ISSN No. 0976-5697

Volume 8, No. 5, May-June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Improvement in Key Project Performance Indicators through deployment of a Comprehensive Test Metrics Advisory Tool

Mandeep Walia
PhD Research Scholar,
Dept. of Comp. Sc. & Applications
Panjab University, Chandigarh, India

Anu Gupta
Professor,
Dept. of Comp. Sc. & Applications
Panjab University, Chandigarh, India

Ravinder Kumar Singla
Professor
Dept. of Comp. Sc. & Applications
Panjab University, Chandigarh, India

Abstract: A wide variety of software metrics focusing on various levels of abstraction and attributes have been recommended by the software research community. Several organizations have implemented comprehensive metrics programs to strengthen management decision making and enable continuous improvement of their software engineering processes. In spite of the enormous efforts made to bring about advancements in the field, industry's adoption of various metrics is still at a basic level and has not changed much over past 20+ years. Moreover, the project manager community still struggles to identify the right set of metrics pertaining to their specific needs and looking for guidelines on how to make the right usage of selected metric sets.

There was a dedicated research effort made by the authors over past 3+ years to bring together various attributes of interest in software testing life cycle phases along the dimensions of effectiveness and efficiency with a special focus on the associations amongst these attributes. This research effort produced a software test metrics advisory tool, which project managers for software testing projects can depend upon as an advisory aid while making selection, usage and interpretation of various attributes and associated metric sets apt for their needs.

This paper provides a view of the improvement which was observed in key performance indicators of 30 software testing projects in an IT organization when this test metrics advisory tool was closely deployed across all these projects. The paper also provides a view of the statistical validation exercise that was conducted to prove that the deployment of the tool actually resulted in improvements in all the identified indicators.

It is highly expected that the research work undertaken by authors will contribute significantly towards filling up a crucial gap in the field of software test measures and metrics. Project managers for software testing projects will be able to effectively leverage the devised advisory tool for seeking answers to various questions running in their minds related to software measurement.

Keywords: project, performance, test, metrics

I.INTRODUCTION

Most practitioners understand and agree that there is a need to establish a software testing process that is cost effective and efficient to meet the market pressures of delivering low cost and quality software. Measurement is a key element of an effective and efficient software testing process as it helps evaluate the quality and effectiveness of the process [6]. Moreover, it helps assess productivity of the personnel involved in testing activities and enables improve software testing procedures, methods, tools and activities. Gathering of software testing related measurement data and proper analysis provides an opportunity for the organizations to learn from their past history and grow in software testing process maturity [1].

A key goal for test managers is to ensure that the software product that gets released finally is a high quality product and this assurance needs to be provided with an optimum usage of resources and costs incurred towards software testing activities [2]. During the course of software testing life cycle, test managers are tasked with making certain crucial decisions [4] viz. devise smarter test strategies to select the most appropriate subset of test cases, which holds highest potential to catch defects, optimally sequence execution of test cases such that the test set executed during initial stages holds highest probability of catching defects. Additionally, test managers are also required to make predictions about final software quality based on defect trends observed till now during the software development life cycle [7].

Hence, in a nutshell, test managers are challenged to solve below mentioned key problems as a part of their key responsibilities:

- Continuously ensure efficiency and effectiveness of software testing process and software product quality [6]
- Making decisions on the right test set selection for execution [6]
- Making decisions on an optimal sequencing of test execution [6]
- Predicting end-software quality based on the defect trends observed till now [6]

Software metrics serve as a key enabler for test managers, while they embark on the journey to make above mentioned key decisions. They surely need a guidance model or framework, which is objective in nature (based on metrics) and can quickly act as an advisory aid for them for quicker and accurate actions related to these key needs of test managers [7].

There was a dedicated research effort made by the authors over past 3+ years to bring together various attributes of interest in software testing life cycle phases along the dimensions of effectiveness and efficiency with a special focus on the associations amongst these attributes. This research effort produced a software test metrics advisory tool, which project managers for software testing projects can depend upon as an advisory aid while making selection, usage and interpretation of various attributes and associated metric sets apt for their needs.

This paper provides a view of the improvement which were observed in key performance indicators of a sample set of 30 software testing projects in an IT organization on adoption of the devised test metrics advisory tool. The paper also provides a view of the statistical validation exercise that was conducted to prove that the deployment of the tool actually resulted in improvements in all the identified indicators.

II. APPROACH TAKEN FOR DEVISING THE TEST METRICS ADVISORY TOOL

Given overall aim of the study being devising a comprehensive software testing metrics model cum tool that can be used by test managers managing software testing projects as an advisory aid, the overall research roadmap was laid down in terms of various intermediate milestones called objectives.

Below are the high level objectives, which were set to meet the research goal:

- Analysis of various phases in software testing life cycle for their associated purpose and key objectives
- Understanding of key measurable attributes in all these phases and identification of associations amongst them
- Enumeration of existing and formulation of new metrics for measuring efficiency and effectiveness aspects of the software testing process and metrics for measuring intermediate/predicting final product quality
- Analysis of situations in which those metrics need to be collected and what decisions are supported by using them
- Identification and association of customer perspectives and vendor team's perspectives with the metrics in an outsourced environment
- Lay down a set of guidelines, which software test managers can use as an advisory aid while making a choice of metrics amenable for their specific needs
- Integrate identified metrics and guidelines in the form of a comprehensive software testing metrics model
- Develop an easy to use Metrics Dashboard, which test managers can use to enhance their decision making

The research study being an exploratory and of formulative kind in nature, had a flexible design, which allows required changes based on the findings and developments during the course of the study.

The exhaustive list of attributes and corresponding metrics derived post an exhaustive literature survey and data collection exercise conducted over a wider practitioner community were consolidated in to a comprehensive metrics model. The next step in the research process was to carry out a statistical validation of the effectiveness of the devised test metrics model.

III.RESULTS VALIDATION & STATISTICAL ANALYSIS

We used a 2-pronged approach for carrying out statistical validation of the devised test metrics advisory model:

- Validation Mechanism-1: Deploy the metrics model over a sample set of software testing projects. Validate that there has been a significant improvement in the vital indicators of project performance for these projects

- Validation Mechanism-2: Gather feedback from the project managers for these projects to validate the research hypothesis and associated research questions

IV.VALIDATION THROUGH IMPROVEMENT IN KEY PROJECT PERFORMANCE INDICATORS (KPIS)

For carrying out validation through this mechanism, a set of 30 software testing projects were selected in a leading software organization. While making the selection of these projects, following project characteristics were taken in to account in order to ensure a homogeneity in the environment amongst all the projects and to ensure that there is no biasing involved:

Table I: Characteristics of short-listed Projects

Project Characteristic	Project State with respect to the characteristic
Project Type	Software testing projects being executed for market leading software product organizations in an outsourced environment
Team Structuring	Software Testing ownership lies with vendor, whereas Software Development with customer, multi-cultural teams
Project Duration	Typically 6 months, with test cycle durations of 2 weeks each
Financial Budget	(300 – 350) KUSD for 6 months period
Organizational Environment & Support	The project teams have the right environment for getting the job done. This covers everything from office space to desks and chairs to software development/testing tools
Type of Software Products under Consideration	3-tiered web-based products based on Java/J2EE technologies with comprehensive functionalities
Complexity of work	Medium to High

Communication needs	High with geographically distributed Test and Development teams
Development Methodology	Semi-agile, iterative model
Process Maturity	All projects adhering to the practices of and operating at CMMI Level-5
Team Skills	Fair
Team Size	8-12 team members including project manager, test leads and test analysts
Test Bank Size	5000 – 8000 Test Cases for each project
3rd Party Dependencies	No 3rd party dependencies as such

We then deployed the Software Testing Metrics Model over all of these projects. Prior to the roll out, the test managers managing these projects were educated through formal training sessions conducted in batches. The test managers were also provided a Single Point of Contact to help with any queries they might have during the execution of their respective test programs. Following key performance indicators of these projects were observed pre and post deployment of the Test Metrics model to gauge whether adoption of the model brought forward any significant benefits to the project performance [5]:

- Project Quality KPIs
  - Number of Defects per Function Point (FP): Having more defects per FP implies that the amount of rework in the project would be significant. Any kind of rework in the project consumes additional cost and causes project schedule realignment or slippage [5]
  - Customer Reported Issues per Function Point (FP): Here, customer can be external or internal. More customer reported defects signify that the internal quality assurance and quality control practices deployed in the project are not up to the mark [5]
- Project Cost KPIs
  - Project Budget Variance: The extent to which the actual project budget has varied from the planned one [5]
- Project Timeliness KPIs
  - Cycle Time (Calendar Days): Cycle time is the time required to complete a certain activity or task [5]
  - On-Time Completion Percentage: extent of timely completion of various project activities and tasks [5]

Summary of Observations

The observations along the chosen KPIs for all the 30 projects pre and post deployment of test metrics model were recorded. These observations were then taken through required statistical tests of analysis to validate the hypothesis. Below is a summary of the statistical outcome for each of these project performance KPIs:

- Project Quality KPIs - Number of Defects per Function Point (FP)

To validate whether there has been a significant reduction in the No. of Defects per Function Point (FP), take No. of Defects per Function Point (FP) during pre-deployment phase of test metrics model as X and the number post-deployment as Y and then taking the null hypothesis that the mean of difference is zero, we can write:

H0: m1 = m2 which is equivalent to testing H0: D = 0, where D = m1 – m2

Ha: m1 > m2 (as we want to conclude that there has been a significant reduction in No. of Defects per Function Point (FP) post deployment of test metrics model)

As we are having matched pairs, we used paired t-test and worked out the test statistic t. The paired t–test assumes that the differences between pairs are normally distributed [3]. We first tested the differences for normality using SPSS statistics using Shapiro-Wilk Test of Normality.

We use paired t-test in SPSS and work out the test statistic t as under:

Paired Samples Test									
		Paired Differences					T	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	No of Defect PRE-No of Defect POST	.58633	.41296	.07540	.43213	.74053	7.777	29	.000

As Ha is one-sided, we shall apply a one-tailed test (in the right tail because Ha is of greater than type) for determining the rejection region at 5 per cent level which comes to as under, using the table of t-distribution for 29 degrees of freedom:

R: t > 1.699

The observed value of t is 7.777 which is in the rejection region and thus, we reject H0 and conclude that the difference in no. of defects prior and post deployment of test metrics model is significant i.e., deployment of test metrics model has been actually effective.

Also, we observe that the Sig (2-Tailed) value is less than .05, we can conclude that there is a statistically significant difference between no. of defects pre and post deployment of test metrics model.

Project Quality KPIs - Customer Reported Issues per Function Point (FP)

To validate whether there has been a significant reduction in the No. of Customer Reported Issues per Function Point (FP), take No. of Customer Reported Issues per Function Point (FP) during pre-deployment phase of test metrics model as X and the number post-deployment as Y and then taking the null hypothesis that the mean of difference is zero, we can write:

H0: m1 = m2 which is equivalent to testing H0: D = 0, where D = m1 – m2

Ha: m1 > m2 (as we want to conclude that there has been a significant reduction in No. of Defects per Function Point (FP) post deployment of test metrics model)

As we are having matched pairs, we used paired t-test and worked out the test statistic t. The paired t–test assumes that the differences between pairs are normally distributed [3]. We first tested the differences for normality using SPSS statistics using Shapiro-Wilk Test of Normality.

We used paired t-test in SPSS and work out the test statistic t as under:

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	CUSTDEFECTSPRE - CUSTDEFECTSPOST	.29983	.21510	.03927	.21951	.38015	7.635	29	.000

As Ha is one-sided, we shall apply a one-tailed test (in the right tail because Ha is of greater than type) for determining the rejection region at 5 per cent level which comes to as under, using the table of t-distribution for 29 degrees of freedom:

R: t > 1.699

The observed value of t is 7.635 which is in the rejection region and thus, we reject H0 and conclude that the difference in no. of customer reported defects prior and post deployment of test metrics model is significant i.e., deployment of test metrics model has been actually effective.

Also, we observe that the Sig (2-Tailed) value is less than .05, we can conclude that there is a statistically significant difference between no. of customer reported defects pre and post deployment of test metrics model.

Project Cost KPIs - Project Budget Variance

To validate whether there has been a significant reduction in Percentage of Project Budget Variance, let us take Percentage of Project Budget Variance during pre-deployment phase of test metrics model as X and the Percentage of Project Budget Variance post-deployment as Y and then taking the null hypothesis that the mean of difference is zero, we can write:

H0: m1 = m2 which is equivalent to testing H0: D = 0, where D = m1 – m2

Ha: m1 > m2 (as we want to conclude that there has been a significant reduction in Percentage of Project Budget Variance post deployment of test metrics model)

As we are having matched pairs, we used paired t-test and worked out the test statistic t. The paired t–test assumes that the differences between pairs are normally distributed [3]. We first tested the differences for normality using SPSS statistics using Shapiro-Wilk Test of Normality.

We used paired t-test in SPSS and work out the test statistic t as under:

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	BUDGETVARPRE - BUDGETVARPOST	2.92333	2.07334	.37854	2.14913	3.69753	7.723	29	.000

As Ha is one-sided, we shall apply a one-tailed test (in the right tail because Ha is of greater than type) for determining the rejection region at 5 per cent level which comes to as under, using the table of t-distribution for 29 degrees of freedom:

R: t > 1.699

The observed value of t is 7.723 which is in the rejection region and thus, we reject H0 and conclude that the difference in Percentage of Project Budget Variance prior and post deployment of test metrics model is significant i.e., deployment of test metrics model has been actually effective.

Also, we observe that the Sig (2-Tailed) value is less than .05, we can conclude that there is a statistically significant difference between Percentage of Project Budget Variance pre and post deployment of test metrics model.

Project Timeliness KPIs - Cycle Time (Calendar Days)

To validate whether there has been a significant reduction in Cycle Time, let us take Cycle Time during pre-deployment phase of test metrics model as X and the Cycle Time post-deployment as Y and then taking the null hypothesis that the mean of difference is zero, we can write:

H0: m1 = m2 which is equivalent to testing H0: D = 0, where D = m1 – m2
Ha: m1 > m2 (as we want to conclude that there has been a significant reduction in Cycle Time post deployment of test metrics model)

As we are having matched pairs, we used paired t-test and worked out the test statistic t. The paired t–test assumes that the differences between pairs are normally distributed [3]. We first tested the differences for normality using SPSS statistics using Shapiro-Wilk Test of Normality.
We use paired t-test in SPSS and work out the test statistic t as under:

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	CYCLETIMEPRE - CYCLETIMEPOST	5.90000	4.26170	.77808	4.30866	7.49134	7.583	29	.000

As Ha is one-sided, we shall apply a one-tailed test (in the right tail because Ha is of greater than type) for determining the rejection region at 5 per cent level which comes to as under, using the table of t-distribution for 29 degrees of freedom:
R: t > 1.699
The observed value of t is 7.583 which is in the rejection region and thus, we reject H0 and conclude that the difference in Cycle Time prior and post deployment of test metrics model is significant i.e., deployment of test metrics model has been actually effective.
Also, we observe that the Sig (2-Tailed) value is less than .05, we can conclude that there is a statistically significant difference between Cycle Time pre and post deployment of test metrics model.

Project Timeliness KPIs - On-Time Completion Percentage
To validate whether there has been a significant improvement in On-Time Completion Percentage, let us take On-Time Completion Percentage during pre-deployment phase of test metrics model as X and On-Time Completion Percentage post-deployment as Y and then taking the null hypothesis that the mean of difference is zero, we can write:
H0: m1 = m2 which is equivalent to testing H0: D = 0, where D = m1 – m2
Ha: m1 < m2 (as we want to conclude that there has been a significant improvement in On-Time Completion Percentage post deployment of test metrics model)

As we are having matched pairs, we used paired t-test and worked out the test statistic t. The paired t–test assumes that the differences between pairs are normally distributed [3]. We first tested the differences for normality using SPSS statistics using Shapiro-Wilk Test of Normality.
We use paired t-test in SPSS and work out the test statistic t as under:

Paired Samples Test

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	ONTIMECOMPCPRE - ONTIMECOMPCPOST	-6.30000	5.10679	.93237	-8.20691	-4.39309	-6.757	29	.000

As Ha is one-sided, we shall apply a one-tailed test (in the left tail because Ha is of less than type) for determining the rejection region at 5 per cent level which comes to as under, using the table of t-distribution for 29 degrees of freedom:
R: t < -1.699
The observed value of t is -6.757 which is in the rejection region and thus, we reject H0 and conclude that the difference in On-Time Completion Percentage prior and post deployment of test metrics model is significant i.e., deployment of test metrics model has been actually effective.

Also, we observe that the Sig (2-Tailed) value is less than .05, we can conclude that there is a statistically significant difference between On-Time Completion Percentage pre and post deployment of test metrics model.

V. VALIDATION THROUGH MANAGER FEEDBACK

As a confirming validation mechanism, we gathered feedback from project managers for all the 30 shortlisted software testing projects to validate the questions related to research hypotheses.

Cross-questions to validate following research questions were posed for investigation amongst the test project managers:

R1: Which are the attributes of interest associated with various phases of software testing life cycle that are measurable?

R2: Which are the different types of associations amongst those attributes?

R3: What are the possible metrics and indicators that can be associated with the identified attributes?

R4: Which are the typical characteristics of projects in which certain metrics should be applicable?

R5: What should be an optimum set of metrics to be adopted by a project given its environment and performance requirements?

R6: How the answers to all of above questions be integrated together in the form of a comprehensive metrics model?

A questionnaire based on five-point Likert scale was used for conducting a survey amongst these project managers.

The data on the ordinal scale was transformed in to a numeric form for carrying out a statistical analysis.

We used below mapping for carrying out this transformation:

Strongly Agree = 5; Agree = 4; Neither Agree nor Disagree = 3; Disagree = 2; Strongly Disagree = 1

We also carried out an analysis of descriptive statistics on the result variable “FINALOPINION” (the variable “FINALOPINION” represents the mean of the responses from various respondents against each of the questions) against is the mean and observe below results:

Descriptives

			Statistic	Std. Error
VAR00001	Mean		4.3833	.07064
	95% Confidence Interval for Mean	Lower Bound	4.2389	
		Upper Bound	4.5278	
	5% Trimmed Mean		4.3704	
	Median		4.5000	
	Variance		.150	
	Std. Deviation		.38693	
	Minimum		4.00	
	Maximum		5.00	
	Range		1.00	
	Interquartile Range		.50	
	Skewness		.441	.427
	Kurtosis		-1.160	.833

We observe that the mean value 4.3833 is way above 4, which is the equivalent of Likert scale choice “Agree”. In addition, standard deviation is at minimal levels of .38693.

Hence, we conclude that, overall final opinion is in favor of the software test metrics model.

VI. CONCLUSION

Software testing life cycle is a phased approach to execute various quality assurance and quality control activities to ensure a high quality software product/application with optimized costs and a reduced time to market. Each of the stages in the life cycle have their own purpose and associated objectives. Each of these objectives have their underlying attributes of interest. A systematic study has been carried out of various phases in the software testing life cycle and an exhaustive list of objectives and attributes is laid down for each of these stages. All the identified attributes and metrics have been consolidated in the form of a software test metrics advisory tool. The tool, when adopted enabled test managers managing software testing projects choose an appropriate subset of these objectives and associated attributes based on the specific needs of the project environment in which they operate.

REFERENCES

[1] Afzal, Wasif, "Metrics in software test planning and test design processes." (2007).

[2] Ammann, Paul, and Jeff Offutt. Introduction to software testing. Cambridge University Press, 2008.

[3] http://www.dmi.units.it/~inverniz/adir/statpaired.html
(General theoretical reference)

[4] Lazic, Ljubomir, and Nikos Mastorakis, "Cost effective software test metrics." WSEAS Transactions on Computers 7.6 (2008): 599-619.

[5] Roper, Marc, “Software testing”, McGraw-Hill, Inc., 1995.

[6] Pressman, Roger S, "Software engineering: a practitioner's approach", Palgrave Macmillan, 2005.

[7] Walia Mandeep, Gupta Anu and Singla Ravinder Kumar, "Misapplications of Well-known Test Metrics",
International Journal of Software and Web Sciences, 12(1), March-May, 2015, pp. 65-75