

Identifying a generic N-Tier architecture framework for web-based applications

Y. PRASANTH, V. GOWTHAMI UTTAM, G. ANUSHA and G.SINDHU PRIYA

Koneru Lakshmaiah College of Engineering, Green Fields,
Vaddeswaram P.O., Guntur - 522 502 (India).

(Received: November 02, 2009; Accepted: December 19, 2009)

ABSTRACT

N-Tier architecture is becoming increasingly important because of its scalability, availability and flexibility. There are different frameworks for testing tools which perform performance, load, stress, navigation, regression testing etc. The better framework should be in such a way that it should perform all the testing mechanism. For that purpose we are studying three of the testing tools Win Runner, LoadRunner and QTP which performs different testing mechanisms having several features. This paper proposes a framework which performs all the testing mechanisms

Key words: N-Tier architecture, Framework, WinRunner, LoadRunner, Quick Test Professional.

INTRODUCTION

N-tier is a client-server architecture in which the presentation, the application processing, and the data management are logically separate process. Tier Architecture is the simplest, single tier on single user, and is the equivalent of running an application on a personal computer. The entire required components to run the application are located within it.

The tier architectures have been classified as 2-Tier Architectures, 3-tier architectures and finally N-tier architecture. Though there are different types of tier architecture we consider particular tier architecture based on the web application we are considering. Most commonly we use 3-Tier architecture. Each tier has its own advantages and disadvantages. The N- tier Architectures provide finer granularity(i.e. ability of an application to be broken into small granules) so that flexibility can be increased .N-tier architecture is typically defined by the Presentation ,Business Logic, Data Access Logic Layer and Database Layers

N-Tier Architecture Challenges have exposed the inability of traditional architectures to

efficiently utilize existing IT infrastructures. The challenges include Quality of Service (Dynamic resource allocation, Accounting and management, Infrastructure management, Heterogeneous legacy integration, Cluster support, Multiplatform Java technology, multilevel security model), Scalability, Availability and Predictability. Section 2 deals with

Testing automation and testing tools, Section 3 deals with system test planning, Section 4 deals with test factors Vs testing techniques, Section 5 deals with test case generation, Section 6 deals with testing tools, Section 7 deals with proposed framework, Section 8 deals with conclusion.

Test automation and testing tools

In this phase, the need for automation and availability of testing tools is taken into account. At present in software testing environment, we are faced with ever-diminishing release schedules, resources utilization, return investment, minimal resources and mandate to ensure quality applications². We need to test applications or solution as quickly as possible. The automated testing solutions can quickly and efficiently verify, be cost effective, validate, and ensure that the

applications are working accordingly, as per the client requirements. These solutions save resources and money in testing effort and are beneficial to the Software Development Life Cycle (SDLC) and Software Test Life Cycle (STLC). The important benefits of test automation include higher test coverage levels, greater reliability, shortened test cycles, ability to do multi user testing at no extra cost and customer of end-user satisfaction in the software.

System test planning

A test plan describes the intended scope, approach, resources and schedule of testing activities. It defines test items, features to be tested and not to be tested. It also specifies the personnel who performs each task in further test process, and in case of any risk, its required contingency plans are also included. A test plan includes tricky decisions on staffing, resources and allocation of time to test design. Decision on use of automation tools is a part of the plan too. We are using tools Win Runner, and Quick Test tool for functional testing and loadrunner to perform load and stress testing. It is necessary to have the right tool selection for the intended need. Risk analysis from test perspective is another aspect of test plan that needs utmost care.

If the requirements are rapidly changing then it is better to have a well defined and revisited test plan. This is essential as the changes in development plans or requirements may lead to changes in the test strategy altogether. The failures in testing are mainly because of not performing the plan revision upon changes.

Test case generation

A Test case is a group of tests that is to be executed to check the functionality of a specific object or an operational logic. Test case describes the user input and the system response with some preconditions to determine the correctness and completeness of the functionality. Test case is usually associated with at least one business function/requirement that is being validated. It requires a specific test data to be developed for input during the test case execution. The test case execution may be governed by preconditions that are required or setup before the execution of the

test case such as database support, printer setup or data that should exist at the start of the test case execution. Finally, we conclude that a test case validates one or more criteria to certify that an application is structurally and functionally ready to be implemented into the business production environment

The test cases will be effective and reusable, if they are designed by following certain rules. A test case must Be designed such that they are reusable, Give consistent results independent of testers, Specify what the tester has to perform and the response of the system, Be specific to the testing element or logic etc, Be divided into different test cases to keep them short and avoid confusions within the testers, Be divided into positive and negative scenarios, Be uniquely identified for revisiting, Hold the bug id for tracking purpose, Start with the word 'verify' or 'Check', Contain a maximum of 10 to 15 steps, Be reviewed by the superiors to ensure the coverage of whole functionality, Be approved by the client.

Test case Design methods

- ' Functional specifications based test cases design
- ' Use cases based test case design
- ' Application based test cases design

Functional specification based Test case Design

- ' Analyze the requirements in terms of entry point, exit point, normal flow, alternative flows, input required, expected outputs and exceptions
- ' Analyze the dependency in between requirements
- ' Understand the behavior by exploring the corresponding application/product.
- ' Determine the order or sequence of requirements in which specific transactions must be tested
- ' Determine the preconditions or setup necessary to execute a test procedure such as put the application build in ready state, database configuration or other requirements
- ' Determine the high risk requirements
- ' Author the test case titles or test case scenarios
- ' Review the test case titles or test case

- scenarios
- Finally write the test cases documents

Testing tools

In this paper we considered three testing tool frameworks for identifying proposed framework for N-Tier architecture (Win runner, Load runner, Quick test Professional)

Win Runner

Win Runner, a very powerful automated testing tool developed by Mercury Interactive Company is used for Functional Regression Testing. It automates functional testing process to ensure that the applications work as is expected from them. It records operations as we perform them manually in your application build and generates a test with required observations as checkpoints.

Salient Features

- Win Runner is a functional testing tool
- It supports various technologies like VB , JAVA , Power Builder , .Net, VC++ , HTML , Delphi, ActiveX and ERP/Siebel Technologies for automated functional testing
- It runs on windows family operating systems only
- It converts our manual functional tests into Test Script Language (TSL) programs.

Testing Process

Testing with Win Runner includes 4 main stages
Selecting manual functional test cases to be automated

The automation test process starts with the selection of manual test cases to be automated by a Testing Engineer.

Creating automated test scripts in TSL

Create tests using both recording and programming. While recording tests, insert checkpoints where you want to check the behavior of the Application under Test (AUT).

Running automated test scripts on build

When you run a test, Win Runner emulates a user by entering mouse and keyboard input into your application. Each time the Win Runner encounters a checkpoint in the test , it compares the current response of our application to its

expected response

Analyzing automated test script results

When a test run ends, the test engineer can examine the test results. Win Runner list all the major events that occurred during the run such as checkpoints

Quick test professional

QTP is Mercury automated keyword-driven testing solution like Win Runner. Quick Test Pro provides everything we need to quickly create, run and debug the test scripts. Automated testing with Quick Test Pro improves the speed of the testing process than Win Runner. We can create tests by using Quick Test Pro that checks all aspects of our application. Build and then run these tests every time on your application build changes for regression. As Quick Test Pro runs tests, it simulates a human user by moving the mouse cursor in an application build screen, like window or web page, clicking Graphical User Interface (GUI) objects, and entering keyboard input.

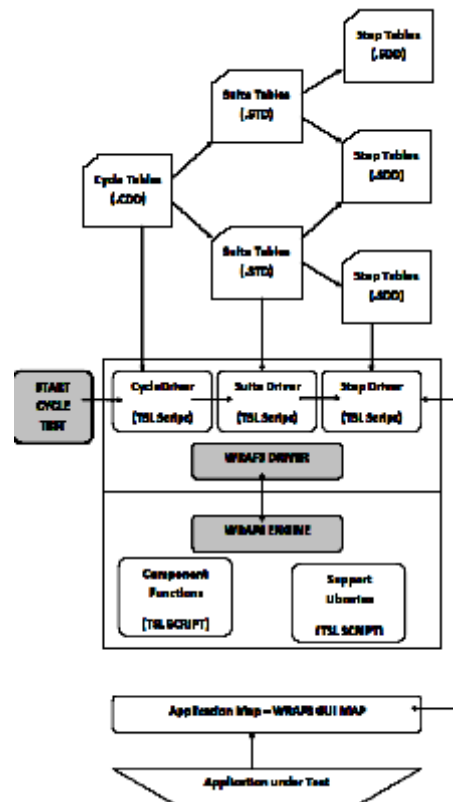


Fig. 1: Win runner framework architecture

Features

- ✓ Functionality testing tool
- ✓ Derived from Win Runner
- ✓ Support all Win Runner supported technologies and .Net, SAP, People Soft, Oracle application, XML and multimedia.
- ✓ Recognize objects and windows in your application build during creating test scripts like as Win Runner.
- ✓ Create automation test scripts in VBScript

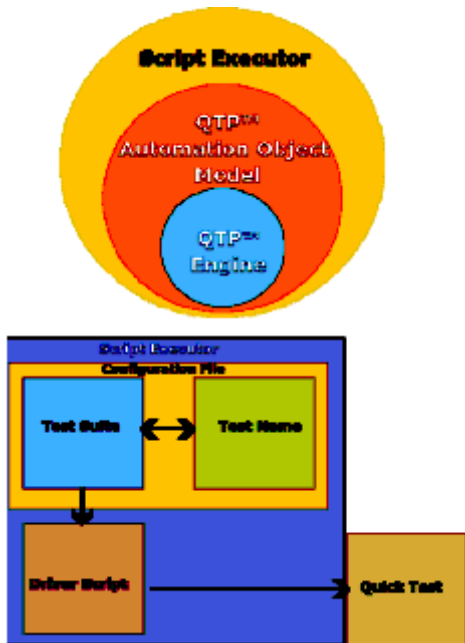


Fig. 2: QTP framework architecture

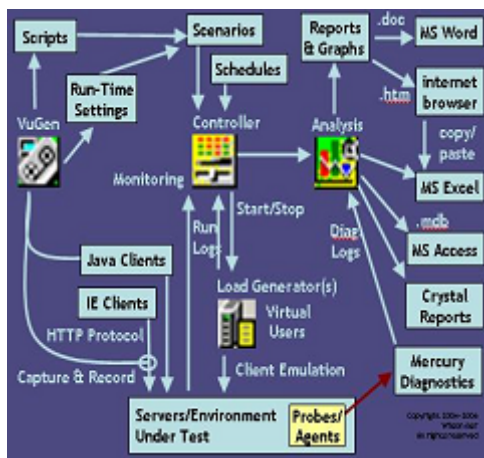


Fig. 3: Load Runner architecture

Testing Process

The QuickTest testing process consists of the following main phases to convert our selected manual functional test cases into automated test scripts

Preparing base state

Before creating a test, we can confirm that our application build and QuickTestPro are set to match the needs of test. Make sure, our application build displays the elements on which we want to record and that our application-build options are set as we expect for the purpose of our test, called Base state.

Recording actions on application build

As we navigate through our application build with respect to corresponding manual test case actions, Quick Test Pro graphically displays how each step is performed as a row in the Keyword View and displays VBScript program in the Expert View.

Enhancing recorded test script

This is about inserting checkpoints with our expected values to determine whether our application build is functioning correctly. By replacing fixed values with parameters, we can check how our application performs the same operations with multiple sets of test data. Adding logic and conditional or loop statements enable us to add sophisticated checks to our test.

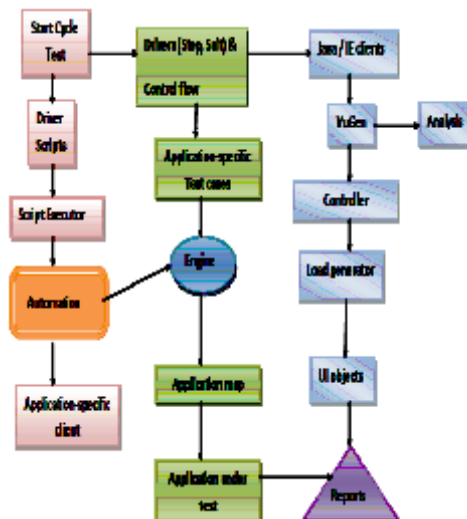


Fig. 4:

Debugging test script

We debug a test to ensure that it operates smoothly, without runtime errors. This debugging phase is an optional phase before running our test script on application build.

Running test script

We can run a test script to check the behavior of our application build. While running, QuickTestPro opens the application build and performs each step in our test script

Analyzing test results

We can examine the test results to pinpoint defects in our application build after running our test script.

Reporting defects

The detected defects can be reported to the point of contact by using an available tracking system

Load runner

Load Runner is the industry-standard

performance and load testing product for predicting system behavior and performance. The performance testing is performed to determine how fast the system performs operations under a particular workload in software engineering. Performance testing can be used in different purposes; it can demonstrate that the system meets performance criteria in terms of expected performance benchmarks or customer expectations.

Need of performance Testing

The quality of software is always defined in terms of good functionality and optimal performance. An organization that produces quality software with optimal performance will only be attracted and get good reputation. The term performance testing is just a theoretical concept whereas load and stress testing are practically used in software testing process. Web load testing is the measure of an entire web application's ability to sustain concurrent users, while maintaining adequate response time. The failure of production server due to performance problem is a huge loss to the customer business. Because of this reason,

Table 1: Test factors vs. Testing techniques

S. No	Test factor	Testing techniques
1	Authorization	Security Testing
2	Access Control	Security Testing
3	Audit trail	Functionality Testing
4	Correctness	Functionality Testing
5	Continuity of Processing	Integration Testing
6	Coupling	Inter Systems Testing
7	Data Integrity	Input domain Testing
8	Ease of use	<ul style="list-style-type: none"> ˆ User Interface Testing ˆ Manual Support Testing
9	Ease of operation	Installation Testing
10	Reliability	<ul style="list-style-type: none"> ˆ Recovery Testing ˆ Stress Testing
11	Performance	<ul style="list-style-type: none"> ˆ Load / Stress Testing ˆ Storage Testing ˆ Data Volume Testing
12	Portable	<ul style="list-style-type: none"> ˆ Compatible Testing ˆ Configuration Testing
13	Service levels	Functionality Testing
14	Maintainable	Compliance Testing
15	Methodology	Compliance Testing

Table 1:

S.No	Feature	Win Runner	LOADRUNNER	QTP
1	Components	3 different components	4 different components	3 different components
2	Frame work	Application Map – WRAFS GUI MAP, step driver, suit driver, WRAFS control flow and driver	Java clients , IE clients , Controller, Vuser host, Load generation, System Under test	Script Executor, QTP Automation Object Model , QTP engine
3	Type of testing tools	Functional testing tool, here we using C Script.	Performance testing tools, here we are using C# Script.	Functional testing tool, Here we are using VB Script.
4	Supported Platforms	Windows 98, Windows NT, Windows Me, Windows 2000, Windows XP	Microsoft Windows: Windows 3.x, Windows NT, Windows 95 UNIX: Sun OS, Solaris, Yes	Windows 2000-Service Pack 3 or Service Pack Windows XP-Service Pack 1 or Service Pack 2 Windows 2003 Server Yes
5	Platform dependency	Yes	Yes	Yes
6	Language in which the tool is written	C Script	C# script	VB Script
7	Working of components	To interact with the Application Under Test, it first learns the application. It does this through the use of its GUI Map editor Test tables are formulated in spread sheets with the knowledge of the application under test, the vocabulary followed in the WinRunner Automation Framework	The Controller organizes , drives , manages , and monitors the load test The Virtual User Generator captures end-user business processes and creates an automated performance testing script , also known as a virtual user script The Analysis helps u view, dissect and Compare the performance results The LoadGenerators create the load running by virtual users	SCRIPT EXECUTOR core use is to perform the same tasks multiple times or on multiple tests/components, quickly configuring QTP according to your needs for a particular execution of automated script.
8	Record/Replay	ODBC & Excel Connectivity	-	ODBC & Excel Connectivity
9	Numerous Checkpoints	Analog	-	Analog
10	Browsers	Internet Explorer • Netscape • AOL	Internet Explorer • Netscape • AOL	Microsoft Internet Explorer 5.01–6.0 Netscape Navigator 4.06–4.7x Netscape 6.1, Netscape 6.22, and Netscape 6.23 , AOL 5.0 and 6.0

organizations usually resolve performance problems before deploying the software into production³. This way of resolving performance problems before deploying the software into production is more cost efficient than resolving after deployment.

Features and Benefits¹

- Real-time performance monitors help quickly identify and resolve bottlenecks before deployment.
- LoadRunner uses sophisticated correlation, analysis and reporting capabilities to help improve application performance.
- Supports the most common enterprise environments: Including ERP/CRM, Web, J2EE, .NET, wireless, and streaming media.

Testing Process

Study System under Test

It is very important to understand the customer intentions and requirements regarding software performance. The major activities of this phase are:

- Application walkthrough to explore
- Access to the application
- Gathering all requirements such as performance goals, performance testing completion criteria, and software and hardware specifications of the production server.

Create Vuser Script

In this phase, major roles will do the following activities:

- Preparing load test strategy document
- Recording Vuser test scripts
- Enhancing Vuser test scripts

Execute Tests

In this phase, test engineer runs the script for multiple users with respect to the customer requirements. The important tasks to be done in this phase are:

- Test environment setup
- Application setup
- Test data collection
- Test execution
- Deriving the execution results and monitoring transaction flow

Analyze Test Results

After executing Vuser test scripts for multiple users, the testing team will collect test results and analyze them. To resolve performance bottlenecks, it is an important step in collecting and processing the data. The activities to be done in this phase are:

- Analyze the application server and database server resources, like CPU usage and memory usage.
- Analyze client side metrics like elapsed time and response time
- Identify the performance bottlenecks

In this paper we compare the properties and feature of each of the frameworks. Comparison is as follows as in the table. Until now we identified the framework analysis i.e., about the tool framework, architecture features and its comparison with other architectures

Proposed framework for n-tier application

N-Tier application includes all the processing layers in which it is related to the database and different layers. For this .Net applications are the best examples. So considering one of the application and testing use cases finally out of the considered frameworks Quick Test Pro is the best one but both has different features and several limitations. Instead of all these tools if there is a tool which performs both functional and performance testing we can easily perform all the tests for the web based application

The proposed architecture uses in different aspects. This is an automated testing proposed framework. The working of it will be the Script Executor will enhance, analyze, run, debug the test scripts and report defects, use is to perform the same tasks multiple times or on multiple tests/components First of all the cycle test will be started, transferred to Step, Suit & Cycle Driver States which will perform TSL Script, contain the argument of the global array with the details of the Cycle Table, Current Suit Table & Current Step Table. After the execution of the component functions, the Status Code is set to pass or fail based upon the execution results and then the control is passed to the Step Driver. The execution go back to Step Driver script for the same Step Table file and read the next record and this process

continues till all the records of that Step Table are processed. After that the control goes back to the Suite Table and next Record is read. The process continues till all the records in that Suite Tables are processed. After that control goes back to Cycle Table and its next record is read. The process continues till all the records in Cycle Tables are processed. The remaining components will organize, drive, manage, and monitor the load test, captures end-user business processes and creates an automated performance testing script, helps u view, dissect and compare the performance results, creates the load running by virtual user. The user can write his application – specific test cases so that they will be recorded, numerous checkpoints can also be generated.

In this framework, we are performing different testing strategies. The importance of testing is that to test whether the application responds quickly, handle the expected user load, and handle number of transactions for business & stability are to be checked. In this framework the component testing is performed to find the behavior and performance of each tier, the load testing to find out whether the system can handle the expected load upon deployment under real-word conditions, stress testing to find the applications breaking point also to apply testing that measures whether the applications environment is properly configured to handle

expected or potentially unexpected high transaction volumes, volume testing to find the stability of the system with respect to handling large amounts of data over extended time periods, functionality test to check the whether the business processes function work properly after implementation & load test objectives are to check stability, performance, functionality under load. After the application is tested, reports will be generated. This will perform effectively, easy to understand, has more extra features than remaining tools and user-friendly.

CONCLUSION

The main aim of this paper is to identify the generic N-Tier architecture for web application. So for this, we used three testing tools and performed system testing process. For N-Tier architecture the framework should include all the testing strategies and support performance and functional testing tools. We studied some of the testing tools features and a framework for N-Tier architecture should be a web-based management tool which will be very effective for the efficient management of the testing process and if organization believes in a process oriented approach to testing, the framework should be of immense use. In this paper we developed a new framework which performs all testing strategies

REFERENCES

1. Hung Q. Nguyen, Bob Johnson, and Michael Hackett *Testing Applications on the Web: Test Planning for Mobile and Internet-Based Systems, Second Edition*, John Wiley & Sons, Inc., (2003).
2. Bezier B., *Software Testing Techniques*, 2nd Ed, Van No strand Reinhold (1990).
3. Daniel S. Soper. A Framework for Automated Web Business Intelligence Systems. In: proceedings of the 38th Hawaii International Conference on System Sciences, Hawaii, IEEE (2005).
4. W. F. Cody. T. Kreulen, V. Krishna & W. S. Spangler. The integration of business intelligence and knowledge management. *Ibm Systems Journal*, **41**(4): (2002).
5. W. Pohs, G. Pinder, C. Dougherty, and M. White, the Lotus Knowledge Discovery System: Tools and Experiences [J] *IBM Systems Journal* **40**(4).
6. Dean Prichard. Generating surrogate data for time series with several simultaneously measured variables [J]. *Phys.Rev.Lett* (1994).
7. W. Retschitzegger, and W. Schwinger, "Towards Modeling of Data Web Applications - A Requirements' Perspective", *Proc. of the Americas Conference on Information Systems (AMCIS)* Long Beach California, Vol. I (2000).
8. G. Rossi, D. Schwabe, D., and A. Garrido, "Designing Computational Hypermedia Applications", *Journal of Digital Information (JODI)*, **1**(4): (1995).