



Warfighter's Mission-Critical System: Automated Testing and Test Management

H. Ferhan Kilical, Ph.D.

Technical Fellow, Electronic Systems

(Test , Test Automation, SOA, Performances Test
and Agile Methodologies)



Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE APR 2010		2. REPORT TYPE		3. DATES COVERED 00-00-2010 to 00-00-2010	
4. TITLE AND SUBTITLE Warfighter's Mission-Critical System: Automated Testing and Test Management				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Northrop Grumman Electronic Systems,1580-A West Nursery Road,Linthicum,MD,21090				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES Presented at the 22nd Systems and Software Technology Conference (SSTC), 26-29 April 2010, Salt Lake City, UT. Sponsored in part by the USAF. U.S. Government or Federal Rights License					
14. ABSTRACT Warfighter's mission-critical system: Agile automated testing and test management Do any of these problems sound familiar? ? Too many failed scripts ? Slipping schedules ? Automation tools that never get off the shelf ? Incomplete test coverage ? Do more with less, and ? Be creative in performing the business Then, come and listen to the story of Warfighter's mission critical application testing. This case study reveals how testers used automated functional, performance, and Service Test scripts for a major mission-critical application that had to meet the most rigorous quality standards. Also, working in an agile environment, the team managed end-to-end requirements and defects and performed functional, SOA and performance testing. With risk based and automated test strategies, the team was able to do automated regression and smoke tests in a fast-paced development ? agile environment and managed test results, test sets, and test-related artifacts. It was a very successful project, completed on time with very limited resources.					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 33	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

Warfighter's mission-critical system: Agile automated testing and test management

Do any of these problems sound familiar?

- Too many failed scripts
- Slipping schedules
- Automation tools that never get off the shelf
- Incomplete test coverage
- Do more with less, and
- Be creative in performing the business

Then, come and listen to the story of Warfighter's mission critical application testing.

This case study reveals how testers used automated functional, performance, and Service Test scripts for a major mission-critical application that had to meet the most rigorous quality standards.

Also, working in an agile environment, the team managed end-to-end requirements and defects and performed functional, SOA and performance testing.

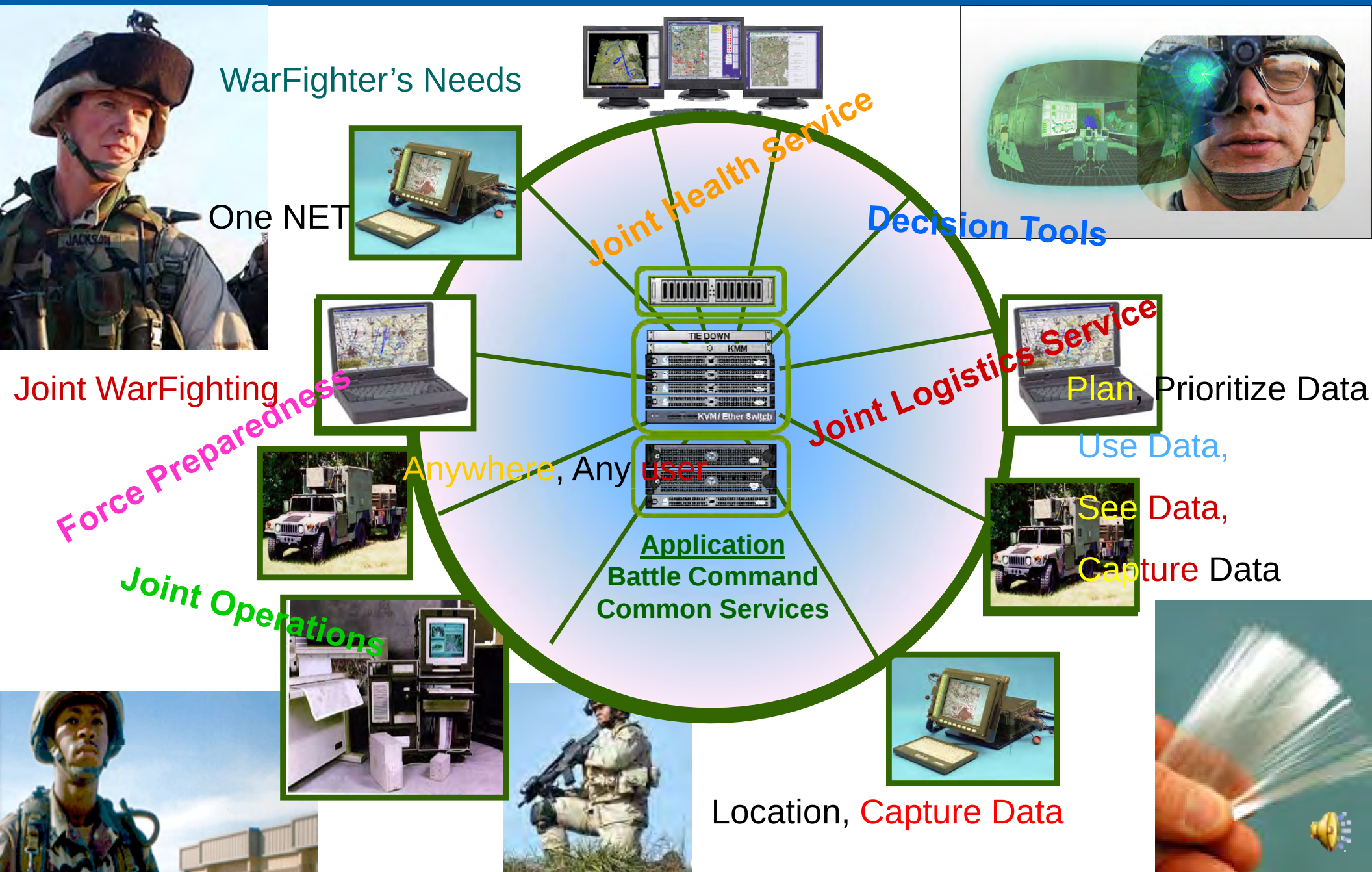
With risk based and automated test strategies, the team was able to do automated regression and smoke tests in a fast-paced development – agile environment and managed test results, test sets, and test-related artifacts. It was a very successful project, completed on time with very limited resources.



Acknowledgements



- Special thanks to the project team at Mission Systems, NG, Herndon.



General Info for Testing & Integration in Government Contracting and Test Automation

- **Keeping Track of:**
 - Requirements
 - Defects
 - Test cases, test processes and test plan
- **Managing Testing Cycles**
 - Problems in infrastructure and scalability
 - Ongoing implementations and significant development initiatives
 - Significant degree of customization and integration
 - Limited resource availability
 - Integration with Complex Portal and Identity Management Solutions
- **Weak Testing Methodology**
 - Manually intensive
 - People - Not process driven
 - No automated testing capabilities
 - Not trusting anyone else to test

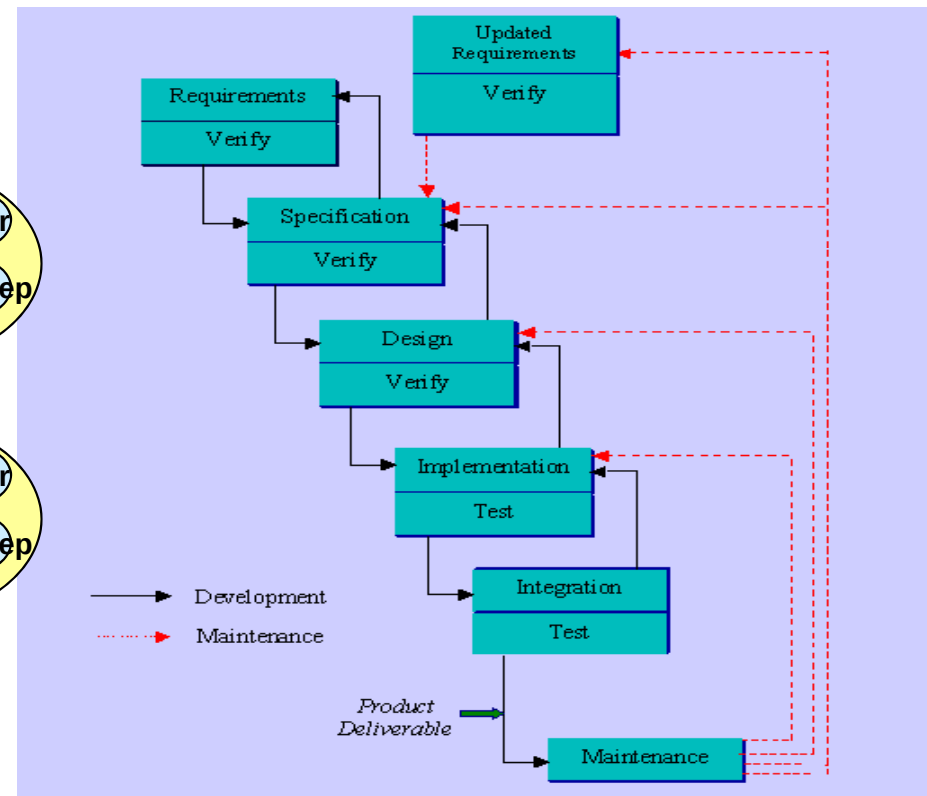


- When a major government contractor delivers software, that software must comply with the most rigorous quality standards
- By enabling both automation and proper test management, we benefit from critical advantages not offered by manual testing
- If in “agile” development model: Development cycles are short and tests are conducted at the same time as the coding
- Things can get particularly complicated when the team is testing SOA applications and performance-scalability testing



Why Testing is Very Crucial in Agile & Waterfall Development?

Waterfall Model



Agile Model

Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5
Analysis	Analysis	Analysis	Analysis	Analysis
Coding	Coding	Coding	Coding	Coding
Testing	Testing	Testing	Testing	Testing

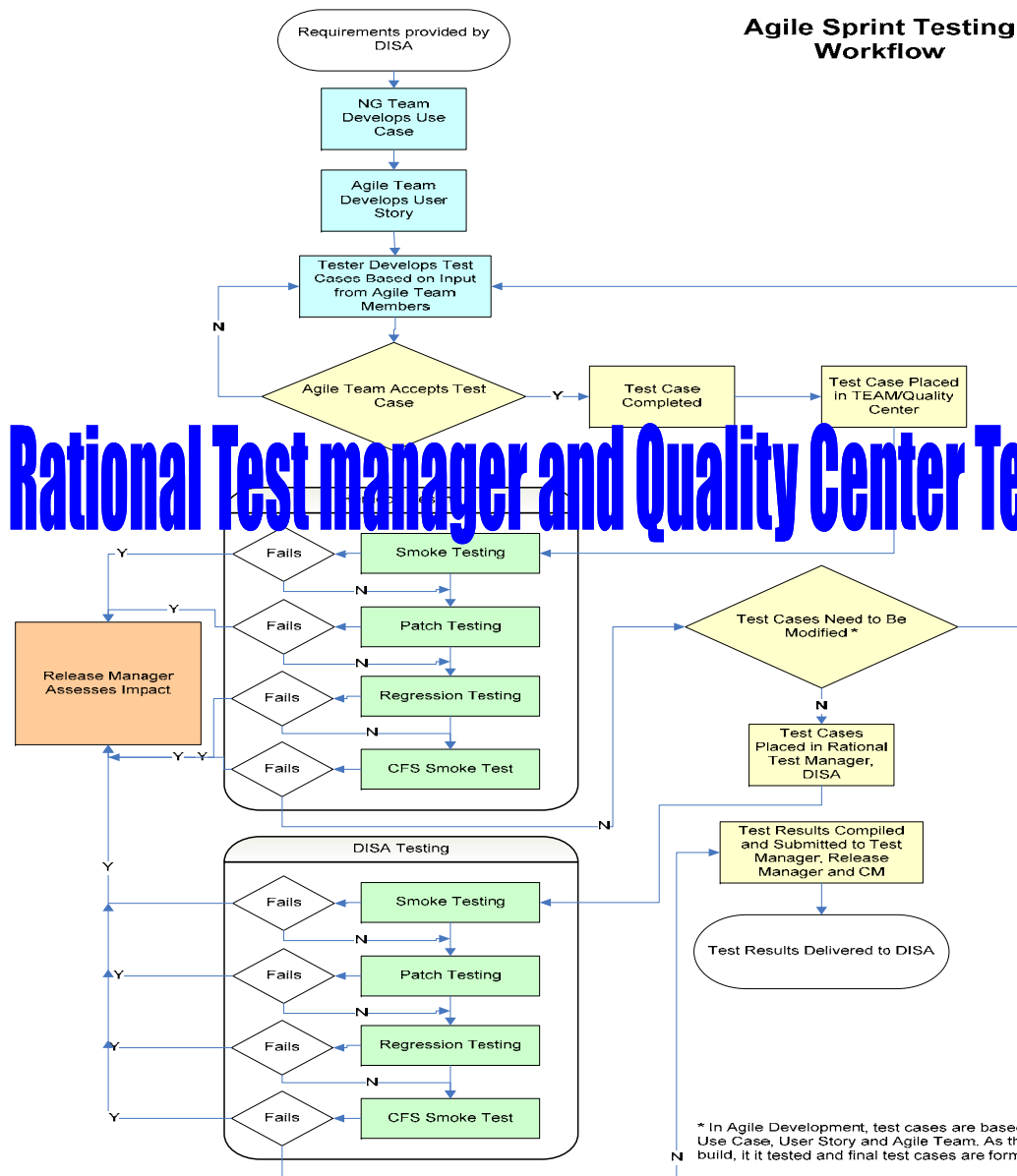
We Did Not Have Processes and Tools in Place

- The client had **Rational tools (Req Pro, ClearQuest and Test Manager)** but they were not properly used.
- And a lot of **Excel sheets and Word docs**. Never ending story of not being able to control requirements and defects ...
- We bought **Quality Center**, and we made Rational tools and Quality Center tools talk to one another. We faced challenges with Firewalls and security.
- Meanwhile, we started working on the **processes** with the client and our NG internal processes for test.
- We started putting together IMS, built partnership with the client .



Agile Workflow and Tools

Agile Sprint Testing Workflow



We had to make Rational Test manager and Quality Center Test Plan talk to one another.



Automating Test Scripts

- Functional Testing
- Performance, Load, Stress Testing
- Service Test
- Security and SA Type of Testing
- Important Features in Relation to Automation:
 - Test Case, Requirement correlation
 - Defects, Defect Management and its correlation to requirements and test cases if possible
 - Test Scheduling IMS and test schedules and customer testing



Purpose of Automated Testing

- Checks virtually any functionality in application.
- Provides consistently re-usable tests to be run each time a site or application changes
- Shortens testing time especially regression testing.
- Tracks all test runs, logs, test results in one, shared repository.

Major benefits are:

- Reusability
- Consistency
- Productivity
- Team work environment



- Tests that need to be run for every build, sometimes referred to as Sanity tests (Smoke and Regression tests).
- Tests that use multiple data values for the same actions are known as Data-Driven tests (Equals to, $=>$, $<=$).
- Identical tests that need to be executed using different browsers (We are using IE6,7 and FF).
- Mission critical pages (Certain pages need to be checked all the time).



- **Sprints Testing:** For every sprint, test team will have a baseline. The baseline consists of tests created as a result of sprint requirements that will be fulfilled.
- **Smoke Testing:** For every sprint, we review the new test cases and adjust standard smoke tests to reflect any needed changes.
- **Regression Testing:** For every sprint, we review the new test cases, and based on the requirements and development, we complete a regression test. That full regression test includes all the sprint baseline regression and smoke tests. The regression test is fully automated with *QTP*.
- **Load Testing:** During sprints at Herndon, based on the needs, we develop LoadRunner scripts for performance and tuning. At the end of each iteration, our goal is to have a set of LoadRunner scripts that will allow us to see the performance, load and scalability for major business rules and transactions or identify bottlenecks...
- **Service Testing:** During sprints at Herndon, based on the needs, we develop services tests scripts. Our goal is to run these scripts under load as well as security.



- # Believe it or not testing takes a lot of time and resources!

- **Test Plan**
 - Living document updated throughout iteration
 - User stories augment the test plan
 - Delivered at the end of each iteration
- **Test Cases**
 - Written throughout the Agile process
 - Input to Rational prior to end of each sprint
 - Automate QTP, LR and ST
- **Test Results**
 - At Herndon with Agile Teams
 - Delivered at the end of each sprint
- **System Test Report**
 - Living document updated throughout iteration
 - Updated at the end of each sprint
 - Delivered at the end of each iteration



Script Development

- Developed global scripts that can be called from one script to the other
- Scripts were grouped into test sets for different purposes. Such as quick regression test sets, quick check for critical areas or known issues
- With one script we were able to test the system with different browsers at the same time. E.g., IE6, IE8, FF3, etc.
- The same scripts were used for executing tests at different suits. So with one script we were able to run several tests depending on the situation we were in for that particular day.



Calling Scripts from Other Test Sets

Calling scripts from other test sets, excluding log in and log out, preparing test sets and making sure users have the right privileges to perform certain business rules

The screenshot displays the HP Virtual User Generator (VuGen) interface. The main window shows a script titled "NG_701_AV_DODAAclp_Rprt_Mar5_09 - Web (HTTP/HTML)". The script contains several actions, including "vuser_init", "DODAAclp", "vuser_end", and "globals.h". A "Parameter Properties" dialog box is open, showing the "PrimaryCSA" parameter type. The dialog includes a table with columns for "Username", "Password", "URL", and "PrimaryCSA". The table contains several rows of test data. The "File path" field is set to "D:\PerfTest701\Scripts\Global_Data\Username.dat". The "Select column" dropdown is set to "By number", and the "Select next row" dropdown is set to "Same line as Username". The "File format" is set to "Comma", and the "Column delimiter" is set to "Comma". The "First data line" is set to "1". The "Replay Log" window at the bottom shows the execution of the script, including the "vuser_init" action and the "DODAAclp" action. The log shows the script starting at 3:22 PM and the "DODAAclp" action completing successfully at 3:23 PM.

Username	Password	URL	PrimaryCSA
1	tester100	GLOBAL	GLOBAL
2	tester01	GLOBAL	NORTHCOM
3	tester02	GLOBAL	GLOBAL
4	tester03	GLOBAL	GLOBAL
5	tester04	GLOBAL	GLOBAL
6	tester05	GLOBAL	GLOBAL
7	tester06	GLOBAL	GLOBAL

Quality Center, Schedule QTP Scripts, Defects

The screenshot displays the Mercury Quality Center 9.0 web interface within a Microsoft Internet Explorer browser. The interface is divided into several sections:

- Top Navigation Bar:** Includes links for BACK, NEXT, TOOLS, HELP, and LOGOUT. The domain is GCSSDOMAIN, Project: GCSSTest, User: administrator.
- Left Sidebar:** Contains a tree view for Test Sets, Requirements, and Test Plan. The Test Plan section is expanded, showing a list of test scripts.
- Main Content Area:** Displays the 'AirfieldFacilityDetails' test set. Below this, there is a table of defects.
- Defect Details Panel:** A pop-up window showing details for Defect 63. It includes fields for Build Reference, Detected By, Status, Subject, Assigned To, and various dates and times.
- Summary Table:** A table at the bottom of the defect list showing the status and assignee for each defect.

Defect ID	Status	Assignee
62	Closed	mgautam
63	Closed	mgautam
64	Closed	mgautam
65	Closed	mgautam
66	Closed	pkeith
67	Closed	inguyen
68	Closed	mgautam
69	Closed	randerso
70	Closed	randerso
71	Closed	randerso
72	Closed	randerso
73	Closed	randerso
74	Closed	randerso
75	Closed	mgautam
76	Closed	randerso
77	Closed	randerso
78	Closed	randerso
79	Assigned	dobbie
80	Closed	inguyen
81	Closed	mgautam
84	Assigned	dobbie

Defect 63 Details:

- Defect: 63
- WB - Error when click on delete Aggregate.
- Build Reference: v6.1 Build 3
- Detected By: pkeith
- Status: Closed
- Subject: WB
- Assigned To: mgautam
- Associated Requirement: [Empty]
- Capability: [Empty]
- Fix Due Date: [Empty]
- NG_Rational_ID: [Empty]
- Reproducible: Y
- Use Case: [Empty]
- Planned Closing Version: [Empty]
- Estimated Fix Time: [Empty] (Days)
- Actual Closing Date: 8/18/2007

Summary: WB - Error when click on delete A...

Description: Select "Aggregate Watchboard Library." Clicked on "Delete Aggregate Watchboard" link under Manage Contents on the left, the error shows like this: Error in template wb_showDeleteItem.shtml: Error: The method "getContexts()" was not found.. Line: " <SCRIPT>if(getSize(getContexts())!=1): wb_showDeleteItem.shtml:68.

Comments: What is the error relieved? Jake Jacques <jacques>, 6/29/2007: Now that Build 4 is available for testing, Defect assigned to original tester for verification of fix. Dwayne Keith <pkeith>, 8/16/2007: Created the aggregate WB and as per TC deleted the 8 aggregates and fix in page.

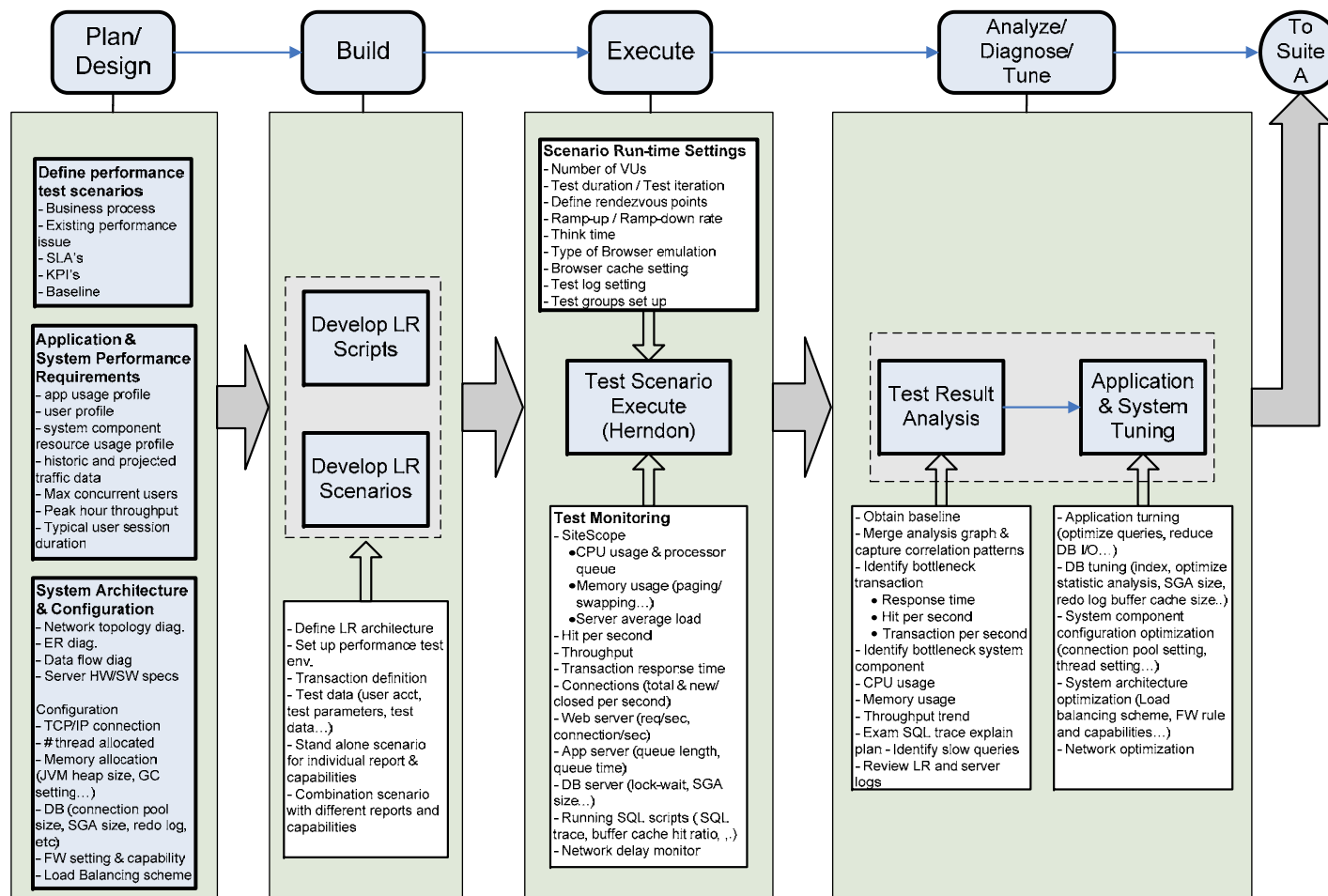
Questions a Performance Test Should Answer:

- Does the application respond quickly enough for the intended users?
- Will the application handle the expected user load and beyond?
- Will the application handle the number of transactions required by the business?
- Is the application stable under expected and unexpected user loads?
- Are you sure that users will have a positive experience on go-live day?



Performance Test Processes at NG Herndon

Performance & Load Test Process in the Application and at Northrop - Herndon



[illegible]

LoadRunner Controller Monitors and SiteScope

- SiteScope Monitors:

The screenshot displays the HP Virtual User Generator (VuGen) interface. The main window shows a script with the following code:

```

lect_EngineeringTab");
anager/gcss/{PrimaryCSA}?_nfpb=true&_pageLabel=engineering_1",
pmanager

```

A "Parameter Properties - [PrimaryCSA]" dialog box is open, showing a table of parameters:

	Username	Password	CertUser	PrimaryCSA
1	tester100	QWvert_008 tester_100	GLOBAL	
2	tester01	QWvert_004 tester_01	NORTHCOM	
3	tester02	QWvert_005 TestUser02	GLOBAL	
4	tester03	QWvert_005 TestUser03	GLOBAL	
5	tester04	QWvert_008 TestUser04	DECCD	
6	tester05	QWvert_008 TestUser05	DECCD	
7	tester06	QWvert_008 TestUser06	DECCD	

The dialog also shows options for "Select column" (By number: 4), "File format" (Comma), and "Update value on" (Each iteration). The "Replay Log" at the bottom shows the following messages:

```

Virtual User Script started
Starting action vuser_init
Web Turbo Replay of LoadRunner 9.10.4
Run Mode: HTML [MsgId: MMSG-2600]
Run-Time Settings file: D:\PerfTest
vuser_init.c(33): Notify: Transaction
vuser_init.c(40): web_add_auto_header("Accept-Language") was successful [MsgId: MMSG-26002]
vuser_init.c(46): Notify: Parameter Substitution: parameter "HOST" = "158.114.68.208"
vuser_init.c(46): Warning: Client Certificate was requested by SSL server [MsgId: MMSG-26000]
vuser_init.c(46): t=355ms: 408-byte response headers for "https://158.114.68.208/gcssportal" (RelFrameId=1, Internal ID=1)
vuser_init.c(46): HTTP/1.1 200 OK\r\n
vuser_init.c(46): X-Backside-Transport: OK OK\r\n
vuser_init.c(46): Cache-Control: no-cache="set-cookie"\r\n
vuser_init.c(46): Date: Wed, 18 Mar 2009 22:14:38 GMT\r\n
vuser_init.c(46): Content-Type: text/html\r\n
vuser_init.c(46): ConsentRequired: true\r\n
vuser_init.c(46): Set-Cookie: def_cius_JSESSIONID=Rb5vJByTVEMhpf19yBnBZrtChpGLnqMLHrhgy6GfKsFVvc25cmH!11914
vuser_init.c(46): 24462; path=/\r\n

```



Challenges of Testing for Agile and SOA

- **A more versatile test-bed environments**
 - It may be difficult to model the whole set of end-to-end software that probably span many different servers
 - Ability to simulate unavailable components
- **Transition for testers – process-centric testing team**
 - Broad knowledge of business processes
 - Understanding the intricacies of domino effects on business transactions
 - Cross-functional teams environment
 - Understand and diagnose underlying technology and connectivity
- **Location and identification of web services (Geographic locations...)**
- **Availability of web services components: Applications, Middleware, Supporting hardware, teams – development, system admin, network, etc.**
- **Locating and isolating defects are difficult:**
 - **Defects in service components would cause domino effects to applications that utilize those services**
 - **Capture and analysis of all SOAP messages that are passed from one component to another is overwhelming**
 - **Service components do not have GUI**



Testing Aspects and Service test

Positive Testing - Generates a full positive test for the selected services. It tests each operation of the selected service.

Standard Compliance – Tests the service's compliance with industry standards such as WS-I and SOAP.

Service Interoperability – Tests the service's interoperability with all supported Web Services.

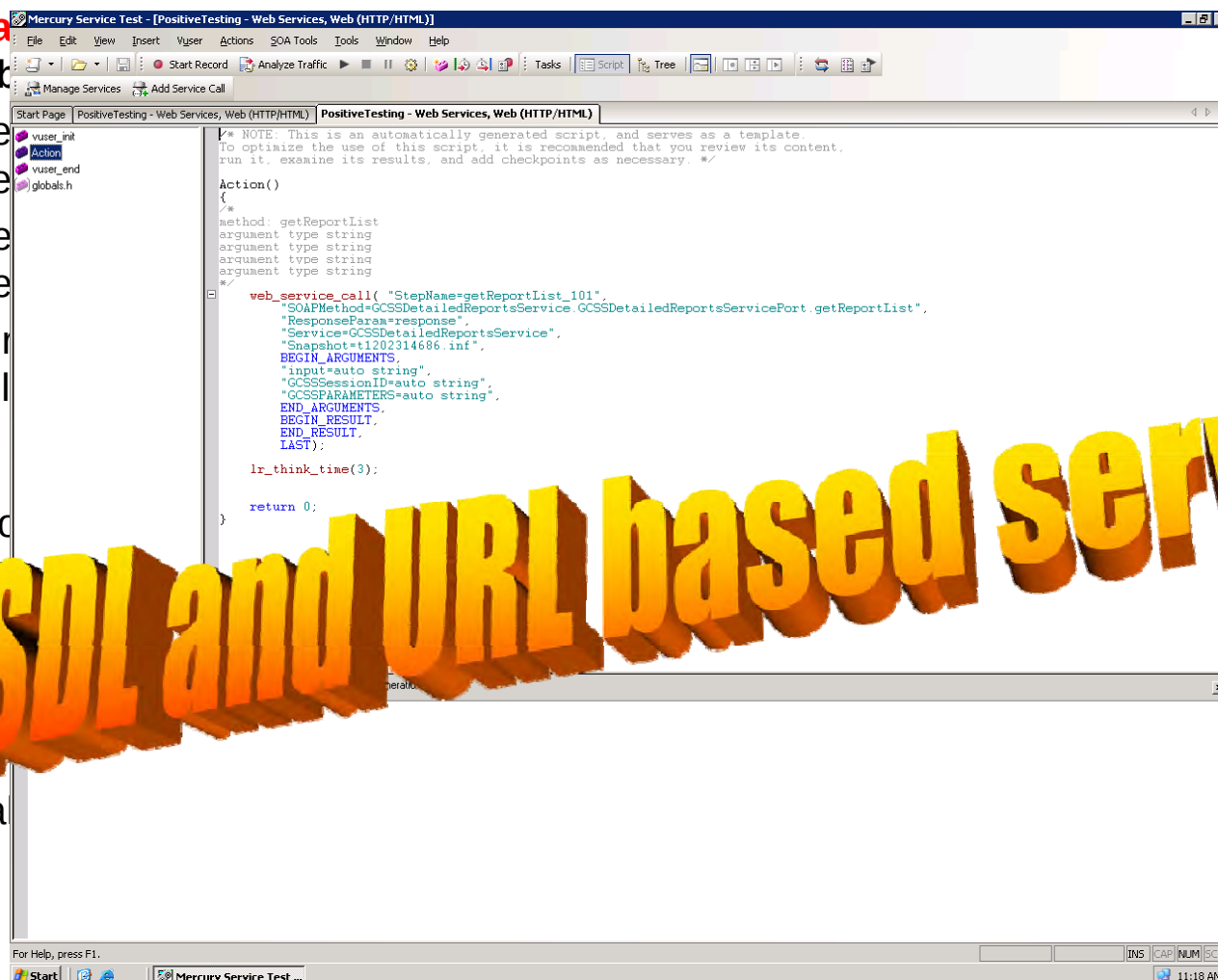
- .NET Framework
- .NET Framework
- Axis 1.3 Web Services
- Axis 1.3 Web Services
- Generic Mercury Sol
- Mercury Sol

Security Testing –

- SQL Injection
- unauthorized
- Cross-site
- distributed

Boundary Testing

- Extreme Values
- Null Values



WSDL and URL based service

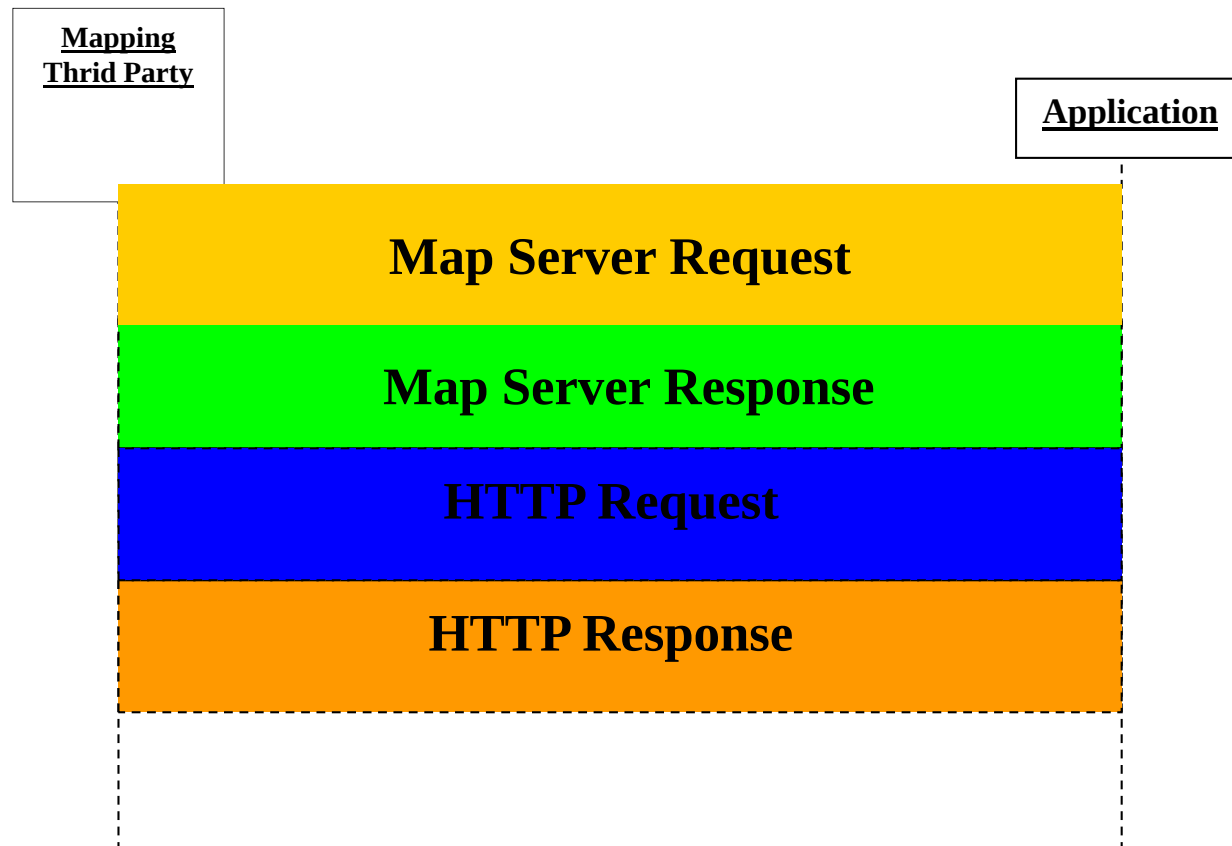
with all
ations using
ations using
sing Generic
es it an
site that will
nique.



Map Service Interface Description

The application has three Mapping Interfaces:

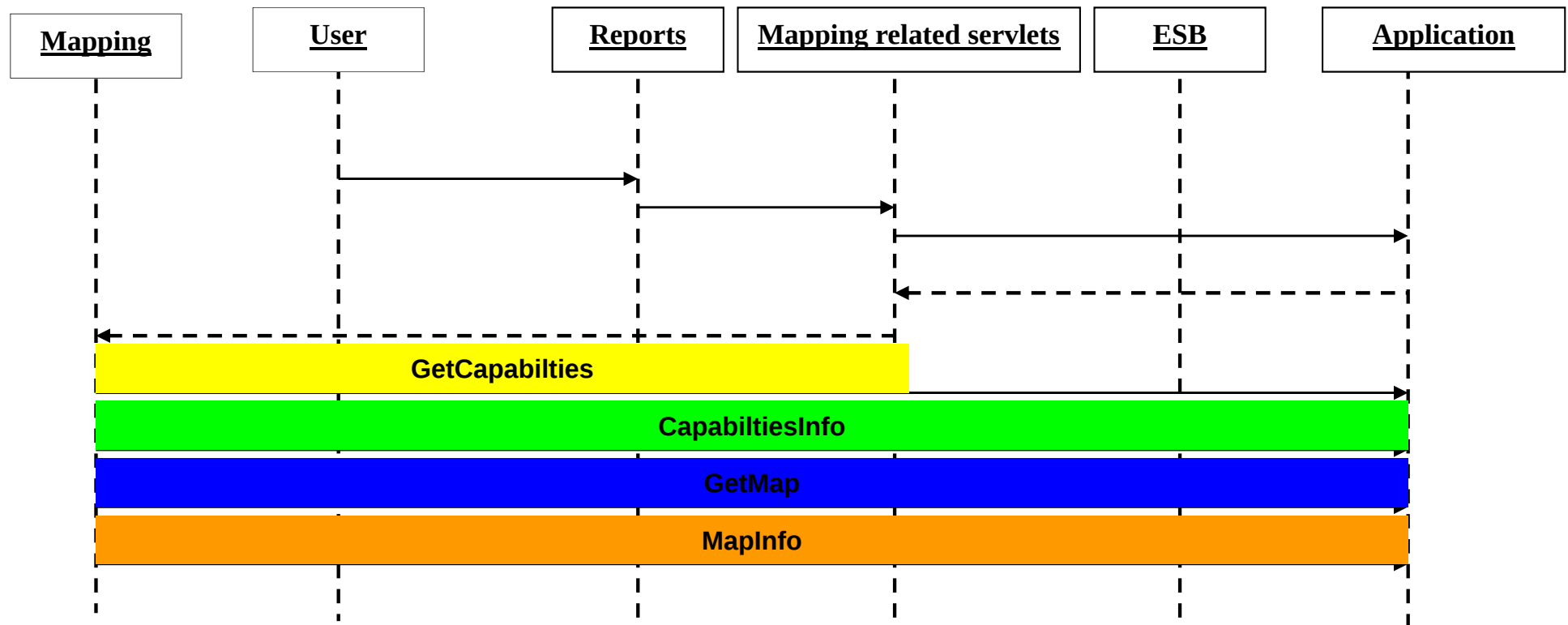
- Map Server
- Reporting Detail Request from Palanterra
- Reporting Mapping Call

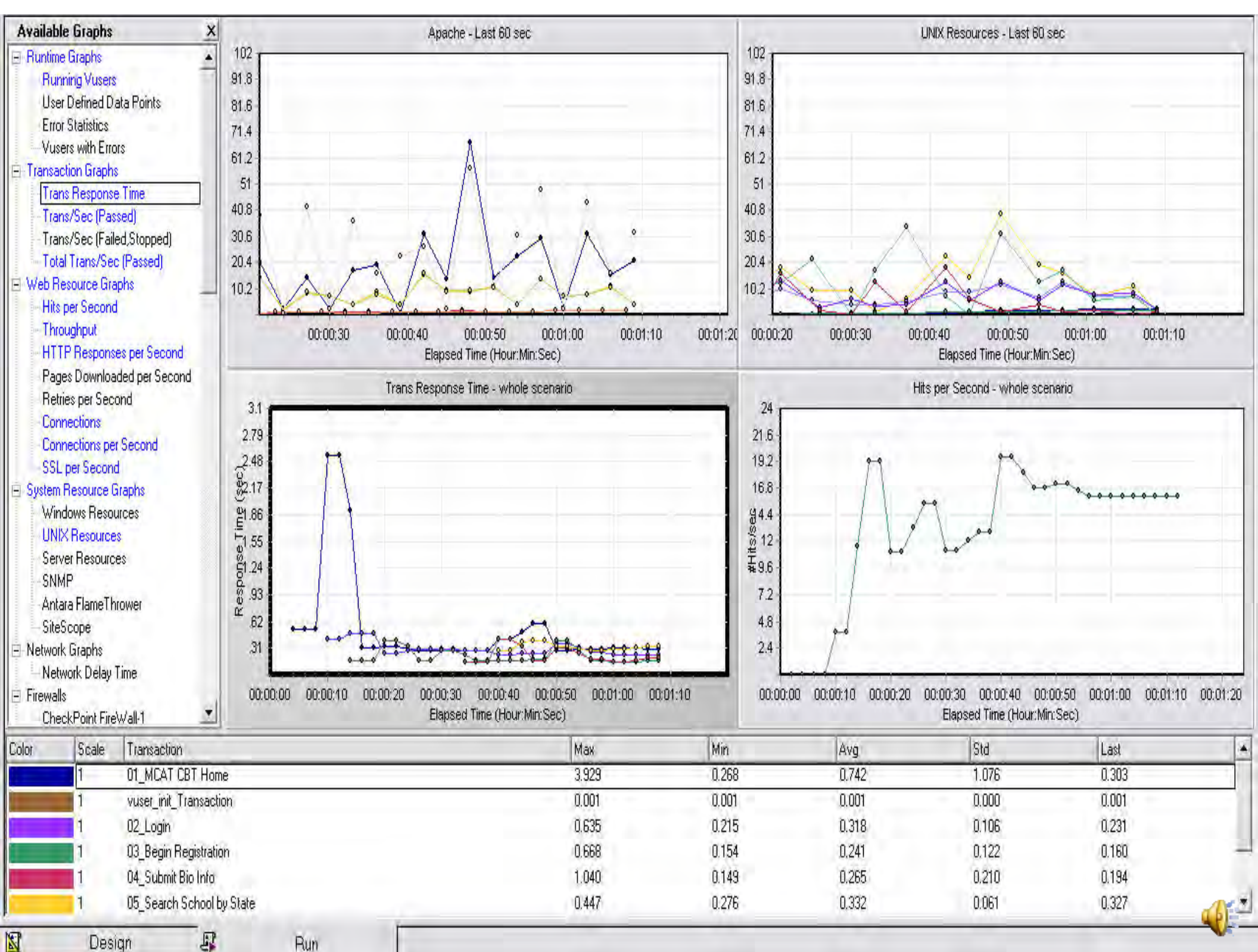


Reporting Mapping Call

Report Mapping Call is the single internal Mapping interface in the application.

When a user clicks the “Map” button within the results page of a report, a call is made to the Mapping application.





With Service Test We are Able to:

- **Develop scripts:**
 - without a GUI
 - using multiple protocols. In enterprise world we have to deal with a lot of multiple protocols. This feature is very helpful.
 - by WSDL, UDDI, File and URL. This is a very helpful feature, too.
- **These scripts can be executed in LoadRunner for performance**
- **We can analyze traffic over the network**
- **We can set Security Policies that includes tokens, SAML, and so on**



Practice for Successful SOA Testing Strategy

Start early in the life cycle:

- Testing client applications – Start the end-to-end testing and tuning 6 months before the deployment.

Create an automated and repeatable testing process.

Create an assembly-oriented test plan

High number of permutations and combination of paths!

- Test the application from its initial state and could be testing into stages with incremental increase the number of components.
- Choose an appropriate set of test cases to support end-to-end testing of the business process and end-user experience.



With the use of the tools we were able to:

- Prioritize testing priorities based on business risk
- Access testing assets anytime, anywhere via a browser interface
- Create an end-to-end quality management infrastructure
- Manage manual and automated tests.
- Accelerate testing cycles by scheduling and running tests automatically, unattended, 24x7
- Manage multiple versions of requirements, tests, test scripts and business components
- Enforce standardized processes and best practices
- Analyze application readiness at any point in the testing process with integrated graphs and reports



With the use of the tools we were able to:

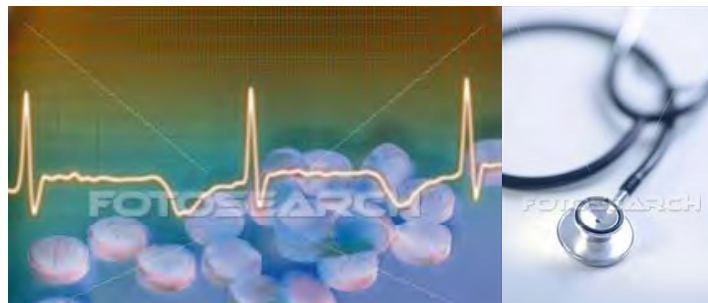
- 50 to 70% decrease in actual testing time (efficient and faster)
- Able to cope with huge amount of testing and captured defects at early stages of development
- Able to produce contractual documents such as RTM – Requirement Traceability Matrix, defect reports, test reports, test plans and the like in a timely manner.
- Able to produce metrics for defects such as defect assignment density, defect aging and other metrics related to defects
- Were able to capture change orders from third parties and teams and sub contractors in Service Test
- Able to capture data from test equipment
- Able to capture test results
- Had a chance to uncover and eliminate performance bottlenecks in architecture and overall performance of the system
- Most important, we did not have to have the same set up that we have at Herndon. Instead of sending testers to classified lab we are planning to send software to the lab. This is the plan...

Test Test Test



Purpose: integrate and test all system components prior to official delivery to the government.

- ✓ All testing related documentation has been completed and is up to date
- ✓ Successful completion of smoke, patch, and regression tests
- ✓ Performance-Load-Stress test baselines obtained
- ✓ SLAs are met
- ✓ All the test results were delivered into government CM
- ✓ All defects are documented in CM tool
- ✓ Final system test report submitted to the PMO
- ✓ Installation and build guide with all the updates completed and delivered



Results from test indicated that the software is ready for Acceptance test

Requirements Validation

Results we would like to see:

- Release Requirements testing
 - Completed 99.13% testing of all testable requirements
 - ✓ Executed 99.63% testing against IE 6.0
 - ✓ Executed 98.50% testing against IE 7.0
 - ✓ Remaining 0.93% of testing could not be functionally tested
- Regression testing
 - Completed 100% of planned regression testing
 - ✓ Executed 95.05% testing against IE 6.0
 - ✓ Executed 92.95% testing against IE 7.0

