Volume 4, Issue 8, August 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Framework for Evaluating Importance of Quality Assurance Metrics in Software Project Management

Akrati Koul*
CSE Department, WCAS,
Muscat, India

Dr. Sushila Madan
Associate Prof. LSR College,
Delhi University, India

Abstract— Software metrics provide a quantifiable basis for measuring the progress of a project. It also aids in planning and predicting the future of the software development projects. Therefore the quality of software can be controlled and improved easily. Good Quality leads to higher productivity, which has brought software metrics to the forefront.

Over the time, many metrics have been developed; resulting in continuous improvement in the arena of successful software project management [2].This paper examines the realm of software engineering to evaluate the impact of software metrics on software quality. A real time implementation of metrics has been conducted in a software organization giving an insight regarding the various software development metrics that are commonly followed.

These experiences can yield tremendous benefits and betterment in software product quality and reliability [3].

Keywords— Software metrics; Software quality; Customer Satisfaction; Statistical tools; Metrics Analysis; Quality Assurance metrics

I. INTRODUCTION

Project management is a methodical approach to planning and guiding project processes from start to finish. According to the Project Management Institute [10], the processes are guided through five stages: initiation, planning, executing, controlling and closing. Project management can be applied to almost any type of project and is widely used to control the complex processes of software development projects. There is no magic formula for ensuring that a project is successful but there are well proven techniques available to help plan and manage projects [11] . Project management has emerged as a vital discipline for successful projects in the competitive world.

According to Dr. Kerzner, "There are three main interdependent constraints for every project; time, cost, and scope. This is also known as Project Management Triangle" [1]. Project Management Success depends on balancing the core project components of Scope, Cost and Time. The project that can achieve the balance on these components shall definitely deliver good quality products.

The International standards ISO9000 [4,5], IEEE [6] and Baldrige [7] emphasize on customer perceived quality and expect that customer satisfaction be strongly linked to all functions of a business [8].

II. BASICS

A. Quality Assurance Metrics

A metric can be defined as any type of measurement used to gauge some quantifiable component of performance in a project. [12]

Metrics are collected for measuring the progress of a project against its planned effort, schedule, cost, resource usage, and error rates, and of establishing a baseline which will aid in planning and forecasting future projects.

Metrics collected and reported for software projects include effort variance, schedule variance, productivity, defect density etc.

Variance is the difference between what was planned and the actual value. Two variance values related to the project tracking method known as "Earned Value" are the "cost variance" and the "schedule variance", both of which are expressed in monitary terms. If effort variance is defined in a similar manner as the other two, then it would be the difference between the Effort actually expended to a given date less the Effort planned to be expended by the same date.

Effort Variance: The purpose of the metric is to measure the amount of effort being spent on the project. This is

accomplished by illustrating the amount of actual hours worked compared to the amount of planned hours and displaying their variance. All time (including all overtime) must be logged for the projects.

Effort Variance = [Actual Effort - Estimated Effort]/Estimated Effort

Schedule Variance: The schedule variance is computed as the ratio between estimated and actual elapsed time. The difference is calculated for both original as well as revised estimates. The calculation is performed at end of the module level phase-wise, and also at the project level.

Schedule Variance = [Actual Elapsed time - Estimated Elapsed time]/Est. Elapsed time

Productivity: It is the ratio of output to input. Lines of code per developer month, Function Points per developer month, Test cases executed/tester month. The productivity metric is defined as the ratio of the size of the code developed, to the effort in person days required to develop the product. Effort is counted from the commencement of the project to client shipment including software engineering, testing, and reviews, re-work and project management.

Productivity = Size (LOC or FP) / Effort (in person days)

Defect density: It is the number of defects divided by some measure of size. The residual defect density is the number of defects remaining in a product at any stage if its development divided by the size.

Defects density can be calculate in three perspectives i.e.

In Process Defect Density: defects that are captured before sending the delivery to the client. These defects are sum total of review defects/ observations and testing defects.

The formula for calculating in process defect density is: ***Total Pre Shipment Review Defects/ observations + Total Testing Defects / Size (size can either be in FP or CSC)***

Delivered Defect Density: defects that are captured/ reported after the delivery is sent to client are called delivered defects.Delivered defect density should be calculated for defects identified in the following two stages:

Defects found during implementation and acceptance testing and After acceptance - during a specified period of time, say quarterly, [subsequent to the acceptance of the software product by the customer].

The formula for calculating in delivered defect density is: ***Total Delivered Defects / Size (size can either be in FP or CSC)***

Weighted Defect Density: weighted defect density is calculated considering the severity of review defects/ observations, testing defects and client reported defects. It is calculated for the entire engagement. Defects can be categorized as Fatal, Major and Minor.

The weights are:

Fatal defect/ observation – 10,

Major – 5,

Minor – 1 and

Client reported defects/ delivered defects – 15 (irrespective of the severity of the defect).

The formula for calculating weighted defect density is: **(Total no of delivered defects * 15 + Total no of fatal defects * 10 + Total no of major defects * 5 + Total no of minor defects *1) / Size (size can either be in FP or CSC)**

III. EVALUATION

A. Implementation

The Quality Assurance metrics are an integral part of the Quality Management Plan. The purpose of Quality Management Plan is to facilitate the project and thereby ensure that the review of software products is planned so as to verify the compliance with the applicable procedures and standards. This procedure also aims to track quality assurance and quality control activities in the project.

This process is used to quantitatively manage the engagement execution so as to achieve the projects established quality and process performance objectives as defined by the organization.

TABLE I:
PHASE WISE DISTRIBUTION OF PERCENTAGE EFFORTS ON SOFTWARE PROJECTS ACROSS THE ORGANIZATION

Activities / Scope	**Project**	**Issue/Task Based**
	SDLC	**SDLC**
Planning & Tracking* (PMP+QMP+Confi.Mgt)	20.11	16.38
SRS	9.46	4.12
Design (HLD + LLD)	10.11	10.14
Coding	40.12	26.99
Testing (Test Plan + Test case + UT+IT+ST)	15.12	23.99
Prototyping	0.84	0
Deployment	1.2	8.24
Maintenance / Warranty / Support	3.04	10.14

Note:
These percentages are derived using the data from all the closed Development and Issue/task based engagements for all the practices across the organization.
*** Planning and tracking effort includes, time spent on research (for the engagement), waiting / idle time for response / query resolution from the client**.

The main steps in the metrication initiative include:

- Define and collect primary, additional, and derived metrics for the projects.
- Analyze the metrics data to set organizational baselines for each engagement category.
- Establish and monitor goals and control limits at organization level for each engagement category.
- Establish and monitor project level goals and control limits for each engagement.
- Review the goals and control limits periodically.
- Review the metrics procedure periodically to add/delete defined metrics.

TABLE II:
METRICS COLLECTED FROM ORACLE PROJECTS

	Engagement Type	Center Line	UNCL	LNCL
Effort Variance	Issue/Task Based Development	0	0	0
	Issue/Task Based Modifications	-0.55	2.4	-3.5
	Projects	14.16	66.9	-38.58
Schedule variance	Issue/Task Based Development	-2	2.3	-5.3
	Issue/Task Based Modifications	0	1	-1.9
	Projects	-2.5	109.69	-114.68
Productivity	Issue/Task Based Development (CSC / Person day)	7.9	36	0
	Issue/Task Based Modifications (CSC / Person day)	16.3	66	0
	Projects (FP / Person day)	1.83	2.8	0.85
Defect Density	Projects (Defects / FP)	30.54		

*** UNCL: Upper Natural Control Limit, LNCL: Lower Natural Control Limit**

Note:
1. Control Limits for Effort and Schedule Variance for project type of Engagement are carried forward from the previous Organizational
Baseline Report as only one project closed during the period Jan 2013 to April 2013.
2. Productivity for Project type of Engagement is carried forward from previous Organizational Baseline Report as data is not available.
3. Defect Density for Issue/Task based and Project type of Engagement is carried forward from the previous Organizational Baseline Report as the data is not available.

B. Metrics Analysis

Data collated shall be evaluated to identify the underlying reasons for the results obtained to devise corrective and preventive actions for the project(s)/functions and the organizations.

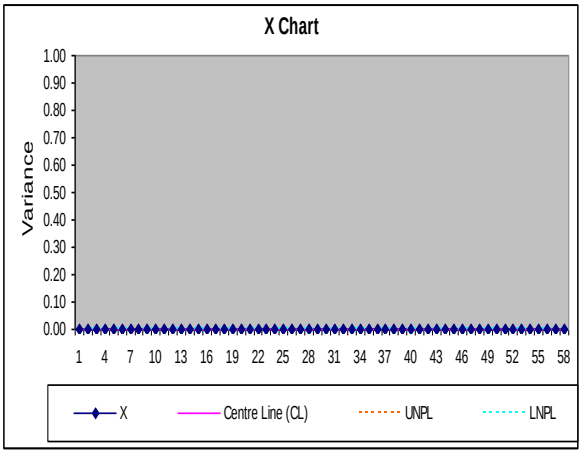
The relevance of data towards 'Organizational Baseline Report' (OBR) generation shall be considered. Data from projects that are one of its kinds may not be taken into consideration.

Some of the statistical tools shall be used for analysis, namely:

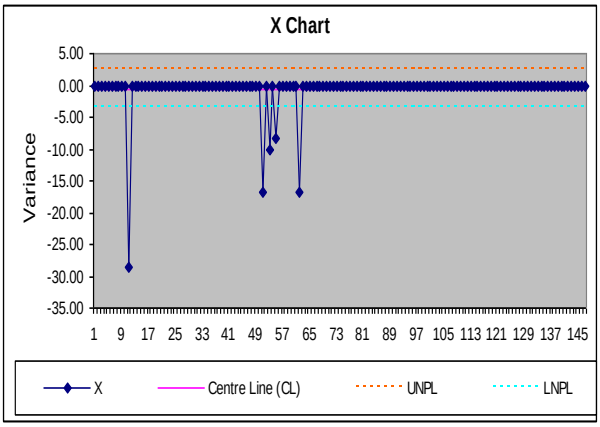
- Pareto Analysis
- Control Charts
- Scatter Diagrams
- Cause & Effect Diagram
- Run Chart
- Brainstorming

For the Oracle projects in the above said organization, Control Charts have been plotted to provide a statistical view of some of the metrics results. A control chart shows the effects of alterations to the process and helps to correct any errors in real time. One can also predict the range of possible future results.

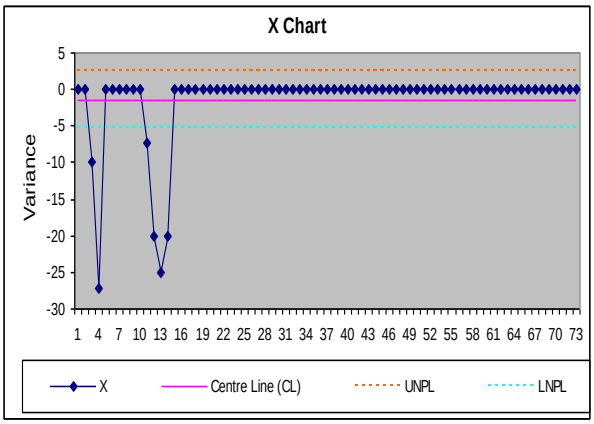
Effort Variance – Development Issues / Tasks – X chart



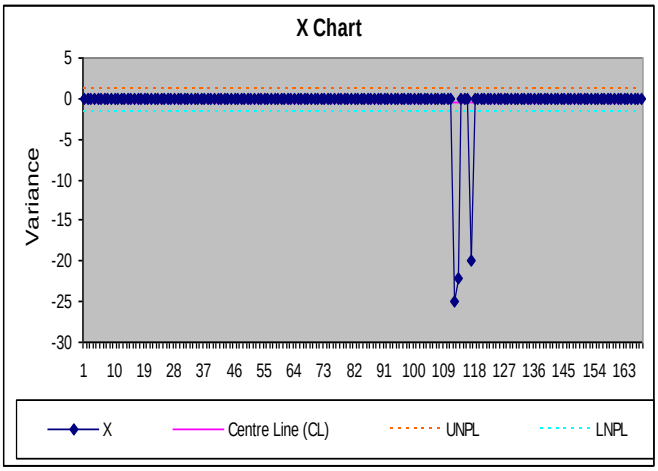
Effort Variance – Modification Issues / Tasks – X chart



Schedule Variance – Development Issues / Tasks – X chart



Schedule Variance – Modification Issues / Tasks – X chart



All the requirements for the project shall be recorded and tracked. The status of each requirement shall be tracked and monitored for completeness using Traceability matrix. The total number of requirements received at the start of the project and the number of requirement changes received during the execution of the project shall be monitored. The number of requirements changes, and the planned and actual effort for these is summarized and status shared with senior management on periodic basis.

- Test Cases

Number of test cases prepared for each level of testing (UT, IT, ST) vis-à-vis the size of the source code, will indicate the sufficiency of testing conducted.
Number of test cases executed/passed/failed shall also be summarized.

- Test Cycles

This measure assesses the adequacy of previous testing phases. The number of test cycles required for integration and system testing phases in the project life cycle shall be planned and documented in the Project Management Plan. The actual number of test cycles performed shall be recorded in the Test Summary Log.

- Risks

The risk assessment for the project shall be done and documented in the Risk Management plan (part of PMP), starting from project initiation. The actual risks encountered during project execution and their impact on schedule/effort, along with the mitigation strategies shall be recorded and monitored throughout the Project life cycle.
Major risks with high impact are shared/escalated to senior management periodically.

- Change Requests

The number of change requests initiated, the number of change requests processed, the number of closed and open CRs, and the effort estimated and expended in incorporating the changes shall be collated.

- Issue/problem Reports

The number of IPRs (Issue Problem Reports) that were logged by the client/user and the effort spent on their resolution shall be recorded.

- Client Satisfaction Levels

Client feedback shall be solicited through a client feedback form at the end of the each project. The Delivery partner/Head MIS/nominated authorities shall send the Client Feedback form to the client. The feedback thus received shall be collated and analysed by MIS.

IV. CONCLUSIONS

When we measure something, we can comprehend it in a better way and can further improve upon it. Metrics help in gauging the progress, quality and health of a software project life cycle. Metrics can also be leveraged to evaluate past performance, current status and envisage future trends. Effective metrics are simple, objective, and measureable and have easily accessible underlying data. [13]

ACKNOWLEDGMENT

This work has been partially supported by the real time examples and samples taken up from Software projects and Issue task based engagements of Computer Associates Technologies, Muscat, Oman. CA Technologies, Inc. is a Fortune 500 company and one of the largest independent software corporations in the world. [9]

REFERENCES

[1] B.Boehm, R.ross, *theory-w software project management; Principles and examples*, IEEE Transactions on software engineering

[2] Program Management Office (PMO), *Project Management Metrics Guidebook*, Revision date Feb 23, 2001, Revision 1.0

[3] Stephen. H. Kan, *"Metrics and methods in software quality engineering*", Second edition, Addison Wesley publication, Sep 20, 2002

[4] ISO, '*Quality Management and Quality Assurance Standards'*, International Standard, ISO/IEC 9001:1991

[5] Ince, D,*'ISO 9001 and Software Quality Assurance*', Quality Forum, McGraw Hill, isbn: 0-07-707885-3, 1994

[6] IEEE, '*Standard for a Software Quality Metrics Methodology'*,P-1061/D20,IEEE Press, New York, 1989

[7] Brown G,'Baldrige Award Winning Quality: How *to Interpret the Malcolm Baldrige Award Criteria*', Milkwaukee, Wi: ASQC Quality Press,1991

[8] Xenos M., Christodoulakis D., *Measuring perceived software quality*, Information and Software Technology Journal, Butterworth Publications, Vol.39, Issue 6, June 1997.

[9] http://en.wikipedia.org/wiki/CA_Technologies

[10] http://searchcio-midmarket.techtarget.com/definition/project-management

[11] http://www.ruskwig.com/project_management.htm

[12] http://www.cprime.com/community/articles/metrics_in_pm.html by Crystal Lee, PMP

[13] http://www.infosys.com/engineering-services/white-papers/Documents/comprehensive-metrics-model.pdf

