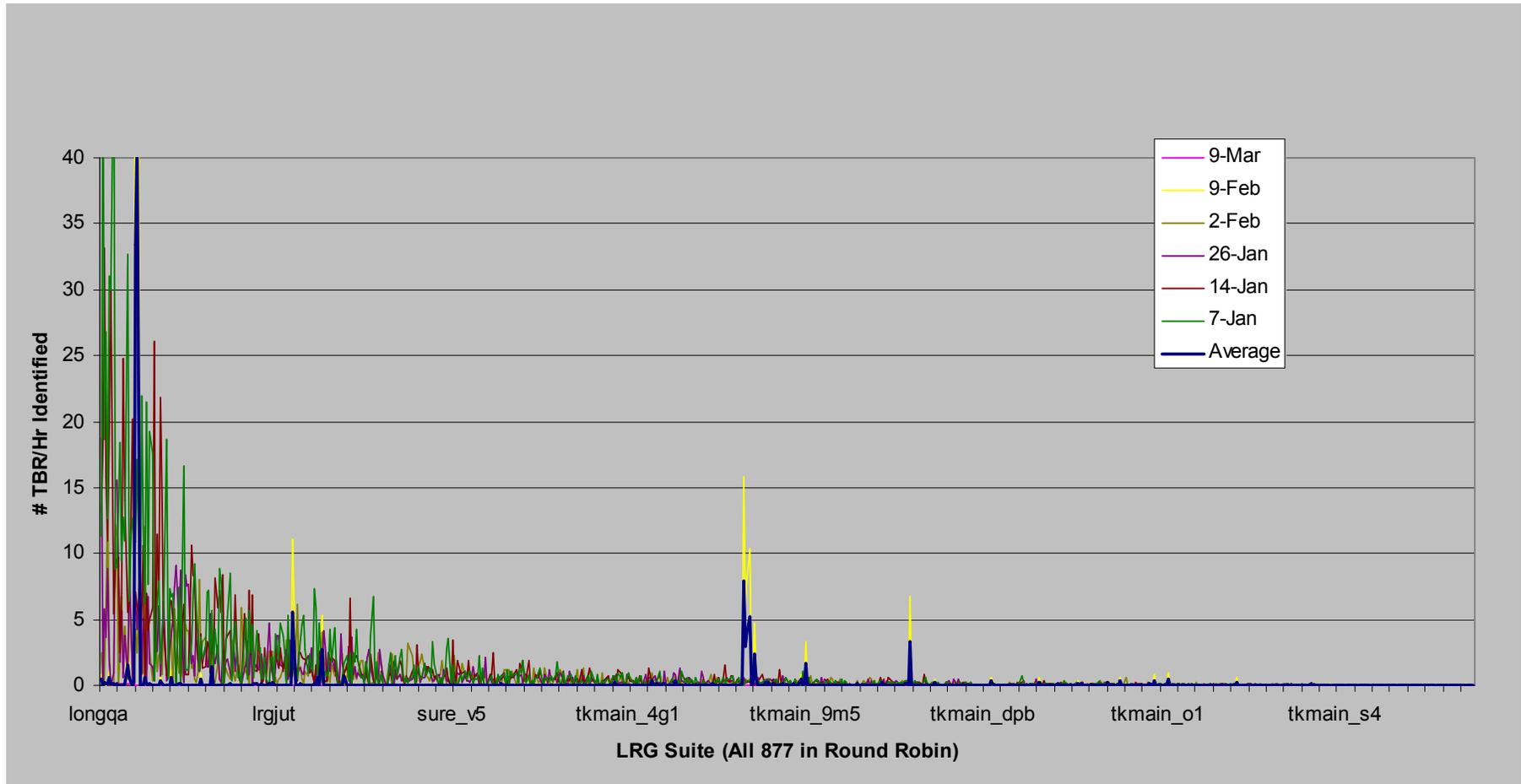
Database Replay vs. SPA

	<i>SQL Performance Analyzer</i>	<i>Database Replay</i>
What is it?	<ul style="list-style-type: none">• Predict SQL performance deviations before end-users can be impacted	<ul style="list-style-type: none">• Replay real database workload on test system
What Purpose?	<ul style="list-style-type: none">• Assess impact of change on SQL response time	<ul style="list-style-type: none">• Assess impact of change on workload throughput
How it works?	<ul style="list-style-type: none">• Single, isolated execution of SQL with production context	<ul style="list-style-type: none">• Replay with production context including concurrency, synchronization & dependencies
When to use?	<ul style="list-style-type: none">• Unit testing of SQL with the goal to identify the set of SQL statements with improved/regressed performance	<ul style="list-style-type: none">• Comprehensive testing of all sub-systems of the database server using real production workload

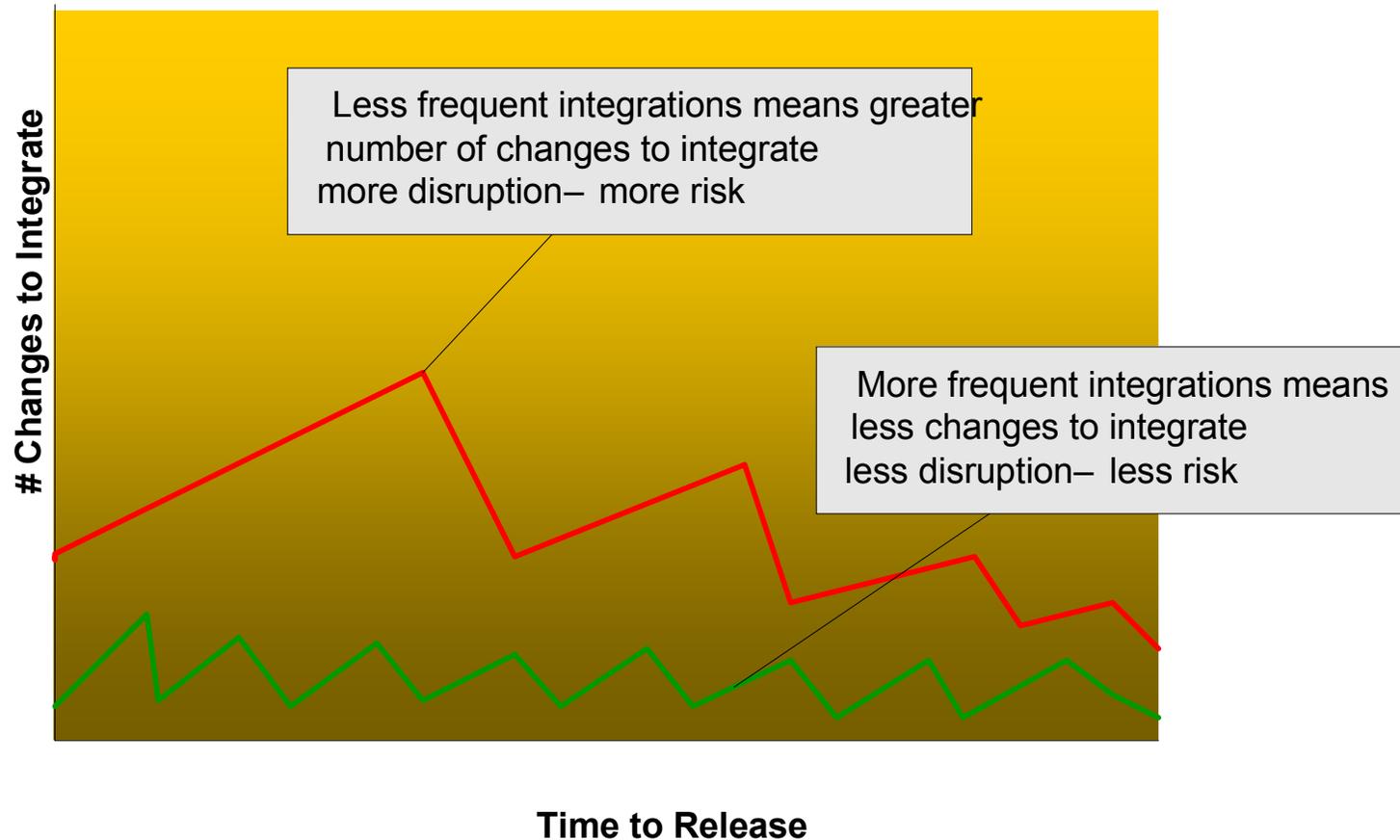
SPA and Database Replay are complementary

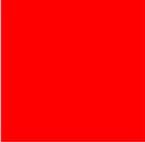
Testing at Scale – QA Yield

Typical QA Yield Distribution Jan 7 - Mar 9



Frequent Integration Cycles Reduce Risk





The “So What” slide...

- A minimum amount of carefully selected testing can provide huge ROI and “reduce risk” significantly
- Testing is like exercise...
... a small amount regularly has much greater benefit than a large amount occasionally
- Aligning your testing to actual production issues experienced significantly increases the focus, reduces risk and consequently the ROI of your testing
- “Plain vanilla” instances benefit from the experiences of the whole product user community – if you customize – then you embark on a journey “...to boldly go where no App has gone before...”



The Vision for Testing...

- Increase the scope of coverage while reducing the domain specific expertise required
 - Tools that understand the context of an application
For example – the key configuration parameters, primary data structure and flows
 - Direct API communication between tools and applications – thus removing the need for intermediate script or object programming
- Close the loop between post go-live issues and pre-release testing
 - Integration between production conditions through monitoring and integrated incident tracking and testing tools to provide a solid objective basis for determining focus areas for testing
- Reduce “ the cost of rediscovery” by leveraging user community knowledge (Wikipedia model?) to improve service levels in all aspects, including testing



Summary

Test the business critical areas first:

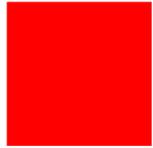
- Will the system provide the essential functionality?
- Will the system hold up and perform under a basic load?

Oracle does extensive testing in the areas of:

- Oracle generic
- Customer generic

To mitigate risk during the many changes that impact an instance and ensure maximum ROI the customer must test:

- Customer specific – which includes:
 - Configuration
 - Customizations

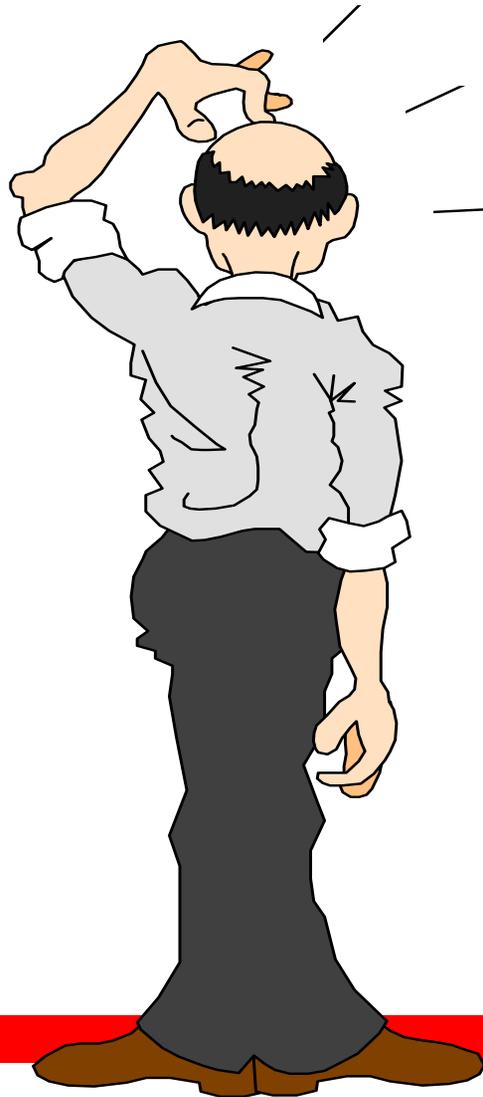


For More Information

Contact:

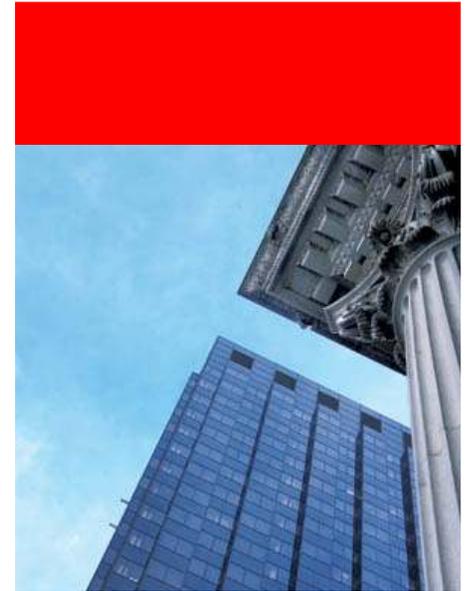
alex.collier@oracle.com – Service Engineering

Who has the First Question?





Appendix





ORACLE IS THE INFORMATION COMPANY

ORACLE®