WebSphere Continuous Test

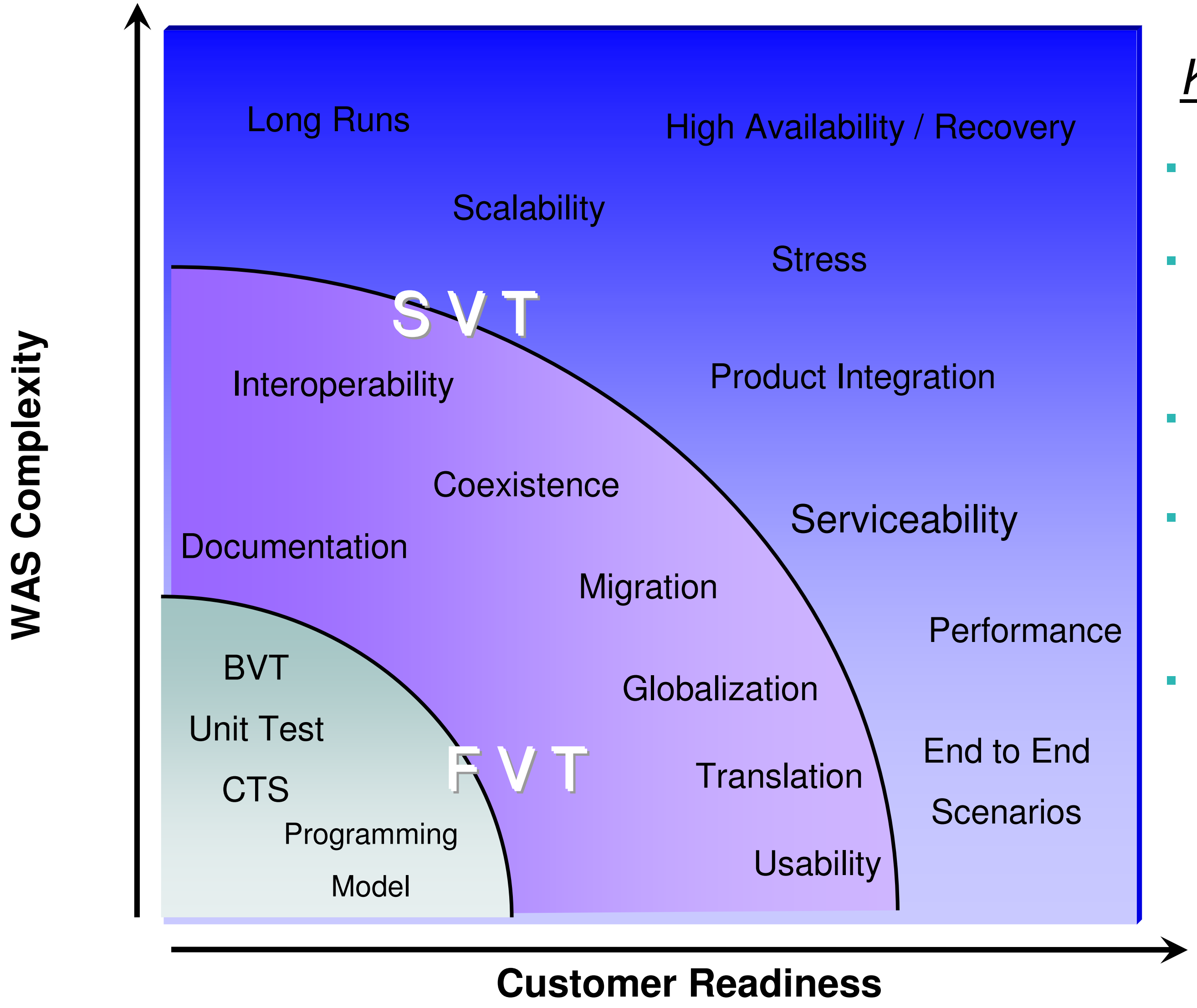
UK WebSphere User Group
March 4th, 2008

Tim Vanderham
WebSphere Senior Development Manager
Systems Management, Console, and Consumability
vandy@us.ibm.com

Agenda

- WebSphere Testing (prior to release)
  - Testing overview
  - Stack compatibility testing release to release
  - Fix pack testing (service stream)
  - Large topology and scale
  - SWG Integration testing
  - Future directions (Personas)
- WebSphere Environment Test Best Practices
  - Test methodology
  - Critical “hotspots” to monitor
  - Tools to understand the JVM
- Questions

WebSphere Product Test
Phased Approach



Key Phases

- **Unit Test:**
  - Developers testing basic capability of a specific new function/component
- **Function Verification Test (FVT)**
  - Tests code for a single function or interaction of functions, using a variety of parameters and sequence of events
- **Globalization Test**
  - Tests functions run correctly in non-English environments
- **Performance Test**
  - Tests the product meets performance objectives based on customer requirements and industry competitiveness
- **System Verification Test**
  - Focus on testing end to end customer usage patterns across all components in the system and all supported operating platforms
  - Long runs, stress, recovery, bad apps, user experience all play a part in this phase

WAS Quality Focus areas – Development cycle

- **WebSphere Early Design Program**
  - Allow customers and partners to have early view into use cases driving development and allow for input
- **Iterative Development cycle**
  - Incrementally construct a product's functionality one or more use cases (or scenarios) at a time with an “almost” full development cycle for better risk management and to get early feedback
- **Improved test automation**
  - Require automated test for new function
  - Increase testing efficiency and coverage from BVT to SVT
- **Improved robustness of System Verification Testing**
  - Limits, HA, Stress, Long runs, Error paths, Bad applications
  - Replicate customer usage patterns (ie industrial, financial, telco….)
- **APAR analysis for top APAR generating components**
  - Analyze PMRs/APARs for top components
  - Initiate specific actions to address design/code/doc issues

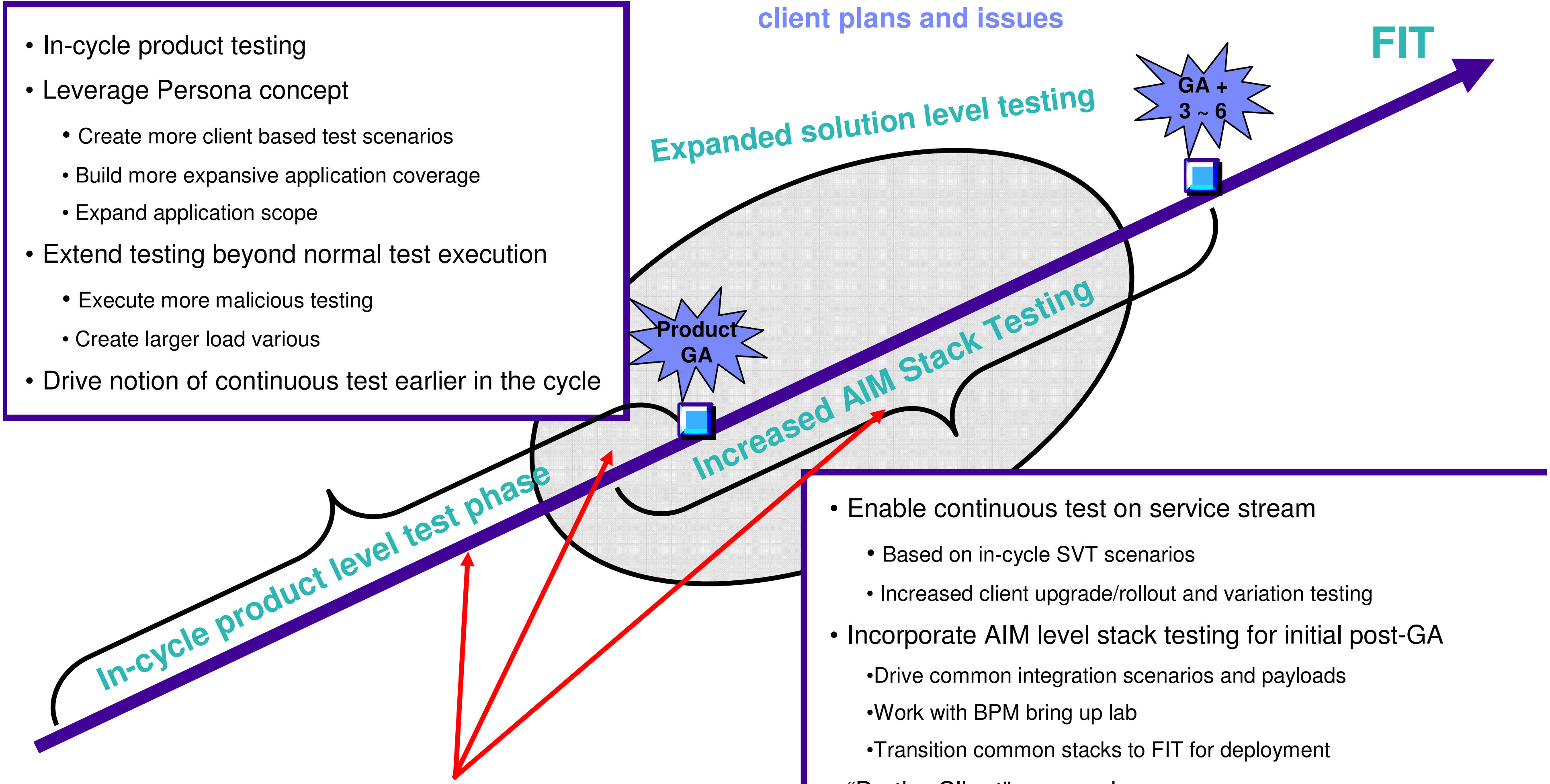
WebSphere Testing Overview

- **Phases of testing**
  - Unit testing
  - Build verification test
    - 900+ builds a week with 200+ automated build tests
  - Compatibility Test Suite
    - **23,000+ Java certification tests**
  - Function verification test
    - **35,000+ function tests**
  - System verification test
    - **1100+ complex customer scenario tested**
  - Performance test
    - **3 large complex end to end benchmarks executed in various configurations and scenarios**
  - Globalization verification test
    - Complete GVT on 8 distinct languages
  - Post GA Testing
    - JDK and OS certification testing
  - Service stream testing
    - Long Runs and stress testing for each fixpack (Started in 3Q 2006)

Stack Evolution of Test

- In-cycle product testing
- Leverage Persona concept
  - Create more client based test scenarios
  - Build more expansive application coverage
  - Expand application scope
- Extend testing beyond normal test execution
  - Execute more malicious testing
  - Create larger load various
- Drive notion of continuous test earlier in the cycle

Leverage Large Topology Environments
Capture additional test variations form client plans and issues

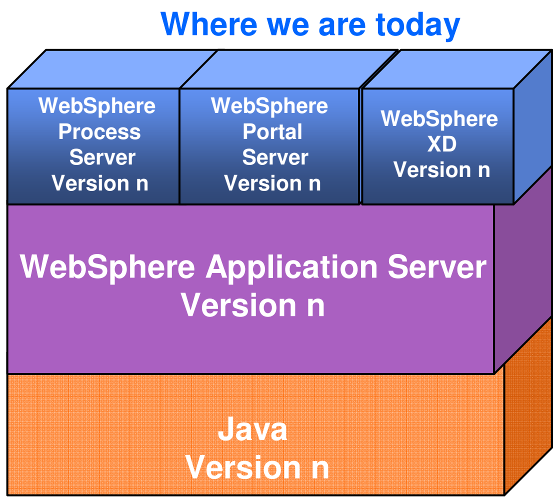

FIT
GA + 3 ~ 6
Expanded solution level testing
Product GA
Increased AIM Stack Testing
In-cycle product level test phase

- Enable continuous test on service stream
  - Based on in-cycle SVT scenarios
  - Increased client upgrade/rollout and variation testing
- Incorporate AIM level stack testing for initial post-GA
  - Drive common integration scenarios and payloads
  - Work with BPM bring up lab
  - Transition common stacks to FIT for deployment
- “Be the Client” approach
  - Don’t cheat, use resources available to clients
  - Provide input to needed collateral

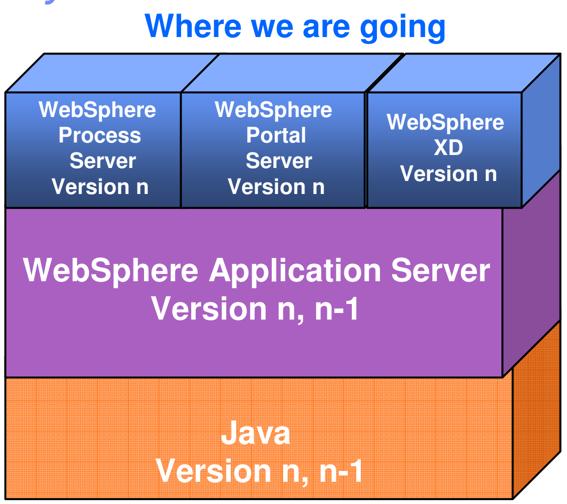
Incorporate critical SWG products

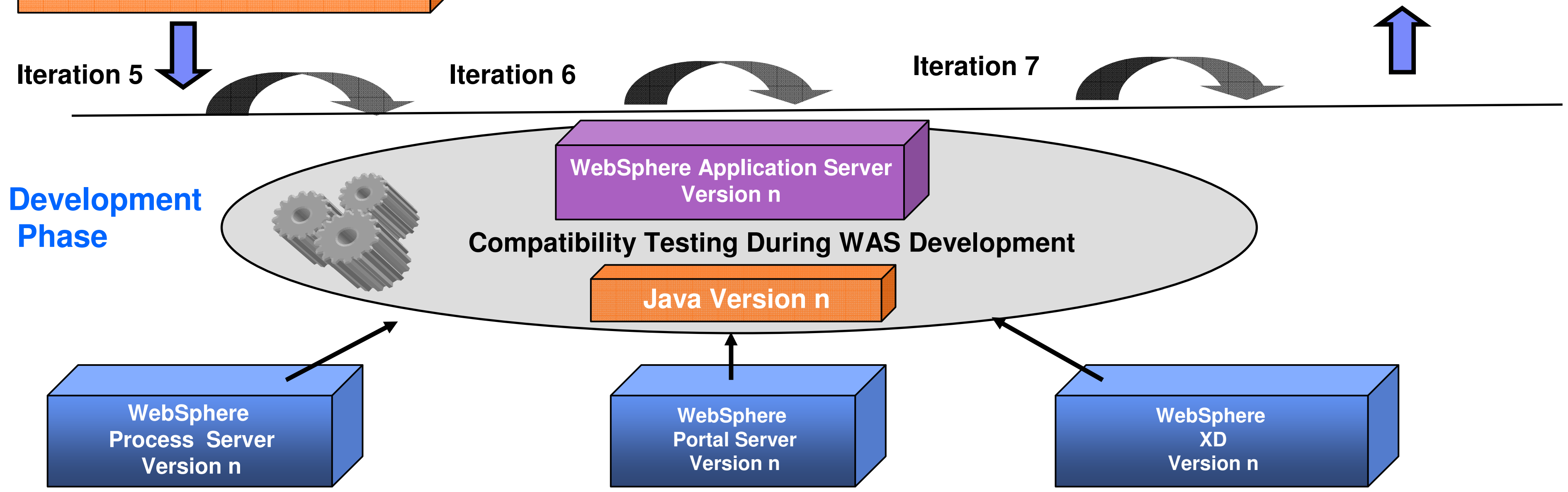
SWG Stack Alignment

Improved consumability through release to release compatibility


Where we are today
WebSphere Process Server Version n
WebSphere Portal Server Version n
WebSphere XD Version n
WebSphere Application Server Version n
Java Version n

- Testing key SWG stack products during WAS development cycle
- Realizing benefits of Agile development
  - Stable code at the end of each iteration
  - Leveraging agile development practices to deliver earlier code drops
- Leverage technology like OSGi to provide an extensible core run-time
- Driven by customer feedback and usage patterns


Where we are going
WebSphere Process Server Version n
WebSphere Portal Server Version n
WebSphere XD Version n
WebSphere Application Server Version n, n-1
Java Version n, n-1


Iteration 5
Iteration 6
Iteration 7
Development Phase
WebSphere Application Server Version n
Compatibility Testing During WAS Development
Java Version n
WebSphere Process Server Version n
WebSphere Portal Server Version n
WebSphere XD Version n

WebSphere Service Stream Continuous Test

- Increase testing coverage on service stream Fix Packs
  - Cover all service streams
    - Feature Packs
    - Java SDK updates
  - Test each Fix Pack on a variety of hardware
- Long Runs
  - Continuous 7-days under load in ND client based environment
    - 80+% CPU utilization,
    - 500+ clients
    - 150+ transactions/sec
- Various application workloads rotated across Fix Pack test cycles
  - SM Continuous Deployments
    - Stresses administration via continually deploying / redeploying applications
  - DayTrader
    - Coverage: EJB 3.0, JPA, JSP 2.1, Servlet 2.5, Platform Messaging
  - Garage Sale
    - Coverage: EJB 3.0, JPA, JSP 2.1, Servlet 2.5, JSF, Web Services Features
  - GasNet
    - Coverage: Shareable Local Transaction containment, platform messaging
  - Acme
    - Coverage: Security (JACC), Web Services Features, Thin clients, JNLP
  - Trade -- Performance benchmarking application
  - ERWW
    - Coverage: JDBC 4.0, EJB3.0, JPA, SQLJ, Platform Messaging, JACC

Service Stream – SVT Continuous Long Run Testing

- Executing upgrades and monitoring environments based on client scenarios
  - Monitored during long runs
    - Heap size
    - CPU utilization
    - Various component logs

- Driving increased size and complexity of Long Runs
  - Combining multiple stress scenarios in for full shared environment simulation
    - Example
      - SM/Trade
      - GasNetwork/GasNetwork
  - Feature packs within one cell for coexistence validation
  - Increasing number of application servers per cell
  - Collaborating with Development / Support / Clients to define additional testcase and workload variations

- Stack product validation testing for WAS fix packs
  - WXD, WPS, Commerce, and Portal

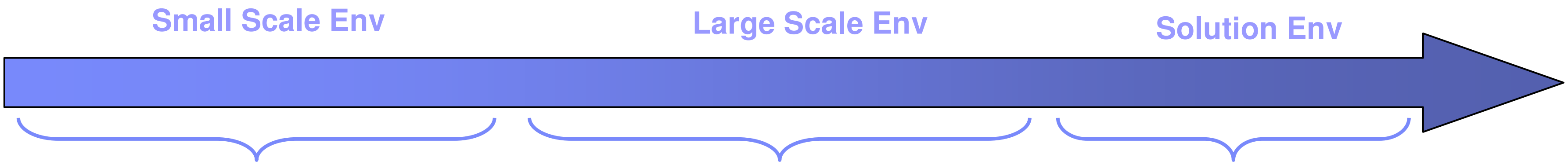
- Including additional products, latest Fix Packs to increase coverage
  - Recent additions
    - ITCAM v6.1
    - Oracle 11g
    - AIX v6.1

WebSphere Large Topology Testing

- Operating Principles:
  - Focus on complex client topologies and usage scenarios
  - Remain agile in adapting to focus on “hot spots”
  - Continuously executing, updated and maintained
  - Catalyst for driving improvements in development and test
- Manipulating multiple levers creates more moving parts and overall stress
  - Large Scale Migration under load – security enabled, high load, large scale, web/ejb traffic, mixed cell
  - Mixed Cell environments – multiple WSAS versions within one cell, Systems Management stress, application load, security enabled
  - XD environments – Mixed and non mixed cells, high load, ODR, dynamic placement, app load, malicious testing
- Defects to date addressed as a result of multiple conditions
  - Scale of the environment
  - Complex configurations
  - Continuous usage scenarios
  - Malicious testing and misbehaving applications
  - Some defects found not attributable to Large Topology Test uniqueness

WebSphere Large Topology Stack Integration

- Integrate products with large scale client adoption
- Collaborate with development and test teams of stack products
- Proactively address concerns



Small Scale Env

Large Scale Env

Solution Env

Integrate

Discover integration issues with a small scale environment

- Example config: 1 DMGR / 3 nodes
- Utilize recommended best practices and recommended applications
- Integration products
  - WPS
  - Portal
  - DataPower

Scale

Utilize stable large scale environments

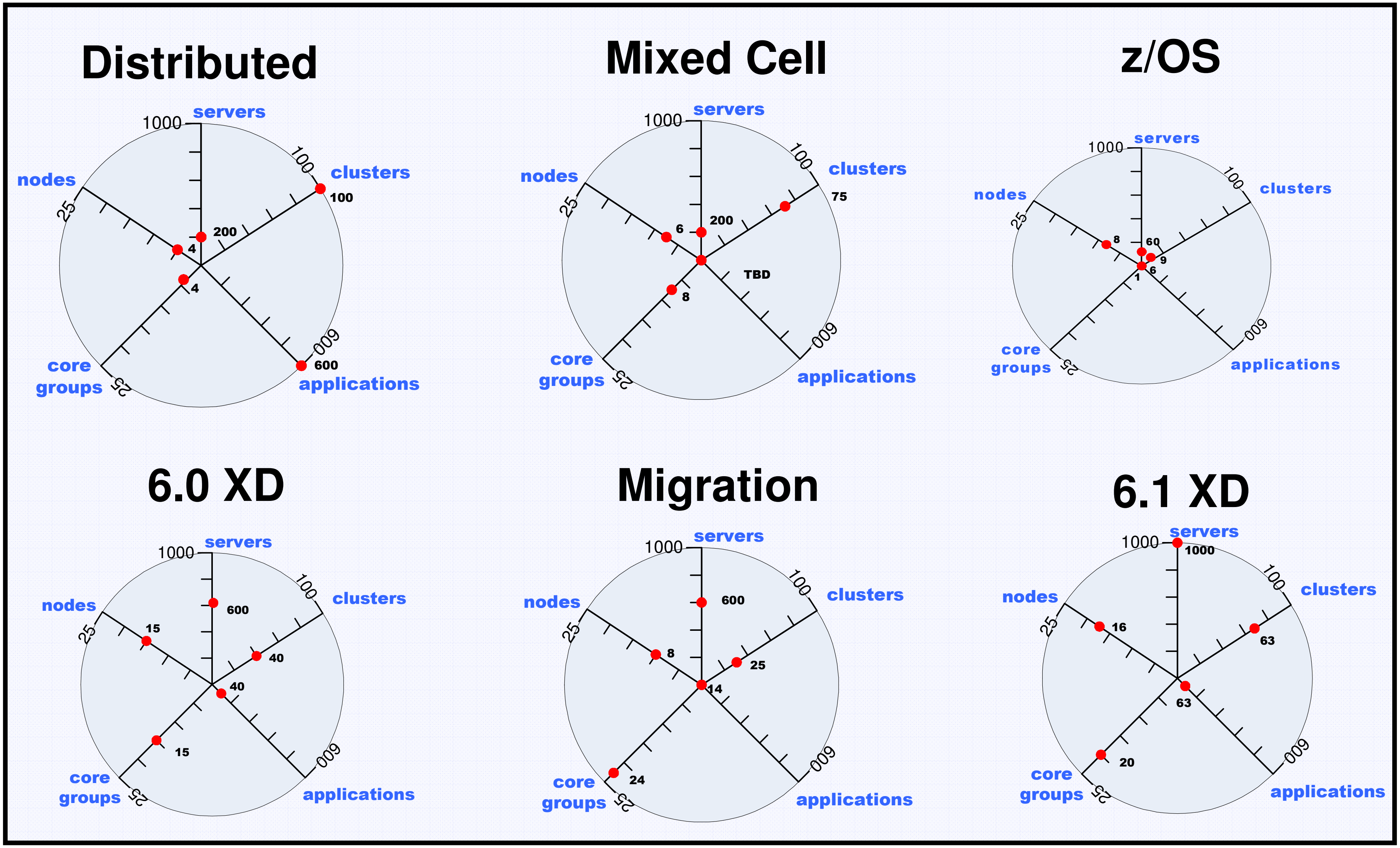
- Example config: 150-400+ AppServers within one cell
- Utilize various applications, usage patterns, load, and malicious apps
- Scaled products
  - WSAS
  - ITCAM
  - XD

Solution

Combine multiple products into one single large scale solution

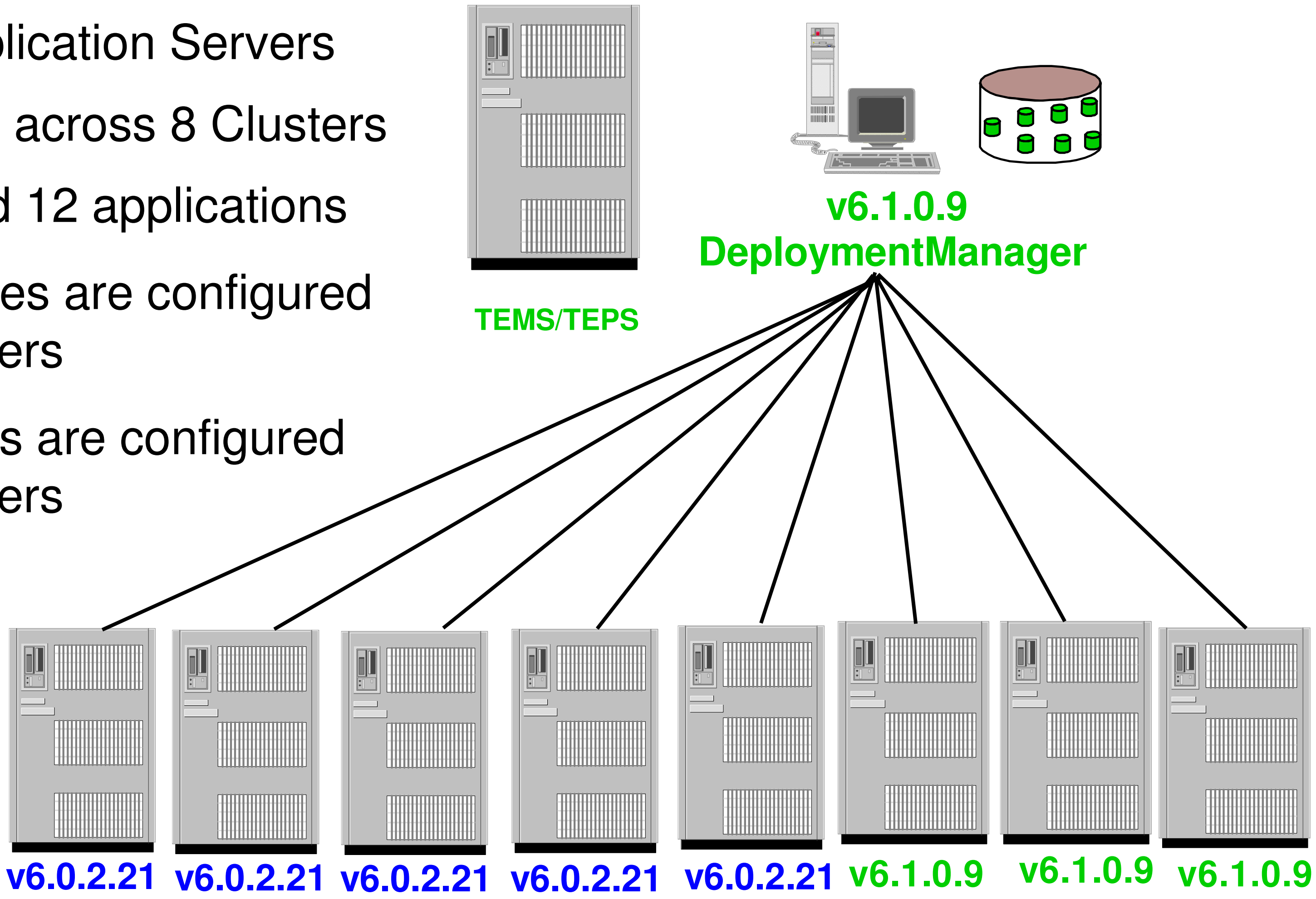
- Include ITCAM in solutions as monitoring utility
- Known client adoptions
  - XD/ITCAM
  - WPS/Portal/ITCAM
  - WPS/WSRR/ITCAM

Example of Current Environments



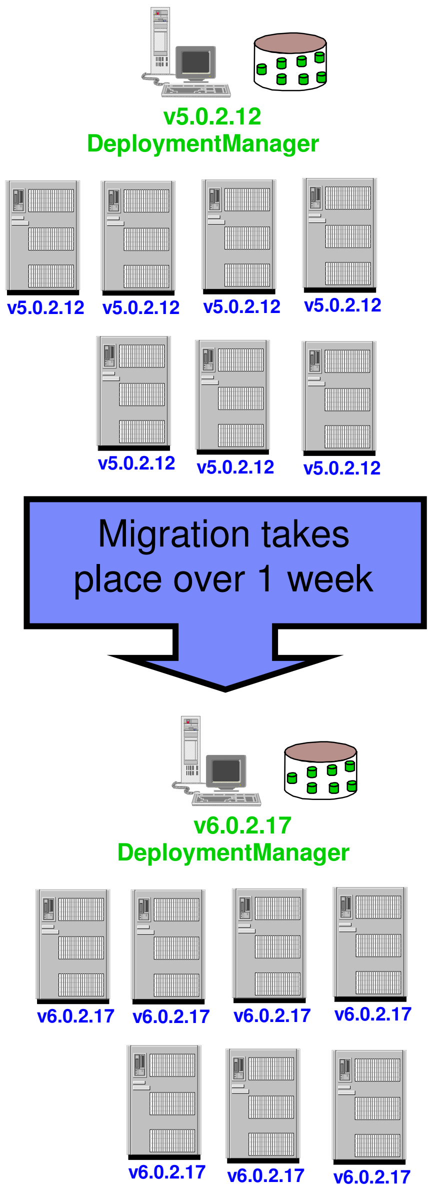
z/OS Mixed Cell environment

- 70 Total Application Servers
  - Spanned across 8 Clusters
  - Deployed 12 applications
- 6.0.2.21 nodes are configured with 42 Servers
- 6.1.0.9 nodes are configured with 28 Servers



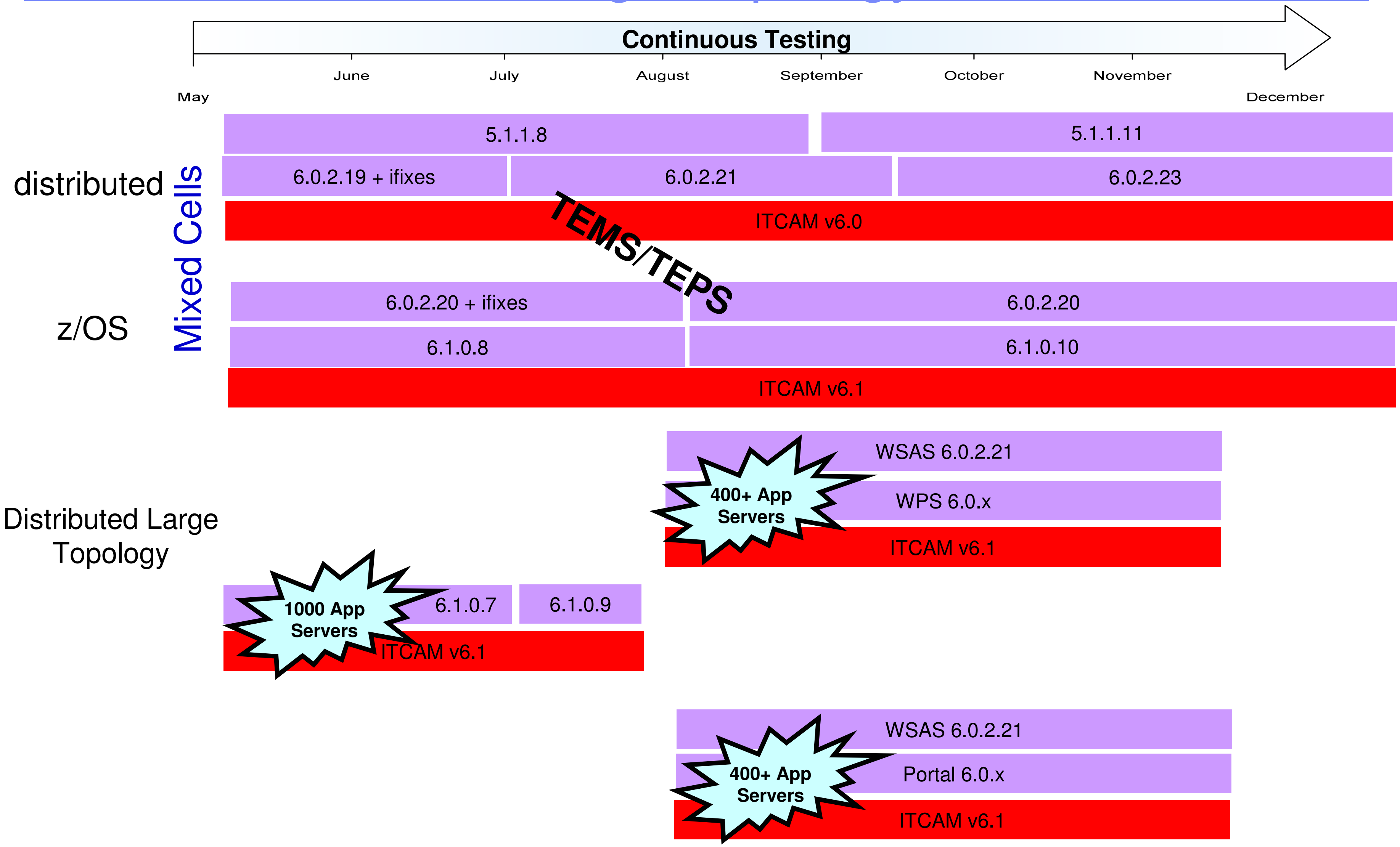
Large Scale Migration Under Load

- Migrate 8 Nodes of pure V5 to V6
  - V5 Nodes: WAS 5.0.2.12 with JDK SR8 + 2 fixes
  - V6 Nodes: WAS 6.0.2.17 with JDK SR6 + fix bundle
- General Overview
  - Environment contained 285 application servers
  - Maintain application availability thru migration window
  - Heavily leveraging HA Manager and CoreGroupBridge
- Completed specific multiple times
  - Completed with 285 servers
  - Completed with 520 servers
  - Provided documentation to clients for specific migration tasks
- Environment leveraged to recreate issues and validate documentation



Migration takes place over 1 week

Customer Focused Large Topology Test Timeline ‘07



Continuous Testing

May
June
July
August
September
October
November
December

distributed

Mixed Cells

5.1.1.8
5.1.1.11
6.0.2.19 + ifixes
6.0.2.21
6.0.2.23
ITCAM v6.0

TEMS/TEPS

z/OS

6.0.2.20 + ifixes
6.0.2.20
6.1.0.8
6.1.0.10
ITCAM v6.1

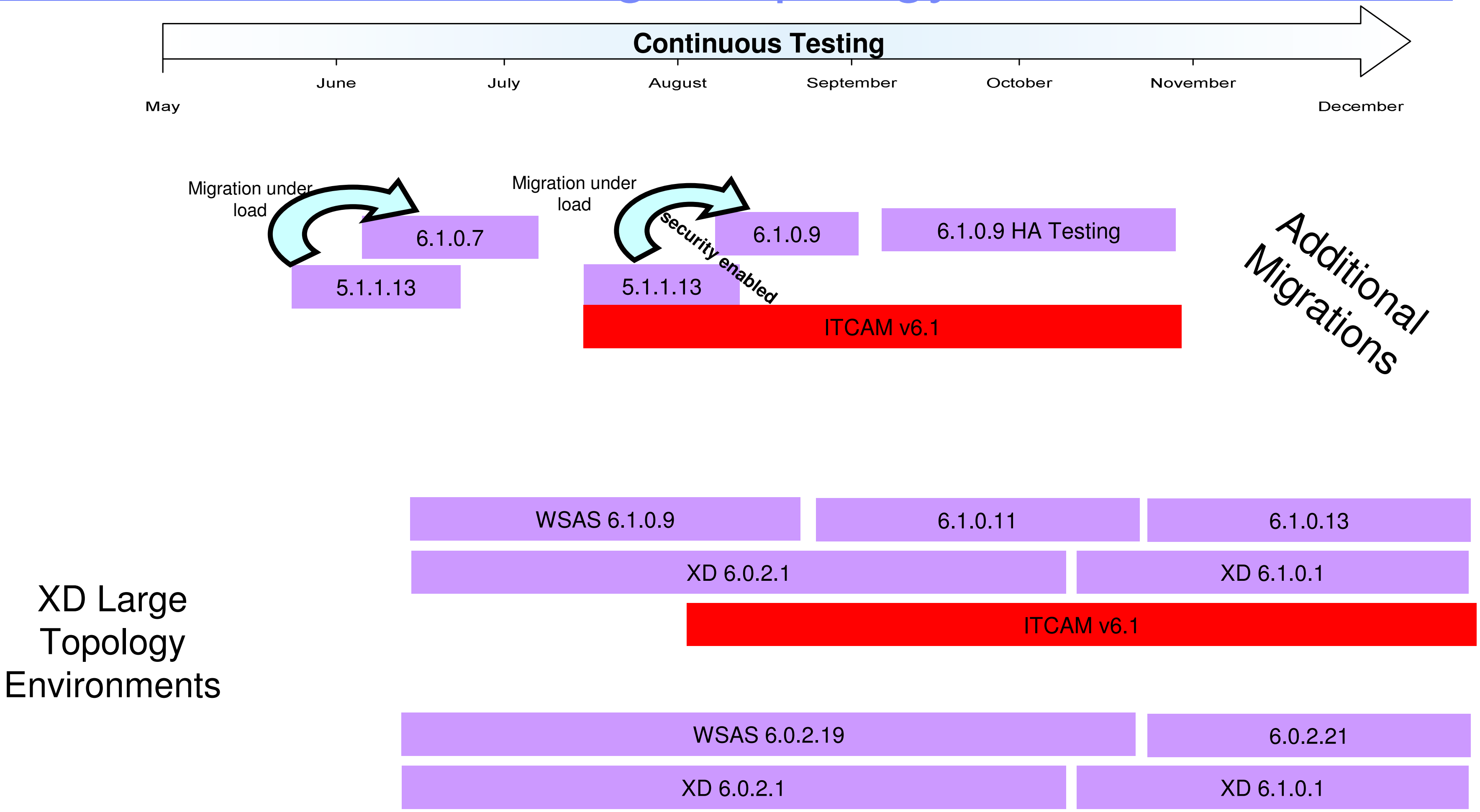
Distributed Large Topology

400+ App Servers
WSAS 6.0.2.21
WPS 6.0.x
ITCAM v6.1

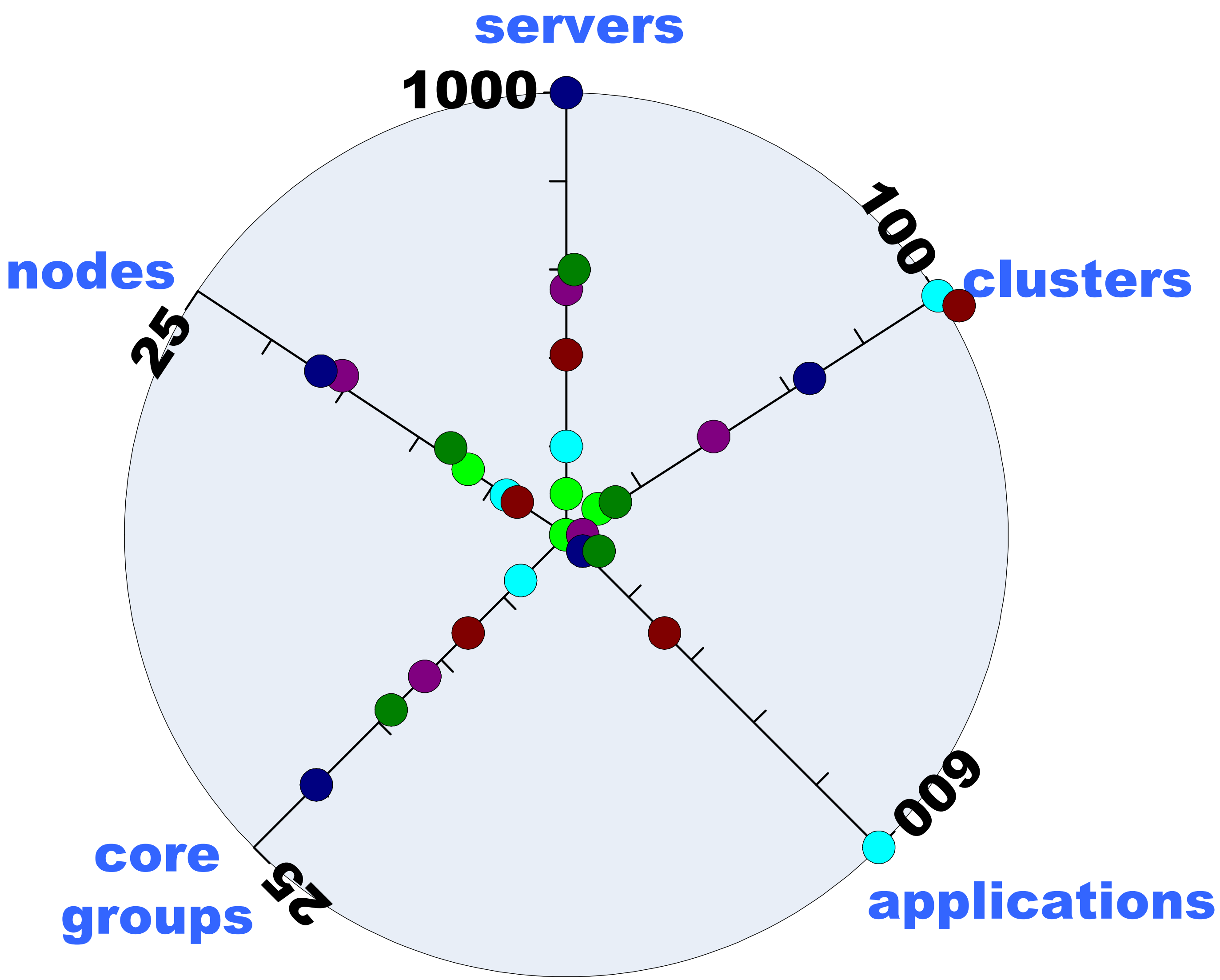
1000 App Servers
6.1.0.7
6.1.0.9
ITCAM v6.1

400+ App Servers
WSAS 6.0.2.21
Portal 6.0.x
ITCAM v6.1

Customer Focused Large Topology Test Timeline '07



All Topologies



SPECjAppServer2004 benchmarking leadership overtime

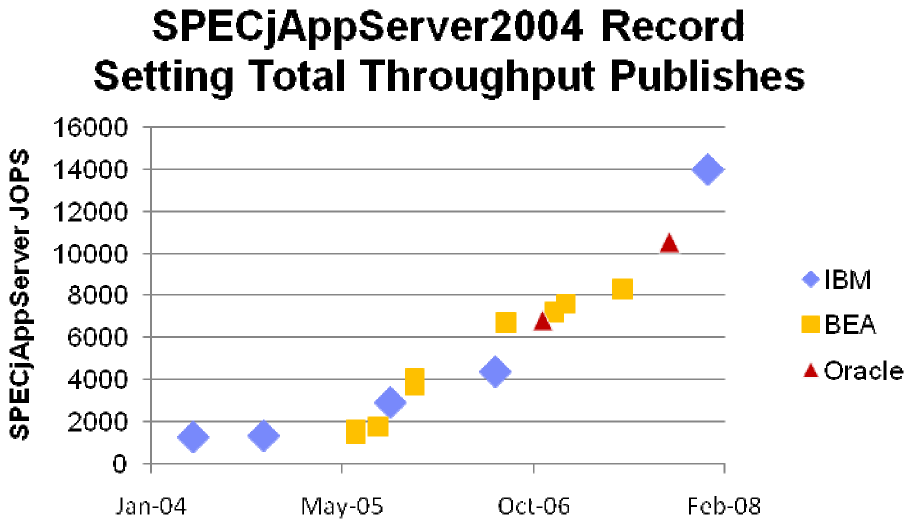
SPECjAppServer2004 Leadership Timeline

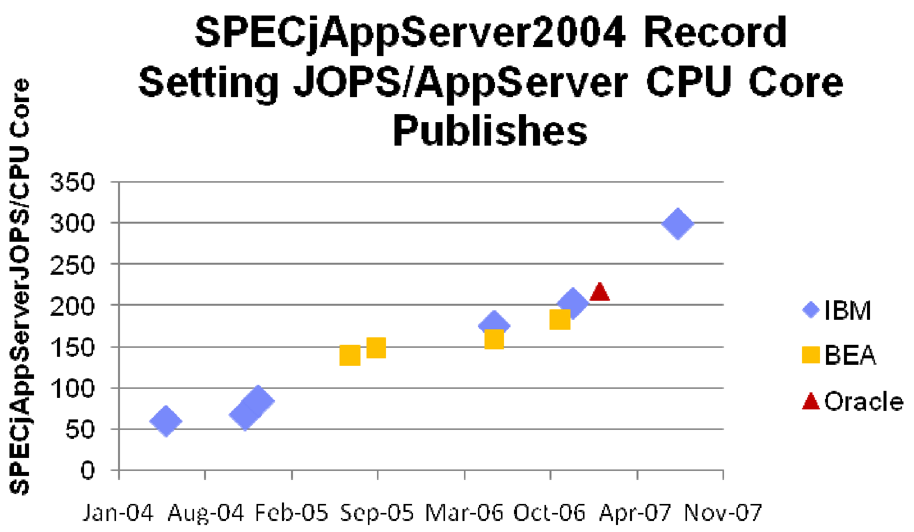
- **April 2004** – SPECjAppServer2004 released. IBM first to publish on benchmark
- **May 2004 – July 2005** – IBM Continues to publish higher throughput numbers while no other competitors submit
- **October 2005** – IBM publishes record setting total configuration result numbers almost 65% better than the competition
- **July 2006** - IBM again publishes record setting total configuration results.
- **December 2006** – IBM publishes dominate SPECjAppServer2004 JOPS/Core results
- **November 2007** – IBM publishes industry leading SPECjAppServer2004 JOPS/Core over 36% faster than the nearest competition
- **January 2008** – **IBM releases total configuration results that top the nearest competitor be 33% and produce and astounding 75,000 database transactions per second**

"IBM WebSphere Application Shatters Industry Benchmark, Powers SOA" – CNN Money

CNNMoney.com
A Service of CNN, Fortune & Money

2/15/2008



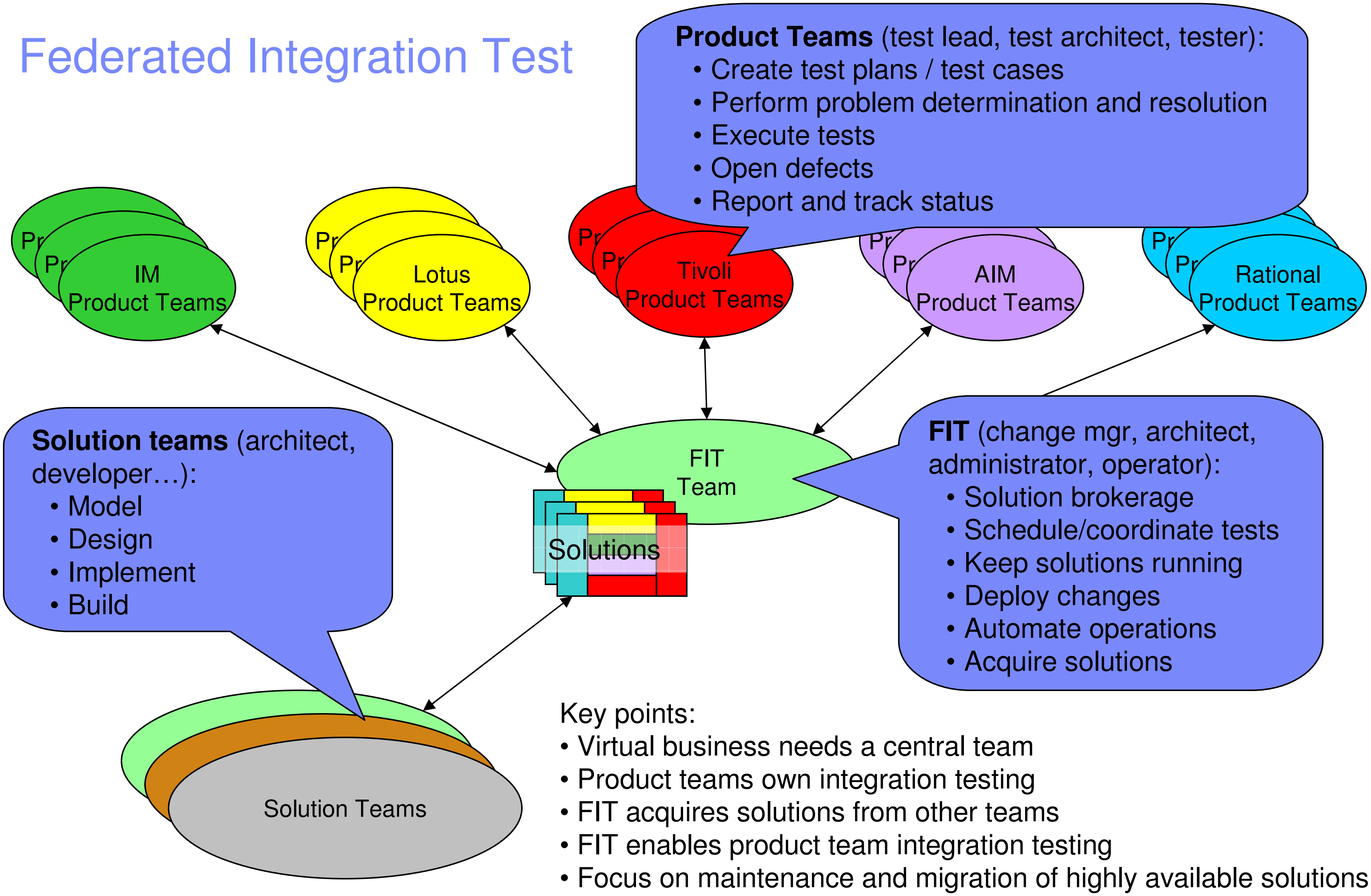


SWG Federated Integration Test (FIT)

Current focus on service stream

- Building a collaborative test lab that is centrally managed but enables brand product teams to do their own integration and/or service stream testing
- All tests are owned by individual product teams and executed with assistance from the FIT Core team
  - Product and core test teams open defects against their own product as well as other products (as appropriate)
- FIT environments is always ready to be used by brand product teams
  - i.e. continuously running systems in which you update/add infrastructure, products and applications
- Provides additional system level test suite to individual product testing

Federated Integration Test



Product Teams (test lead, test architect, tester):

- Create test plans / test cases
- Perform problem determination and resolution
- Execute tests
- Open defects
- Report and track status

Solution teams (architect, developer…):

- Model
- Design
- Implement
- Build

FIT (change mgr, architect, administrator, operator):

- Solution brokerage
- Schedule/coordinate tests
- Keep solutions running
- Deploy changes
- Automate operations
- Acquire solutions

Key points:

- Virtual business needs a central team
- Product teams own integration testing
- FIT acquires solutions from other teams
- FIT enables product team integration testing
- Focus on maintenance and migration of highly available solutions

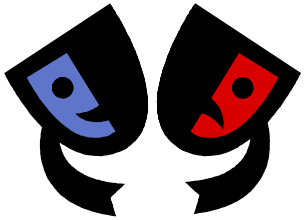
Roadmap to a better understanding of WebSphere customers

Customers aren’t all alike, but they are not all completely different from one another either

A small number of customer archetypes share many common characteristics

1. Identify and prioritize these customer types,
2. Devise a standard way to describe the types, and
3. Provide sufficient details to be useful to testers (and designers)

Introducing WebSphere Customer Personas



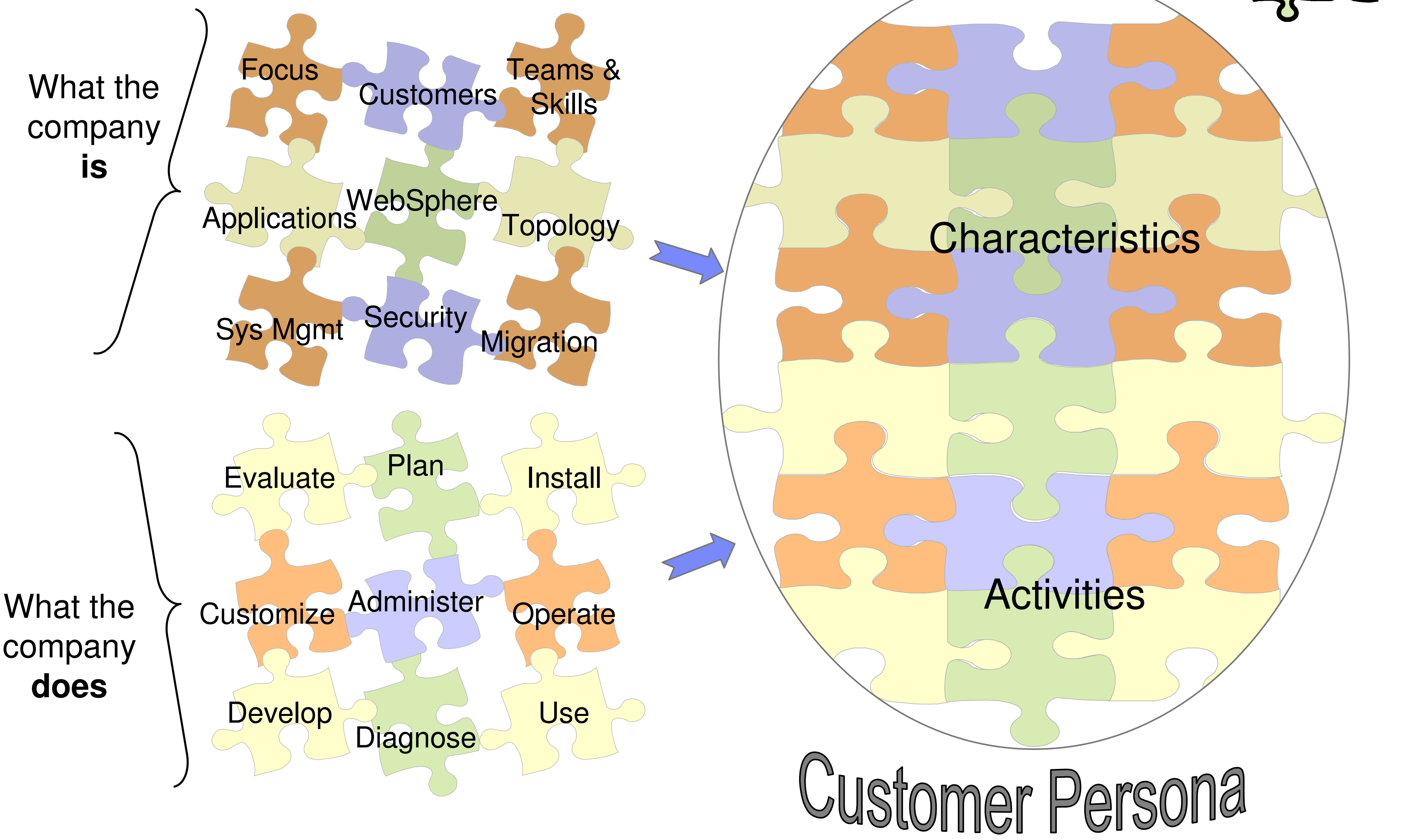
A **concise descriptive model** of a company, what it wishes to accomplish, and why.

A **composite archetype** based on behavioral and descriptive data gathered from many actual companies that share related usage patterns.

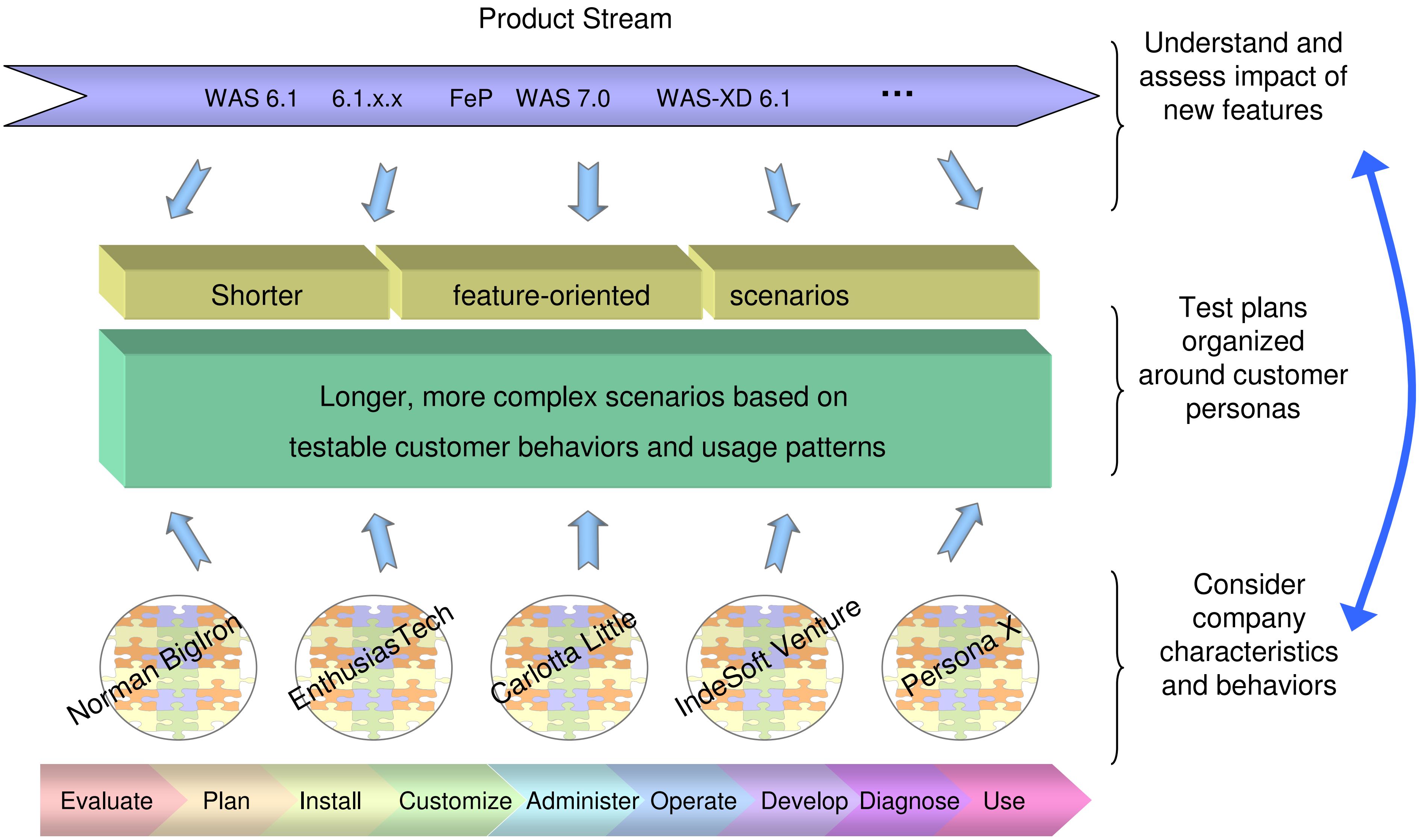
Our company personas are intended to provide WebSphere development and testers with a consumable source of customer information that can serve as a context for writing more customer-oriented test scenarios.

- **Better understanding of customers = more effective design and testing**

Persona composition



Developing test plans



Product Stream

WAS 6.1 6.1.x.x FeP WAS 7.0 WAS-XD 6.1 …

Shorter feature-oriented scenarios

Longer, more complex scenarios based on

testable customer behaviors and usage patterns

Norman BigIron

EnthusiasTech

Carlotta Little

IndeSoft Venture

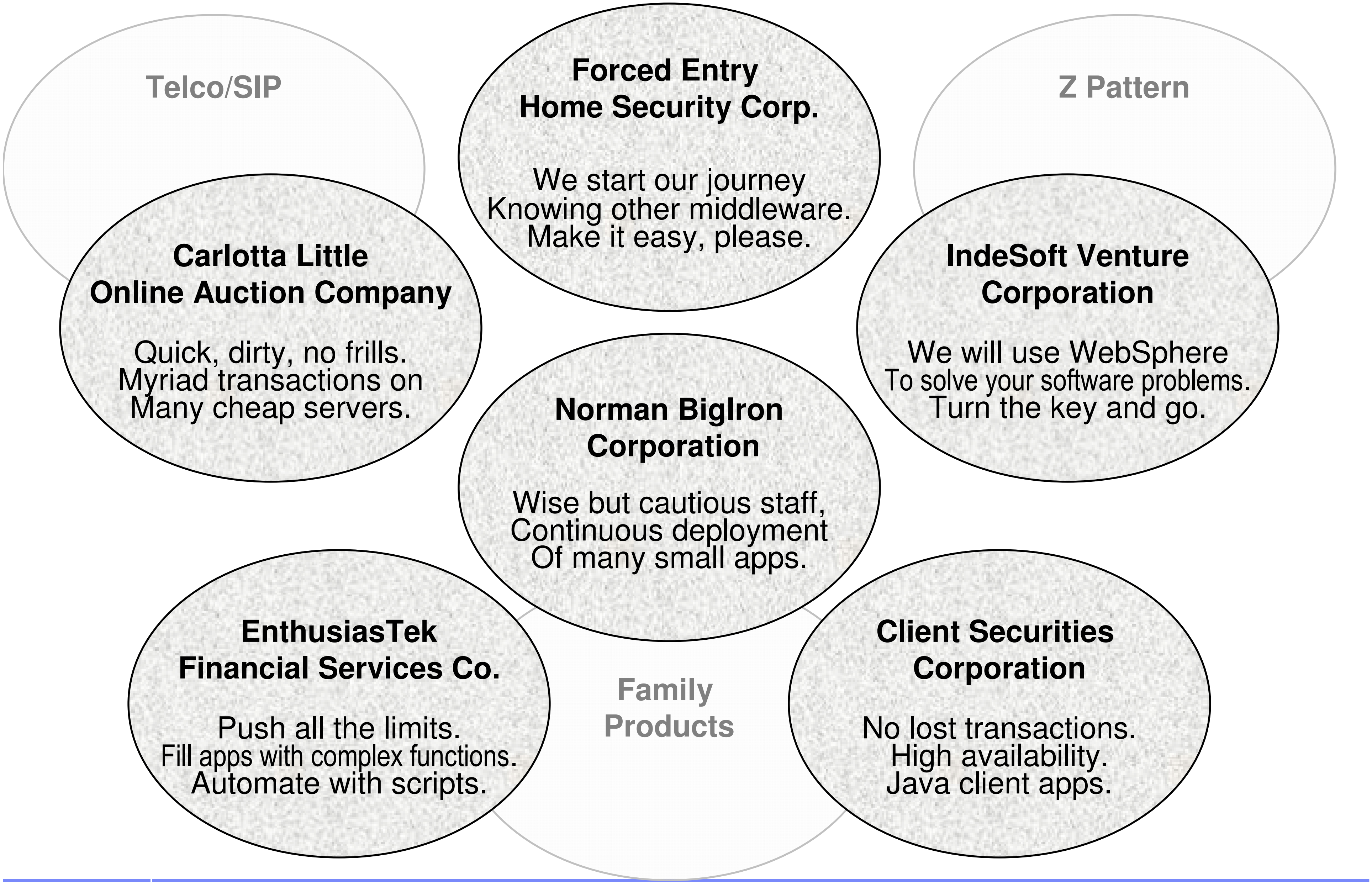
Persona X

Evaluate Plan Install Customize Administer Operate Develop Diagnose Use

Understand and assess impact of new features

Test plans organized around customer personas

Consider company characteristics and behaviors



Telco/SIP

Forced Entry
Home Security Corp.

We start our journey
Knowing other middleware.
Make it easy, please.

Z Pattern

Carlotta Little
Online Auction Company

Quick, dirty, no frills.
Myriad transactions on
Many cheap servers.

IndeSoft Venture
Corporation

We will use WebSphere
To solve your software problems.
Turn the key and go.

Norman BigIron
Corporation

Wise but cautious staff,
Continuous deployment
Of many small apps.

EnthusiasTek
Financial Services Co.

Push all the limits.
Fill apps with complex functions.
Automate with scripts.

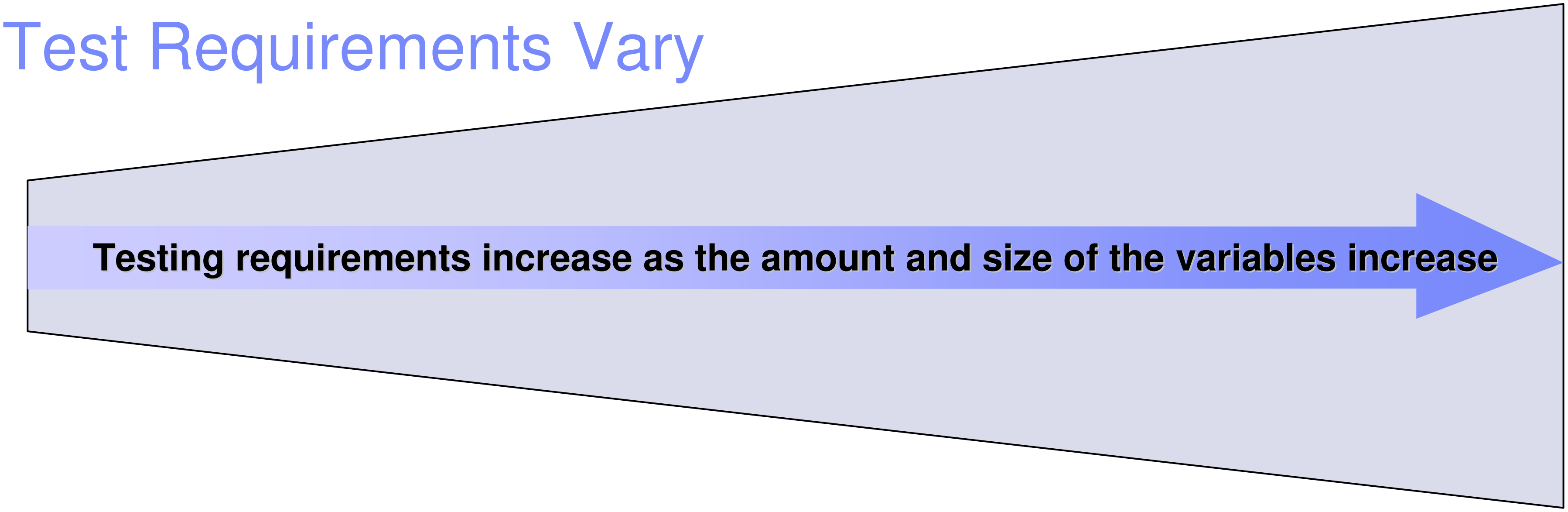
Family
Products

Client Securities
Corporation

No lost transactions.
High availability.
Java client apps.

WebSphere End User Testing Best Practices

Test Requirements Vary



Testing requirements increase as the amount and size of the variables increase

Small

Minor application changes

New application promotions

Runtime configuration changes

Product related iFixes
- Anywhere in the environment

Medium

Product fixpack updates

Significant application changes

New application deployments

Large

WAS version upgrade
- Any product version upgrade

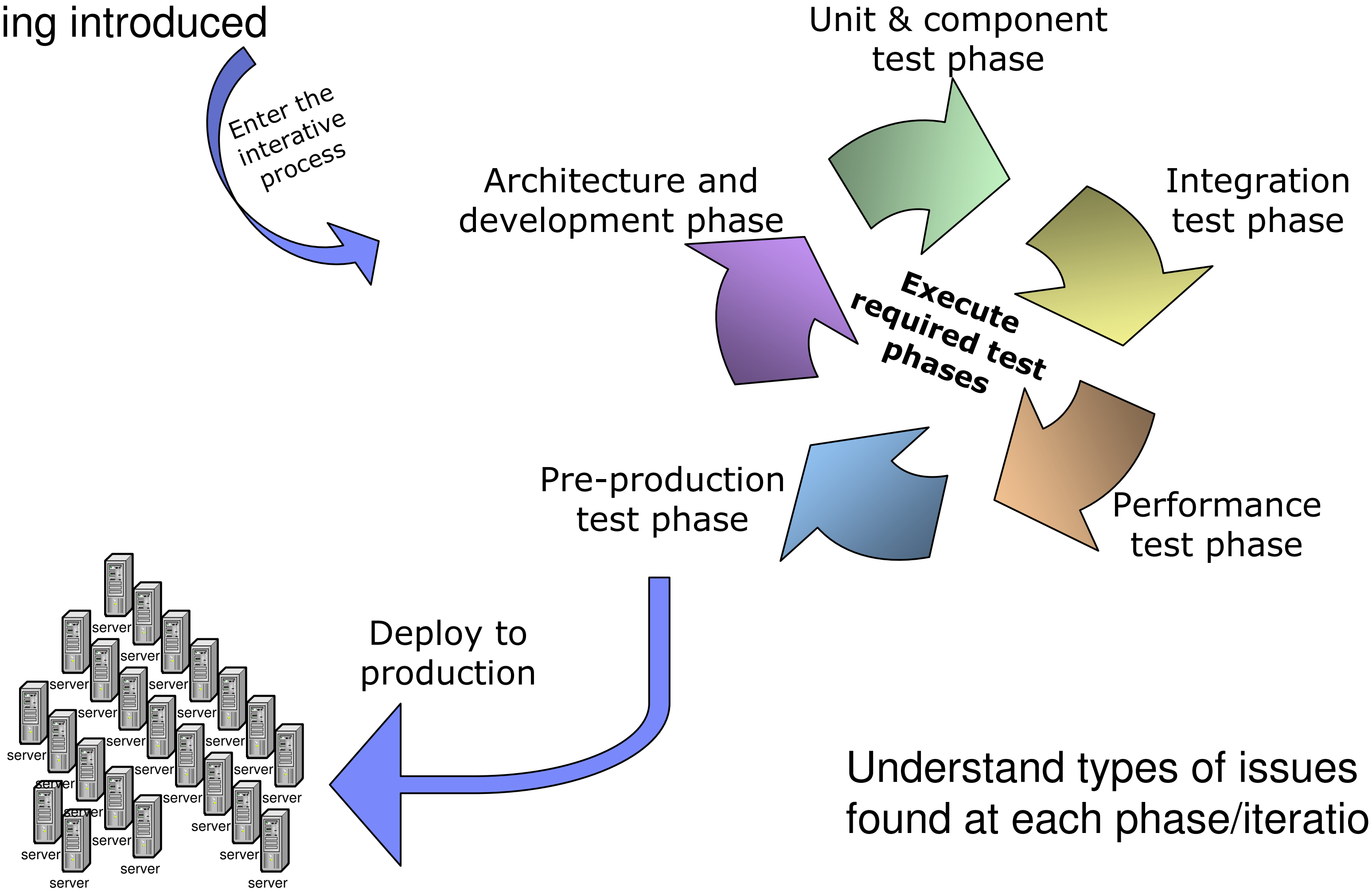
Operating system change

Major application changes and interop modifications

Test environments must represent production

Test Phases Allow for Iterative Planning

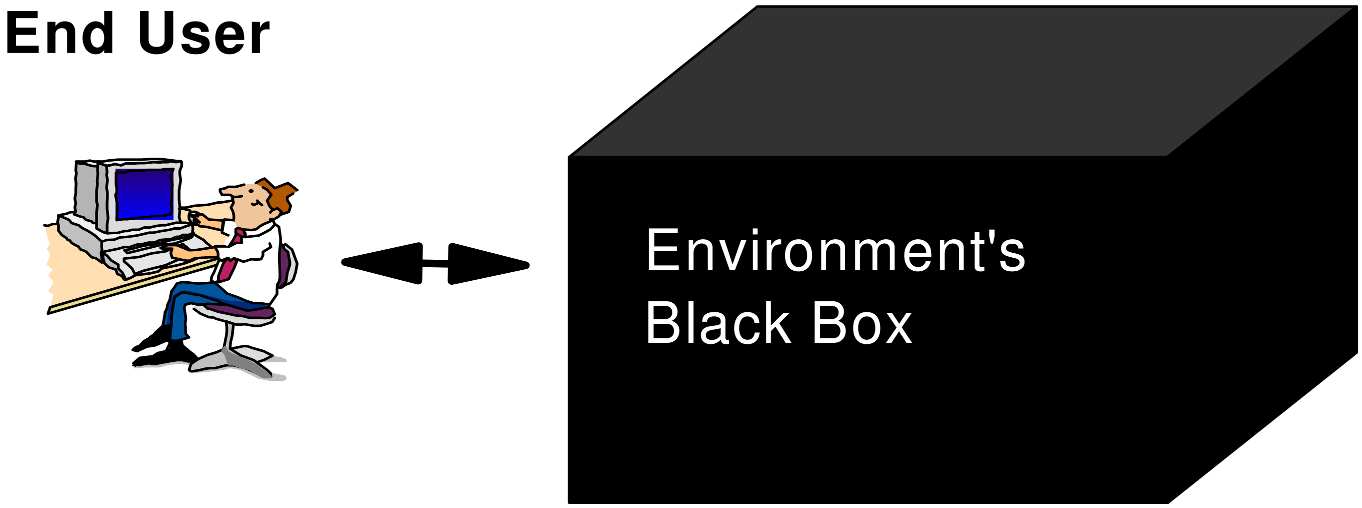
Understand the magnitude of change being introduced



Understand types of issues found at each phase/iteration

Testing Planning: Black Box

- What is in the box?
- Where are the integration points?
- What is expected?
- How is it used?
- What is new or changing?

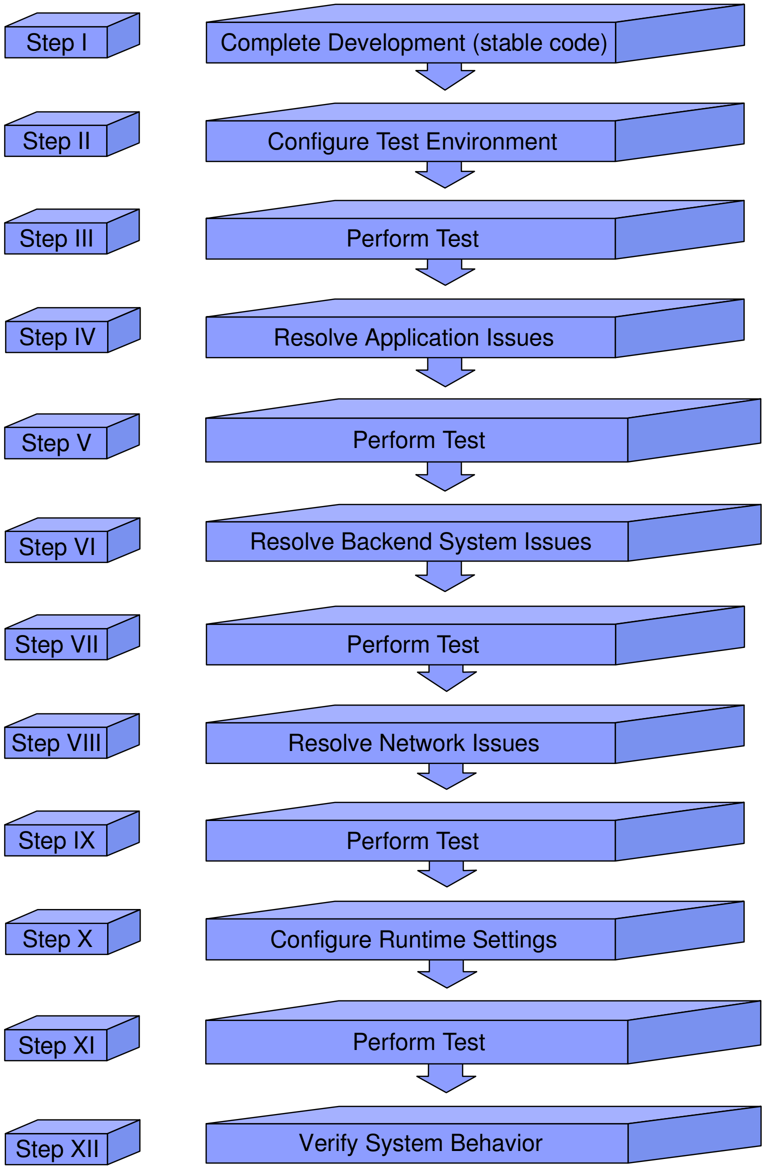


End User

Environment's Black Box

Testing Life Cycle

- Life cycle can be applied within or across the phases
- Iterations and specific tests depend upon application and environment
- Mutiple changes adds test complexity and overall risk
- Understand expected results at each phase and measure



WebSphere Tuning Misconceptions (Performance and Beyond)

- Set the "Magic Knob"
- Add more memory to the system
- Add another server and cluster the application server
- Customer XYZ used these WebSphere settings
- Just write the code and then deploy
- BOTTOM LINE
  - YOU CAN'T TUNE OR CLONE YOUR WAY OUT OF A POORLY DESIGNED APPLICATION
  - NOT EVERY APPLICATION CAN BE TUNED THE SAME WAY

When to Think About Application Integration & Performance

- Architecture design
- Development period
- Unit/Component test phase
- Integration test phase
- Performance test phase
- Pre-production test phase
- Deployment and rollout
- Bottom line
  - **Always keep integration and performance in mind**

What does Testing Entail….(partial list)

- Integration testing
  - Backend systems (DBs, Messaging, Legacy, etc)
  - Stack products and vendor applications/products
  - Additional environment applications (solutions)
- Load testing
  - Simulate user activity
  - Based on actual business patterns
    - Current business transaction rates
    - User loads increased by 10 to 20 percent
  - Identify load level with acceptable response times
  - Provide metrics on performance bottlenecks
- Stress testing
  - Evaluate performance at levels well beyond estimated loads
  - Burst loading effects
  - Sustained tests at extremely high loads
- Failover testing
  - Crash and recovery testing
- User acceptance testing under load

Testing Methodology Imperatives

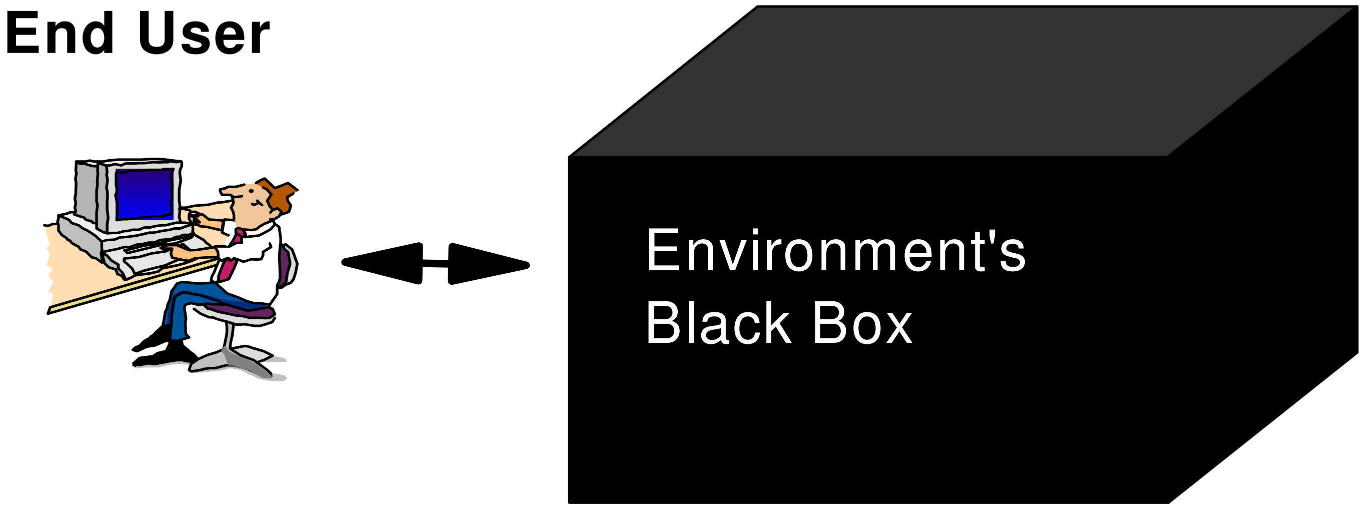
- Test from an end-user perspective
  - End-to-end response times
  - Overall view of application behavior
- Use tools to monitor each piece of the puzzle
  - Specific application code paths
  - Web Server
  - Application Server
  - Server resource use
  - Network health
- Correlate results from tools to determine the problem piece
- Represent deployment
  - Environment (configuration and size/scale) (at least proportional)
  - Application(s)
  - Number of users / transactions
  - Scenarios

Testing Methodology Imperatives (cont)

- Test all key business processes
  - Include most intensive business logic
  - Include database activity
  - Typically about 70 to 80 percent code coverage
  - Look for processes with different code paths
- Follow a disciplined and repeatable process
  - Follow a set of processes and procedures
  - Document every run and modification
  - Make controlled and limited changes simultaneously
  - Validate results
  - Ensure problems are real
  - Easier to replicate

Testing Planning: Black Box

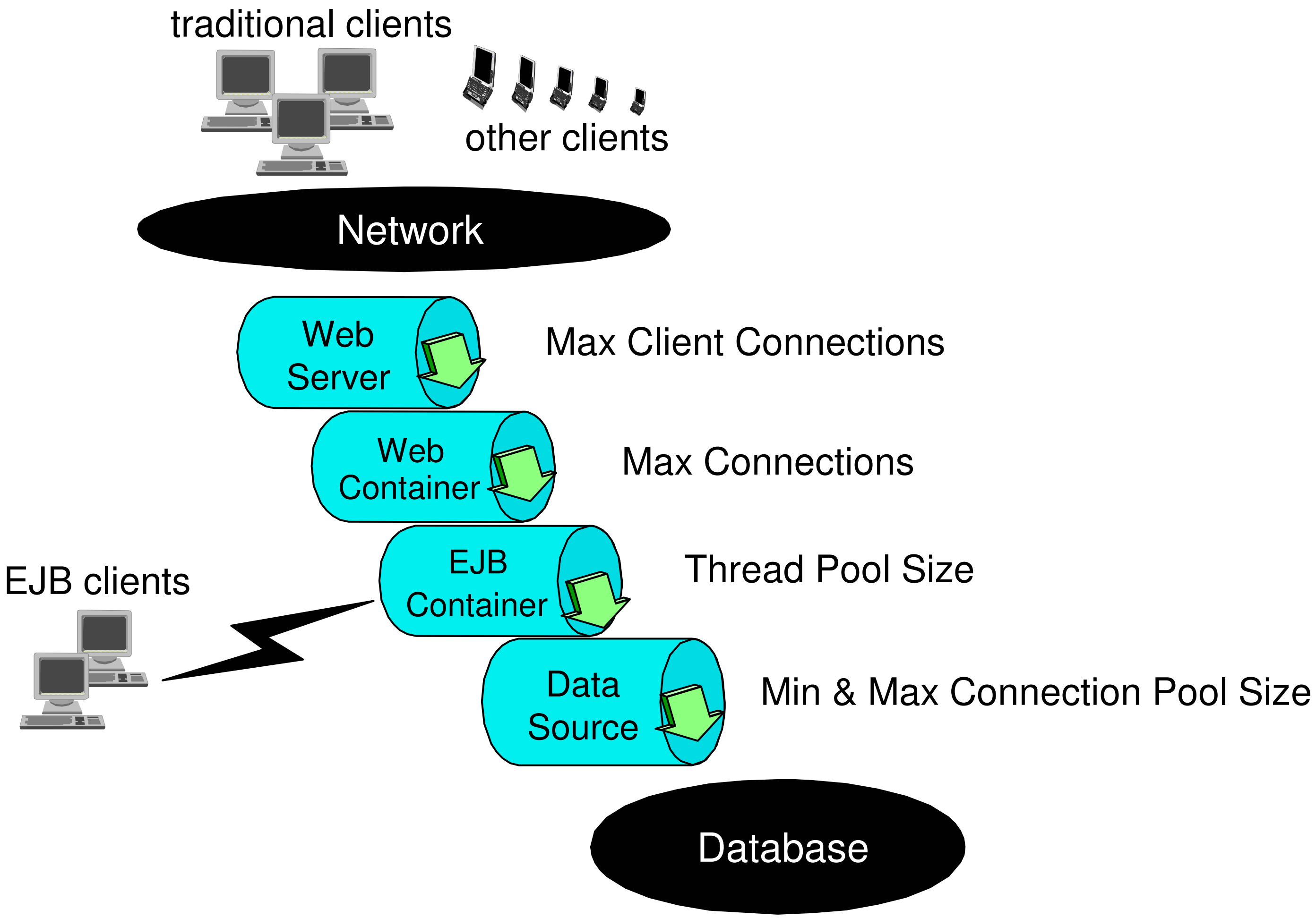
- What is in the box?
- Where are the integration points?
- What is expected?
- How is it used?
- What is new or changing?



End User

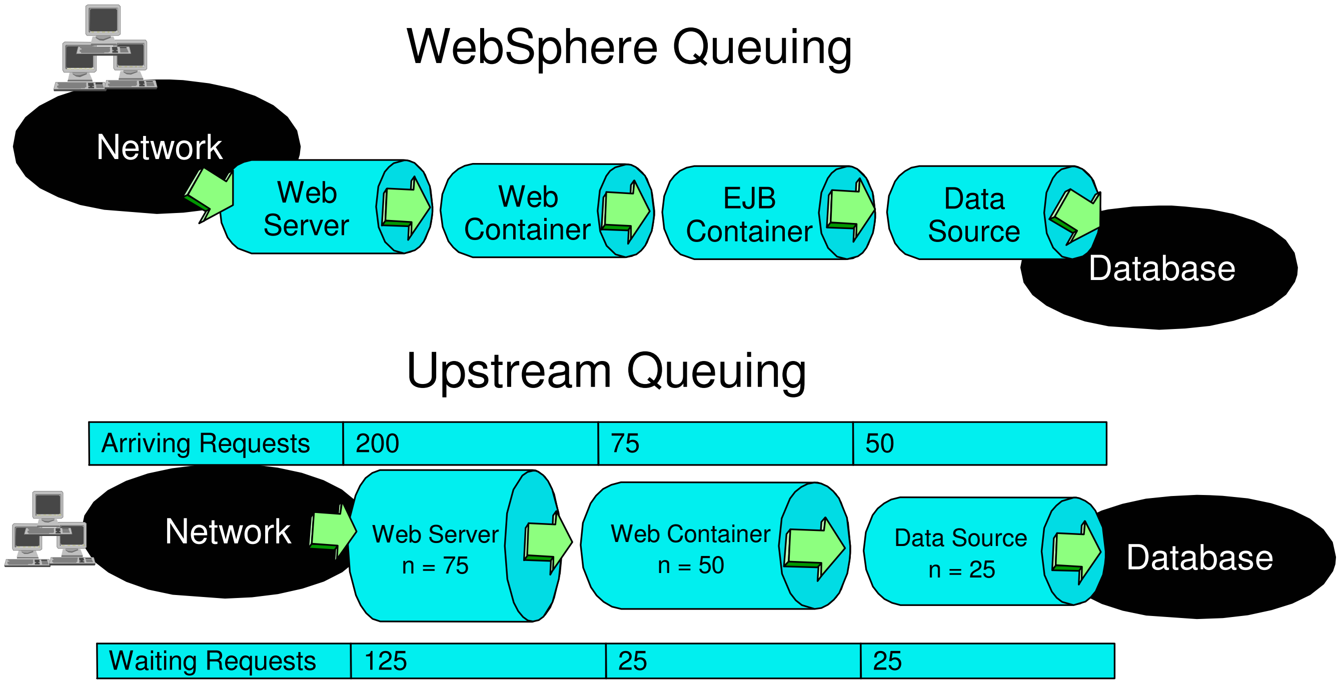
Environment's Black Box

WebSphere Queue Settings



WebSphere Upstream Queuing

- Upstream queuing attempts to allow more work to be done by limiting the number of connections at each tier of the application



Arriving Requests	200	75	50
Waiting Requests	125	25	25

Performance Monitoring Infrastructure (PMI) Data

Servlet Session Manager
- Avg # of active HTTP sessions
- Session size

System Data
- CPU Utilization
- Memory usage

JDBC Connection Pools
- % pool in use
- Avg. wait time
- JDBC times

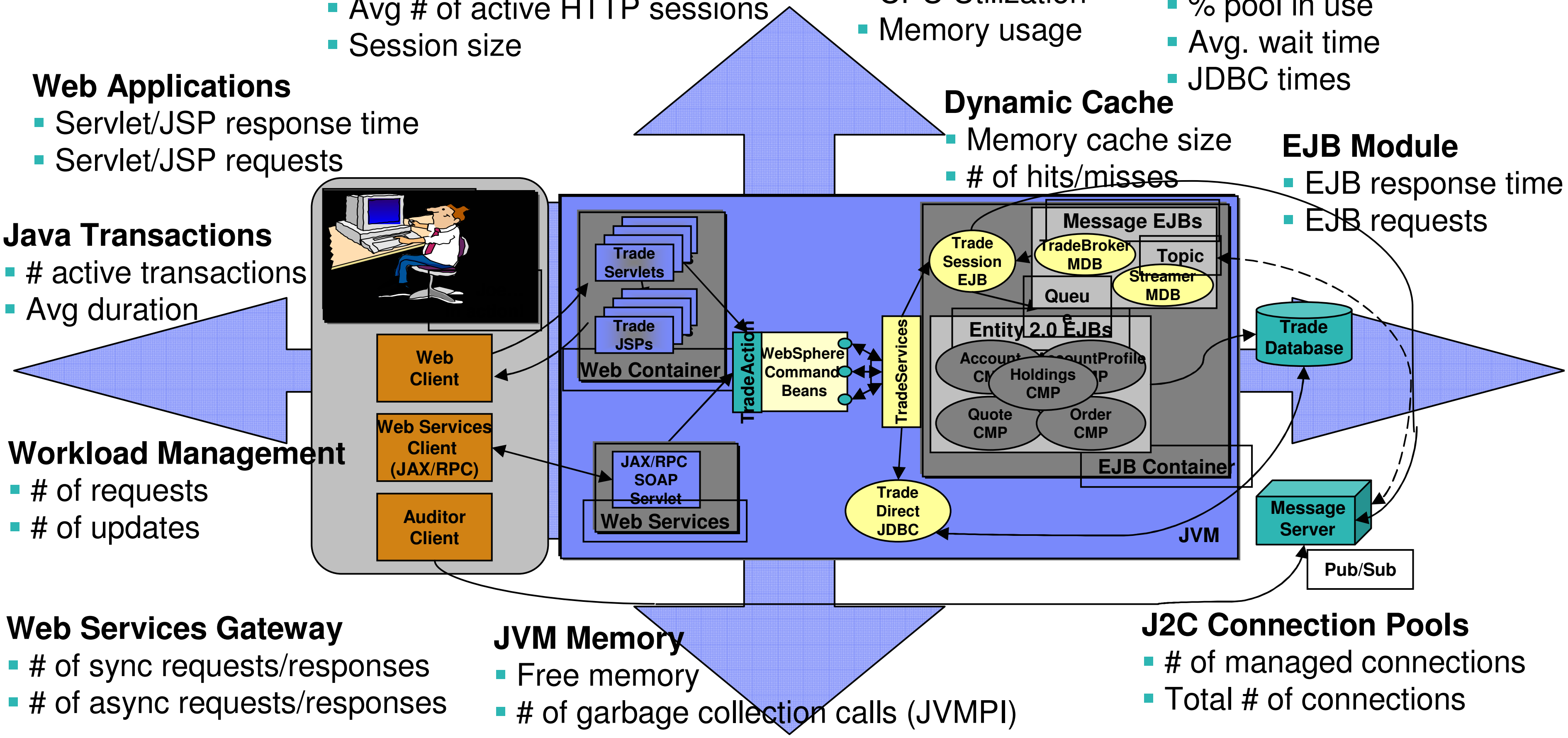
Web Applications
- Servlet/JSP response time
- Servlet/JSP requests

Dynamic Cache
- Memory cache size
- # of hits/misses

EJB Module
- EJB response time
- EJB requests

Java Transactions
- # active transactions
- Avg duration

Workload Management
- # of requests
- # of updates



Web Services Gateway
- # of sync requests/responses
- # of async requests/responses

JVM Memory
- Free memory
- # of garbage collection calls (JVMPI)

J2C Connection Pools
- # of managed connections
- Total # of connections

Web Services
- # of loaded web services
- # of requests delivered/processed

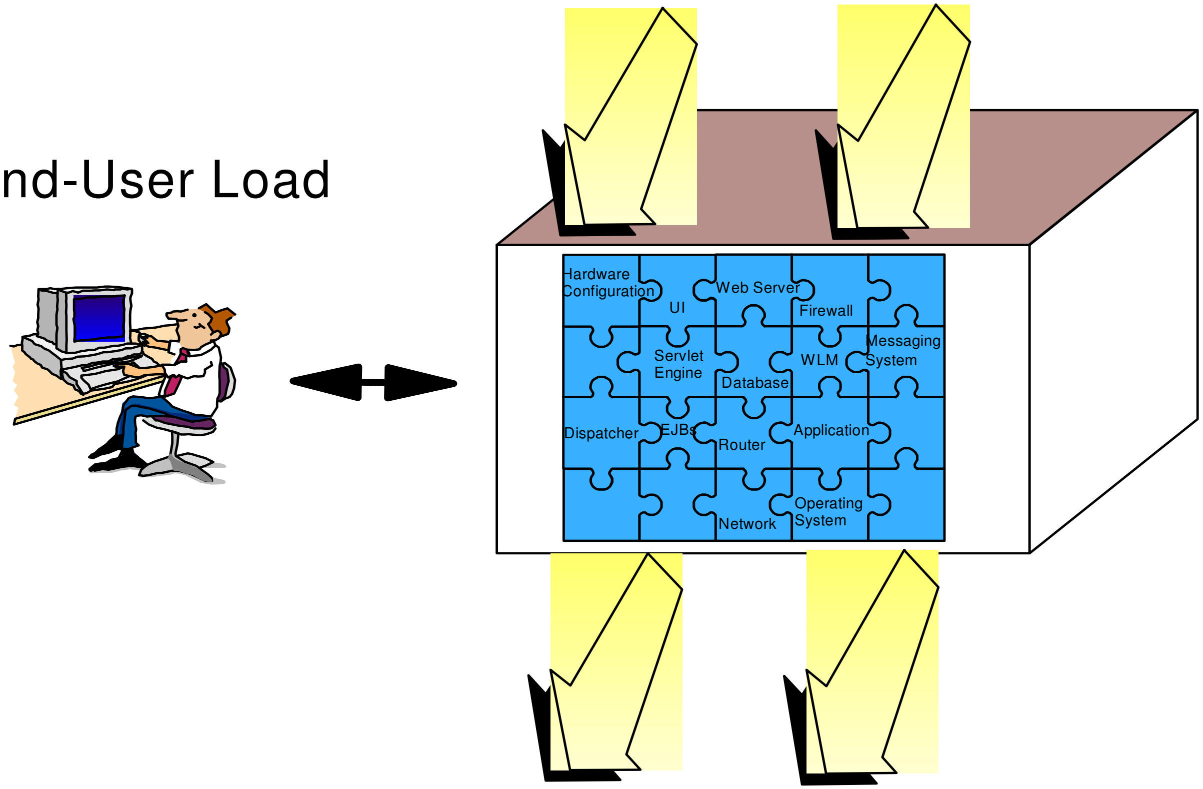
Thread Pools
- Web container and ORB
- Active threads
- % pool maxed

Object Request Broker (ORB)
- Object reference lookup time
- # of requests

Testing Execution: White Box

Monitor for Results

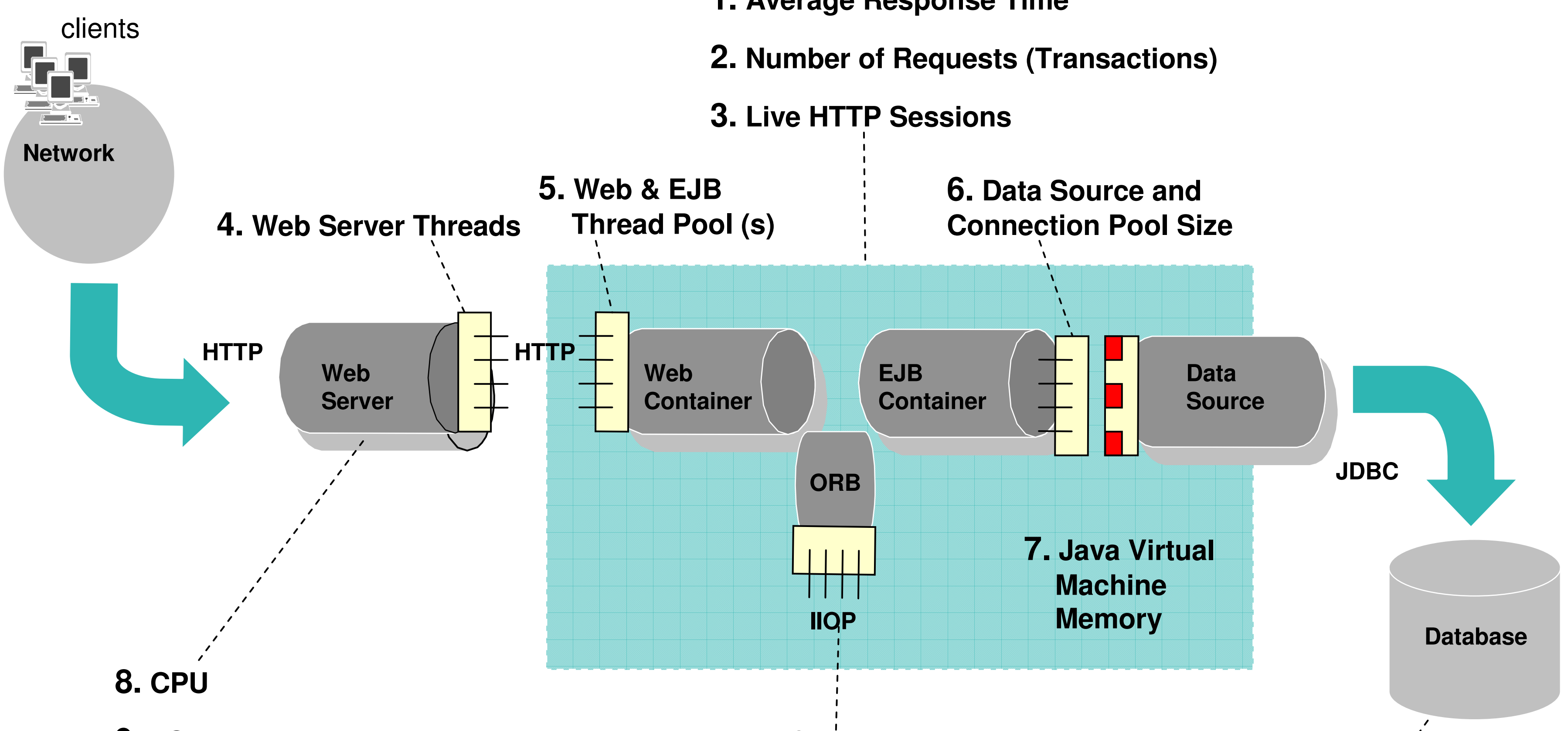
Drive End-User Load



Capture and Correlate Results

What should you monitor?

End-to-End Monitoring -- “Top Ten Metrics”



Servlets and EJBs

1. Average Response Time
2. Number of Requests (Transactions)
3. Live HTTP Sessions

4. Web Server Threads

5. Web & EJB Thread Pool (s)

6. Data Source and Connection Pool Size

7. Java Virtual Machine Memory

clients

Network

HTTP

Web Server

HTTP

Web Container

EJB Container

Data Source

ORB

IIOP

JDBC

Database

8. CPU

9. I/O

10. Paging

8. CPU

9. I/O

10. Paging

8. CPU

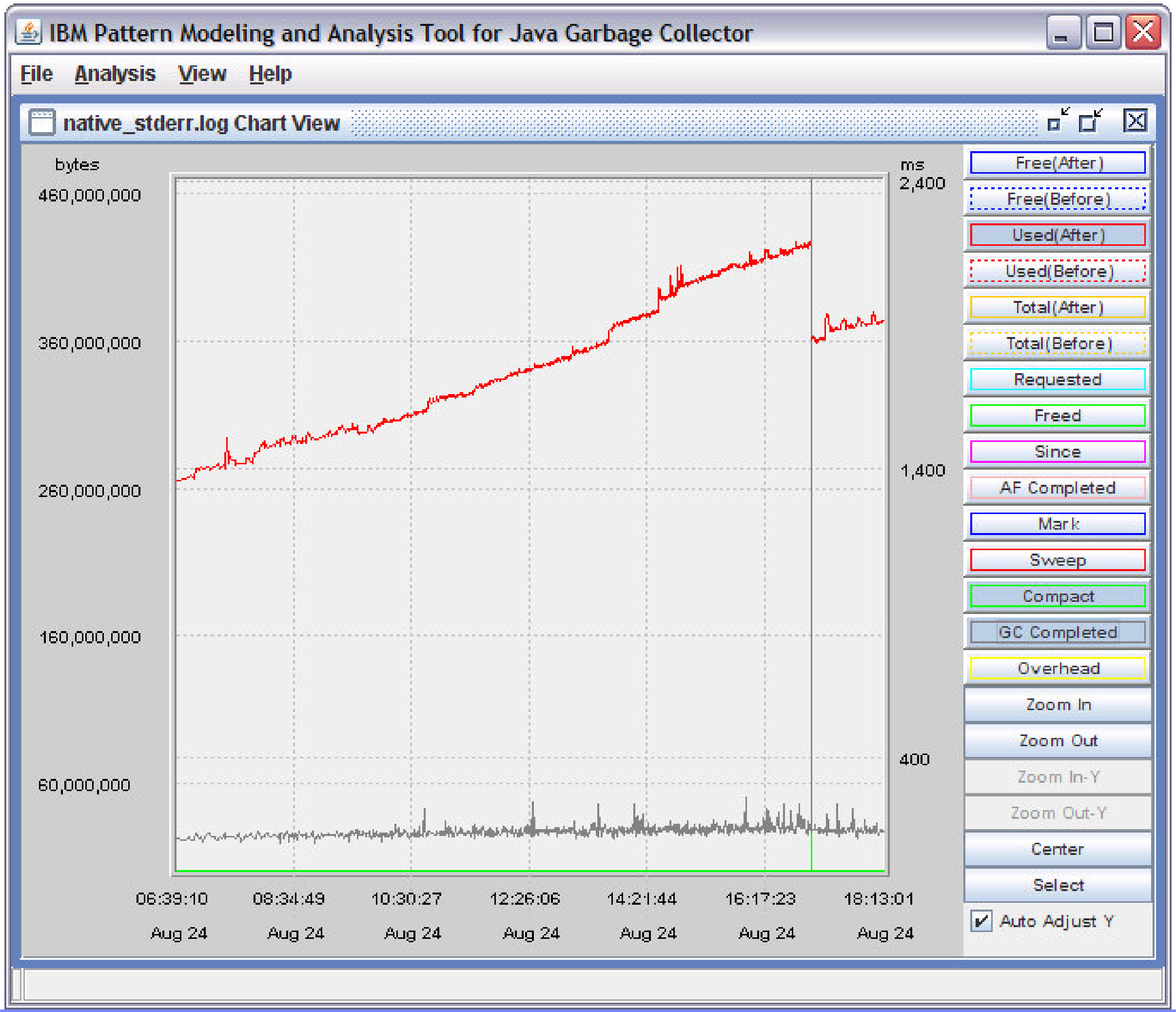
9. I/O

10. Paging

JVM Garbage Collection Analysis

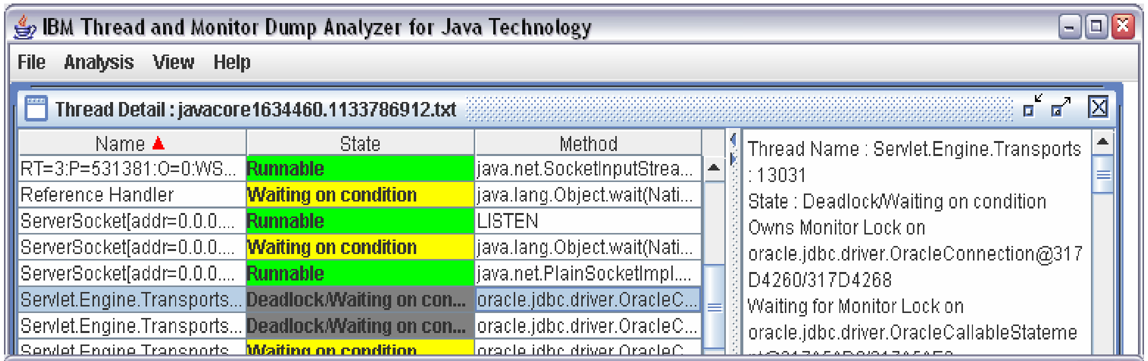
- http://www.alphaworks.ibm.com/tech/pmat

- Monitor characteristics
- Understand patterns
- Identify leaks and spikes
- Leverage all information
  - Frequency
  - Compaction
  - Size
  - Large Objects
  - And more……



Thread and Monitor Dump Analysis

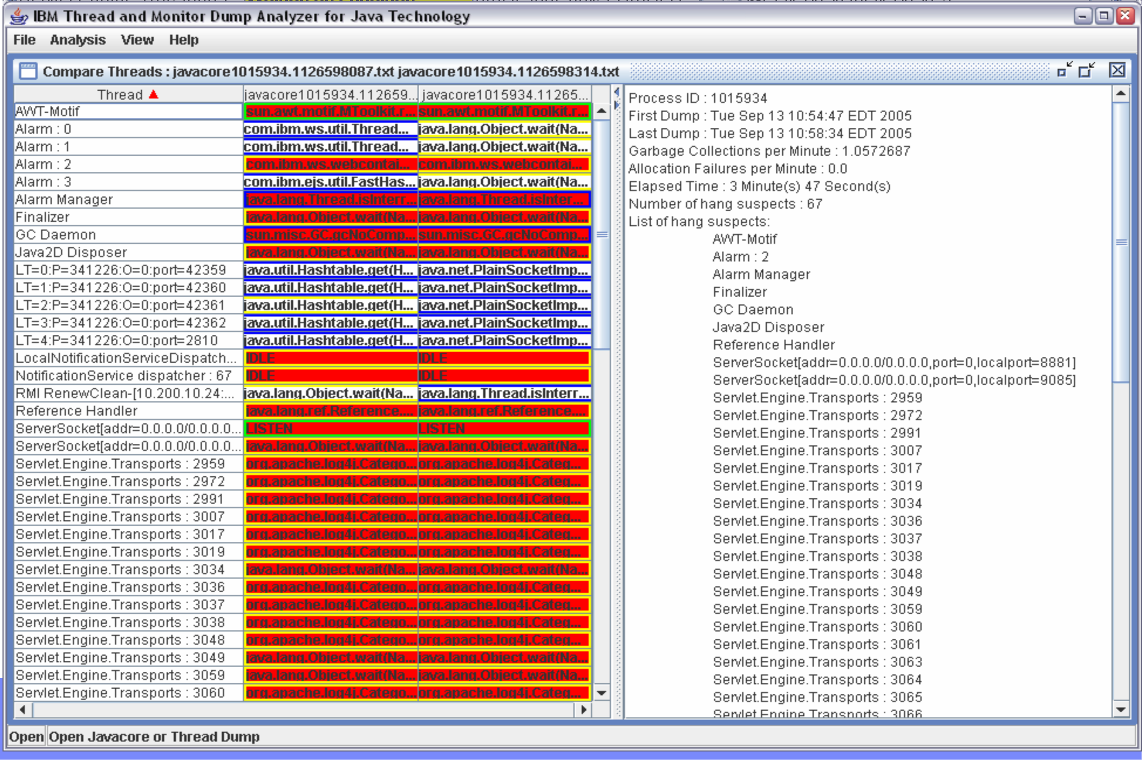
- http://www.alphaworks.ibm.com/tech/jca
- Identify thread contention & non-optimal behavior
- Understand performance issues and bottlenecks
- Compare thread dumps to better diagnose issues



Thread Detail : javacore1634460.1133786912.txt

Name ▲	State	Method
RT=3:P=531381:O=0:WS...	Runnable	java.net.SocketInputStrea...
Reference Handler	Waiting on condition	java.lang.Object.wait(Nati...
ServerSocket[addr=0.0.0....	Runnable	LISTEN
ServerSocket[addr=0.0.0....	Waiting on condition	java.lang.Object.wait(Nati...
ServerSocket[addr=0.0.0....	Runnable	java.net.PlainSocketImpl....
Servlet.Engine.Transports...	Deadlock/Waiting on con...	oracle.jdbc.driver.OracleC...
Servlet.Engine.Transports...	Deadlock/Waiting on con...	oracle.jdbc.driver.OracleC...
Servlet.Engine.Transports	Waiting on condition	oracle.jdbc.driver.OracleC

Thread Name : Servlet.Engine.Transports : 13031
State : Deadlock/Waiting on condition
Owns Monitor Lock on oracle.jdbc.driver.OracleConnection@317D4260/317D4268
Waiting for Monitor Lock on oracle.jdbc.driver.OracleCallableStateme



Compare Threads : javacore1015934.1126598087.txt javacore1015934.1126598314.txt

Thread ▲	javacore1015934.112659...	javacore1015934.11265...
AWT-Motif	sun.awt.motif.MToolkit.r...	sun.awt.motif.MToolkit.r...
Alarm : 0	com.ibm.ws.util.Thread...	java.lang.Object.wait(Na...
Alarm : 1	com.ibm.ws.util.Thread...	java.lang.Object.wait(Na...
Alarm : 2	com.ibm.ws.webcontai...	com.ibm.ws.webcontai...
Alarm : 3	com.ibm.ejs.util.FastHas...	java.lang.Object.wait(Na...
Alarm Manager	java.lang.Thread.isInterr...	java.lang.Thread.isInter...
Finalizer	java.lang.Object.wait(Na...	java.lang.Object.wait(Na...
GC Daemon	sun.misc.GC.gcNoComp...	sun.misc.GC.gcNoComp...
Java2D Disposer	java.lang.Object.wait(Na...	java.lang.Object.wait(Na...
LT=0:P=341226:O=0:port=42359	java.util.Hashtable.get(H...	java.net.PlainSocketImp...
LT=1:P=341226:O=0:port=42360	java.util.Hashtable.get(H...	java.net.PlainSocketImp...
LT=2:P=341226:O=0:port=42361	java.util.Hashtable.get(H...	java.net.PlainSocketImp...
LT=3:P=341226:O=0:port=42362	java.util.Hashtable.get(H...	java.net.PlainSocketImp...
LT=4:P=341226:O=0:port=2810	java.util.Hashtable.get(H...	java.net.PlainSocketImp...
LocalNotificationServiceDispatch...	IDLE	IDLE
NotificationService dispatcher : 67	IDLE	IDLE
RMI RenewClean-[10.200.10.24:...	java.lang.Object.wait(Na...	java.lang.Thread.isInterr...
Reference Handler	java.lang.ref.Reference....	java.lang.ref.Reference....
ServerSocket[addr=0.0.0.0/0.0.0.0...	LISTEN	LISTEN
ServerSocket[addr=0.0.0.0/0.0.0.0...	java.lang.Object.wait(Na...	java.lang.Object.wait(Na...
Servlet.Engine.Transports : 2959	org.apache.log4j.Catego...	org.apache.log4j.Categ...
Servlet.Engine.Transports : 2972	org.apache.log4j.Catego...	org.apache.log4j.Categ...
Servlet.Engine.Transports : 2991	org.apache.log4j.Catego...	org.apache.log4j.Categ...
Servlet.Engine.Transports : 3007	org.apache.log4j.Catego...	org.apache.log4j.Categ...
Servlet.Engine.Transports : 3017	org.apache.log4j.Catego...	org.apache.log4j.Categ...
Servlet.Engine.Transports : 3019	org.apache.log4j.Catego...	org.apache.log4j.Categ...
Servlet.Engine.Transports : 3034	java.lang.Object.wait(Na...	java.lang.Object.wait(Na...
Servlet.Engine.Transports : 3036	org.apache.log4j.Catego...	org.apache.log4j.Categ...
Servlet.Engine.Transports : 3037	org.apache.log4j.Catego...	org.apache.log4j.Categ...
Servlet.Engine.Transports : 3038	org.apache.log4j.Catego...	org.apache.log4j.Categ...
Servlet.Engine.Transports : 3048	org.apache.log4j.Catego...	org.apache.log4j.Categ...
Servlet.Engine.Transports : 3049	java.lang.Object.wait(Na...	java.lang.Object.wait(Na...
Servlet.Engine.Transports : 3059	java.lang.Object.wait(Na...	java.lang.Object.wait(Na...
Servlet.Engine.Transports : 3060	org.apache.log4j.Catego...	org.apache.log4j.Categ...

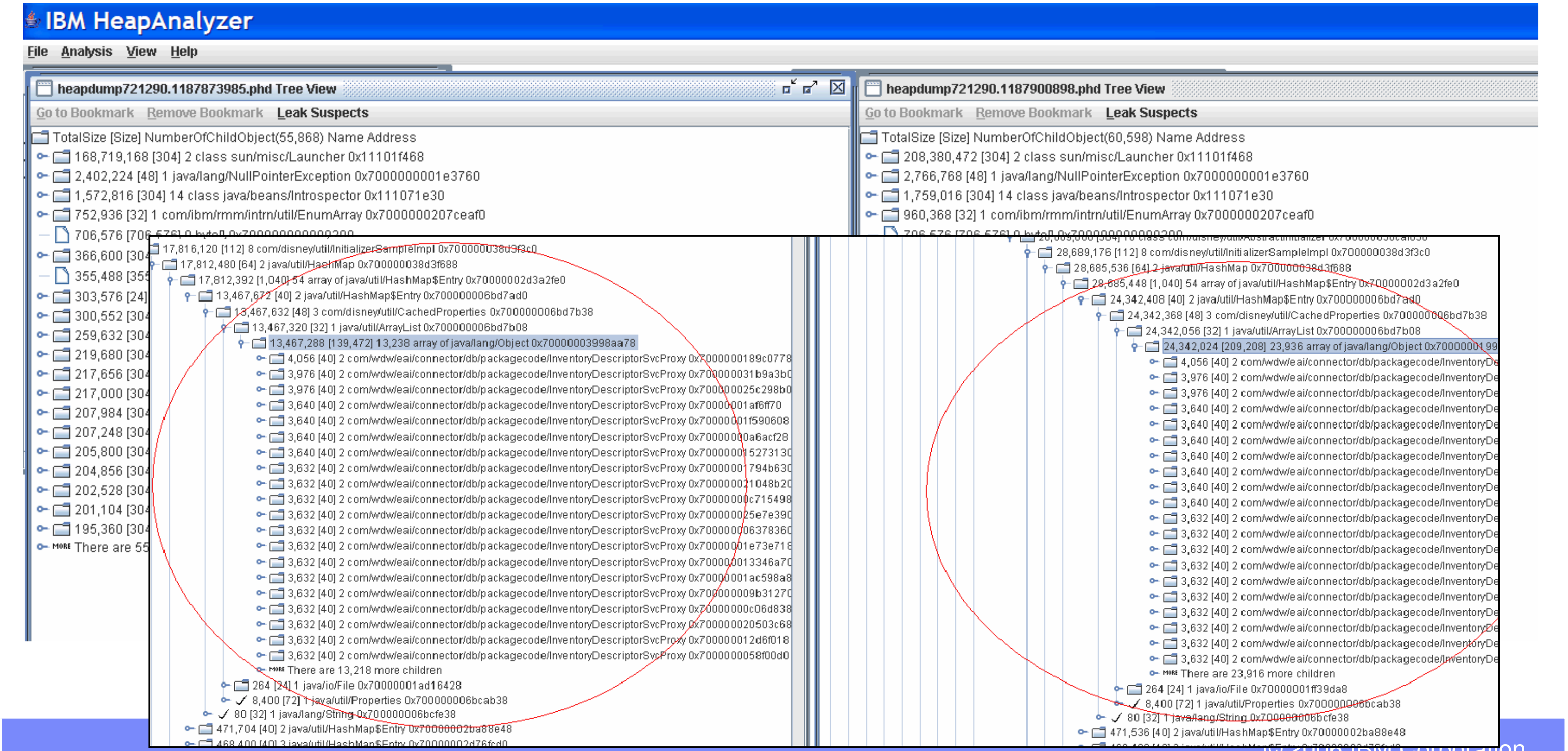
```
Process ID : 1015934
First Dump : Tue Sep 13 10:54:47 EDT 2005
Last Dump : Tue Sep 13 10:58:34 EDT 2005
Garbage Collections per Minute : 1.0572687
Allocation Failures per Minute : 0.0
Elapsed Time : 3 Minute(s) 47 Second(s)
Number of hang suspects : 67
List of hang suspects:
            AWT-Motif
            Alarm : 2
            Alarm Manager
            Finalizer
            GC Daemon
            Java2D Disposer
            Reference Handler
            ServerSocket[addr=0.0.0.0/0.0.0.0,port=0,localport=8881]
            ServerSocket[addr=0.0.0.0/0.0.0.0,port=0,localport=9085]
            Servlet.Engine.Transports : 2959
            Servlet.Engine.Transports : 2972
            Servlet.Engine.Transports : 2991
            Servlet.Engine.Transports : 3007
            Servlet.Engine.Transports : 3017
            Servlet.Engine.Transports : 3019
            Servlet.Engine.Transports : 3034
            Servlet.Engine.Transports : 3036
            Servlet.Engine.Transports : 3037
            Servlet.Engine.Transports : 3038
            Servlet.Engine.Transports : 3048
            Servlet.Engine.Transports : 3049
            Servlet.Engine.Transports : 3059
            Servlet.Engine.Transports : 3060
            Servlet.Engine.Transports : 3061
            Servlet.Engine.Transports : 3063
            Servlet.Engine.Transports : 3064
            Servlet.Engine.Transports : 3065
            Servlet.Engine.Transports : 3066
```

Open Open Javacore or Thread Dump

Heap Analysis

- http://www.alphaworks.ibm.com/tech/heapanalyzer

- Understand and identify leaking objects
- Pinpoint exact issues for application developers to resolve



heapdump721290.1187873985.phd Tree View

```
TotalSize [Size] NumberOfChildObject(55,868) Name Address
  168,719,168 [304] 2 class sun/misc/Launcher 0x11101f468
  2,402,224 [48] 1 java/lang/NullPointerException 0x7000000001e3760
  1,572,816 [304] 14 class java/beans/Introspector 0x111071e30
  752,936 [32] 1 com/ibm/rmm/intrn/util/EnumArray 0x7000000207ceaf0
  706,576 [706,576] 0 byte[] 0x700000000000200
  366,600 [304
  355,488 [355
  303,576 [24]
  300,552 [304
  259,632 [304
  219,680 [304
  217,656 [304
  217,000 [304
  207,984 [304
  207,248 [304
  205,800 [304
  204,856 [304
  202,528 [304
  201,104 [304
  195,360 [304
  There are 55
```

```
17,816,120 [112] 8 com/disney/util/InitializerSampleImpl 0x700000038d3f3c0
  17,812,480 [64] 2 java/util/HashMap 0x700000038d3f688
    17,812,392 [1,040] 54 array of java/util/HashMap$Entry 0x70000002d3a2fe0
      13,467,672 [40] 2 java/util/HashMap$Entry 0x700000006bd7ad0
        13,467,632 [48] 3 com/disney/util/CachedProperties 0x700000006bd7b38
          13,467,320 [32] 1 java/util/ArrayList 0x700000006bd7b08
            13,467,288 [139,472] 13,238 array of java/lang/Object 0x70000003998aa78
              4,056 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDescriptorSvcProxy 0x7000000189c0778
              3,976 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDescriptorSvcProxy 0x700000031b9a3b0
              3,976 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDescriptorSvcProxy 0x700000025c298b0
              3,640 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDescriptorSvcProxy 0x70000001af6ff70
              3,640 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDescriptorSvcProxy 0x70000001f590608
              3,640 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDescriptorSvcProxy 0x70000000a6acf28
              3,640 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDescriptorSvcProxy 0x700000015273130
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDescriptorSvcProxy 0x70000001794b630
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDescriptorSvcProxy 0x700000021048b20
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDescriptorSvcProxy 0x70000000c715498
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDescriptorSvcProxy 0x700000025e7e390
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDescriptorSvcProxy 0x700000006378360
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDescriptorSvcProxy 0x70000001e73e718
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDescriptorSvcProxy 0x700000013346a70
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDescriptorSvcProxy 0x70000001ac598a8
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDescriptorSvcProxy 0x700000009b31270
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDescriptorSvcProxy 0x70000000c06d838
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDescriptorSvcProxy 0x700000020503c68
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDescriptorSvcProxy 0x700000012d6f018
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDescriptorSvcProxy 0x7000000058f00d0
              There are 13,218 more children
          264 [24] 1 java/io/File 0x70000001ad16428
          ✓ 8,400 [72] 1 java/util/Properties 0x700000006bcab38
        ✓ 80 [32] 1 java/lang/String 0x700000006bcfe38
      471,704 [40] 2 java/util/HashMap$Entry 0x70000002ba88e48
      468,400 [40] 3 java/util/HashMap$Entry 0x70000002d76fcd0
```

heapdump721290.1187900898.phd Tree View

```
TotalSize [Size] NumberOfChildObject(60,598) Name Address
  208,380,472 [304] 2 class sun/misc/Launcher 0x11101f468
  2,766,768 [48] 1 java/lang/NullPointerException 0x7000000001e3760
  1,759,016 [304] 14 class java/beans/Introspector 0x111071e30
  960,368 [32] 1 com/ibm/rmm/intrn/util/EnumArray 0x7000000207ceaf0
  706,576 [706,576] 0 byte[] 0x700000000000200
```

```
28,689,176 [112] 8 com/disney/util/InitializerSampleImpl 0x700000038d3f3c0
  28,685,536 [64] 2 java/util/HashMap 0x700000038d3f688
    28,685,448 [1,040] 54 array of java/util/HashMap$Entry 0x70000002d3a2fe0
      24,342,408 [40] 2 java/util/HashMap$Entry 0x700000006bd7ad0
        24,342,368 [48] 3 com/disney/util/CachedProperties 0x700000006bd7b38
          24,342,056 [32] 1 java/util/ArrayList 0x700000006bd7b08
            24,342,024 [209,208] 23,936 array of java/lang/Object 0x70000001 99
              4,056 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDe
              3,976 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDe
              3,976 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDe
              3,640 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDe
              3,640 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDe
              3,640 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDe
              3,640 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDe
              3,640 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDe
              3,640 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDe
              3,640 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDe
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDe
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDe
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDe
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDe
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDe
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDe
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDe
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDe
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDe
              3,632 [40] 2 com/wdw/eai/connector/db/packagecode/InventoryDe
              There are 23,916 more children
          264 [24] 1 java/io/File 0x70000001ff39da8
          ✓ 8,400 [72] 1 java/util/Properties 0x700000006bcab38
        ✓ 80 [32] 1 java/lang/String 0x700000006bcfe38
      471,536 [40] 2 java/util/HashMap$Entry 0x70000002ba88e48
```

Summary

Everything is Important

Test, Test, Test

Questions and Answers