**Table of Contents**

[Acknowledgement](#_Toc420605942) 2

[Summary](#_Toc420605942) 3

[About BLACKROCK 4](#_Toc420605942)

[Introduction To Assigned Tasks](#_Toc420605942) 11

[Software Testing 12](#_Toc420605944)

[What is Automation? 13](#_Toc420605947)

[Software Testing Tools 14](#_Toc420605950)

[Software Testing - Methods 14](#_Toc420605951)

[Black-Box Testing 14](#_Toc420605952)

[White-Box Testing 15](#_Toc420605953)

[Grey-Box Testing 16](#_Toc420605954)

[A Comparison of Testing Methods 17](#_Toc420605955)

[Software Testing – Levels 19](#_Toc420605956)

[Functional Testing 19](#_Toc420605957)

[Unit Testing 20](#_Toc420605958)

[Integration Testing 20](#_Toc420605961)

[Regression Testing 21](#_Toc420605963)

[About APG-Testing](#_Toc420605942) 23

[Tools and Technologies](#_Toc420605963) 24

A[utomated Testing Process: 25](#_Toc420605964)

[Framework 26](#_Toc420605965)

[Significance of Record and Playback: 30](#_Toc420605969)

[Object Repository: 31](#_Toc420605972)

[Example of Automation](#_Toc420605972) 33

[What is ALM?](#_Toc420605972) 36

[QTP-QC Integration](#_Toc420605972) 45

[VB Script](#_Toc420605972) 48

[What is Disaster Recovery?](#_Toc420605972) 51

DR Checkout at Blackrock53

[Project Schedule](#_Toc420605972) 56

[Conclusion and Future Scope of Work](#_Toc420605972) 57

[References](#_Toc420605972) 58

**ACKNOWLEDGEMENT**

As understanding of the study like this is never the outcome of the efforts of a single person, rather it bears the imprint of a number of persons who directly or indirectly helped me in completing the present study. I would be failing in my duty if I don’t say a word of thanks to all people.

Regarding this project, first and foremost, I would like to acknowledge and extend my heartfelt gratitude to **Mrs. Amanpreet Kaur** for her expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion. Her keen vital encouragement, superb guidance, and constant support are the motive behind this project work.

I owe my gratitude to **Mr. Vaibhav Masih**, Vice-President (APG-QA), to provide me the privilege to be a part of this project under BlackRock, Inc. and his guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project

Last, but not the least special thanks to all my teammates in BlackRock, especially **Parminder Singh**, **Abhinav Rajput** and **Sanket Khanolkar** for kind co-operation, encouragement and indispensable support which helped me throughout in this project. A very special thanks to **Arnab Rakshit** for being a supportive and sensible partner in my project.

**SUMMARY**

**Disaster recovery** (DR) involves a set of procedures to enable the recovery or continuation of vital technology infrastructure and systems following a natural or human-induced disaster. Disaster recovery focuses on the IT or technology systems supporting critical business functions, as opposed to business continuity, which involves keeping all essential aspects of a business functioning despite significant disruptive events. Disaster recovery is therefore a subset of business continuity.

Significant negative events, such as equipment failures, as well as hurricanes, earthquakes and other natural disasters, can include anything that puts an organization’s operations at risk, so to reduce the influence of occurrence of any such events is high priority task which needs to be addressed.

**ABOUT BLACKROCK**

**Ideas, Innovations and Growth**

Beginning with its founding in 1988, **BlackRock’s** story over a quarter of a century is about a commitment to putting client’s first, innovative thinking, passion for performance and a remarkable collaboration.

* BlackRock is the world’s largest Asset Managerand a premier provider of Risk managementand advisory services to institutional, intermediary, and individual clients worldwide.
* To a broad base of clients with portfolios totalling approximately US$13 trillion. As of June 1, 2014, BlackRock's Assets Under Management across equity, fixed income, cash management, alternative investment, real estate, advisory strategies and the industry-leading iShares® exchange-traded funds. Through BlackRock Solutions, it offers risk management, strategic advisory and enterprise investment system services
* BlackRock is a truly global firm that combines the benefits of worldwide reach with local service and relationships. They manage assets for clients in North and South America, Europe, Asia, Australia, the Middle East and Africa. Headquartered in New York, BlackRock employs more than 11,000 talented professionals and maintains offices in 30 countries around the world. Blackrock’s client base includes corporate, public, union and industry pension plans; governments; insurance companies; third-party mutual funds; endowments; foundations; charities; corporations; official institutions; sovereign wealth funds; banks; financial professionals; and individuals worldwide.

That’s why investors of all kinds entrust Blackrock with trillions of dollars, and it’s why companies, institutions and global governments come to us for help meeting their biggest financial challenges.

**Mission and Principles**

Our mission is to create a better financial future for our clients. As we pursue that mission, we’re guided by our culture and a set of principles that ensure we never forget what we stand for.

1. **We are fiduciary to our Clients.**
2. **We are passionate about Performance.**
3. **We are One BlackRock**
4. **We are Innovators.**

**Aladdin Platform Overview**

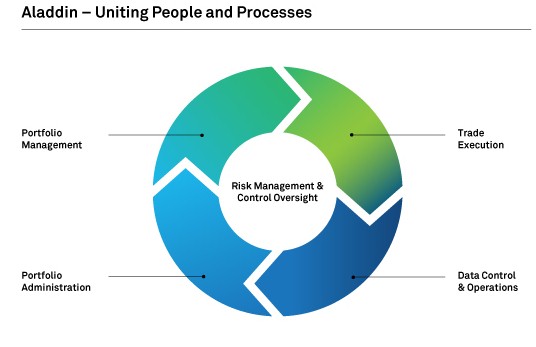
Aladdin is an operating system for investment managers that connect the information, people and technology needed to manage money in real time. The Aladdin platform combines sophisticated risk analytics with comprehensive portfolio management, trading and operations tools on a single platform to power informed decision-making, effective risk management, efficient trading and operational scale.

More than just technology, Aladdin powers [Collective Intelligence](https://www.blackrock.com/aladdin/benefits/organizations) by providing tools to help your organization communicate effectively, address problems quickly, and make informed decisions at every step of the investment process.

### Aladdin helps:

* **Technology Teams** - Benefit from a robust suite of highly flexible, real-time management tools through the Aladdin dashboard, so that you can see what you need when overseeing intraday activity or external communications across different business functions.
* **Work Smarter –** Aladdin handles the extensive data processing needed to support investment management, helping put the information you need to be effective at your fingertips.
* **Move Faster –** You can move fast on opportunities with Aladdin's ability to process enormous amounts of data quickly and on a single system.
* **Portfolio Managers** - Receive automated trade confirmations right in your dashboard, as Aladdin is fully integrated with Omgeo OASYS/CTM, Mark it SERV and SWIFT. Confirmed trades are sent directly to custodians through SWIFT, helping to support settlements and cash reconciliations.
* **Risk Managers** - Access Aladdin's comprehensive suite of sophisticated models, built over the past 20 years by a dedicated financial modelling team.

**ALADDIN OVERVIEW**

****

**BLACKROCK PRODUCT AND SERVICES**

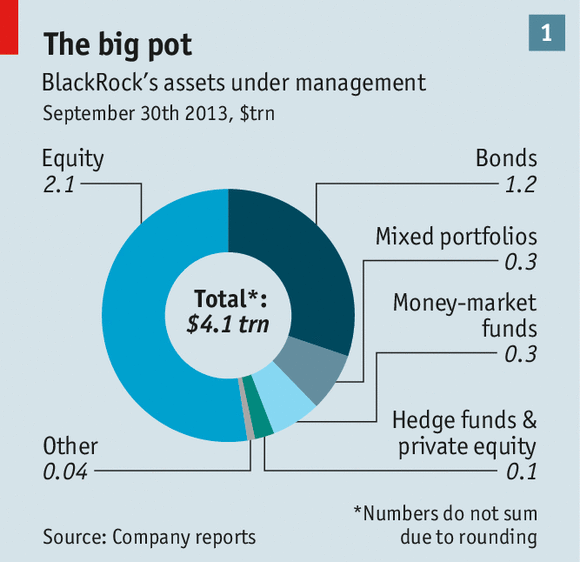


Fig 2. BlackRock assets under management

1. **MUTUAL FUNDS** : An investment vehicle that is made up of a pool of funds collected from many investors for the purpose of investing in securities such as stocks, bonds, money market instruments and similar assets. Mutual funds are operated by money managers, who invest the fund's capital and attempt to produce capital gains and income for the fund's investors. A mutual fund's portfolio is structured and maintained to match the investment objectives stated in its prospectus.
2. **IShares ETFs**: ETFs are the Exchange Traded Funds and represents a security that tracks an index, a commodity or a basket of assets like an index fund, but trades like a stock on an exchange. ETFs experience price changes throughout the day as they are bought and sold. A large provider of exchange-traded funds, managed by the investment management company BlackRock. iShares, Inc. funds began trading in the United States and the United Kingdom in 2000, and are listed on major exchanges such as the NYSE Euronext, Chicago Board Options Exchange, Nasdaq and NYSE Arca. In 2010, iShares, Inc. represented 46% of the U.S. ETF market, with over 440 funds and excess of $480 billion in assets under management.
3. **Close Ended Funds** : Close ended funds raises a prescribed amount of capital only once through an IPO by issuing a fixed number of shares, which are purchased by investors in the closed-end fund as stock. Unlike regular stocks, closed-end fund stock represents an interest in a specialized portfolio of securities that is actively managed by an investment advisor and which typically concentrates on a specific industry, geographic market, or sector. The stock prices of a closed-end fund fluctuate according to market forces (supply and demand) as well as the changing values of the securities in the fund's holdings.
4. **Multi Asset Income Funds** : Multi Asset Funds represents a combination of asset classes (such as cash, equity or bonds) used as an investment. A multi-asset fund investment would contain more than one asset class, thus creating a group or portfolio of assets. For example, a multi-asset class investor might hold bonds, stocks, cash and real property, whereas a single-class investor might only hold stocks. The weights and types of classes will vary according to the individual investor. The BlackRock Multi-Asset Income Fund seeks out the best income opportunities around the world and across asset classes, carefully balancing the trade-offs between yield and risk.
5. **Alternatives** : Alternative investments – such as real estate, private equity, hedge funds, commodities, infrastructure and renewable energy – should no longer be regarded as 'alternative'. In an environment where returns from traditional investments are not enough to meet future liabilities and correlations among asset classes and regions are high, alternative investments can provide investors with a vital source of uncorrelated returns. Combined with a commensurate focus on risk management, a core alternatives allocation has the potential to help investors construct more risk-efficient portfolios, increase return over time and ultimately reach the outcomes they need.

**INTRODUCTION TO ASSIGNED TASKS**

* Understanding the Testing Process Workflow : Understanding Testing Analyzing Test Scenario, Designing Test steps, Creating Test Scripts
* Learning Quick Test Professional : Understand the working of the tool and executing Test Scripts using Object Repository and Function Libraries
* Learning Application Lifecycle Management Tool: Understanding the test Management Tool and its various tabs. Using these to manage test cases and execute on different remote machines
* Understanding the Disaster Recovery Process, Framework for DR Checkout and highlight improvement areas in existing scripts.
* Automate Logging-In into Virtual Machines using Batch Files and VBS Files.
* Understanding the business of Blackrock as well as the Testing Process
* Gaining Insight into Finance Domain

**Software Testing**

This section describes the different types of testing that may be used to test a software during SDLC.

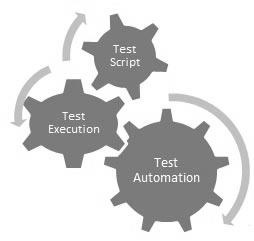
**Manual Testing**

Manual testing includes testing a software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. There are different stages for manual testing such as unit testing, integration testing, system testing, and user acceptance testing.

Testers use test plans, test cases, or test scenarios to test a software to ensure the completeness of testing. Manual testing also includes exploratory testing, as testers explore the software to identify errors in it.

**Automation Testing**

Automation testing, which is also known as Test Automation, is when the tester writes scripts and uses another software to test the product. This process involves automation of a manual process. Automation Testing is used to re-run the test scenarios that were performed manually, quickly, and repeatedly.



Apart from regression testing, automation testing is also used to test the application from load, performance, and stress point of view. It increases the test coverage, improves accuracy, and saves time and money in comparison to manual testing.

**What is Automation?**

It is not possible to automate everything in a software. The areas at which a user can make transactions such as the login form or registration forms, any area where large number of users can access the software simultaneously should be automated.

Furthermore, all GUI items, connections with databases, field validations, etc. can be efficiently tested by automating the manual process.

**When to Automate?**

Test Automation should be used by considering the following aspects of a software:

* Large and critical projects
* Projects that require testing the same areas frequently
* Requirements not changing frequently
* Accessing the application for load and performance with many virtual users
* Stable software with respect to manual testing
* Availability of time

**How to Automate?**

Automation is done by using a supportive computer language like VB scripting and an automated software application. There are many tools available that can be used to write automation scripts. Before mentioning the tools, let us identify the process that can be used to automate the testing process:

* Identifying areas within a software for automation
* Selection of appropriate tool for test automation
* Writing test scripts
* Development of test suits
* Execution of scripts
* Create result reports
* Identify any potential bug or performance issues

**Software Testing Tools**

The following tools can be used for automation testing:

* HP Quick Test Professional
* Selenium
* IBM Rational Functional Tester

**Software Testing - Methods**

There are different methods that can be used for software testing. This chapter briefly describes the methods available.

**Black-Box Testing**

The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

The following table lists the advantages and disadvantages of black-box testing.

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| * Well suited and efficient for large code segments. * Code access is not required. * Clearly separates user's perspective from the developer's perspective through visibly defined roles. * Large numbers of moderately skilled testers can test the application with no knowledge of implementation, programming language, or operating systems. | * Limited coverage, since only a selected number of test scenarios is actually performed. * Inefficient testing, due to the fact that the tester only has limited knowledge about an application. * Blind coverage, since the tester cannot target specific code segments or error-prone areas. * The test cases are difficult to design. |

**White-Box Testing**

White-box testing is the detailed investigation of internal logic and structure of the code. White-box testing is also called **glass testing** or **open-box testing**. In order to perform **white-box** testing on an application, a tester needs to know the internal workings of the code.

The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

The following table lists the advantages and disadvantages of white-box testing.

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| * As the tester has knowledge of the source code, it becomes very easy to find out which type of data can help in testing the application effectively. * It helps in optimizing the code. * Extra lines of code can be removed which can bring in hidden defects. * Due to the tester's knowledge about the code, maximum coverage is attained during test scenario writing. | * Due to the fact that a skilled tester is needed to perform white-box testing, the costs are increased. * Sometimes it is impossible to look into every nook and corner to find out hidden errors that may create problems, as many paths will go untested. * It is difficult to maintain white-box testing, as it requires specialized tools like code analyzers and debugging tools. |

**Grey-Box Testing**

Grey-box testing is a technique to test the application with having a limited knowledge of the internal workings of an application. In software testing, the phrase the more you know, the better carries a lot of weight while testing an application.

Mastering the domain of a system always gives the tester an edge over someone with limited domain knowledge. Unlike black-box testing, where the tester only tests the application's user interface; in grey-box testing, the tester has access to design documents and the database. Having this knowledge, a tester can prepare better test data and test scenarios while making a test plan.

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| * Offers combined benefits of black-box and white-box testing wherever possible. * Grey box testers don't rely on the source code; instead they rely on interface definition and functional specifications. * Based on the limited information available, a grey-box tester can design excellent test scenarios especially around communication protocols and data type handling. * The test is done from the point of view of the user and not the designer. | * Since the access to source code is not available, the ability to go over the code and test coverage is limited. * The tests can be redundant if the software designer has already run a test case. * Testing every possible input stream is unrealistic because it would take an unreasonable amount of time; therefore, many program paths will go untested. |

**A Comparison of Testing Methods**

The following table lists the points that differentiate black-box testing, grey-box testing, and white-box testing.

|  |  |  |
| --- | --- | --- |
| **Black-Box Testing** | **Grey-Box Testing** | **White-Box Testing** |
| The internal workings of an application need not be known. | The tester has limited knowledge of the internal workings of the application. | Tester has full knowledge of the internal workings of the application. |
| Also known as closed-box testing, data-driven testing, or functional testing. | Also known as translucent testing, as the tester has limited knowledge of the insides of the application. | Also known as clear-box testing, structural testing, or code-based testing. |
| Performed by end-users and also by testers and developers. | Performed by end-users and also by testers and developers. | Normally done by testers and developers. |
| Testing is based on external expectations - Internal behavior of the application is unknown. | Testing is done on the basis of high-level database diagrams and data flow diagrams. | Internal workings are fully known and the tester can design test data accordingly. |
| It is exhaustive and the least time-consuming. | Partly time-consuming and exhaustive. | The most exhaustive and time-consuming type of testing. |
| Not suited for algorithm testing. | Not suited for algorithm testing. | Suited for algorithm testing. |
| This can only be done by trial-and-error method. | Data domains and internal boundaries can be tested, if known. | Data domains and internal boundaries can be better tested. |

**Software Testing – Types**

There are different levels during the process of testing. In this chapter, a brief description is provided about these levels.

Levels of testing include different methodologies that can be used while conducting software testing. The main levels of software testing are:

* Functional Testing
* Non-functional Testing

**Functional Testing**

This is a type of black-box testing that is based on the specifications of the software that is to be tested. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for. Functional testing of a software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

There are five steps that are involved while testing an application for functionality.

|  |  |
| --- | --- |
| **Steps** | **Description** |
| I | The determination of the functionality that the intended application is meant to perform. |
| II | The creation of test data based on the specifications of the application. |
| III | The output based on the test data and the specifications of the application. |
| IV | The writing of test scenarios and the execution of test cases. |
| V | The comparison of actual and expected results based on the executed test cases. |

An effective testing practice will see the above steps applied to the testing policies of every organization and hence it will make sure that the organization maintains the strictest of standards when it comes to software quality.

**Unit Testing**

This type of testing is performed by developers before the setup is handed over to the testing team to formally execute the test cases. Unit testing is performed by the respective developers on the individual units of source code assigned areas. The developers use test data that is different from the test data of the quality assurance team.

The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

**Limitations of Unit Testing**

Testing cannot catch each and every bug in an application. It is impossible to evaluate every execution path in every software application. The same is the case with unit testing.

There is a limit to the number of scenarios and test data that a developer can use to verify a source code. After having exhausted all the options, there is no choice but to stop unit testing and merge the code segment with other units.

**Integration Testing**

Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. Integration testing can be done in two ways: Bottom-up integration testing and Top-down integration testing.

|  |  |
| --- | --- |
| **S.N.** | **Integration Testing Method** |
| 1 | Bottom-up integration  This testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds. |
| 2 | Top-down integration  In this testing, the highest-level modules are tested first and progressively, lower-level modules are tested thereafter. |

In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing. The process concludes with multiple tests of the complete application, preferably in scenarios designed to mimic actual situations.

**Regression Testing**

Whenever a change in a software application is made, it is quite possible that other areas within the application have been affected by this change. Regression testing is performed to verify that a fixed bug hasn't resulted in another functionality or business rule violation. The intent of regression testing is to ensure that a change, such as a bug fix should not result in another fault being uncovered in the application.

Regression testing is important because of the following reasons:

* Minimize the gaps in testing when an application with changes made has to be tested.
* Testing the new changes to verify that the changes made did not affect any other area of the application.
* Mitigates risks when regression testing is performed on the application.
* Test coverage is increased without compromising timelines.
* Increase speed to market the product.

**ABOUT APG-TESTING**

APG stands for Aladdin Product Group. As discussed, Aladdin is a platform where all the applications that promote trading reside. Testing as the name suggests tests the apllications from functional point of view as well as domain wise. There are 3 teams in the testing group

1. **Web Testing Team:** Thisteam is responsible for testing of all the Blackrock websites that are public. Blackrock has 2 websites:

a) **Blackrock.com:** This website hosts and supports the product mutual funds

b) **iShares.com:** This Website hosts and supports the product iShares

2. **Functional Testing:** This team is responsible for testing of allBlackrock applications that promote trading. Every time, an application gets on boarded, the application needs to be tested. Also the existing applications need to be maintained with the enhancements. The Functional team manually uses the test cases to test the functionality from domain perspective. Due to confidentiality, these applications cannot be listed down.

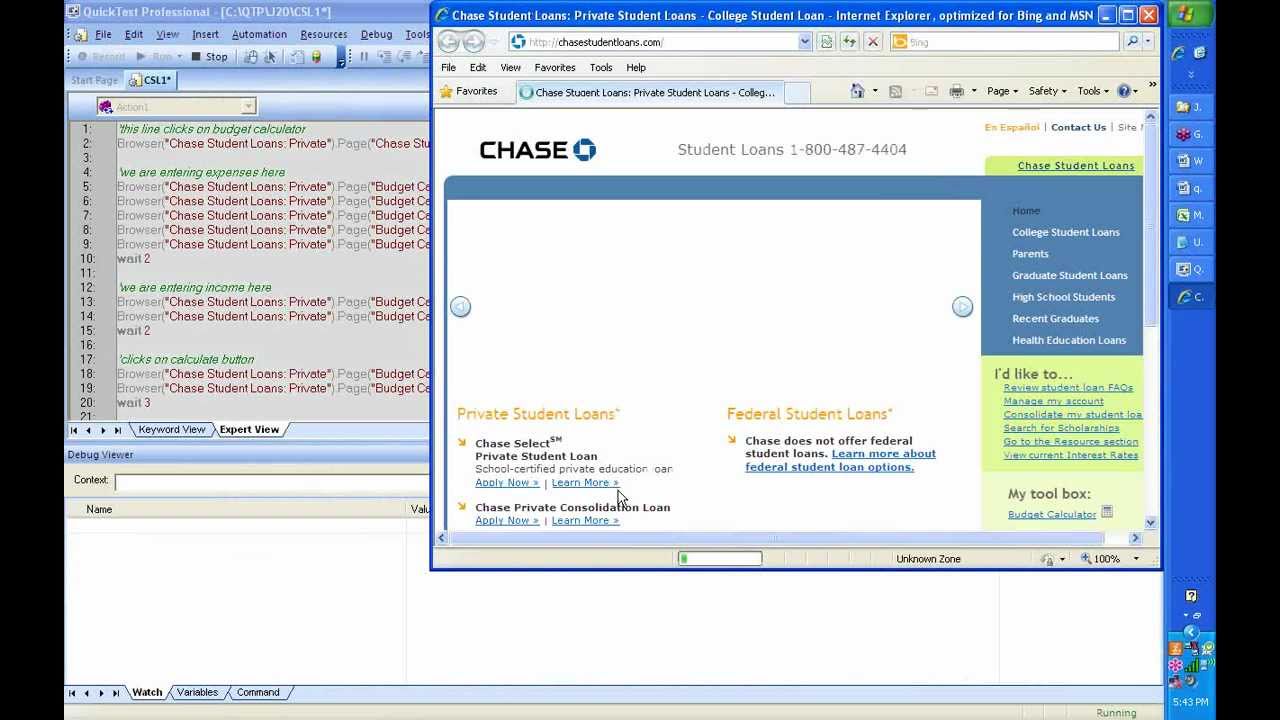
3. **Automation Services Team (AST):** This team focuses on automating the whole test flow. The functional team works upon the manual testing. Once this is done, this team automates the flow. The team uses third party tools like: Selenium and QTP.

I am working under the AST team in a strategic project called DR Checkout.

**TOOLS AND TECHNOLOGIES USED**

**What is QTP?**

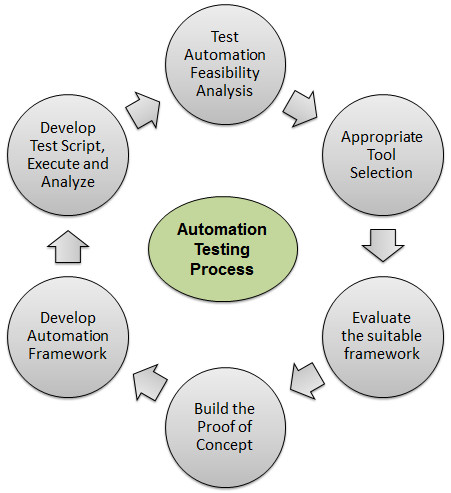
[](http://www.google.com/url?sa=i&rct=j&q=&esrc=s&frm=1&source=images&cd=&cad=rja&uact=8&ved=0CAcQjRw&url=http://www.varshneyinfotech.net/qtp-training-mumbai&ei=Z7tUVcDUHMykgwTa0oDADA&bvm=bv.93112503,d.eXY&psig=AFQjCNHZDs2dGJj7-jmyQSifygSAbyXJDw&ust=1431702733463748)

[](http://www.google.com/url?sa=i&rct=j&q=&esrc=s&frm=1&source=images&cd=&cad=rja&uact=8&ved=0CAcQjRw&url=http://www.youtube.com/watch?v%3D91XSG7qPp0o&ei=iv5VVcDzG6fbsASn2IG4Cg&bvm=bv.93564037,d.cWc&psig=AFQjCNGLByBLV6chaFvei1txYXNKh7xBcQ&ust=1431785076213542)

**HP Unified Functional Testing** (**UFT**) software, formerly known as **HP QuickTest Professional** (**QTP**) provides [functional](http://en.wikipedia.org/wiki/Functional_testing) and [regression test](http://en.wikipedia.org/wiki/Regression_testing) automation for software applications and environments. It uses the Visual Basic Scripting Edition ([VBScript](http://en.wikipedia.org/wiki/VBScript)) scripting language to specify a test procedure, and to manipulate the objects and controls of the application under test. It works by identifying the objects in the application user interface or a web page and performing desired operations (such as mouse clicks or keyboard events).

**Automated Testing Process:**

For any automated tool implementation, the following are the phases/stages of it. Each one of the stages corresponds to a particular activity and each phase has a definite outcome.



* **Test Automation Feasibility Analysis** - First step is to check if the application can be automated or not. Not all applications can be automated due to its limitations.
* **Appropriate Tool Selection** - The Next most important step is the selection of tools. It depends on the technology in which the application is built, its features and usage Like Java, Pearl, Web etc.
* **Evaluate the suitable framework** - Upon selecting the tool the next activity is to select a suitable framework. There are various kinds of frameworks and each framework has its own significance.
* **Build the Proof of Concept** - Proof of Concept(POC) is developed with an end to end scenario to evaluate if the tool can support the automation of the application. As it is performed with an end to end scenario which will ensure that the major functionalities can be automated.
* **Develop Automation Framework** - After building the POC, framework development is carried out, which is a crucial step for the success of any test automation project. Framework should be build after diligent analysis of the technology used by the application and also its key features.
* **Develop Test Script, Execute and Analyze** - Once Script development is completed, the scripts are executed, results are analyzed and defects are logged, if any. The Test Scripts are usually version controlled.

**Framework**

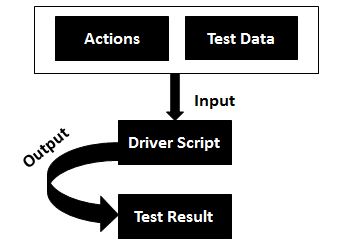
A Framework defines a set of guidelines/best practices that enforces a set of standards which makes it easy to use for the end users to work with. There are different types of automation frameworks and the most common ones are listed below:

* Keyword-Driven Framework
* Data-Driven Framework
* Hybrid Framework

**Keyword-Driven Framework**

Keyword driven testing is a type of functional automation testing framework which also known as table-driven testing or action word based testing.

In Keyword-driven testing we use a table format, usually a spreadsheet, to define keywords or action words for each function that we would like to execute.



**Advantages:**

* It is best suited for novice or a non-technical tester.
* Enables writing tests in a more abstract manner using this approach.
* Keyword driven testing allows automation to be started earlier in the SDLC even before a stable build is delivered for testing.
* There is a high degree of reusability.

**Disadvantages:**

* Initial investment in developing the keywords and its related functionalities might take longer.
* It might act as a restriction to the technically abled testers.

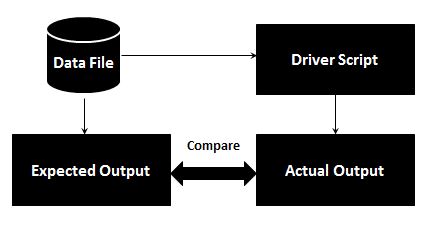
**Data Driven Framework**

Data-driven testing is creation of test scripts where test data and/or output values are read from data files instead of using the same hard-coded values each time the test runs. This way testers can test how the application handles various inputs effectively. It can be any of the below data files.

* Datapools
* Excel files
* ADO objects
* CSV files
* ODBC sources

**Flow Diagram:**

Data Driven Testing can be best understood by the following diagram:



**Advantages:**

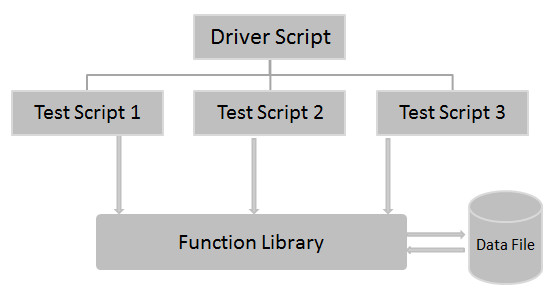
* Data driven framework results in less amount of code.
* Offers greater flexibility for maintaining and fixing the scripting issues.
* Test Data can be developed

**Disadvantages:**

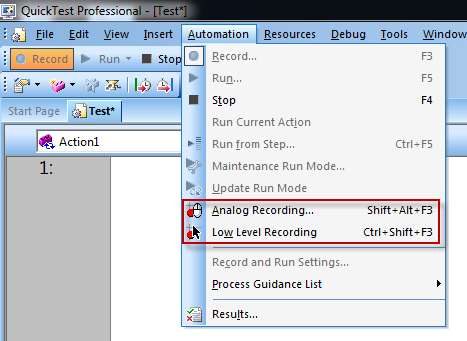
* Each script needs to be different to understand different sets of data.

**Hybrid Framework**

Hybrid Framework is a combination of Keyword driven and data Driven framework that can be best described using the following flow diagram.



**Significance of Record and Playback:**

[](http://www.google.com/url?sa=i&rct=j&q=&esrc=s&frm=1&source=images&cd=&cad=rja&uact=8&ved=0CAcQjRw&url=http://testautomateandenjoy.blogspot.com/2012/08/qtp-recording-modes.html&ei=KAhWVd72AcungwTG7wE&bvm=bv.93564037,d.cWc&psig=AFQjCNF4srGtnB3xvEGt6_wzLCFyhH0aPg&ust=1431787936221710)

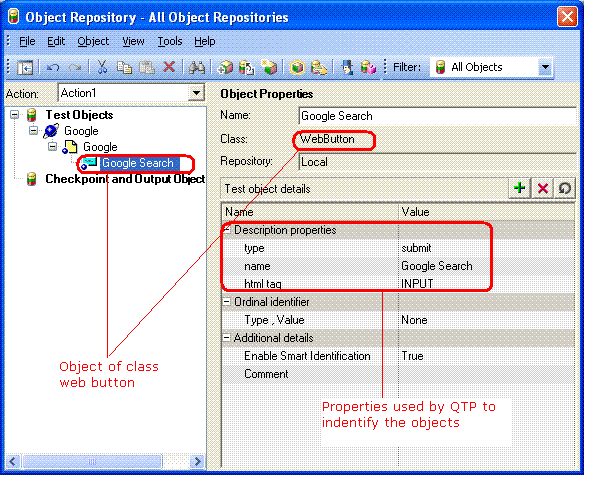
* It is used as the preliminary investigation method to verify if UFT can support the technology/application.
* Used to create a test a basic functionality of an application or feature that does not require long-term maintenance.
* It can be used for recording both mouse movements and keyboard inputs.

**Modes of Recording:**

* **Normal Recording**: This is the default recording mode that records the objects and the operations performed on the application under test.
* **Analog Recording**: This records not only the keyboard actions but also the mouse movements relative to the screen or the application window.
* **Low-Level Recording**: This records the exact co-ordinates of the objects independent of the fact whether UFT recognizes the object or NOT. It just records the co-ordinates, hence does NOT record mouse movements.
* **Insight Recording**: UFT records operation based on its appearance and NOT based on its native properties.

**Object Repository:**

Object Repository is a collection of object and properties with which QTP will be able to recognize the objects and act on it. When a user records a test, the objects and its properties are captured by default. Without understanding objects and its properties, QTP will NOT be able to play back the scripts.

[](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&frm=1&source=images&cd=&cad=rja&uact=8&ved=0CAcQjRw&url=https://testingideas.wordpress.com/2008/08/26/object-repository-and-descriptive-programming-in-qtp/&ei=eAhWVafjO8vhgwSZgoKwAg&bvm=bv.93564037,d.cWc&psig=AFQjCNHg9ZX13JRW7cDgfDVbdkb6Fv91iA&ust=1431788011044215)

**Advantages:**

* Developing automated tests using VBScript doesn't require a highly skilled coder and relatively easy when compared other object oriented programming languages.
* Easy to use, ease of navigation, results validation and Report generation.
* Readily Integrated with Test Management Tool (Hp-Quality Center) which enables easy scheduling and Monitoring.
* Since it is a Hp product, the full support is provided by HP and by its forums for addressing technical issues.

**Disadvantages:**

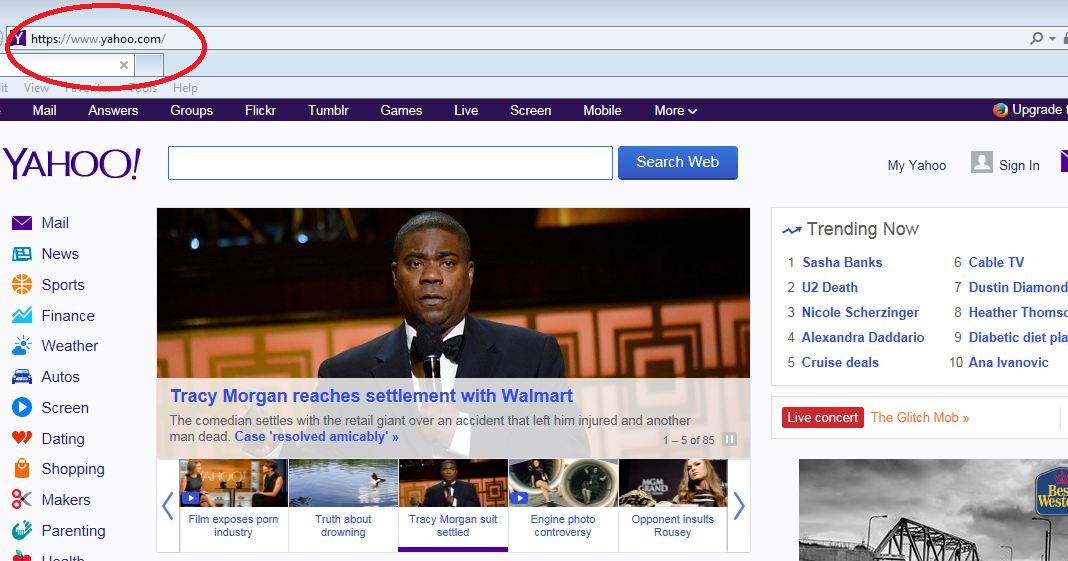
* Not all versions of Browsers are supported and the testers need to wait for the patch to be released for each one of the major versions.
* Having said that it is a commercial tool, the licensing cost is very high.

**EXAMPLE OF AUTOMATING USING QTP**

**Objective: Creating an account in Yahoo**

|  |  |  |
| --- | --- | --- |
| Step No | Description | Expected |
| Step No. 1 | To launch the browser and load www.yahoo.com | [www.yahoo.com](http://www.yahoo.com) is launched |

**Screenshot 1:**



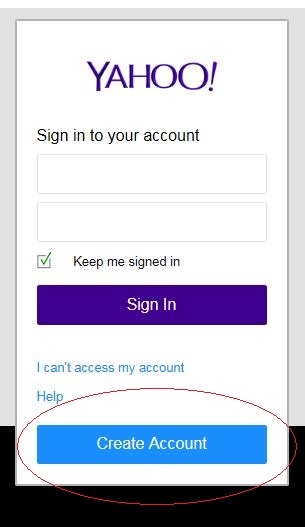
|  |  |  |
| --- | --- | --- |
| Step No | Description | Expected |
| Step No. 2 | To click the sign-in button | The sign-in button is clicked and the login and sign-in page appears |

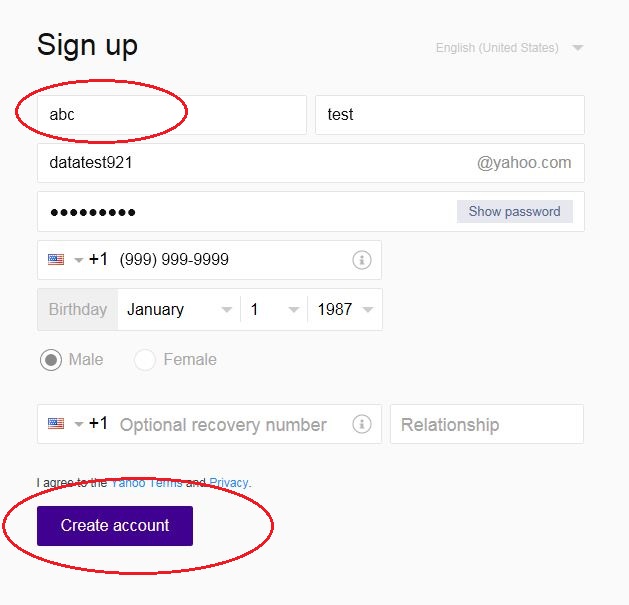
**Screenshot 2:**



|  |  |  |
| --- | --- | --- |
| Step No | Description | Expected |
| Step No. 3 | Click the Create Account button  Fill up the form and click the create account button  Capture the username from the form | The Create Account Button is Clicked and Sign-in form appears.  The form is filled and the button is clicked, account created  Username is captured  say : abc |

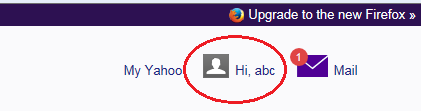
**Screenshot 3:**



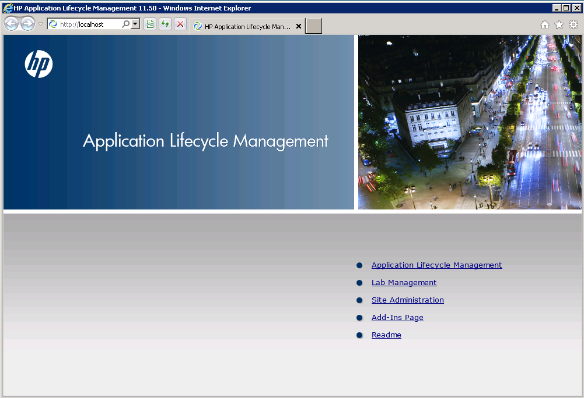
**Screenshot 4: **

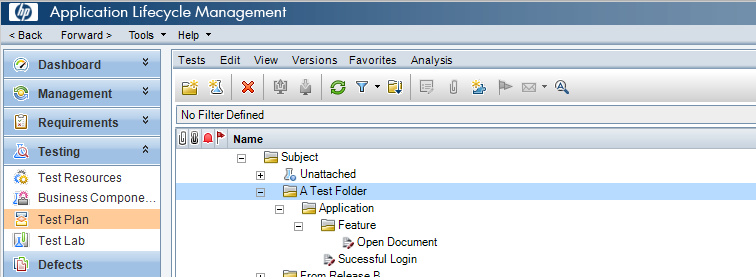
|  |  |  |
| --- | --- | --- |
| Step No | Description | Expected |
| Step No. 4 | Account is created then check if the username is same or not | The username is same. |

**Screenshot 5:**



**What is ALM?**



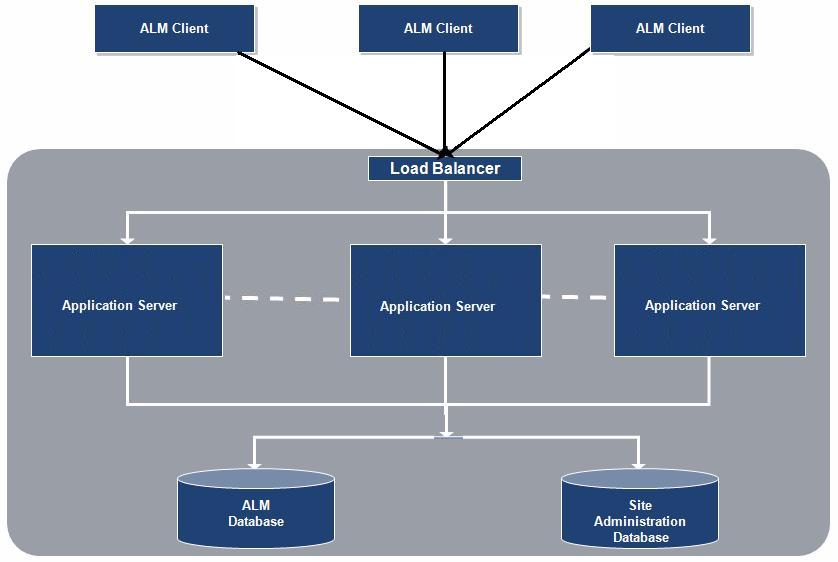
[](http://www.google.com/url?sa=i&rct=j&q=&esrc=s&frm=1&source=images&cd=&cad=rja&uact=8&ved=0CAcQjRw&url=http://www.shooter-smith.co.uk/how-to-export-tests-from-excel-to-hp-qc-alm/&ei=NgxWVbfQCIWVNrqsgNAB&bvm=bv.93564037,d.cWc&psig=AFQjCNFcj9i5zGBgYyKGFJEYK1My2mtuCA&ust=1431788938329017)

HP Quality Center, a test management tool is now popularly known as **A**pplication **L**ife Cycle **M**anagement Tool (ALM) as it is no longer just a test management tool but it supports various phases of the software development life cycle.

**Architecture of QC:**

HP-ALM, an enterprise-wide application that is based on Java 2 Enterprise Edition (J2EE) technology and uses MS SQL Server or Oracle as its back end. There is also a load balancer to effectively cater user's requests.

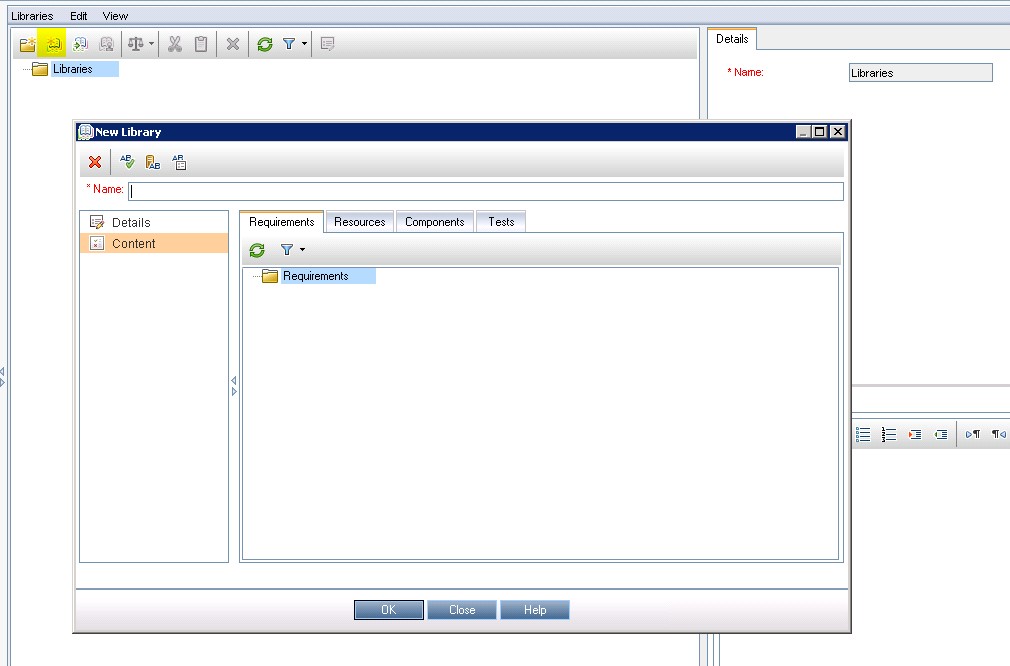
Site Administration Database is hosted on a standalone Database server while other project related data are stored on a separate Database server. The below diagram shows how the set up would be for larger corporations.



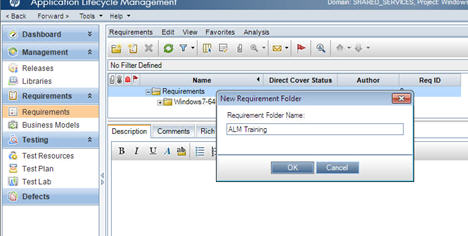
**Libraries**

A library represents a set of entities in a project and the relationships between them which consists of requirements, tests, test resources, and business components

A New Library can be created by clicking on "New Library" Icon in Library Navigation bar. A New window would be as shown below and one can name the library and add resources such as requirements, test plan, components.as requirements, test plan, components.

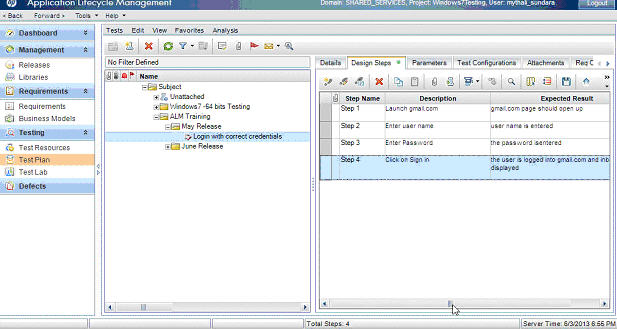


**Requirements**



Defining and Capturing Requirements is one of the key criteria for any software development process. Describing Requirements refers to what needs to be done to meet the objectives during development. Establishing requirements Clearly and correctly upfront is very much emphasized so that there would be minimal/no rework after UAT. The Requirements module enables users to define, manage and track requirements at all stages of the software lifecycle.

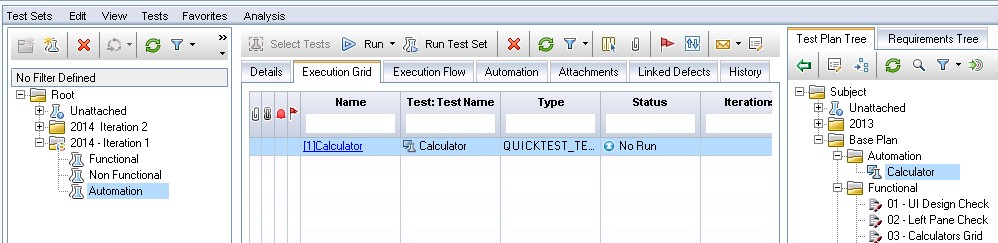
**Test Plan:**



The crucial step in testing any application is developing a clear and a precise test plan. A good test plan enables the team to assess the quality of the application under test at any point in the software development life cycle.

* **Creating Tests**: This module describes how to create folders of test subjects in the test plan tree and also to add tests.
* **Uploading Tests:** Uploading Tests using ALM-MS Excel, ALM-QTP Addins
* **Requirement and Test Coverage:** Enables how to define the relationship between the requirements and tests.
* **Test Configuration:** Specifies the subset of data or a run-time environment that the test should use.

**Test Lab:**



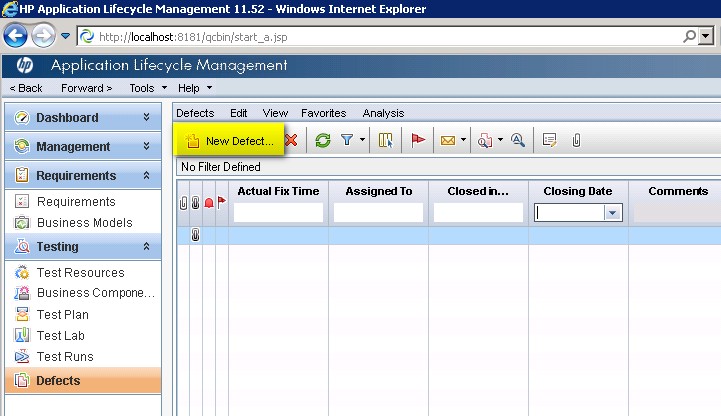
This module helps the testers to execute the created tests. One can schedule, run and analyze, post defects using this module. It basically contains all the Test Suit which holds a reference to the Tests kept in Test Plan

* **Creating Test Set:** This functionality describes how to create and define test sets in the Test Lab module.
* **Test Run Scedules:** Helps the testers to control the execution of test instances in a test set.
* **Test Execution:** Enables testers to execute the test/test set.
* **Test Results:** This Functionality enables testers to analyze the test results.

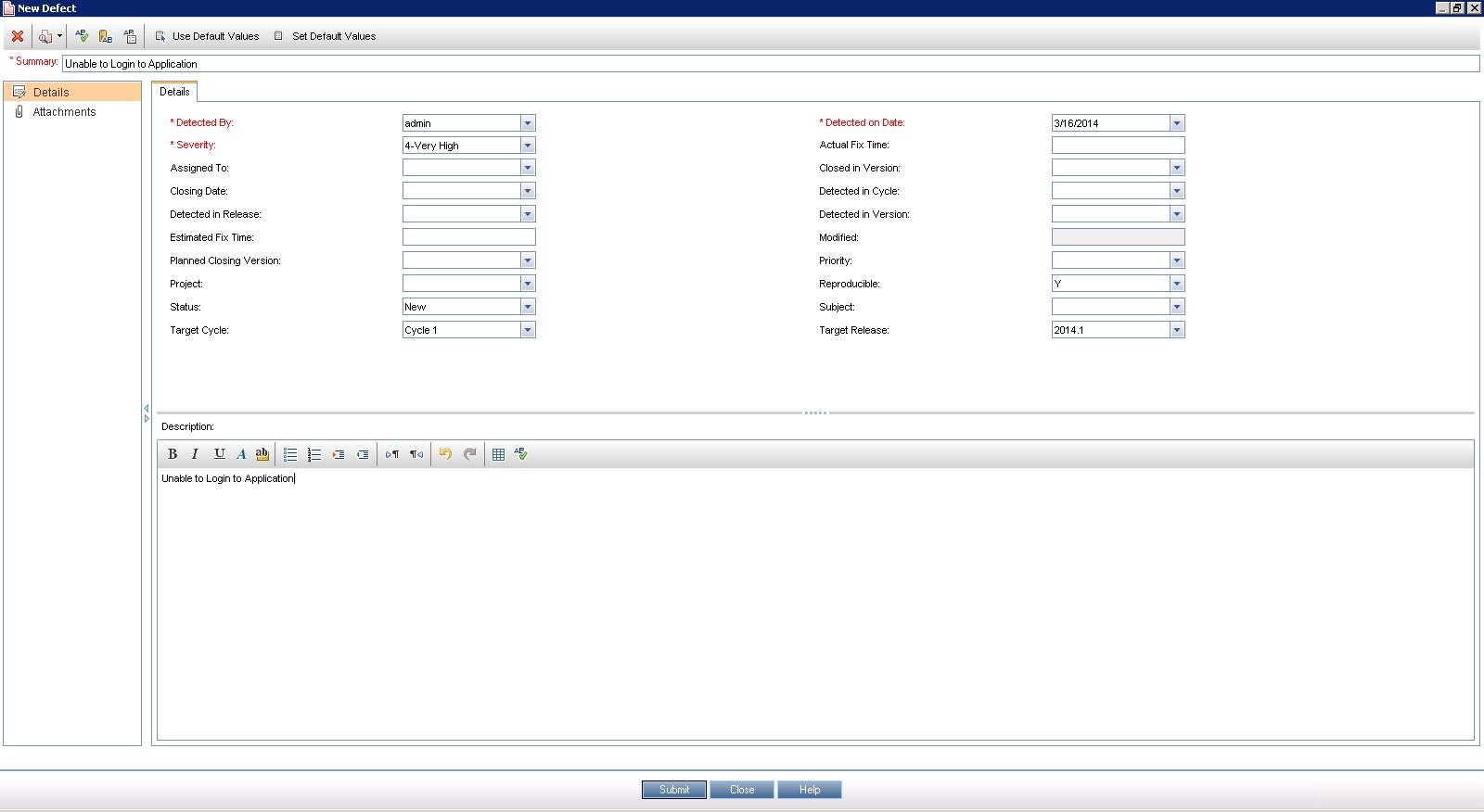
**Defects:**

During test execution, when expected result doesnot match with actual result, a defect should be logged. Now Let us focus various functionalities associated with defects.

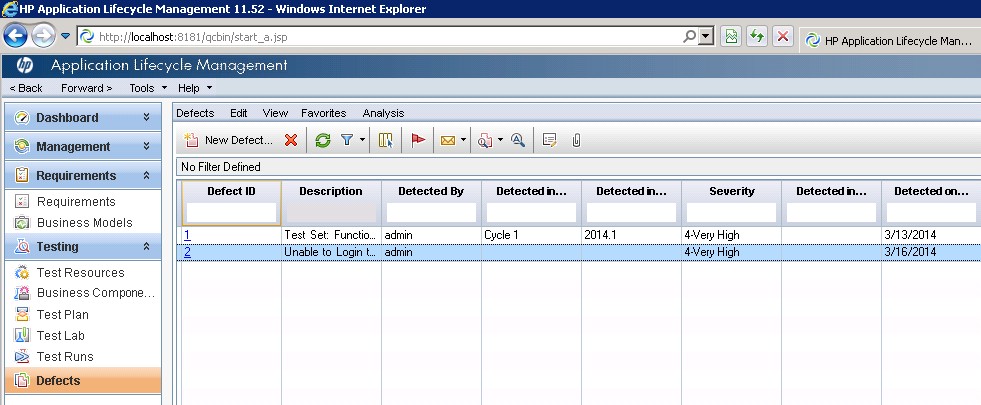
**Navigate to defects Tab in Quality Center and Click on "New Defects".**



**Fill in the mandatory parameters such as defect summary, detected by, Severity and Description.**

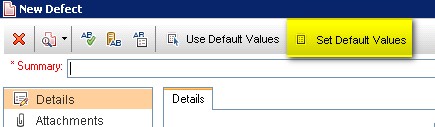


**Once a defect is posted, the same be accessed in Defects Tab as shown below.**

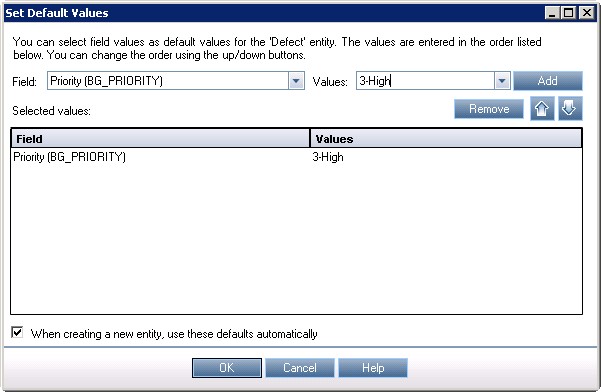


If the defects are logged during the test execution the step and description would be auto populated from test case where as If a defect is logged by clicking on "New Defect" in defect module the steps, description and summary needs to be entered manually by the user.

Many a times testers will NOT be in a position to enter all field values in "New Defect" Window. Instead they can make use of "Set Default Values" Option. It can be accessed from "New Defect" Window as shown below.



The "Set Default Values" dialog opens. We will be able to set a default value for each one of the fields in "Defects" Section which would be prepopulated once we click on "New Defect" button. We can also add/remove Default values using the appropriate button in this dialog box.



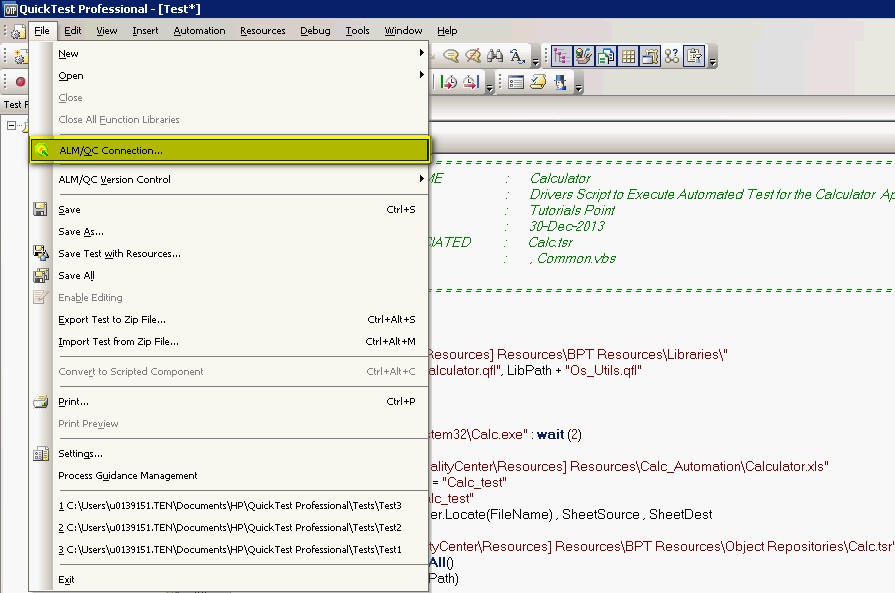
Defects can also be uploaded from Excel to Quality Center and the procedure remains the same as that of uploading "Requirements" and "Test Cases" using HP-MS Excel plugin utility.

All the defects are displayed by default, hence user has to use filter functionalities to display only those that are filtered by user.

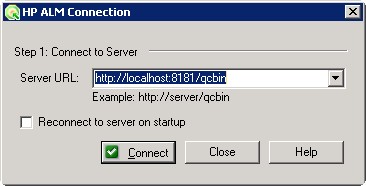
**QTP-QC Integration:**

By Integrating, the QTP automation scripts can be executed right from Quality Center

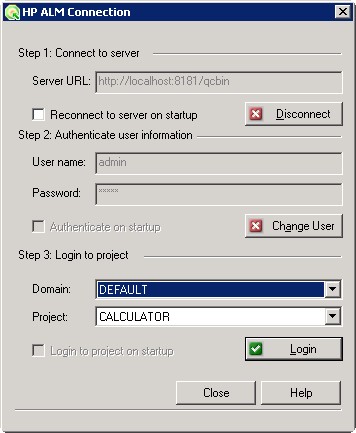
**Step 1**: Establish a connection between QTP and QC by Navigating to "File" >> "ALM/QC Connection".



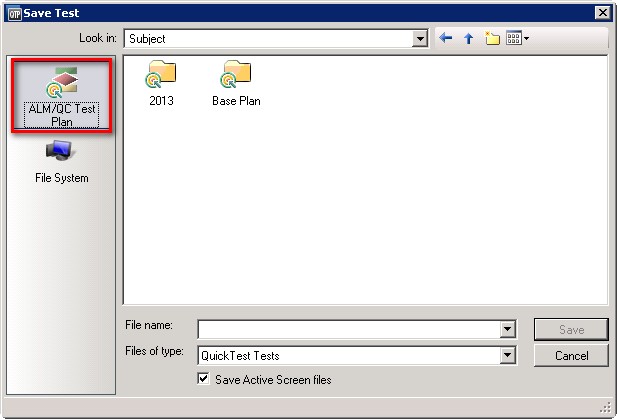
**Step 2** : HP ALM Connection dialog opens. Enter the Server URL as shown



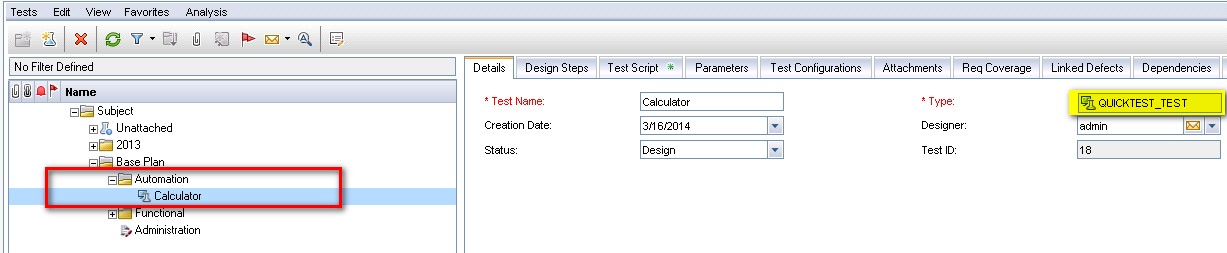
**Step 3** : Once the URL is correct, the credentials dialog opens. Tester has to connect to the relevant project area.



**Step 4** : Once ALM Connection is successful, we need to save the Driver Script in Quality Center. Select ALM/QC Plan as shown below.



**Step 5** : Once the test is saved, we can see the in ALM where the test type is shown as "QuickTest".



**VB Script**

**VBScript** (**Visual Basic Scripting Edition**) is an [Active Scripting](http://en.wikipedia.org/wiki/Active_Scripting) language developed by [Microsoft](http://en.wikipedia.org/wiki/Microsoft) that is modeled on [Visual Basic](http://en.wikipedia.org/wiki/Visual_Basic). It is designed as a "lightweight" language with a fast interpreter for use in a wide variety of Microsoft environments. VBScript uses the [Component Object Model](http://en.wikipedia.org/wiki/Component_Object_Model) to access elements of the environment within which it is running; for example, the FileSystemObject (FSO) is used to [create, read, update and delete](http://en.wikipedia.org/wiki/Create,_read,_update_and_delete) [files](http://en.wikipedia.org/wiki/Computer_file).

Nowadays, VBScript is the primary scripting language for Quick Test Professional (QTP), which is a test automation tool.

**VB**Script stands for **V**isual **B**asic Scripting that forms a subset of Visual Basic for Applications (VBA).

VBA is a product of Microsoft which is included NOT only in other Microsoft products such as MS Project and MS Office but also in Third Party tools such as AUTO CAD.

**Features of VBScript**

* VBScript is a lightweight scripting language, which has a lightning fast interpreter.
* VBScript, for the most part, is case insensitive. It has a very simple syntax, easy to learn and to implement.
* Unlike C++ or Java, VBScript is an object-based scripting language and NOT an Object-Oriented Programming language.
* It uses Component Object Model **(COM)** in order to access the elements of the environment in which it is executing.
* Successful execution of VBScript can happen only if it is executed in Host Environment such as Internet Explorer **(IE)**, Internet Information Services **(IIS)** and Windows Scripting Host **(WSH)**

**VBscript – Version History and Uses**

VBScript was introduced by Microsoft way back in 1996 and the first version was 1.0. The Current Stable version of VBScript is 5.8, which is available as part of IE8 or Windows 7.

The VBScript usage areas are aplenty and not restricted to the below list.

* VBScript is used as a scripting language in one of the popular Automation testing tools – Quick Test Professional abbreviated as **QTP**
* Windows Scripting Host, which is used mostly by Windows System administrators for automating the Windows Desktop.
* VBScript is used for Client side scripting in Microsoft Internet Explorer.

**How it Works?**

VBScript talks to host applications using Windows Script. With Windows Script, browsers and other host applications do not require special integration code for each scripting component. Windows Script enables a host to compile scripts, obtain and call entry points, and manage the namespace available to the developer. With Windows Script, language vendors can create standard language run times for scripting. Microsoft will provide run-time support for VBScript. Microsoft is working with various Internet groups to define the Windows Script standard so that scripting engines can be interchangeable. Windows Script is used in Microsoft® Internet Explorer and in Microsoft® Internet Information Service.

**Disadvantages**

* VBscript is used only by IE Browsers. Other browsers such as Chrome, Firefox DONOT Support VBScript. Hence, JavaScript is preferred over VBScript.
* VBScript has a Limited command line support.

**Syntax Example:**

Dim objFSO, objFSOText, objFolder, objFile  
Dim strDirectory, strFile  
strDirectory = "E:\Desktop\folder1"  
strFile = "\Summer.txt"  
  
' Create the File System Object  
Set objFSO = CreateObject("Scripting.FileSystemObject")  
  
' Create the Folder specified by strDirectory on line 10  
Set objFolder = objFSO.CreateFolder(strDirectory)  
  
' -- The heart of the create file script   
'-----------------------  
'Creates the file using the value of strFile   
' -----------------------------------------------  
Set objFile = objFSO.CreateTextFile(strDirectory & strFile)  
Wscript.Echo "Just created " & strDirectory & strFile  
  
Wscript.Quit  
  
' End of FileSystemObject example

**What is Disaster Recovery?**

**Disaster recovery** (DR) involves a set of policies and procedures to enable the recovery or continuation of vital technology infrastructure and systems following a [natural](http://en.wikipedia.org/wiki/Natural_disaster) or [human-induced](http://en.wikipedia.org/wiki/Man-made_hazards) [disaster](http://en.wikipedia.org/wiki/Disaster). Disaster recovery focuses on the IT or [technology systems](http://en.wikipedia.org/wiki/Technology_systems) supporting critical business functions as opposed to [business continuity](http://en.wikipedia.org/wiki/Business_continuity), which involves keeping all essential aspects of a business functioning despite significant disruptive events. Disaster recovery is therefore a subset of [business continuity](http://en.wikipedia.org/wiki/Business_continuity).

During the 1980s and 90s, customer awareness and industry both grew rapidly, driven by the advent of open systems and [real-time processing](http://en.wikipedia.org/wiki/Real-time_computing) which increased the dependence of organizations on their IT systems. Regulations mandating business continuity and disaster recovery plans for organizations in various sectors of the economy, imposed by the authorities and by business partners, increased the demand and led to the availability of commercial disaster recovery services, including mobile data centers delivered to a suitable recovery location by truck.

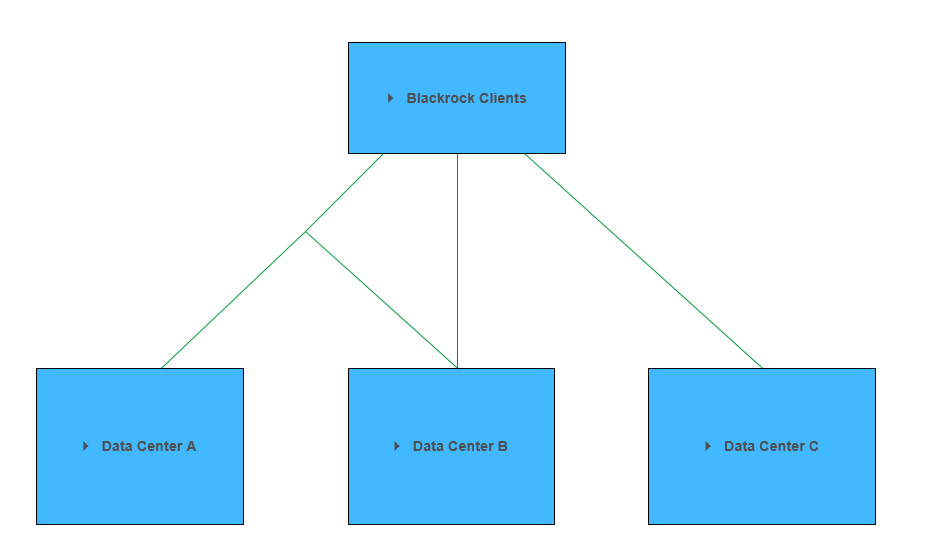
With the rapid growth of the [Internet](http://en.wikipedia.org/wiki/Internet) through the late 1990s and into the 2000s, organizations of all sizes became further dependent on the continuous [availability](http://en.wikipedia.org/wiki/Availability) of their IT systems, with some organizations setting objectives of 2, 3, 4 or 5 nines (99.999%) availability of critical systems. This increasing dependence on IT systems, as well as increased awareness from large-scale disasters such as tsunami, earthquake, flood, and volcanic eruption, spawned disaster recovery-related products and services, ranging from [high-availability](http://en.wikipedia.org/wiki/High-availability) solutions to [hot-site](http://en.wikipedia.org/wiki/Hot-site) facilities. Improved networking meant critical IT services could be served remotely, hence on-site recovery became less important.

The meteoric rise of cloud computing since 2010 continues that trend: nowadays, it matters even less where computing services are physically served, just so long as the network itself is sufficiently reliable (a separate issue, and less of a concern since modern networks are highly resilient by design).

### Classification of disasters

Disasters can be classified into two broad categories. The first is natural disasters such as floods, hurricanes, tornadoes or earthquakes. While preventing a natural disaster is very difficult, risk management measures such as avoiding disaster-prone situations and good planning can help. The second category is man-made disasters, such as hazardous material spills, infrastructure failure, bio-terrorism, and disastrous IT bugs or failed change implementations. In these instances, surveillance, testing and mitigation planning are invaluable.

**DR CHECKOUT AT BLACKROCK**



**Automated DR Process**

Blackrock has developed applications which are used within Blackrock as well as by Clients. These Client Environments are aligned to different Data Centers. **Disaster Recovery** is a **business critical process** for uninterrupted service to our clients. DR Checkout is an automated process which the APG-QA team carries out to ensure that in case of any unforeseen disaster or unexpected error that occurs at any one data center, there should be a reliable switch of connection of all clients of that data center to either of the other data centers. This process has been automated using Quick Test Professional and Application Lifecycle Management which are the most important tools for any testing process. According to the usage of every application by the client, test scripts have been developed. At the end of DR Execution, the test results are analyzed and the steps that failed on the client environment are manually tested. This process is carried out on regular intervals for all clients.

The main purpose of this project is to reduce the time invested in performing testing manually by automation.

Before the introduction of automation the actual time which was required by this process was about 6 to 8 hours but after implementation of the automated scenarios the time will be reduced to 30 minutes which is about **92.57% less**.

In addition to this it will also help to increase the ROI incurred in the automation task and further reduce the man hours required to perform DR and thereby saving cost to company.

The Test Scripts are stored in ALM and triggered onto remote machines. With the use QC-QTP Integration, these scripts run on QTP automatically and generate results which are later analyzed for success and failure percentages.

**Virtual Machine(VM) Management**

The DR is executed on the software VNC (Virtual Network Computing) Viewer, which is similar to the Remote Desktop supported by Windows. The virtual machines we use for this process, need to be in logged-in and unlocked state. We have created batch files to automate the log-in into VM process which will be run at the backend, before we actually execute the DR on those virtual machines. Also, the remote machines we use require a VNC Server to be installed and a VNC Client installed on the local machine.

## 

A.L.M.

Test Lab

QC-QTP Connection

VM+QTP

Results

Analysis

**Failed Scripts**

Manually Executed

**Testing Workflow**

**Project Schedule**

|  |  |  |
| --- | --- | --- |
| **Task Name** | **Start Date** | **End Date** |
| ALM Training | 5-Feb-15 | 9-Feb-15 |
| QTP Training | 9-Feb-15 | 13-Feb-15 |
| Framework Walkthrough | 13-Feb-15 | 16-Feb-15 |
| Understanding and Highlight Improvement Areas | 16-Feb-15 | 30-Mar-15 |
| Automate VM Login | 30-Mar-15 | 15-May-15 |
| Project Completion | 16-May-15 | 16-Jun-15 |

**CONCLUSION and FUTURE SCOPE of WORK**

Disaster Recovery being an essential process for any organization, this project will contribute towards business continuity. The automated DR Process will reduce a lot of manual effort and ensure uninterrupted service to our clients. Also, the project has given me a platform to enhance my technical as well as problem-solving skills.

Following the completion of this project, I will be able to work in different functional as well as automation teams with the knowledge of testing workflow, QTP the automation tool as well as ALM the test management tool. With an insight into the finance domain, I have got an exposure to a completely different horizon which will help me in my professional journey.

**REFERENCES**

* [www.blackrock.com](http://www.blackrock.com)
* [www.w3schools.com](http://www.w3schools.com)
* [www.tutorialspoint.com](http://www.tutorialspoint.com)
* [www.wikipedia.org](http://www.wikipedia.org)