SAP TESTING SERVICE & OFFERING

By

S.VINOTH KUMAR
Registration Number: 71205621059

of

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE

A PROJECT REPORT
Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements
for the award of the degree

of

MASTER OF COMPUTER APPLICATION

JUNE 2008

KUMARAGURU COLLEGE OF TECHNOLOGY

Coimbatore-641006

DEPARTMENT OF COMPUTER APPLICATION

Bonafide Certificate

Certified that this project report titled **SAP TESTING SERVICE & OFFERING** is the bonafide work of **Mr. S.VINOTH KUMAR (Registration Number: 71205621059)** who carried out the research under my supervision. Certified further, that to the best of my Knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

N. Jayakanthan.
Supervisor

Head of the Department

Submitted to Project and Viva Examination held on 1-07-2008

1/7/08
Internal Examiner

External Examiner 1/7/08

patni

Patni Computer Systems Ltd.
55, SDF II, Seepz,
Andheri (E),
Mumbai - 400 096, India
TEL : +91 22 2829 1454
FAX : +91 22 2829 2764
www.patni.com

March 31, 2008

To whomsoever it may concern

This is to certify that Mr. Vinothkumar Subramaniam as partial fulfillment of his MCA , from Kumaraguru College of Technology , has completed his project for Patni under the guidance of Ms.Kavita Ramchandani. He has worked on "SAP TESTING SERVICE & OFFERING". The project traineeship duration has been from 24/12/2007 to 23/6/2008.

Vinothkumar has been very sincere & diligent in his job and his performance on the project has been acceptable.

We wish him a bright career and success in all his future endeavors.

For Patni Computer Systems Ltd.

Pradnya Naidu
Asst. Manager - GRiTHR

ABSTRACT

The project entitled "SAP Testing Service & Offering" is carried out to create a central repository of reusable components for the verification and validation team in SAP for the Patni computer System Ltd, Mumbai. This project is carried out to create reusable components that help the testing team to create test cases for projects. Its main objective is to save the time and effort by eliminating the creation of similar kind of test cases by the testing team which already exists.

This project is developed for the testing purpose in a software company. The Sap applications are processed with the transaction codes for all the process area which are universally common. The test cases are prepared for each process area for different projects based on these transaction codes. The chance of creating similar kind of test cases for new projects are high since its hard to remember those test cases and retrieve them.

This repository is developed to accumulate all the test cases in central place and segregate them according based on transaction codes. Create scenario for those integrated test cases and identify the flow of process. We can identify the components from the flow and automate them through scripts so that when a new project is allotted the testing team can use these components with some customization according to the requirements.

ACKNOWLEDGEMENT

I would like to take this opportunity to say thank you to the people who have helped to make this project.

First I would like to express my heartfelt thanks to our principal **Dr.Joseph V.Thanikal,** B.E., M.E., Ph.D., PDF., CEPIT., for having given me the opportunity to do this project .

I express gratitude and thanks to **Dr.S.Thangasamy**, Ph.D., Dean, Department of Computer Science, for providing moral support towards this project work.

I express my deep sense of gratitude and thanks to **Dr.M.Gururajan**, Ph.D., HOD, Department of Computer Application, for providing moral support towards this project work.

I am indebted to **MR. Hameed Ibrahim**, MCA., Senior Lecturer, Department of Computer Application, for his constant encouragement and support for completing the project on time as schedule.

I sincerely express my humble gratitude to **MR N.Jeyakanthan**, M.C.A., Lecturer, Department of Computer Application, who has been my guide with valuable and holistic suggestions and extended kind of operation and encouragement.

I also express my gratitude and thanks to **MS. Kavita Ramchandani**, Project Leader, Patni Computer System, who has been my guide and gave valuable suggestions and encouragement.

I acknowledge my hearty thanks to all my beloved friends on their valuable co-operation in the proceeding of my work. I wish to credit my special accordance to my parents for their encouragement and prayers to have a successful project.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 SYSTEM OVERVIEW

The project titled "**SAP TESTING SERVICE & OFFERING**" provides development of central repository with reusable components.

SAP is the leading Enterprise Information and Management Package worldwide. SAP idea was to provide customers with the ability to interact with a common corporate database for a comprehensive range of applications. SAP applications, built around their latest R/3 system, provide the capability to manage financial, asset, and cost accounting, production operations and materials, personnel, plants, human resource, and archived documents.

The need for the SAP testing service & offering was felt due to the nature of the work required, which involved creation of test cases for the existing and upcoming sap based projects in the software industry. Since there are no properly documented scenario the testing team had to prepare the similar test cases for different projects which are time consuming and repetitive process.

Automation is the key to increase efficiency, performance and reliability in any system and is being rapidly followed in all walks of life. The problem with the traditional approach of creating test cases for each of the project there is always the possibility of wastage of time and very hard to remain which test cases are similar and available. Automation helps eradicate the root cause and time wastage by completely automating the system and providing the reusable components.

These reusable components in the central repository can be utilized by the testing team for the new projects by slightly customizing them according to their requirement. These reusable components in the central repository cover all the process areas of sap so that any requirement in the projects can be processed with the help of the central repository.

The development of central repository involves creation of reusable components for the sap application projects, exporting and importing the available test cases to the central repository with the help of utility and performing the automation.

The creation of components and test cases involves understanding the sap domain and the process areas. There are six major process areas in the sap which includes some manufacturing, quality management, supply chain management, finance, proposal to revenue, proposal to pay. Separate set of test cases are needed to be created for each process areas in sap application. The test cases are created in the excel sheets by running the application for each process areas separately.

After creation of the test cases they should be uploaded in the repository for further processing. A utility was developed which get the connection to the repository and the excel files. After specifying the required fields it reads the text up to the end of the excel files and upload them in to the repository in the appropriate places.

The testing team may require this test case from the repository for reference if there is a new project in progress. Another utility is developed to import the required test cases from the repository and save them in the local area where the user specified. The process area which the user wants to import can be specified in the utility.

Running tests is the core of the testing process. We run manual tests in the project to locate defects and assess quality. We create test sets and choose which tests to include in each set. A test set is a group of tests in a quality center project designed to achieve specific testing goals.

1.2 COMPANY PROFILE

Patni Computer Systems Limited is a global IT Services provider with revenues of US $ 662.9 million for the year 2007. Founded in 1978 and headquartered in Mumbai – India. Patni employs over 10,000 people and has 26 sales offices across Americas, Europe and Asia-Pacific.

Patni has over two decades of experience in the IT industry with vast experience in system and application development projects, across all major software platforms and environments. Patni has major software competency centers throughout India, with its employees employed on onsite as well as offshore projects. Patni spends a significant part of its revenue on training, ensuring that its employees are constantly updated on new technologies and skills. Committed to quality, Patni adds value to client business through well established and structured methodologies, tools and techniques backed by Six Sigma processes. Patni is ISO 9001:2000 certified and is also assessed at SEI-CMMI Level 5 and P-CMM Level 3.

INDUSTRY SOLUTIONS

Insurance:

Patni is the premier total IT solution provider for the insurance industry, with a solution set that encompasses application development and maintenance, business process consulting, and business process outsourcing. We are able to execute on this vision because of our unparalleled insurance domain expertise, business-driven solutions and implementations, mature global delivery systems, and strong relationships with complementary partners.

Financial Services:

Patni's long-term strategic relationships with leading Financial Services and other organizations have given us an in-depth understanding of today's market forces. Patni provides integrated solutions for the complex business challenges faced by the entire financial services industry including: investment managers, brokerages, financial exchanges, banks, and providers of credit, leasing, and payment services.

Manufacturing:

Patni has decades of experience in servicing Manufacturing clients across a range of industries spanning the industrial, consumer goods and transportation sectors Patni's offerings encompass the entire spectrum of enterprise services viz. Enterprise Resource Planning, Supply Chain Management, Customer Relationship Management and Enterprise Integration. Our focus on RFID based solutions as well as new dimension products make us a valuable partner for a Manufacturing enterprise.

Telecom:

Patni's Telecom Business Unit (TBU) is recognized as a global leader in providing IT services to the telecom industry. The TBU combines deep domain expertise in OSS and BSS systems, intellectual leadership and a global workforce advantage to provide services to leading clients in every segment of the worldwide telecom market, especially communication service providers in mobile, wire line, cable, broadband and media .

Product Engineering:

Patni's Product Engineering Services (PES) practice caters to R&D and complete end-to-end product realization services for Hardware Chipset Vendors, Embedded Software Tool Vendors and companies from industry verticals like Automotive Electronics, Consumer Electronics, Medical Electronics, Industrial Automation, Information Storage, Mobile & Wireless, Office Automation, and Semiconductors.

Energy & Utilities:

Patni provides solutions to several leading energy companies, utility providers and focused ISVs. Our expertise areas include field services, work force management, billing and tariff management, usage measurement, metering systems and customer services.

CHAPTER 2

SYSTEM STUDY AND ANALYSIS

2.1 PROBLEM STATEMENT

In the existing practice, creating test cases for a new project without any properly documented scenarios or diagram and implementing them is time consuming process. Some clients do not provide a detailed document about requirements which may create problem in understanding and may some time lead to irrelevant results.

The testing team often prepares test cases which are sometimes similar to each other for different projects. The tasks involved in the generation of these test cases whenever a new project is assigned is repetitive. Often, due to the repetitive nature of the work and the tedious process involved, the testing team suffers by the time constraint laid for each new project.

These test cases have to be segregated according to their transaction codes for identifying business components. There was a need of creating a central repository arises where all the test cases of six process areas of sap from different projects can be managed in a central place to create reusable components.

After identification they have to be automated by writing script to make them as a reusable component. The testing team can use these components by slightly customizing them according to their requirements in the project without developing test cases. This script can be written with the help of testing tools.

2.2 EXISTING SYSTEM

The test cases for sap process areas for most of the project are prepared by using the transaction codes of sap to perform each process in the application and recorded in excel sheets and for some projects they are prepared in word files. These two different types of test cases are organized in separate ways from each other.

Since the test cases are not organized ,when a new project is assigned the testing team start form the initial part to develop the test cases based on the projects even most of the test cases may be similar to the previously developed test cases as its difficult task to search from huge collection .

After designing the test cases they are grouped in to test set since there may be more than one test case for each flow.Test sets can include both manual and automated tests in a testing tool. Perform operations on the application under test and then actual results with the expected results are compared.

After comparison assign a pass or fail status to each step, depending on whether or not the expected results match the actual results. If a step fails, how the application actually responded is noted. A test fails if one or more steps fail in the process testing.

2.2.1 DRAWBACKS OF THE EXISTING SYSTEM

The drawbacks of the existing system can be summarized as below:

- Increase in redundancy of work
- Time Consuming.
- Scenario for the process are not available
- No Segregation of test cases based on transaction codes.
- To search particular test cases form process is difficult.
- Difficult to integrate the similar test cases.

2.3 PROPOSED SYSTEM

The proposed system would automate the processes of creating the test cases described which would help reduce the overhead incurred by the testing team and make the whole process simple and efficient. The new system has effective and simple way of testing the sap application.

The proposed system will have scenario level flow for each process area in the repository that is derived from the segregation of test cases based on the identical transaction codes to identify the component. Repository also includes utility which will convert the test cases that are designed in the word file to the excel file format and to export and import them from the repository.

The identified components are then automated with script to design them as business components. These components can be used with slight customization in the future projects. Proposed system has been designed to eliminate the major disadvantage of the existing system.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

The expected benefits of the proposed system are as follows:

- Requirement analysis and understanding
- Scenario Scripting
- Scenario Recording (Automation)
- Defect analysis and resolution
- Need less number of resources
- Test cases for most of the processes are present
- Continuous enhancement of the repository
- Saving in the billing Hours.
- Provides Accuracy and Precision.
- Adapt to new processes in future for
- Enhancing and better ROI.

2.4 FEASIBILITY ANALYSIS

Feasibility analysis is the measure of how beneficial or practical the development of the System will be to the project. Once the problem is explained information is gathered about the system to test whether the system is viable Technically, Financially and Operationally. Thus, feasibility study is carried out in three phases as follows:

2.4.1 TECHNICAL FEASIBILITY

Technical Feasibility is the measure of practicality of a specific technical solution and the availability of technical resources and expertise. It centers on the existing computer system (hardware, software, etc.) and to what extent it can support the new addition. This involves financial consideration to accommodate technical enhancement.

The proposed system is to be developed using VB.net 2005 and Quality Center which are some of the leading technologies in the market. These technological resources are easily available and the company/project does not need to acquire any development licenses. Visual studio .NET 2005 and Quality Center are already available with the company. These technologies work well on Microsoft platforms. When take the project size, it's very small. At present, the system is intranet environment. Future expansion is planned but will not affect this project.

2.4.2 OPERATIONAL FEASIBILITY

Operational Feasibility asks if the system will work when it is developed and installed. It checks for the support of the management, the current business methods, user's involvement and their attitude towards the proposed system, etc.

The proposed system has found encouraging support from the employees, team leads, managers and the top management as it will be of great use to them. The team leads and managers of the project are also committed to have the system operational as it will save time and reduce their workload.

Also since the testing teams can make a great use of these reusable components and scenario to design the test cases for the upcoming projects in future, they are very much in favor of implementing the system. The current processes followed in the project would be depicted in the system as it is.

2.4.3 ECONOMIC FEASIBILITY

Economic Feasibility is the measure of the cost-effectiveness of the proposed system. The investment to be made in the proposed system must prove a good investment to the project by returning benefits equal to or exceeding the costs incurred in developing the system.

The proposed benefits of the system will outweigh the costs to be incurred during system developed since the system does not require procurement of additional hardware facilities it is economically feasible. It uses VB.Net 2005 and Quality Center for its development. So it's found that the benefits outweigh costs. In addition capability of the system to incorporate future enhancement will improve the performance to suit the future need of the company/project.

2.5 USERS OF THE SYSTEM

The users of the proposed system have been categorized as below and each of the user categories will have a set of rights which manage their use of the proposed system.

- Testing team
- Team Leads/Managers

Testing team is authorized to the following task:

- Logging to the authorized central repository.
- Perform the task such as planning test, running test, tracking defects.
- Authorized to use the utility for processing.
- Performing Automation.
- Creating reports and graphs by monitoring testing process
- Exporting and importing test cases from repository.

Team leads/Manager is authorized to the following task:

- Team Leads/Managers have full authorization over the repository
- Specify the release of project in quality center.
- Specifying requirements of what needed to be tested in application.
- Changing the severity of a defect.
- Assigning the defect to cycle.
- Authorization for creating and deletion of process area.
- Determining the authority for the user

CHAPTER 3

DEVELOPMENT ENVIRONMENT

3.1 HARDWARE REQUIREMENTS

The hardware support required for deploying the application

Server Configuration

Processor	: Intel Pentium IV Processor / Athlon Processor
Memory	: Minimum 512MB
Hard Disc	: 40GB or More

Client Configuration

Processor	: Intel Pentium III Processor / Athlon Processor
Memory	: Minimum 256MB
Hard Disc	: 20GB or More
CDROM	: 52X-MAX
Printer	: Inkjet Printer

3.2 SOFTWARE REQUIREMENTS

The software support required for deploying the application

Operating System	: Windows XP
Software used	: VB.NET 2005
Back End Tool	: Quality Center

3.3 PROGRAMMING ENVIRONMENT

3.3.1 THE .NET FRAMEWORK

The .NET frame work is a new computing platform that simplifies application development in the highly distributed environment of the internet. To avoid separate runtime environment called the Common Language Runtime (CLR).

OBJECTIVES OF .NET FRAMEWORK

- To provide a consistent Object-oriented programming environment whether object codes is stored and executed locally and internet distributed, or executed remotely.
- To provide a code-execution environment to minimizes software deployment and guarantees safe execution of code.
- Eliminates the performance problems.
- There are different types of application, such as windows-based application and web-based applications.
- To make communication on distributed environment to ensure that code be accessed by the .NET Framework can integrate with any other code.

COMPONENTS OF .NET FRAMEWORK

THE COMMON LANGUAGE RUN TIME (CLR)

The common language runtime is the foundation of the .NET Framework. It manages code at execution time, providing important services such as memory management, and remoting and also ensures more security and robustness. The concept of code management is a fundamental principle of the runtime code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code.

THE .NET FRAMEWORK CLASS LIBRARY

It is a comprehensive, object-oriented collection of reusable type used to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

The .NET Frame work can be hosted by unmanaged components that load the common language runtime into their processes and initiate the execution of managed code, there by creating a software environment that can exploit both managed and managed features. The .NET Framework not only provides several runtime hosts, but also supports the development of third-party runtime hosts.

Internet explorer is an example of unmanaged application that hosts the runtime (in the form of a MIME type extensions).using Internet Explorer to hosts the runtime to enables embeds managed components or windows forms controls in HTML documents.

FEATURES OF THE COMMAN LANGUAGE RUN TIME

The common language run time include manager memory; thread execution, code execution, code safety verification, compilation, and other system devices these are all run on CLR. The features are

- Security
- Robustness
- Productivity
- Performance

SECURITY

The run time enforces code access security. The security features of the run time thus enable legitimate internet deployed software to be exceptionally features rich. With regards to security, managed components are awarded varying degrees of trust, depending on a numbers of factors that include their origin to perform file access operations, registry access operations, or other sensitive functions.

ROBUSTNESS

The runtime enforces code robustness by implementing a strict type and code verification infrastructure called the common type system (CTS). The CTS ensures that all managed code is self describing. The managed environment of the run time eliminates many common software issues.

PRODUCTIVITY

The run time also accelerates developer productivity. For example, programmers can write applications in the development language of choice, yet take full advantage of the run time, the class library, and components written in other languages by other developers.

PERFORMANCE

The run time is designed to enhance performance. Although the common language run time provides many standard run tine services, managed code is never interpreted. A feature is called just in time (JIT) compiling enables all managed code to run in the negative machine language of the system on which it is executing. Finally, the run time can be hosted by high performance, server side applications, such as Microsoft SQL sever and internet information services (IIS).

VB.NET 2005

VB.NET 2005 comes with a number of enhancements. The IntelliSense Code snippets, the Windows Forms designer updates, IntelliSense filtering, debugger data tips, exception Assistant etc make the software a pleasure to work with. The language has been spruced up with generics, unsigned types, Operator overloading etc.

The My Namespace is the most significant enhancement that provides a single reference to commonly used functionalities within the .NET framework. It includes classes like Application, Computer, Forms, Resources, Settings and Users. This enables users to ping a computer with a simple line of code or play a audio file with a one line code.

VB.NET FEATURES

- VB.Net is object oriented
- Multithreading is possible in a better way in VB.Net
- .Net supports ASP.Net
- A new concept , Name Space, New Operators, Just My Code, Support for Generics is added
- Edit and Continue - probably the biggest "missing feature" from Visual Basic. .NET, allowing the modification of code and immediate resumption of execution
- VB.Net is platform independent
- Data types are considered as objects
- Data Source binding, easing database client/server development
- Structure error handling is possible
- Unary operators are available
- Arguments are passed as ByVal by default
- Instead of COM components there are .Net components in VB.Net

3.3.2QUALITY CENTER

Quality Center offers an organized framework for testing applications before they are deployed. Because test plans evolve with new or modified application requirements, you need a central data repository for organizing and managing the testing process. A Quality Center project is a database for collecting and storing data relevant to a testing process.

Quality Center guides you through all phases of the testing process an intuitive and efficient method for scheduling and running tests, collecting results, analyzing the results, and managing test versions. It also features a system for tracking defects, enabling you to monitor defects closely from initial detection until resolution.

QUALITY CENTER TESTING PROCESS

The Quality Center testing process includes five phases architecture which includes

- Specifying Releases.
- Specifying Requirements.
- Planning Tests.
- Running Tests.
- Tracking Defects.

SPECIFYING RELEASE

The testing process in the quality center begins by specifying releases in the Releases module. A release represents a group of changes in one or more applications that will be available for distribution at the same time. The Releases module enables you to define releases and cycles for managing the testing process. Each release can contain a number of cycles.

A cycle represents a development and QA cycle based on the project timeline. The releases and cycles have defined start and end. After creating requirements in the

Requirements module, assign them to releases and cycles and later plan tests to test these requirements and run them as part of test sets, belonging to test set folders. Each test set folder is assigned to a cycle. When logging defects we can assign them to specific cycles. In this way can keep track of the progress and quality of the release.

To specify the release plan, consider these steps:

- Decide how to break up your application releases.
- For each release, determine how to break it down into cycles.
- Analyze the progress and quality of your releases and cycles.

VIEWING RELEASE AND CYCLES

We can view graphs and statistics to display the progress and quality of releases and cycles. To view make sure the release module is displayed in the quality center if not click the Releases button on the sidebar.

- ➢ The progress tab displays the progress of the release based on requirement coverage, elapsed and remaining time, actual and remaining test instances to run.
- ➢ In the quality tab the defect opening rate graph summarizes the number of defects detected in the selected release. Defects are categorized according to their severity. The current status of the defects is not considered.
- ➢ In the quality tab the outstanding defects graph summarizes the number of outstanding defects for the selected release. These defects are categorized according to their severity.
- ➢ The user can view the progress graph and the quality graph together for any specific cycle or release by selecting the quality tab.
- ➢ Releases and cycles each have start dates and end dates. The date range for a cycle must be contained within the date range for the release to which it belongs.

SPECIFYING REQUIRMENTS

After defining releases and cycles, specify requirements in the requirement module. Requirements describe in detail what needs to be tested in the application and provide the test team with the foundation on which the entire testing process is based.

DEFINING REQUIREMENTS

Define the requirements in Quality Center by creating a requirements tree. This is a graphical representation of the requirements specification, displaying the requirements in a hierarchical way in the module. We can also group and then sort requirements in tree, and monitor the task allocation, monitor the progress in meeting requirements, and generate the detailed reports and graphs for projects.

After creating tests in the Test Plan module, we can link requirements to tests. Then we can log defects, and also link requirements to defects in this way, we can keep track of your testing needs at all stages of the testing process. If requirement changes, can quickly identify which tests and defects are affected, and who is responsible for them.

VIEWING REQUIREMENTS

We can change the way requirements are displayed. Display the requirements in the requirements grid in a flat nonhierarchical view. Each line in the grid is displayed as separate requirement. We can set filter condition that corresponds to creation date for displaying the requirement.

CONVERTING REQUIREMENTS

After you create the requirements tree, you use the requirements as a basis for defining your test plan tree in the Test Plan module. You can use the Convert to Tests wizard to assist you when designing your test plan tree. The wizard enables you to convert selected requirements or all requirements in the requirements tree to subjects or tests in the test plan tree.

PLANNING TEST

After defining the requirements, determine the testing goals to be achieved .It can be performed by examining the application, system environment, and testing process to outline the testing strategy for achieving the goals.

DESIGNING TEST STEP

After determining the goals test plan tree is build, which hierarchically divides your application into testing units, or subjects. For each subject in the test plan tree, define tests that contain steps. For each test step, you specify the actions to be performed on your application and the expected result. In addition to creating a test plan tree directly in Quality Center, you can also import test plan data from microsoft excel.

CALLING TEST WITH PARAMETERS

We can increase the flexibility of a test step by adding parameters. To keep track of the relationship between your tests and the requirements defined, we can add links between them. This can help to ensure compliance with the requirements throughout the testing process for the application

When you design test steps, you can include a call to a manual test. When you run the test, the test steps include the steps from the called test as part of the test. The test that you call is a template test. This is a reusable test that can be called by other tests.

A template test can include parameters. A parameter is a variable that replaces a fixed value. You can modify the value of a parameter according to the test that is calling it, or for various instances of the same test. Suppose you have a test that logs in a user with a specific password when you start your application. You need to call this test at the beginning of each test.

REQUIREMENTS COVERAGE

In the Test Plan module, you create requirements coverage by selecting requirements to link to a test. Alternatively, in the Requirements module, you create tests coverage by selecting tests to link to a requirement. A test can cover more than one requirement, and a requirement can be covered by more than one test.

To further ensure compliance with your testing requirements, after you log defects, you can link your requirements and tests to defects. This ensures that if a requirement changes, you can identify which tests and defects are affected, and who is responsible for them.

ANALYZING TEST COVERAGE

After you create tests coverage, you can use the Coverage Analysis view in the requirements module to analyze the breakdown of child requirements according to tests coverage. In the Coverage Analysis column, you can see graphical representation of the result of the test coverage.

AUTOMATED TEST SCRIPT

After you design your tests, you can decide which tests to automate. If you choose to perform tests manually, the tests are ready for execution as soon as you define the test steps. When you automate a test, you can generate a test script and then complete it using other HP testing tools Quick Test Professional or Win Runner.

Automate the tests that will run with each new version of your application to check the stability of basic functionality across the entire application (regression tests). Tests that use multiple data values for the same operation(data-driven tests).Tests that are run many times (stress tests) and tests that check a multi-user client/server system (load tests).

RUNNING TEST

Running tests is the core of the testing process. As the application changes, run manual and automated tests in your project to locate defects and assess quality. You start by creating test sets and choosing which tests to include in each set .A test set is a group of tests in a Quality Center project designed to achieve specific testing goals

DEFINING TEST SET

After you design tests in the Test Plan module, you create a test sets tree. A test sets tree enables you to organize your testing process by grouping test sets in folders and organizing them in different hierarchical levels in the Test Lab module. You assign each test set folder to a cycle. This enables you to group together test sets that will be run during the same cycle and analyze the progress of the cycle as you run your tests.

Test sets can include both manual and automated tests. You can also include instances of the same test in different test sets or add more than one instance to the same test set. To decide which test sets to create, consider about the testing goals you defined at the beginning of the testing process. After you define a test set, you can add test instances to your test set.

SCHEDULING TEST RUN

The execution flow tab enables you to specify a date and time to execute a test instance and set conditions for it. A condition is based on the results of another specified test instance in the execution flow.

By setting conditions, you can postpone the execution of a test instance until another specified test instance finishes running or passes. You can also set the sequence in which to execute the test instances. For example, you can schedule test 2 to run only after test 1 finishes, and test 3 to run only if test 2 passes. Locating and repairing defects is an essential phase in application development. Defects can be detected and submitted by developers and testers in all stages of the testing process.

TRACKING DEFECTS

When you submit a defect to a quality center, it is tracked through these stages: New, Open, Fixed, and Closed. A defect may also be rejected or it may be reopened after it is fixed.

When you initially report the defect to the Quality Center project, it is assigned the status new, by default. A quality assurance or project manager reviews the defect and determines whether or not to consider the defect for repair. If the defect is refused, it is assigned the status rejected. If the defect is accepted, the quality assurance or project manager determines a repair priority, changes its status to open, and assigns it to a member of the development team. A developer repairs the defect and assigns it the status fixed.

If the defect recurs, the quality assurance or project manager assigns it the status reopened. If the defect is repaired, the quality assurance or project manager assigns it the status closed

MATCHING DEFECTS

Matching defects enables you to eliminate duplicate or similar defects in your project. Each time you add a new defect, Quality Center stores lists of keywords from the Summary and Description fields. When you search for similar defects, keywords in these fields are matched against other defects. Keywords must be more than two characters and letter case does not affect your results.

UPDATING DEFECTS

Tracking the repair of defects in a project requires that you periodically update defects. You can do so directly in the defects grid or in the defect details dialog box. Note that the ability to update some defect fields depends on your permission settings. After you have updated defects, you can view the current quality status of your release in the release module.

LINKING DEFECT TO TEST

You can link a test in your test plan to a specific defect in the Defects Grid. This is useful, for example, when a new test is created specifically for a known defect. By creating linkage, you can determine if the test should be run based on the status of the defect. Note that you can also link the defect to other entities, such as requirements

A defect can be linked directly or indirectly to an entity. When you add a defect link to an entity, Quality Center adds a direct link to this entity and indirect links to other related entities in the project.

When we link a defect to a run step, it adds an indirect link to its run, test instance, test set, and test. If the same test is covered by a requirement, an indirect link is also added to the requirement. Note that the indirect linkage is a one-directional flow.

ANALYZING THE TESTING PROCESS

Quality Center reports and graphs help you assess your testing process. You can generate reports and graphs at any time during the testing process from the Requirements, Test Plan, Test Lab, and Defects modules, using default or customized settings. When customizing a report or graph, you can apply filters and sort conditions, and display information according to your specifications. You can also save your settings as favorite views and reload them as needed.

GENERATING REPORTS AND GRAPHS

We can generate a report from the Requirements, Test Plan, Test Lab, and defects modules. Each of these modules contains various report options. After you generate report, you can customize report properties to display information according to the specifications. Quality Center graphs help you analyze the relationships between different types of data in a project. Each Quality Center module contains various graph options. After you generate a graph, you can customize graph properties to display information according to your specifications

CHAPTER 4

SYSTEM DESIGN AND DEVELOPMENT

4.1 ELEMENTS OF DESIGN

System Design is the most creative and challenging phase in the development of any system. Design implies to a description of the final system and the process by which it is developed. The first step is to determine how to export and import the test cases from the excel file to the quality center that will meet the requirements of the proposed system. The next step is to determine how to create reusable components that are needed as an output from the system.

During the design of the proposed system some areas where attention is required are:

- How to convert the test cases to the repository?
- How should the test cases be organized?
- What will be the processes involved in the system?
- How to automate the test cases to components?

The steps carried out in the design phase of the reusable components needed of creating some utility for the test cases to process them in the quality center and then automating those to components.

- DOC TO XLS UTILITY
- EXPORTING UTILITY
- COMPONENT IDENTIFICATION
- IMPORTING UTILITY
- AUTOMATING TO COMPONENTS

4.1.1 DOC TO XLS UTILITY

Some of the test cases are maintained by the functional people in the word document for certain projects. This utility is developed in order to convert all the test cases that are available in a word document into excel files for uploading it in quality center. It will decrease the time and effort required to create these test cases manually in excel file format. This utility is developed in java using eclipse editor.

LOCATING FILE PATH:

When this utility is executed it prompts a graphical user interface with radio buttons and to select an option and performs the operation as follows.

- Select file.
- Select directory.
- Reading the content
- Selecting data
- Excel file creation.

SELECT FILE:

When this option is selected, the user can give the source path of a particular word document in Source Path field that is to be converted using Browse button. It will generate an error message if user selects file other than word document.

SELECT DIRECTORY:

When this option is selected, the user can give the source path of a particular folder which contains all the document files.

READING THE CONTENT:

After locating the files path, it will open the file using File Input Stream and reads the contents character by character till the end of the file. A Word Extractor class of POI package is used in order to remove all the special characters which occur during reading the source file.

SELECTING DATA:

While reading the document file certain constrains are defined which select only the required text data from that file

EXCEL FILE CREATION:

This selected data is copied in the new Excel file using File Output Stream as soon as user clicks on Copy button. Data is copied in different sheets as per the requirement in the same Excel workbook. This file is created on the location specified in the destination path field using Browse button.

DESCRIPTION ABOUT CLASS AND ITS FUNCTION

This java file contains 5 classes which are described as follows

FILE CHOOSER CLASS:

This class extends JPanel class. The JPanel provides general-purpose containers for lightweight components. By default, panels do not add colors to anything except their own background. Thus, this class creates a panel for File Chooser entries and buttons.

COMPO CLASS:

This class implements ActionListener Interface and the methods of the classes are given as follows:

- actionPerformed():-This method contain object of action Eventclass as argument and perform functionality of browsing the files or directories from the given path.
- checkFolder() :- This method performs the functionality of checking for files or folders from the given path.

- get Extension() :-This method performs the functionality of finding the extention of the files
- change Extension() :- This method performs the functionality of changing the extention of the file to .xls file

COPYBUT CLASS:

This class implements ActionListener Interface and contains the method actionPerformed (), method contain object of action Eventclass as argument and changes the extention of the destination path and calls the method of the Test class.

OPTION LISTNER CLASS:

This class implements ActionListener Interface and contains the method actionPerformed (), method contain object of action Eventclass as argument and perform the functionality of selecting the files or folders.

FILEPREVIEWER CLASS:

This class extends JComponent class and implements PropertyChangeListener Interface and the methods of this class are :-

- loadImage():- loadImage() method is a method of ImageIcon class an implementation of the Icon interface that paints Icons from Images. Images that are created from a URL or filename are preloaded to monitor the loaded state of the image. This method loads the image, returning only when the image is loaded.
- propertyChange ():- This method gets called when a bound property is changed and is method of PropertyChangeListener Interface.
- Paint ():- This method is invoked by Swing to draw components.
- Main() :- This is the main method which is executed first while running the code.

4.1.2 EXPORTING UILITY:

The test cases for the different type of projects are prepared in the excel sheets by the testing team. An exporting utility is developed to transfer the test cases from the excel file to the quality center for further processing.

- The utility prompts with a dialog box requesting the user name and the password. It should be the same and match with the quality center. The link that is provided in the utility gets connection with the quality center and authenticate for process.
- In the next dialog box it prompts for the domain name and the project area in the quality center. The user has to specify the repository under which they working und the project name.
- Another dialog box appears requesting the user to specify process area and scenario name under which the test cases are going to be uploaded in the quality center.
- If the user specified scenario is not available it will create a new folder in the same name under the process area and save the test cases under it.
- After the verification a new dialog box is prompted with step number, description, expected result, transaction code, data requirements, subject, and test name.
- The user has to specify the filed names from the excel file in the same sequence according to the request.
- In the subject filed the user has to specify the test case folder name with the number as there may be more than one test case in scenario.
- The test case name should be given in test name field and the same test case name cannot be given if there is more than one test case in a scenario.
- After selecting transfer option the test cases are uploaded in the quality center in the user specified area under the same name.

4.1.3 IMPORTING UTILITY

The main purpose of this utility is to extract all the test cases when the user requires from a particular project or process area from the quality center which is stored in hierarchical tree structure in test plan module.

LOG-IN PAGE

- In the hostname field the user has to give name of the host which has remote access permission to the quality center.
- The user name, password, domain, project should be given same as the quality center.
- In the project field user has to give name of the process area with full path address on which you want to run the utility for downloading the test cases.
- In storage path field user has to give valid path on their local machine where they wanted to store the files.

IMPORTING TEST CASES

- After entering all the required details the user selects the get attachment button to import them from quality center.
- It goes to the connection module gets connected with quality center server and checks for the qcUser and qcPassword. Then connects with the qcDomain, qcProject.
- After getting the connection a message box is prompted do you want to download attachments under subject with option yes or no?
- If the user response is yes it goes to the subject node in the quality center checks the availability attachments in the process area.
- If the attachment status is true then reads the filename retrieves them to the specified path mentioned in the storage path filed.

- After retiring the files it goes to the disconnection module checks the tree structure for the attachments under process area where it will return null and it gets disconnected.
- If the user selects no option to download attachment under subject it prompts with message box to enter the path for downloading attachments.
- If the user log out of the quality center the connection will be lost be lost between the utility resulting in exit of utility.
- The utility can also be exited directly without logging out of quality center.

REPORT

- The user can view the report of the process each time the utility is executed to import the attachments from the quality center.
- The srno field assigns the number to test cases. When the utility is executed it runs from the first test case in the quality center so the user can know the number of test cases available in the repository.
- Path in qc field specifies the process name, scenario name, folder name of the test cases that are stored. If an attachment is imported we can know the path in quality center by referring to this field.
- Test case name field gives the name of the test cases in the quality center. The user can check the test case name and verify whether the correct test cases are imported.
- The hasattachment filed will be true if there is attachment in the repository. The user can also view the path and attachment name respectively.
- The attachment name field gives the name of the attachment and test case number with the extension of the file attached.
- The download path field gives the user select path along with the project name and the test case name that is downloaded in the user system.
- If there is any error occurred while executing the utility its reported to the user in message box with details.

4.1.4 COMPONENT IDENTIFICATION

Business components are testing units that perform specific tasks in a business process. The subject matter expert creates Business components that describe the specific tasks that can be performed in the application, and the condition or state of the application before and after those tasks.

The following are the necessary steps that should be performed in identifying the business components in the quality center.

- First Identify the Scenarios for which the components are to be built in the quality center.
- Create an excel sheet, wherein we mention the Scenario and the transaction codes used. Also note down the number of occurrences of each transaction.
- Identify the different transaction codes used in the scenarios considered and draw a flow diagram for each transaction code.
- The flow diagrams are created with a VISIO software tool as it's convenient to the entire user to perform any editing operation during the enhancement in future.
- For each Transaction code flow diagram, identify the different flows for the transaction and build a component for each flow. If there is a single flow only, one component is enough.
- Since the flow diagrams are drawn in the scenario level they are all uploaded as a attachment in the quality center.
- The purpose of doing this is to reuse the component in other scenarios too where it will be useful. For example, if we have identified 3 flows for a transaction, we build 3 business components for each flow.
- Group different business components created according to the scenario flow. This solves the purpose of creating business components and reusing them wherever necessary.
- You can also define manual steps for the component, and then choose whether to convert it to an automated Quickest keyword-driven component, Quick Test scripted component, or Win Runner component.

4.1.5 AUTOMATING TO COMPONENTS

After identifying the components they have to be automated. Automation is a Microsoft technology that makes it possible to access software objects inside one application from other applications. For example, most applications require users to log in before they can access any of the application functionality. Creating one business component that represents this login procedure that can be used in many tests, resulting in easier and more cost-efficient.

- Once components are identified from the diagram write the description in steps for each component with the flow of process.
- Once all the components are finished with the description part the components can be automated with quick test tool.
- For quick test to access a quality center project, user must connect to the local or remote Web server where quality center is installed.
- The user selects the component goes to the automation tab in the quality center and selects the quick test tool and click run.
- The quick test tool automates them by generating script for the test steps that is located in the components.
- In the Keyword View, quick test displays information about each step and shows the object hierarchy in an icon-based table.
- In the Expert View, quick test displays each step as a VBScript line or statement. In object-based steps, the VBScript statement defines the object hierarchy.
- Automation scripts are especially useful for performing the same tasks multiple times or on multiple tests, or quickly configuring quick test according to your needs for a particular environment or application.
- The user can now manually change the script, so when a new project is allotted for testing these test script or components can be executed with some customization if required instead of creating new test cases.

4.2 DATA FLOW DIAGRAM (DFD)

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed.

The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams.

The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams.

4.2.1 CONTEXT DIAGRAM

The development of DFD'S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The top-level diagram is often called context diagram. It consists of single process bit, which shows the interaction between the system and outside entities. This context-level DFD is then "exploded" to show more detail of the system being modeled.

The Context diagram shows the overall system with the users who will be interacting with it.

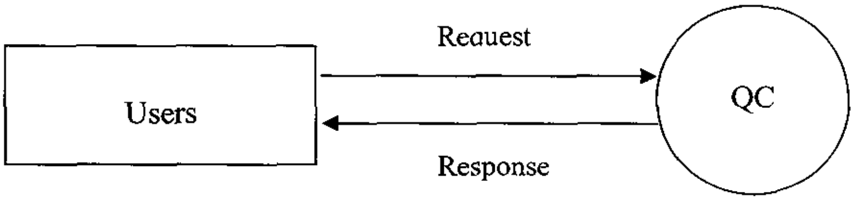


Figure 4.2.1 DFD LEVEL 0 QC

DFD: LEVEL 1 LOGIN

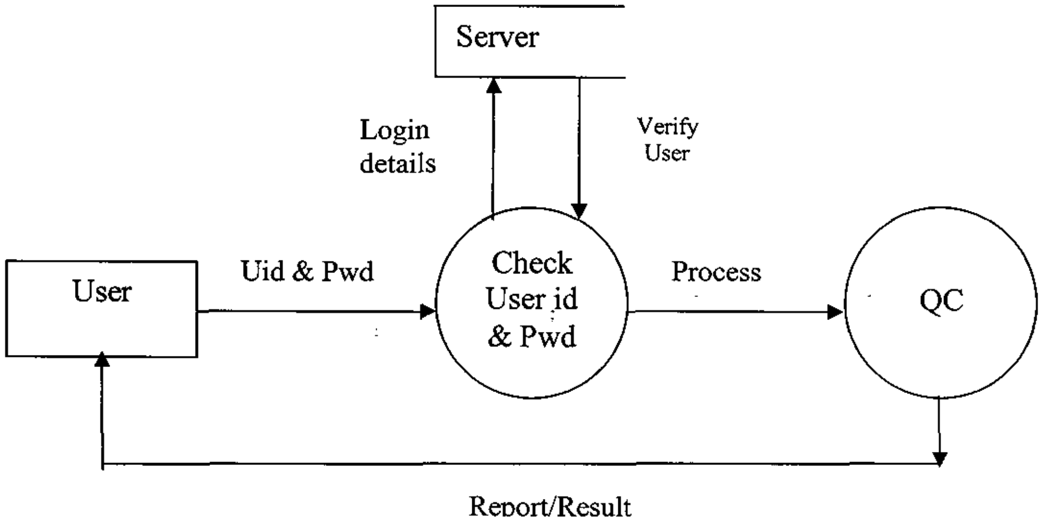


Figure 4.2.2 DFD: LEVEL 1 LOGIN

Login DFD explains the process that will take place when the team leads/tester logs on into the system. Only authorized users are allowed to enter into system by ensuring username and password.

DFD: LEVEL 2 EXECUTING UTILITY

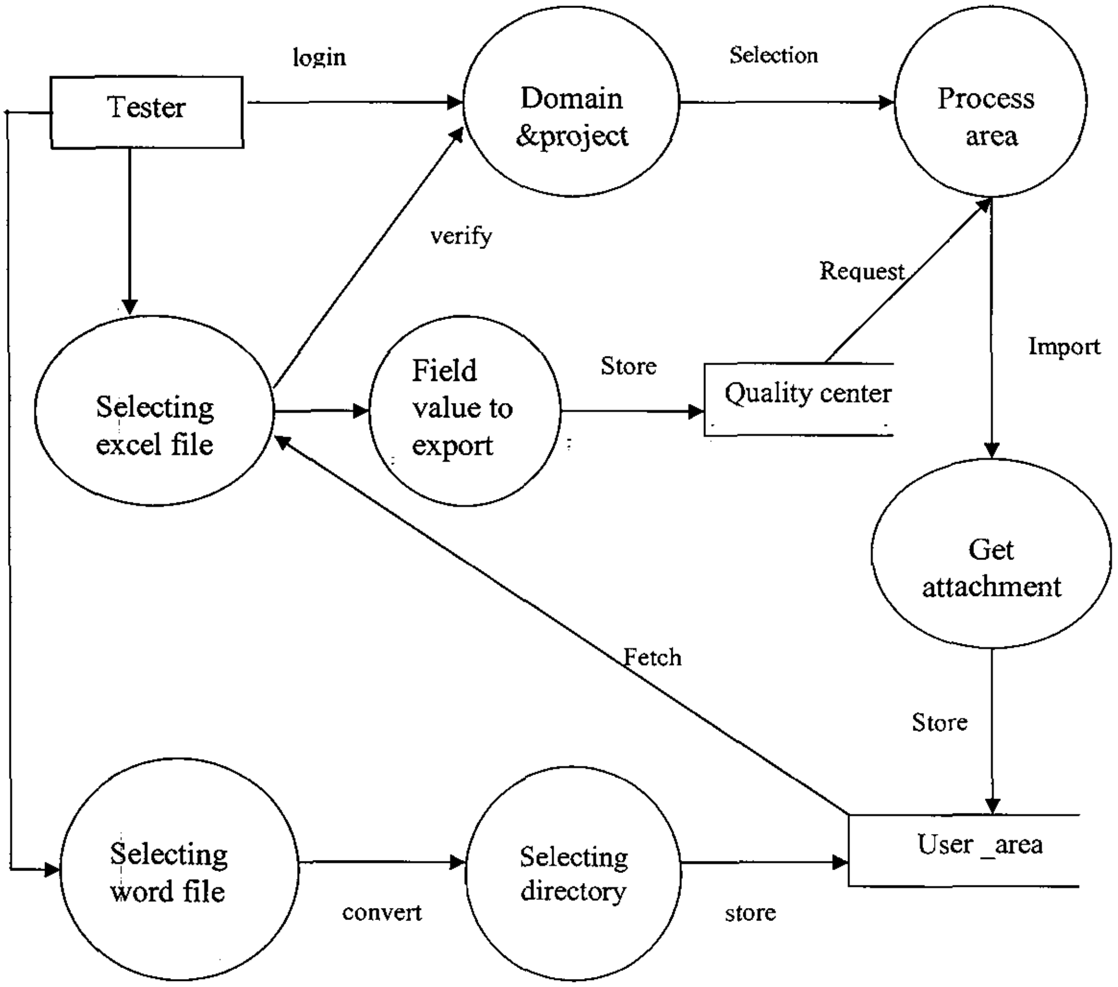


Figure 4.3.3 DFD: LEVEL 2 EXECUTING UTILITY

The utility are used to transfer the test cases from word files to excel files. Those excel files can be exported in to quality center and they can also be imported by the user in specific area for reference from quality center. These processes can be carried out only by the authorized person.

FLOW DIAGRAM FOR TEST CASES

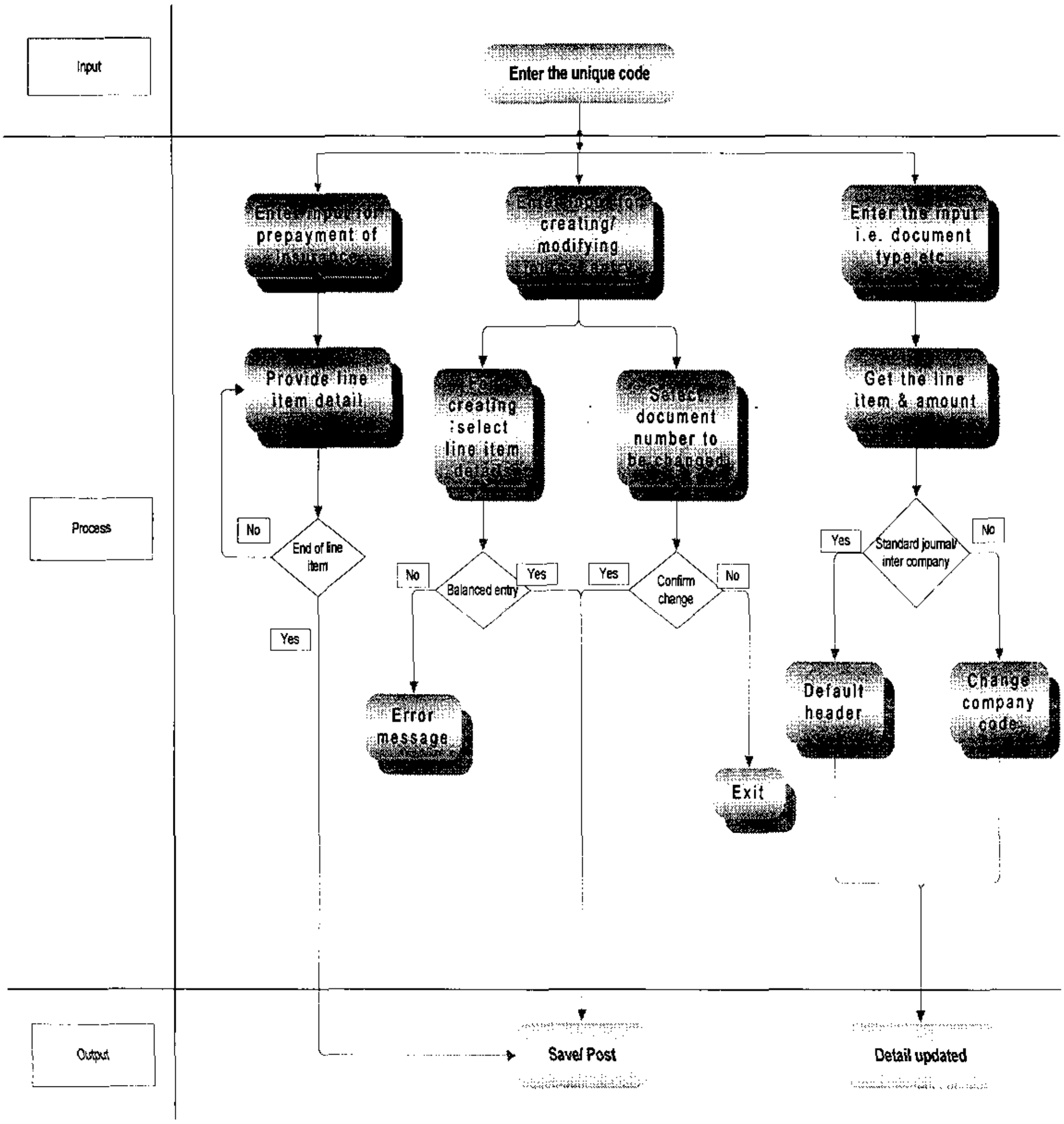


Figure 4.2.4

The flow diagrams are created with a VISIO software tool, by integrating the test cases at scenario level. The code specifies the transaction code in sap. We can identify & automate the components with help of this diagram.

4.3.1 USE CASE DIAGRAM

The diagram below gives the overall context of the SAP testing service & offering. The users of the system are depicted below.

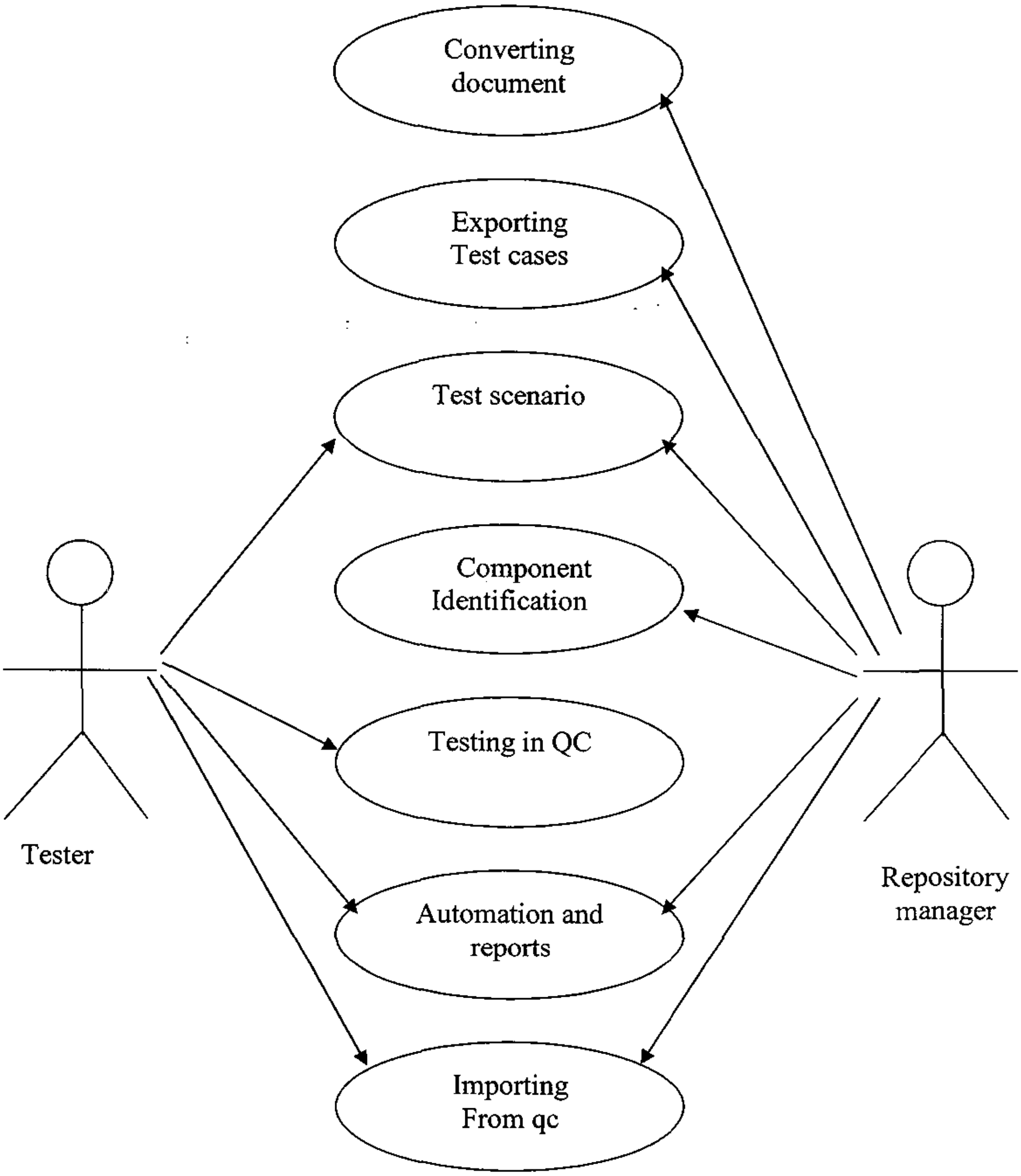


Figure 4.3.1 USE CASE DIAGRAM

CHAPTER 5

SYSTEM TESTING AND IMPLEMENTATION

System Testing and Implementation is the part of the software engineering life cycle, where, the design artifacts are converted to a working application. Once the design is coded into a working application, it has to be verified, validated and tested in detail. The tested product if successful is deployed in the user environment.

5.1 SYSTEM VERIFICATION

System Verification answers the question "Am I building the product right?" It includes the review of interim work steps and interim deliverables during a project to ensure they are acceptable. Verification also determines if the system is consistent, adheres to standards, uses reliable techniques and prudent practices, and performs the selected functions in the correct manner. In data access, it verifies whether the right data is being accessed, in terms of the right place and in the right way.

5.2 SYSTEM VALIDATION

Validation answers the question "Am I building the right product?" This checks whether the developer is moving towards the right product, whether the development is moving towards the actual intended product that was agreed upon in the beginning. Validation also determines if the system complies with the requirements and performs functions for which it is intended and meets the organization's goals and user needs. It is traditional and is performed at the end of the project. In data access, it checks whether we are accessing the right data, in terms of data required to satisfy the requirement.

5.3 TESTING

Testing is a critical element of software quality and assurance and represents the ultimate review of specification design and coding. It is a vital activity that has to be enforced in the development of any system. This could be done in parallel during all the phases of system development. The feedback received from these tests can be used for further enhancement of the system under consideration. The testing phase conducts test using the Software Requirement Specification as a reference and with the goal to see whether the system satisfies the specified requirements.

The main types of tests carried out on SAP testing service & offering are:

- Unit Testing
- Integration Testing
- System Testing
- Security Testing
- Stress Testing
- Regression Testing.

5.3.1 UNIT TESTING

Module or Unit Testing is the process of testing all the program units that make up a system. Unit testing focuses on an individual module thus allowing one to uncover all the errors made logically and while coding in the module.

In SAP testing service & offering each page is tested separately as a unit. Initially the flow of control and data on sap system is checked. When considering a module as a unit, the flow of data and control through the whole module is tested. The result is stored in the test plan. In a page, each control is further tested in unit testing. The process is done in all the pages of the system. Once the errors are rectified, the testing procedure is repeated with same test cases to ensure this hasn't produced new errors. Hence this is a continuous process.

5.3.2 INTEGRATION TESTING

Integration testing tests the process of integrating the various modules of sap to form the complete system. Integration starts with a set of units each individually tested in isolation and ends when the entire application has been built. Integration testing verifies that the combined units function together correctly. It facilitates in finding problem that occur at interface or communication between the individual parts.

SAP testing service & offering follows top-down integration testing. Modules of the sap system taken for testing are linked to the main menu in a sequence as required in the real time operating mode of the system. This process is continued from the page level to module level, finally to the system level. In the final stage, the whole system is taken together and tested for integration. A change in one place should be reflected through out the sap system.

Regression testing is done after each change made into the software. This set can include both positive and negative checks. Negative tests attempt to fail an application to demonstrate that the application is not functioning properly .The whole set of test cases need to be run again to do the regression testing.

5.3.3 SYSTEM TESTING

System testing is actually a serious of different tests, whose primary purpose is to fully exercise the computer-based system. This helps in verifying that all the system elements have been properly integrated and perform the allocated functions. It verifies the entire product after having integrated all software and hardware components, and validates it according to the original project requirement. The system testing takes into consideration the hardware, and the software.

5.3.3.1 SECURITY TESTING

Security testing is important in system testing. The system in no way shall be accessible to unauthorized users. Testing is done to ensure that a user with respective rights can only view the various forms and reports presented by Task Allocation and Management System. If users try to perform something beyond his assigned rights corresponding messages should be displayed. The Task Allocation and Management System in such cases display an error message.

Another security issue involves the sensitive data in the system. The system is highly secure with authentication fixed at various levels of the hierarchy.

One more level of security is concerned with user rights. Each user is applied rights module wise. The menus can be configured to roles. Users can also be configured to roles. Menu items are assigned to users dynamically based on the roles assigned to menu items as well as users. A match is done before displaying the menu to the user. Different Menu items are displayed for user and team lead or manager.

5.3.3.2 STRESS TESTING

Stress Testing executes a system in a manner that demands resources in abnormal quantity, frequency or volume. Task Allocation and Management System was stress tested by all employees in the project and accessing simultaneously to various modules in the system.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

The SAP testing service & offering enables the testing team to get rid of a very tedious and time consuming process of creating test cases which has been followed so far. By automating the test cases in to reusable components the testing team and managers are spared of a cumbersome and repetitive task. It also eliminates the possibility of any error in manual testing system.

When fully deployed, it will transform the entire working of the testing team and managers that deal with creation of test cases for projects in to customization of components based on the requirement with the help of this automation. It makes the entire procedure faster, error free, simple, and efficient and more performance enhancing one. SAP testing service & offering provides uniform utility tools for transferring the test cases in to central repository.

The components and the information in the system should be maintained up to date with periodic updates. The security feature of this repository allows only testing leads and managers to make updates to important and sensitive data. It prevents unauthorized access to important data.

Thus the SAP testing service & offering increases the efficiency, performance of the testing team and managers by enabling them to spend more time on concentrating on quality of the testing process.

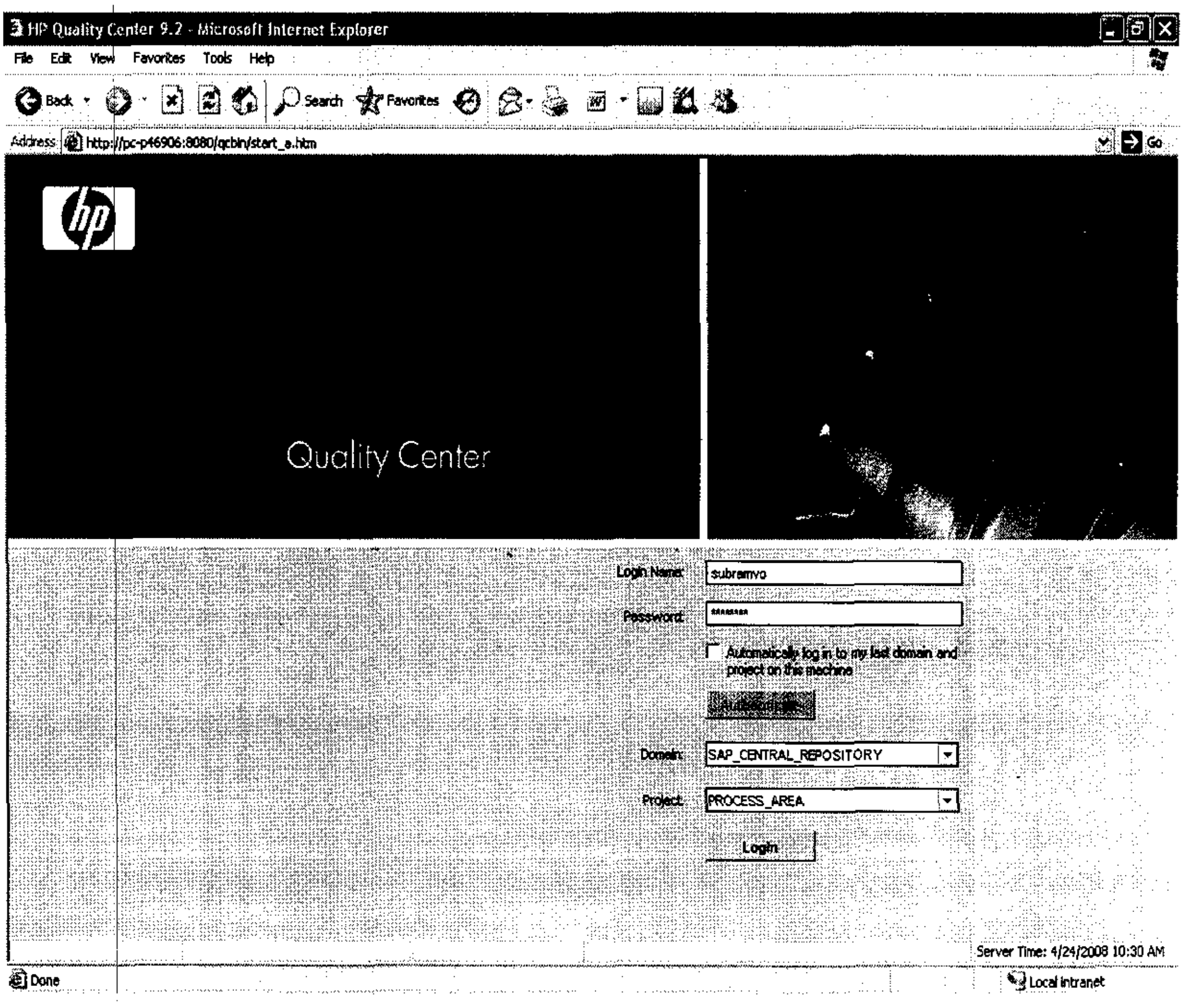
6.2 FUTURE ENHANCEMENT

The following features listed below if introduced could benefit the users of the system.

- Extending the repository by creating more components so that it suits projects without doing customization.
- Using the script for creating test cases for new process where there are no components.
- enhancing the component by incorporating functions (operations) that encapsulate the steps needed to perform a particular task.
- Integrating all the utility in to one single application so that it will be easy for the user to execute them..
- If any new modules want to be added into the system can be added easily.

APPENDIX

MAIN PAGE



HP Quality Center 9.2 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go
Quality Center
Authenticate
Login
Done
Local intranet

Address: http://pc-p46906:8080/qcbin/start_a.htm

Login Name: subramvo

Password: ********

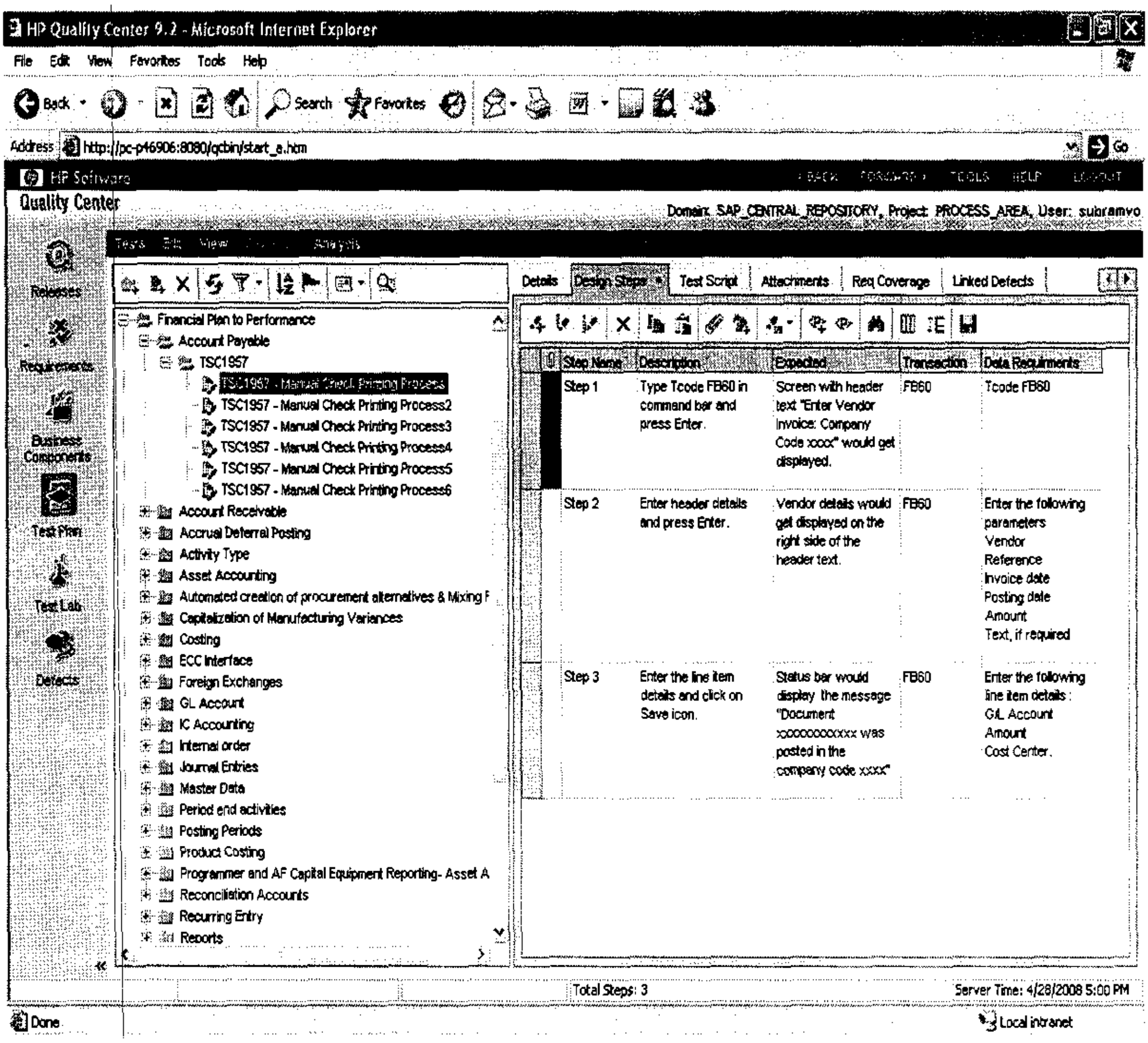
□ Automatically log in to my last domain and project on this machine

Domain: SAP_CENTRAL_REPOSITORY

Project: PROCESS_AREA

Server Time: 4/24/2008 10:30 AM

STRUCTURE OF TEST CASE IN QUALITY CENTER

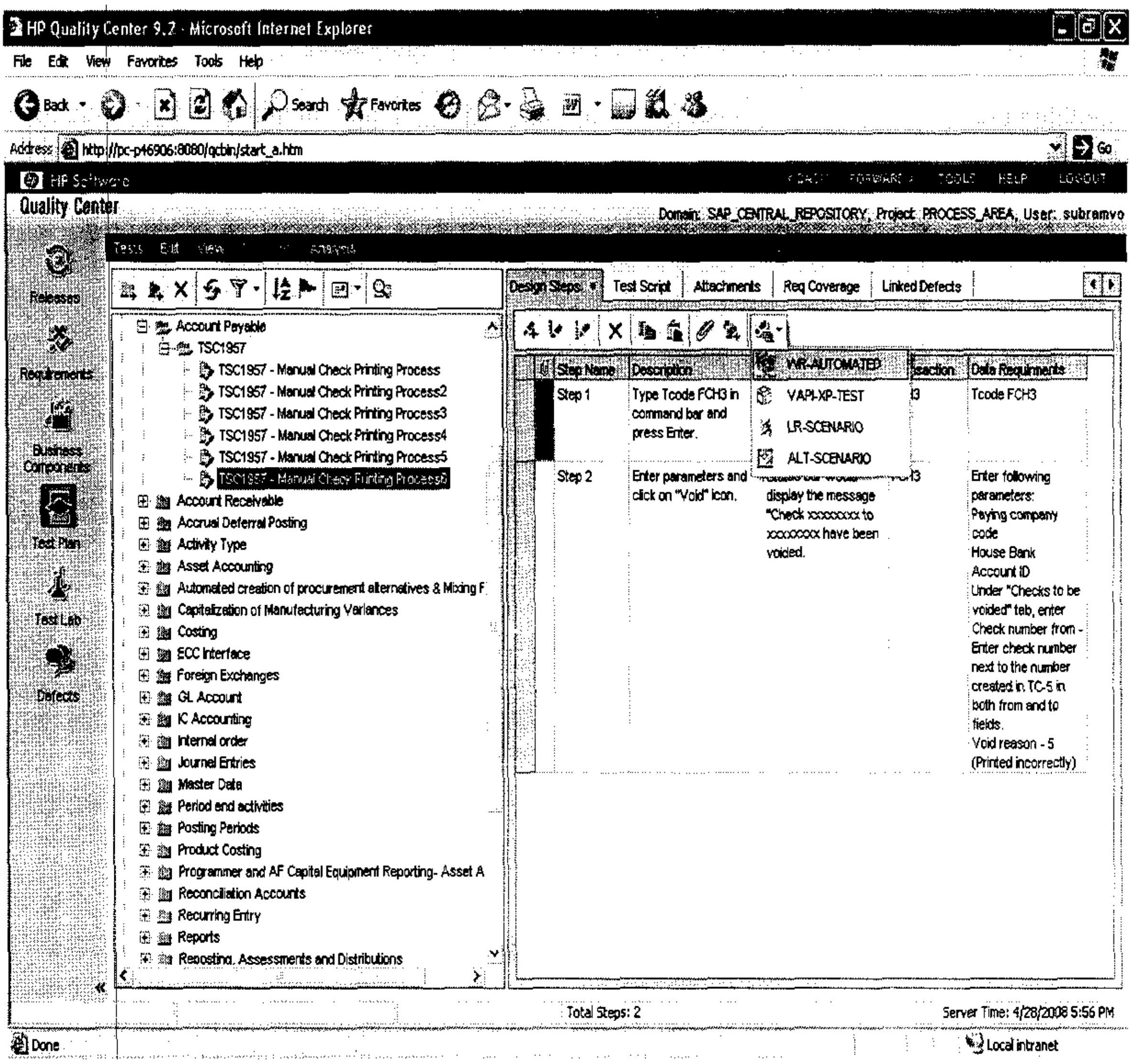

HP Quality Center 9.2 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://pc-p46906:8080/qcbin/start_a.htm
Go
HP Software
Quality Center
Domain: SAP_CENTRAL_REPOSITORY, Project: PROCESS_AREA, User: subramvo
Releases
Requirements
Business Components
Test Plan
Test Lab
Defects
Financial Plan to Performance
Account Payable
TSC1957
TSC1957 - Manual Check Printing Process
TSC1957 - Manual Check Printing Process2
TSC1957 - Manual Check Printing Process3
TSC1957 - Manual Check Printing Process4
TSC1957 - Manual Check Printing Process5
TSC1957 - Manual Check Printing Process6
Account Receivable
Accrual Deferral Posting
Activity Type
Asset Accounting
Automated creation of procurement alternatives & Mixing F
Capitalization of Manufacturing Variances
Costing
ECC Interface
Foreign Exchanges
GL Account
IC Accounting
Internal order
Journal Entries
Master Data
Period end activities
Posting Periods
Product Costing
Programmer and AF Capital Equipment Reporting- Asset A
Reconciliation Accounts
Recurring Entry
Reports
Details
Design Steps
Test Script
Attachments
Req Coverage
Linked Defects
Done
Local intranet

Step Name	Description	Expected	Transaction	Data Requirements
Step 1	Type Tcode FB60 in command bar and press Enter.	Screen with header text "Enter Vendor Invoice: Company Code xxxx" would get displayed.	FB60	Tcode FB60
Step 2	Enter header details and press Enter.	Vendor details would get displayed on the right side of the header text.	FB60	Enter the following parameters Vendor Reference Invoice date Posting date Amount Text, if required
Step 3	Enter the line item details and click on Save icon.	Status bar would display the message "Document xxxxxxxxxxxx was posted in the company code xxxx"	FB60	Enter the following line item details : G/L Account Amount Cost Center.

Total Steps: 3

Server Time: 4/28/2008 5:00 PM

AUTOMATING TEST CASES


HP Quality Center 9.2 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://pc-p46906:8080/qcbin/start_a.htm
Go
HP Software
BACK FORWARD TOOLS HELP LOGOUT
Quality Center
Releases
Requirements
Business Components
Test Plan
Test Lab
Defects
Tests Edit View Analysis
Account Payable
TSC1957
TSC1957 - Manual Check Printing Process
TSC1957 - Manual Check Printing Process2
TSC1957 - Manual Check Printing Process3
TSC1957 - Manual Check Printing Process4
TSC1957 - Manual Check Printing Process5
TSC1957 - Manual Check Printing Process6
Account Receivable
Accrual Deferral Posting
Activity Type
Asset Accounting
Automated creation of procurement alternatives & Mixing F
Capitalization of Manufacturing Variances
Costing
ECC Interface
Foreign Exchanges
GL Account
IC Accounting
Internal order
Journal Entries
Master Data
Period end activities
Posting Periods
Product Costing
Programmer and AF Capital Equipment Reporting- Asset A
Reconciliation Accounts
Recurring Entry
Reports
Reposting, Assessments and Distributions
Design Steps
Test Script
Attachments
Req Coverage
Linked Defects
WR-AUTOMATED
VAPI-XP-TEST
LR-SCENARIO
ALT-SCENARIO
Done
Local intranet

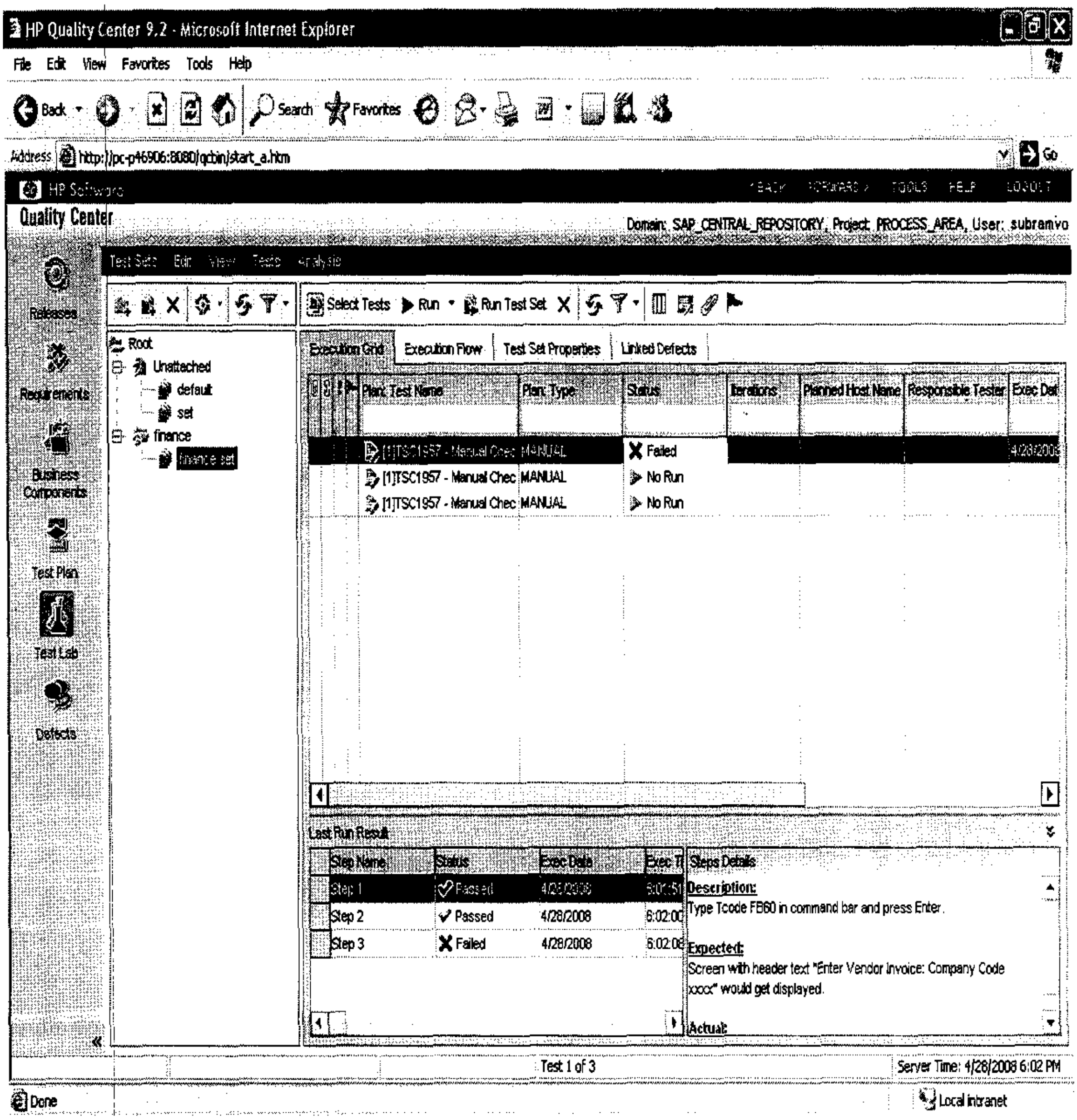
Domain: SAP_CENTRAL_REPOSITORY, Project: PROCESS_AREA, User: subramvo

Step Name	Description		saction	Data Requirements
Step 1	Type Tcode FCH3 in command bar and press Enter.		3	Tcode FCH3
Step 2	Enter parameters and click on "Void" icon.	display the message "Check xxxxxxxx to xxxxxxxx have been voided.	3	Enter following parameters: Paying company code House Bank Account ID Under "Checks to be voided" tab, enter Check number from - Enter check number next to the number created in TC-5 in both from and to fields. Void reason - 5 (Printed incorrectly)

Total Steps: 2

Server Time: 4/28/2008 5:56 PM

TEST RESULT



Domain: SAP_CENTRAL_REPOSITORY, Project: PROCESS_AREA, User: subramvo

Plan: Test Name	Plan: Type	Status	Iterations	Planned Host Name	Responsible Tester	Exec Dat
[1]TSC1957 - Manual Chec	MANUAL	✗ Failed				4/28/2008
[1]TSC1957 - Manual Chec	MANUAL	No Run				
[1]TSC1957 - Manual Chec	MANUAL	No Run				

Step Name	Status	Exec Date	Exec Ti
Step 1	✓ Passed	4/28/2008	6:01:51
Step 2	✓ Passed	4/28/2008	6:02:00
Step 3	✗ Failed	4/28/2008	6:02:08

Steps Details

Description:

Type Tcode FB60 in command bar and press Enter.

Expected:

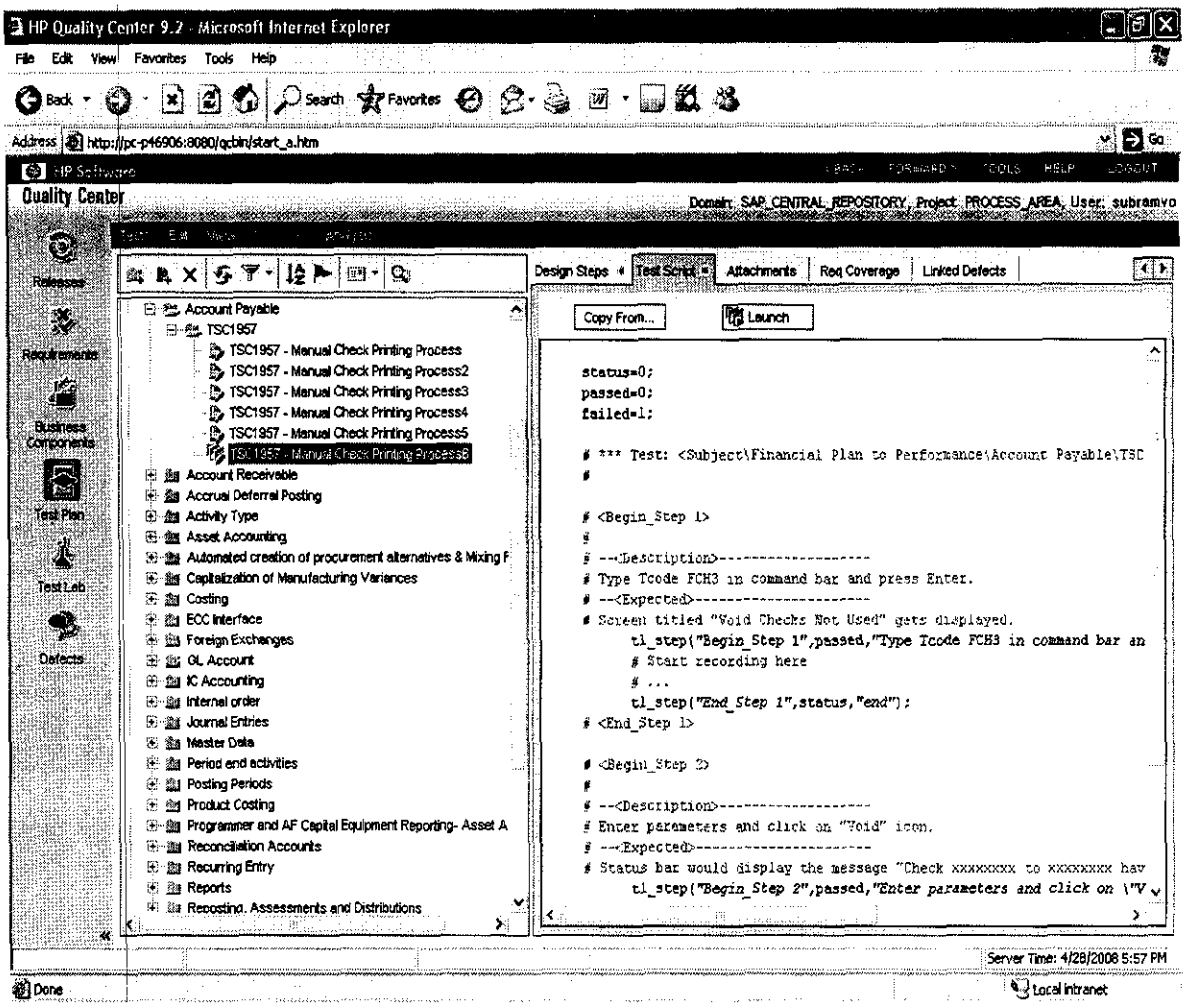
Screen with header text "Enter Vendor Invoice: Company Code xxxx" would get displayed.

Actual:

Test 1 of 3

Server Time: 4/28/2008 6:02 PM

SCRIPT OF AUTOMATED COMPONENTS


HP Quality Center 9.2 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://pc-p46906:8080/qcbin/start_a.htm
Go
HP Software
Quality Center
Domain: SAP_CENTRAL_REPOSITORY, Project: PROCESS_AREA, User: subramvo
Releases
Requirements
Business Components
Test Plan
Test Lab
Defects
Account Payable
TSC1957
TSC1957 - Manual Check Printing Process
TSC1957 - Manual Check Printing Process2
TSC1957 - Manual Check Printing Process3
TSC1957 - Manual Check Printing Process4
TSC1957 - Manual Check Printing Process5
TSC1957 - Manual Check Printing Process6
Account Receivable
Accrual Deferral Posting
Activity Type
Asset Accounting
Automated creation of procurement alternatives & Mixing F
Capitalization of Manufacturing Variances
Costing
ECC Interface
Foreign Exchanges
GL Account
IC Accounting
Internal order
Journal Entries
Master Data
Period end activities
Posting Periods
Product Costing
Programmer and AF Capital Equipment Reporting- Asset A
Reconciliation Accounts
Recurring Entry
Reports
Reposting, Assessments and Distributions
Design Steps
Test Script
Attachments
Req Coverage
Linked Defects
Copy From...
Launch
Server Time: 4/28/2008 5:57 PM
Done
Local intranet

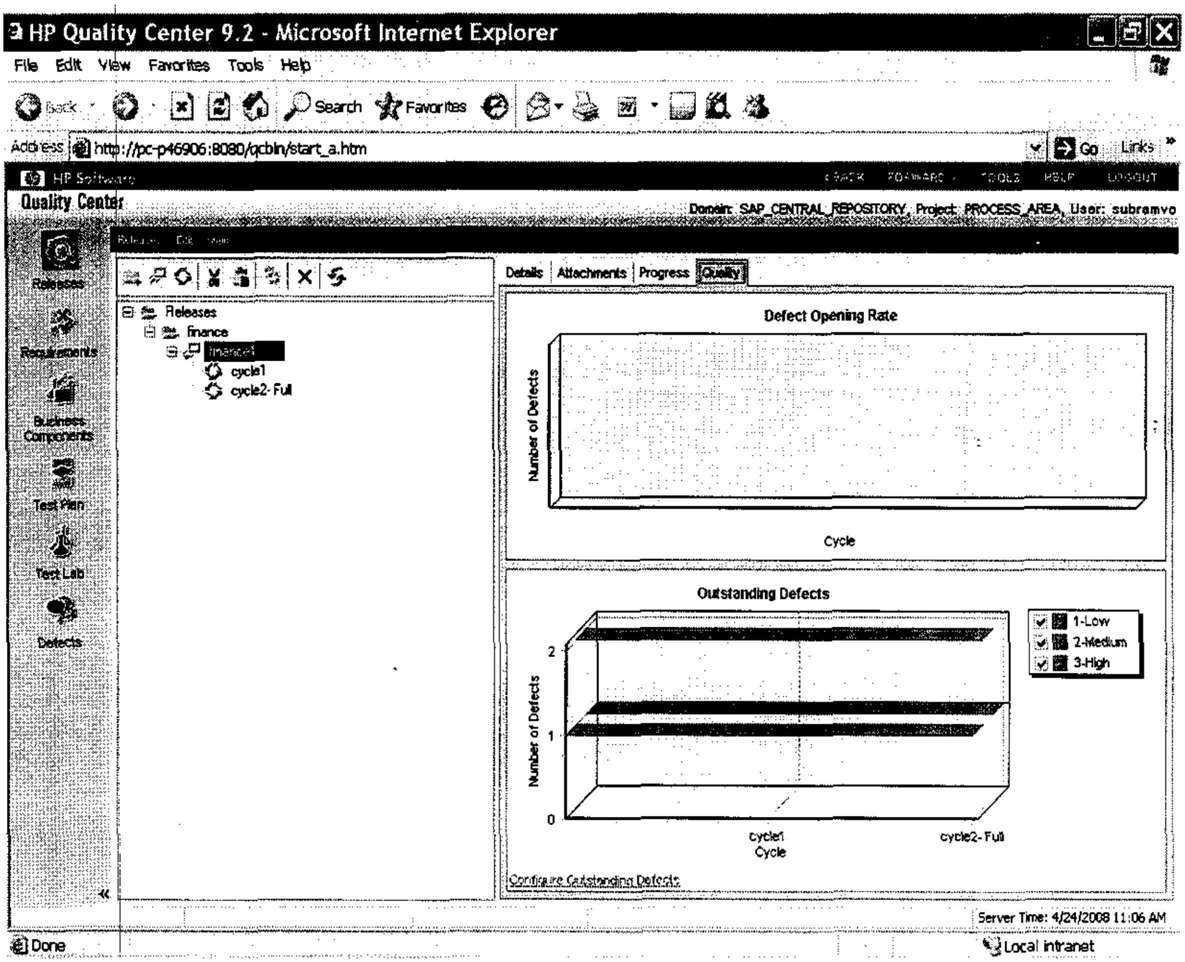
```
status=0;
passed=0;
failed=1;

# *** Test: <Subject\Financial Plan to Performance\Account Payable\TSC
#

# <Begin_Step 1>
#
# --<Description>-------------------
# Type Tcode FCH3 in command bar and press Enter.
# --<Expected>----------------------
# Screen titled "Void Checks Not Used" gets displayed.
     tl_step("Begin_Step 1",passed,"Type Tcode FCH3 in command bar an
     # Start recording here
     # ...
     tl_step("End_Step 1",status,"end");
# <End_Step 1>

# <Begin_Step 2>
#
# --<Description>-------------------
# Enter parameters and click on "Void" icon.
# --<Expected>----------------------
# Status bar would display the message "Check xxxxxxxx to xxxxxxxx hav
     tl_step("Begin_Step 2",passed,"Enter parameters and click on \"V
```

TEST RESULT IN GRAPH

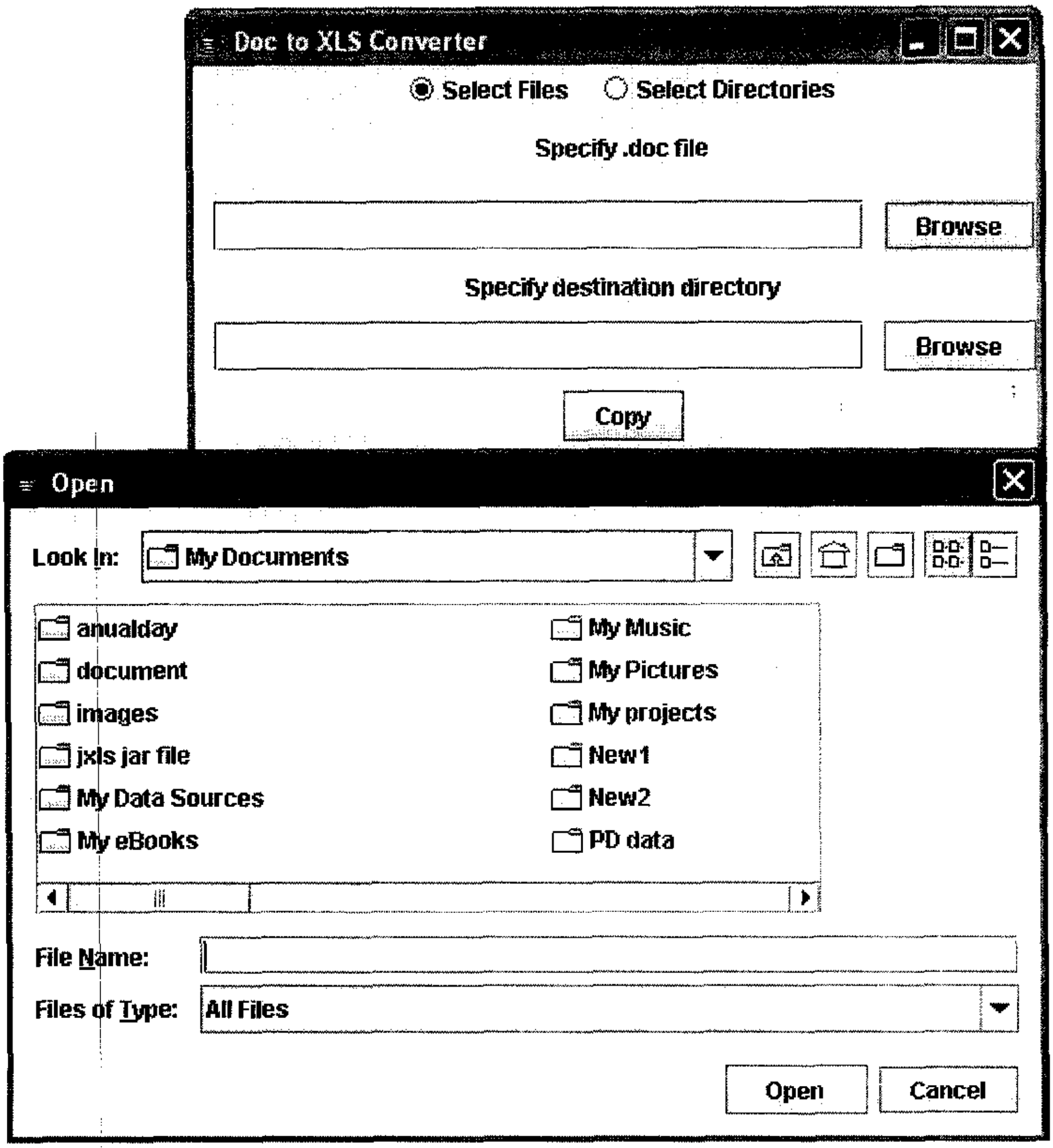


Address http://pc-p46906:8080/qcbin/start_a.htm

Domain: SAP_CENTRAL_REPOSITORY, Project: PROCESS_AREA, User: subramvo

Server Time: 4/24/2008 11:06 AM

WORD TO EXCEL CONVERTING UTILITY

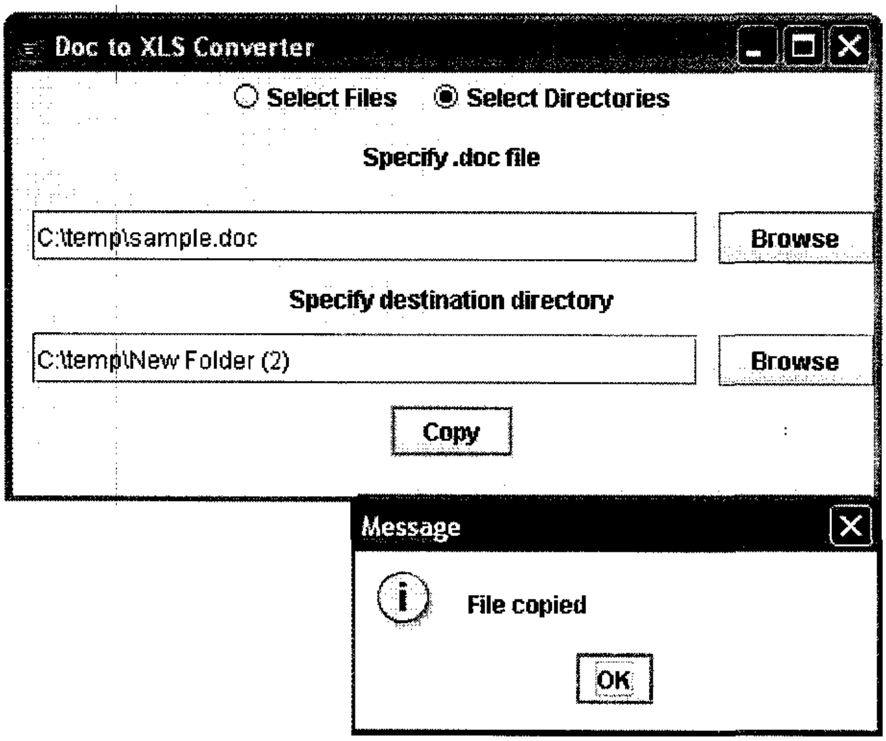


● Select Files ○ Select Directories

Look In: My Documents

Files of Type: All Files

WORD FILE CONVERTED TO EXCEL



Doc to XLS Converter

○ Select Files ● Select Directories

Specify .doc file

C:\temp\sample.doc

Browse

Specify destination directory

C:\temp\New Folder (2)

Browse

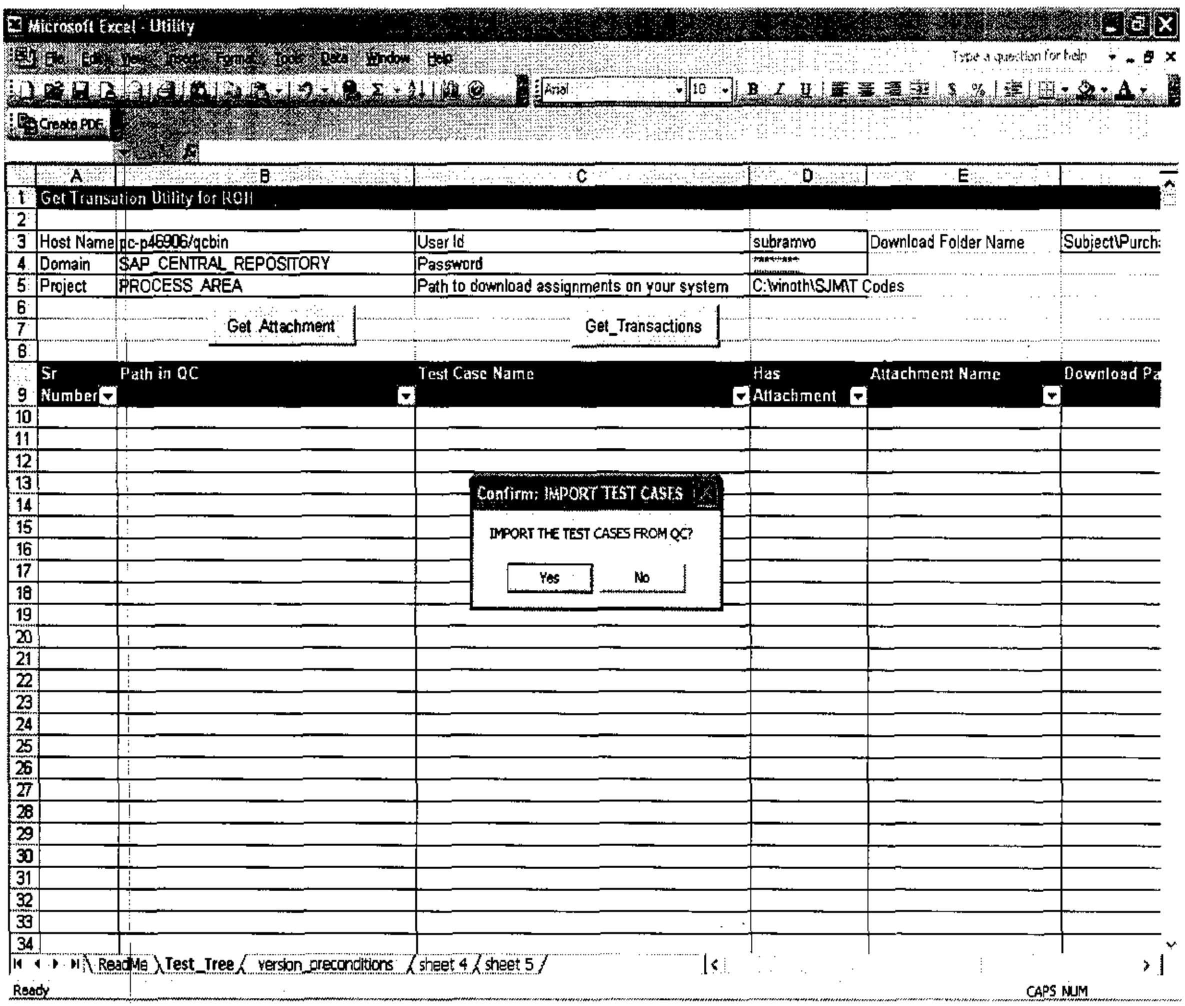
Copy

Message

File copied

OK

IMPORTING UTILITY



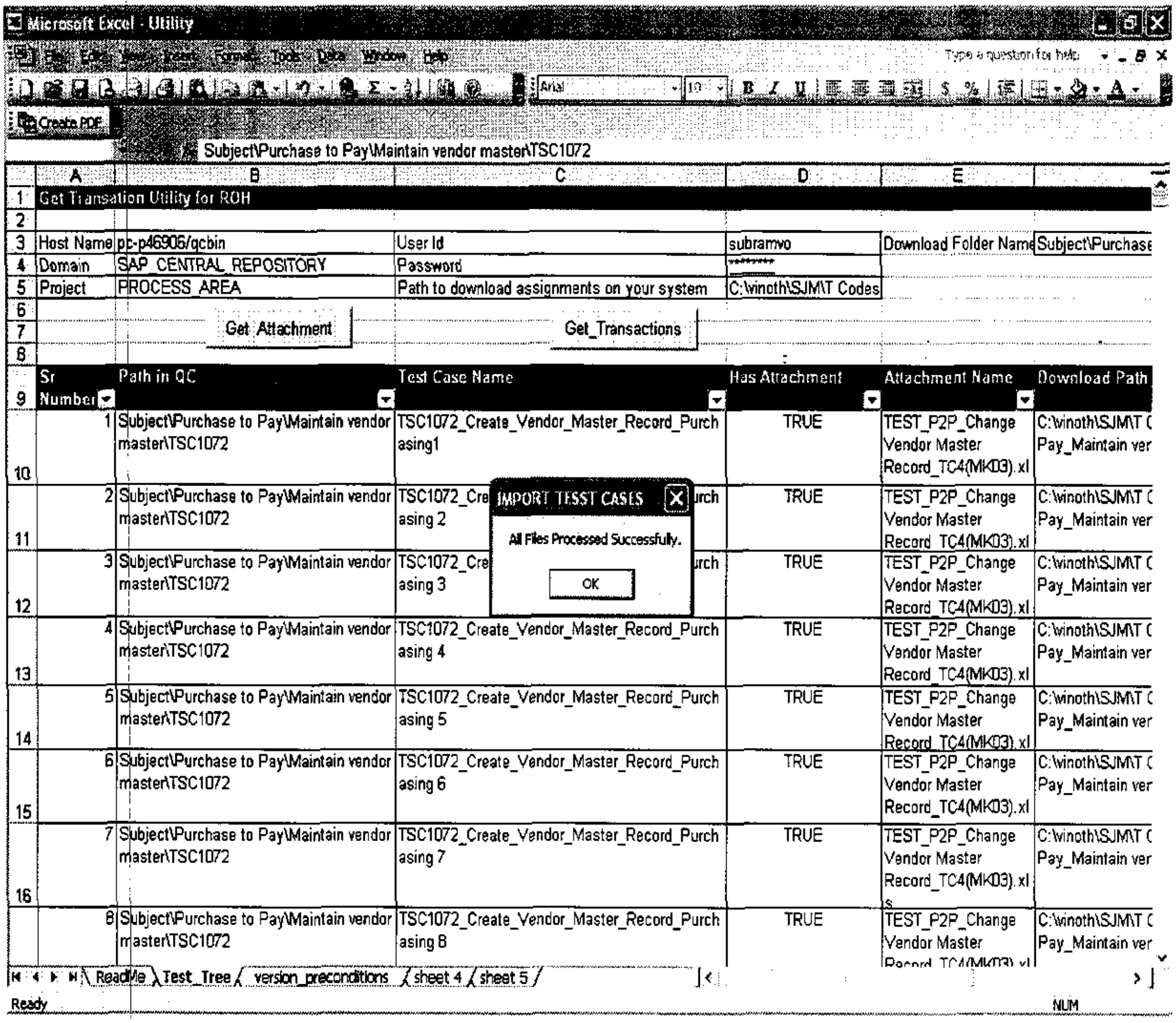
	A	B	C	D	E	
1	Get Transation Utility for ROH					
2						
3	Host Name	qc-p46906/qcbin	User Id	subramvo	Download Folder Name	Subject\Purch
4	Domain	SAP_CENTRAL_REPOSITORY	Password	********		
5	Project	PROCESS_AREA	Path to download assignments on your system	C:\vinoth\SJM\T Codes		
7		Get Attachment		Get_Transactions		
9	Sr Number	Path in QC	Test Case Name	Has Attachment	Attachment Name	Download Pa

Confirm: IMPORT TEST CASES

IMPORT THE TEST CASES FROM QC?

Yes No

TEST CASES IMPORTED



Subject\Purchase to Pay\Maintain vendor master\TSC1072

	A	B	C	D	E	
1	Get Transation Utility for ROH					
2						
3	Host Name	pc-p46906/qcbin	User Id	subramvo	Download Folder Name	Subject\Purchase
4	Domain	SAP_CENTRAL_REPOSITORY	Password	********		
5	Project	PROCESS_AREA	Path to download assignments on your system	C:\vinoth\SJM\T Codes		
6						
7						
8						
9	Sr Number	Path in QC	Test Case Name	Has Attachment	Attachment Name	Download Path
10	1	Subject\Purchase to Pay\Maintain vendor master\TSC1072	TSC1072_Create_Vendor_Master_Record_Purchasing1	TRUE	TEST_P2P_Change Vendor Master Record_TC4(MK03).xl	C:\vinoth\SJM\T C Pay_Maintain ver
11	2	Subject\Purchase to Pay\Maintain vendor master\TSC1072	TSC1072_Cre urch asing 2	TRUE	TEST_P2P_Change Vendor Master Record_TC4(MK03).xl	C:\vinoth\SJM\T C Pay_Maintain ver
12	3	Subject\Purchase to Pay\Maintain vendor master\TSC1072	TSC1072_Cre urch asing 3	TRUE	TEST_P2P_Change Vendor Master Record_TC4(MK03).xl	C:\vinoth\SJM\T C Pay_Maintain ver
13	4	Subject\Purchase to Pay\Maintain vendor master\TSC1072	TSC1072_Create_Vendor_Master_Record_Purchasing 4	TRUE	TEST_P2P_Change Vendor Master Record_TC4(MK03).xl	C:\vinoth\SJM\T C Pay_Maintain ver
14	5	Subject\Purchase to Pay\Maintain vendor master\TSC1072	TSC1072_Create_Vendor_Master_Record_Purchasing 5	TRUE	TEST_P2P_Change Vendor Master Record_TC4(MK03).xl	C:\vinoth\SJM\T C Pay_Maintain ver
15	6	Subject\Purchase to Pay\Maintain vendor master\TSC1072	TSC1072_Create_Vendor_Master_Record_Purchasing 6	TRUE	TEST_P2P_Change Vendor Master Record_TC4(MK03).xl	C:\vinoth\SJM\T C Pay_Maintain ver
16	7	Subject\Purchase to Pay\Maintain vendor master\TSC1072	TSC1072_Create_Vendor_Master_Record_Purchasing 7	TRUE	TEST_P2P_Change Vendor Master Record_TC4(MK03).xls	C:\vinoth\SJM\T C Pay_Maintain ver
	8	Subject\Purchase to Pay\Maintain vendor master\TSC1072	TSC1072_Create_Vendor_Master_Record_Purchasing 8	TRUE	TEST_P2P_Change Vendor Master Record_TC4(MK03).xl	C:\vinoth\SJM\T C Pay_Maintain ver

IMPORT TESST CASES

All Files Processed Successfuly.

OK

REFERENCES

1. HP QUALITY CENTER Tutorial for software version 9.20

2. Testing SAP R/3: A Manager's Step-by-Step Guide

3. Roger S Pressman (2005) 'Software Engineering, A Practitioner's approach', Tata McGraw-Hill

4. Mercury Quick Test Professional Tutorial and user guide for version 9.2

5. Dave Grundgeiger (2001) 'Programming Visual Basic .NET', O'Reilly & Associates

6. MSDN Library Visual Studio 2005 Release.

7. www.google.com

8. www.sap.help.com