Cognizant Technology Solutions

Test Case Point Analysis

White Paper
Version 1.0

April 11, 2001

By
Nirav Patel
Muthukrishnan Govindrajan
Susmita Maharana
Shoba Ramdas

TABLE OF CONTENTS

500 Glenpointe Centre West, Teaneck, NJ 07666 PH: 201-801-0233 Toll Free:1-888-937-3277 http://www.cognizant.com

1 PURPOSE

This is a white paper written on test case point analysis, which basically deals with the estimation of the effort needed for testing projects. The purpose of this article is to provide an introduction to Test Case Point (TCP) Analysis and its application in non-traditional computing situations. The goal of this newly designed estimation technique is to outline all major factors that affect testing projects and to ultimately help projects do an accurate estimation. This metric is technology independent and supports the need for estimating, project management and measuring quality. The target audience for using this approach would be anyone who would want to have a precise testing effort estimation technique for any given application under test. This approach enables separate effort estimates for the different phases of testing viz. Test Case Generation, Automation of Test Cases and Test Case Execution.

500 Glenpointe Centre West, Teaneck, NJ 07666 PH: 201-801-0233 Toll Free:1-888-937-3277 http://www.cognizant.com

2 INTRODUCTION

Effective software project estimation is one of the most challenging and important activity in software project life-cycle. On time project delivery cannot be achieved without an accurate and reliable estimate. Estimates play a vital role in all phases of the software development life-cycle.

There are many popular models for estimation in vogue today. But none of the models determine the efforts needed for separate phases of the SDLC. Organizations specializing in niche areas such as providing only testing services need an estimation model that can accurately determine the size of the Application Under Test (AUT) and in turn the efforts needed to execute it.

Today's e-business applications impose new challenges for software testing. Some of the commonly known challenges are complex application scenarios, extensive third party integration, crunched testing life cycles and increased security concerns. These factors inherently make testing of e-business application far more complex and critical than conventional software testing. In no other projects is performance testing more critical than in web-based projects. Under-estimating a testing project leads to under-staffing it (resulting in staff burnout), under-scoping the quality assurance effort (running the risk of low quality deliverables), and setting too short a schedule (resulting in loss of credibility as deadlines are missed).

Traditionally, estimation of efforts for testing has been more of a ballpark percentage of the rest of the development life cycle phases. This approach to estimation is more prone to errors and carries a bigger risk of delaying the launch deadlines.

TCP analysis is an approach for doing an accurate estimation of functional testing projects. This approach emphasizes on key testing factors that determine the complexity of the entire testing cycle and gives us a way of translating test creation efforts to test execution efforts, which is very useful for regression testing estimation. In other words Test Case Points is a way of representing the efforts involved in testing projects.

500 Glenpointe Centre West, Teaneck, NJ 07666 PH: 201-801-0233 Toll Free:1-888-937-3277 http://www.cognizant.com

3 TCP METHODOLOGY

As stated above, TCP analysis generates test efforts for separate testing activities. This is essential because testing projects fall under four different models. Though in practice, most testing projects are a combination of the four execution models stated below.

Model I **Test Case Generation**	**Model II** **Automated Script Generation**
Model III **Manual Test Execution**	**Model IV** **Automated Test Execution**

Model I – Test Case Generation
This includes designing well-defined test cases. The deliverables here are only the test cases.

Model II – Automated Script Generation
This execution model includes only automating the test cases using an automated test tool. The deliverables include the tool generated scripts

Model III – Manual Test Execution
This execution model only involves executing the test cases already designed and reporting the defects.

Model IV – Automated Test Execution
This phase includes the execution of the automated scripts and reporting the defects.

TCP Analysis uses a 7-step process consisting of the following stages:

1. Identify Use Cases
2. Identify Test Cases
3. Determine TCP for Test Case Generation
4. Determine TCP for Automation
5. Determine TCP for Manual Execution
6. Determine TCP for Automated Execution
7. Determine Total TCP

500 Glenpointe Centre West, Teaneck, NJ 07666 PH: 201-801-0233 Toll Free:1-888-937-3277 http://www.cognizant.com

Given below is an overview of different phases as applied to the four project execution models. Each phase is explained in detail in subsequent sections. Determine TCP for Test Case Generation

To determine the TCP for Test Case Generation, first determine the complexity of the Test Cases. Some test cases may be more complex due to the inherent functionality being tested. The complexity will be an indicator of the number of TCPs for the test case.
Calculate test case generation complexity based on the factors given below.

S.No	Test Case Generation Complexity Factors	Weights *
1	**The number of steps in the case --** Assuming that test cases are atomic and that they test only one condition, the number of steps will be an indicator of the complexity.	2
2	**Interface with other Test Cases** This usually involves calling other test cases from this use case. Ex. Address Book lookup from Compose Mail	1
3	**Establishing a baseline result** e.g. Testing an EMI Calculator would involve validating the formulae used. This would typically be complex	3

Note: The above given weights are just sample weights and should not be mistaken for benchmarks.

-- Determine the complexity based on the following table:

Number of Steps	Weight
<5	1
5-10	2
>10	3

* The standard weights range from 0 to 3.

0 ----> Not Applicable
1 ----> Simple
2 ----> Average
3 ----> Complex

Calculate the sum of the above weights and determine the complexity of each test case.

Calculate the Rough Test Case Points for test generation by using the table below.

Test Case Type	Complexity Weights	Test Case Points
Simple	< 9	6
Average	10 – 16	8
Complex	> 16	12

Calculate Test Case Points by using the below formulae

500 Glenpointe Centre West, Teaneck, NJ 07666 PH: 201-801-0233 Toll Free:1-888-937-3277 http://www.cognizant.com

Rough TCP-G = (# of Simple Test Cases X 6) + (# of Average Test Cases X 8) + (# of Complex Test Cases X 12)

The TCP estimates above might be affected by certain application level parameters. For example, if the AUT is a vertical portal (e.g. Insurance), then creating test cases would involve an understanding of the insurance business. This might increase the effort required. To factor the impact of such parameters, we will increase the TCP by an Adjustment Factor. Some of the Application level parameters that might have an influence on the TCP are listed in the table below:

Sl. No.	Factors	Adjusted Weight
1	**Domain Complexity**	**0.1**
2	**Integration with other devices such as WAP enabled devices, etc.**	**0.1**
3	**Multi-lingual Support**	**0.05**
	Total Factor	**0.25**

The weights in the table are only indicative and will be subject to the application for which estimation is being carried out.

Adjustment Factor = 1 + Total Factor = 1 + 0.25 = 1.25

Each of these factors is scored based on their influence on the system being counted. The resulting score will increase the Unadjusted Test Case Point count. This calculation provides us with the Test Case Point Generation count.

TCP-G = Rough TCP-Generation XAdjustment Factor

3.1 Determine TCP for Automation

From the list of Test Cases derived from Section 3.2, identify the test cases that are good candidates for automation. Some test cases save a lot of effort if done manually and is not worth automating. On the other hand, some test cases cannot be automated because the test tool does not support the feature being tested.

Certain cases are straightforward and are quite easy to automate whereas certain cases involving dynamic data are difficult to automate. Next, determine the test case automation complexity based on the factors given below.

S.No	Test Case Automation Complexity Factors	Weights *
1	**Interface with other Test Cases** This usually involves calling other test cases from this use case. Ex. Address Book lookup from Compose Mail	2
2	**Interface with external application** This is interaction with an application that is outside the system boundary. Ex. Credit Card validations through an independent gateway	0
3	**External Application Validation** This involves testing of third party applications to validate the use case. Ex. Checking Word, PDF reports generated by the system	0
4	**Data Driven** This is usually helpful for testing the use case with positive and negative data.	2

500 Glenpointe Centre West, Teaneck, NJ 07666 PH: 201-801-0233 Toll Free:1-888-937-3277 http://www.cognizant.com

	Ex. User Registration	
5	**Links** # of links to be tested for broken/orphaned links. Typically, dynamic lists like catalog items are rated as complex.	1
6	**Numerical Computations** This involves validation of arithmetical calculations. e.g. Testing an EMI Calculator would involve validating the formulae used. This would typically be complex.	1
7	**Check Points** This involves modifying the test scripts to check for validations. Ex. Page titles, buttons, labels, null values, max char, numeric etc	1
8	**Database Check Points** This involves cross checking the database values for integrity. Ex. Check database after user registration for proper values.	0
9	**UI Components** These are usually Applets, ActiveX etc	0

Note: The above given weights are just sample weights and should not be mistaken for benchmarks.

* The standard weights range from 0 to 3.

0 ----> Not Applicable
1 ----> Simple
2 ----> Average
3 ----> Complex

Calculate the sum of the above weights and determine the complexity of each test case.

From the complexity, calculate the Test Case Points for test automation by using the table below.

Test Case Type	Complexity Weights	Test Case Points
Simple	**< 9**	**6**
Average	**10 – 16**	**8**
Complex	**> 16**	**12**

TCP-A (Single Scenario) = (# of Simple Test Cases X 6) + (# of Average Test Cases X 8) + (# of Complex Test Cases X 12)

e-biz applications need to be tested on various configurations because they can be accessed from anywhere and from an uncontrolled environment. A scenario is basically a combination of a browser, operating system and hardware.

Sometimes, the application by its nature of usage might demand testing on different scenarios. The TCPs identified above need to be modified to factor the impact of multiple scenarios.

The extra effort needed can be obtained by answering the following two questions:

- Can the scripts generated for a single scenario be run on multiple scenarios?
- Do the scripts generated for a single scenario need to be re-generated because the automation tool does not support a scenario?

If the answer to the first question is Yes, the efforts will not be increased. But if the answer to the second question is Yes, then additional effort will be expended in regenerating the scripts.

500 Glenpointe Centre West, Teaneck, NJ 07666 PH: 201-801-0233 Toll Free:1-888-937-3277 http://www.cognizant.com

To arrive at the additional Test Case Points, we will identify the test cases (from section 3.2) that need to be re-generated for every additional scenario.

The TCP-A (single scenario) will be added for every scenario for which regeneration is required.

For example, assume that the test case checks for an applet condition in Internet Explorer 5.0 and Windows NT and the TCP-A comes out to be 6. Further, the test tool does not support the same script for Netscape Navigator 4.7 and Solaris. So the script needs to be regenerated. In this case, the Total TCP-A comes out to be 12 for the 2 different scripts.

3.2 Determine TCP for Manual Execution

To determine the TCPs for Manual Execution, first calculate the manual test case execution complexity based on the factors given below.

S.No	Manual Execution Complexity Factors	Weights *
1	**Pre-conditions** This usually involves setting up the test data. It also includes the steps needed before starting the execution. E.g. to test whether the printing facility is working fine, the pre-conditions would include opening the application, inserting a record into the database, generating a report and then checking the printing facility.	2
2	**Steps in the Test Case +** If the steps themselves are complex, the manual execution effort will increase. Please refer to the legend below to determine the complexity factor.	1
3	**Verification** The verification itself might be complex. For example, checking the actual result might itself involve many steps. Let's say in a test case, we need to check the server logs, it might need opening up the log analyzer and verifying the statistics.	2

Note: The above given weights are just sample weights and should not be mistaken for benchmarks.

+

Number of Steps	Weight
<5	1
5-10	2
>10	3

* The standard weights range from 0 to 3.

0 ----> Not Applicable
1 ----> Simple
2 ----> Average
3 ----> Complex

Calculate the sum of the above weights and determine the complexity of each test case.

500 Glenpointe Centre West, Teaneck, NJ 07666 PH: 201-801-0233 Toll Free:1-888-937-3277 http://www.cognizant.com

Calculate the Test Case Points for manual execution by using the table below.

Test Case Type	Complexity Weights	Test Case Points
Simple	**< 9**	**6**
Average	**10 – 16**	**8**
Complex	**> 16**	**12**

TCP-ME = (# of Simple Test Cases X 6) + (# of Average Test Cases X 8) + (# of Complex Test Cases X 12)

Test Cases need to be manually executed in all the scenarios. To arrive at the additional Test Case Points, the TCP-ME (single scenario) will be added for every scenario for which manual execution is required.

3.3 Determine TCP for Automated Execution

To determine the TCP for Automated Execution, calculate the automation test case execution complexity based on the factors given below.

S.No	Test Case Complexity Factors	Weights *
1	**Pre-conditions** This usually involves setting up the test data. It also includes the steps needed before starting the execution. E.g. to test whether the printing facility is working fine, the pre-conditions would include opening the application, inserting a record into the database, generating a report and then checking the printing facility.	2

Note: The above given weights are just sample weights and should not be mistaken for benchmarks.

The standard weights range from 0 to 3.

0 ----> Not Applicable
1 ----> Simple
2 ----> Average
3 ----> Complex

Calculate the sum of the above weights and determine the complexity of each test case.

Calculate the Test Case Points for automated execution by using the table below.

Test Case Type	Complexity Weights	Test Case Points
Simple	**< 9**	**6**
Average	**10 – 16**	**8**
Complex	**> 16**	**12**

TCP-AE = (# of Simple Test Cases X 6) + (# of Average Test Cases X 8) + (# of Complex Test Cases X 12)

3.4 Calculate Total TCP

The Total TCP is computed by summing up the individual TCPs for Test Case Generation, Test Automation and Test Execution.

500 Glenpointe Centre West, Teaneck, NJ 07666 PH: 201-801-0233 Toll Free:1-888-937-3277 http://www.cognizant.com

TCP-T = TCP-G + TCP-A + TCP-ME + TCP-AE

The total TCP is indicative of the size of the testing project.

500 Glenpointe Centre West, Teaneck, NJ 07666 PH: 201-801-0233 Toll Free:1-888-937-3277 http://www.cognizant.com

4 EFFORT CALCULATION

To translate the Test Case Points into the total person months involved, based on your prior experience estimate the number of test case points per person month.

500 Glenpointe Centre West, Teaneck, NJ 07666 PH: 201-801-0233 Toll Free:1-888-937-3277 http://www.cognizant.com

5 GLOSSARY

TCP: Test Case Point
AUT: Application Under Test
SDLC: Software Development Life-Cycle
TCP-G: Test Case Points for Generation of Test Cases
TCP-A: Test Case Points for Automation of Test Cases
TCP-ME: Test Case Points for Manual Execution of Test Cases
TCP-AE: Test Case Points for Automated Execution of Test Cases
TCP-T: Total Test Case Points

500 Glenpointe Centre West, Teaneck, NJ 07666 PH: 201-801-0233 Toll Free:1-888-937-3277 http://www.cognizant.com