www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 6 June, 2014 Page No. 6742-6745

Review on Software Testing Metrics

[1]Ms. Neha Thapar , [2] Mr. Anil K Mishra

[1]M.tech, Department of Software Engineering Institute of Technology & Management University.

nehathapar2906@gmail.com

[2] Assistant Professor, Department of Computer Science & Engineering

Institute of Technology & Management University,

Abstract:

Metrics are defined as standards of measurement. Test metrics provides the visibility into the readiness of the product, and gives clear measurement of the quality and completeness of the product. Metrics allow for deeper understanding of the performance of the application and its behavior. Metrics help organization to obtain the information it needs to continue to improve its productivity, reduce errors and improve acceptance of processes, products and services and achieve the desired goal. This paper focuses on various types of testing metrics, and introduces the most commonly used software metrics and projected their use in constructing models of the software development process.

Keywords: Software testing, Software testing metrics.

I. INTRODUCTION

Software Metrics is a Measurement Based Technique which is applied to processes, products and services to supply engineering and management information and working on the information supplied to improve processes, products and services, if required. Software Testing Metrics are experimental measures that could be used to measure different characteristics of a software system or the software development process. In recent years software testing technologies have emerged as a dominant software engineering practice which helps in effective cost control, quality improvements, time and risk reduction etc.
We need metrics because it helps us in understanding the type of improvement required and helps in taking decisions on process or technology change. The importance of metrics is to improve the quality and productivity of products and services through customer satisfaction. Metrics provides enhancement for current processes.

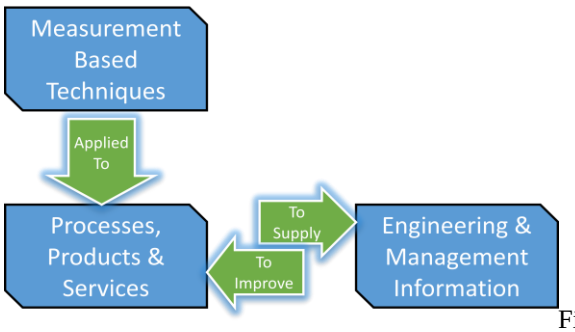

Fig. 1: Software Metrics

Important Factors of Testing Metrics are:
1. Only that data is composed that needs to take decisions or to amend approach.
2. The data which is imprecise must be avoided and complex data must be handled accurately.
3. Decisions are not based exclusively on data that is a variable or can be manipulated.
4. Metrics should be selected on the significance of stakeholders rather than the ease of data collection.
5. Difficult metrics data must be handled vigilantly.

The Software Metrics that the QA team produced are concerned with the test activities that are part of the Test Phase and so are formaly known as Software Testing Metrics.
QA recognises two subsets of Software test metrics:

Fig.2: Breakdown of Software Metrics

Test Process Metrics: These procedures provide information about preparation for testing, test execution and test progress. They don't provide information about the test state of the product and are mostly of use in measuring advancement of the Test Phase.
Test Product Metrics: These measures provide information about the test state of the product and are generated by test execution and code fixes. Using these metrics we can

establish the products test state and indicative level of quality, useful for product release decisions.

1.1 SOFTWARE TESTING METRICS LIFE CYCLE

The procedure of setting up the metrics life cycle is:
1. Categorize the metrics
2. Prioritizing metrics
3. Classifying the metrics that may be project specific
4. Identifying the required data for metric/setup process to confine the data
5. Communicating the stakeholders
6. Capturing and verifying data
7. Analyzing and dispensation of data
8. Reporting

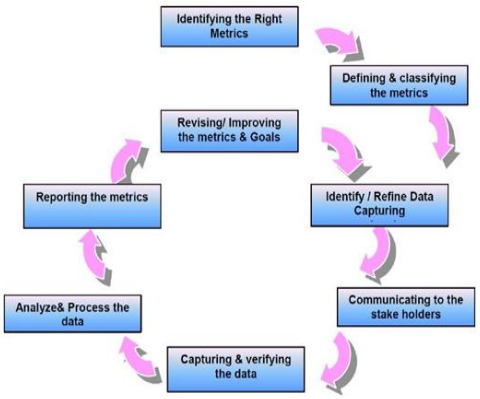

Fig.3 Software Metric Lifecycle

Step1: Identifying the right metrics can be done by:
a) Deciding the spectators.
b) Classify the metrics which confine the status of each type of testing.
c) Setting up easy mechanisms for data collection.
d) Evaluate the value of each metric.
e) Discover the problem or area where necessity is required.
Step2: Define and Classify metrics:
a) Provide definition of each metric.
b) Characterize goal for each metric.
c) Verify that goals are realistic by comparing them with industry standards.
Step3: Identify the data capturing mechanism:
a) The data is classified into base and derived metrics.
Base metrics: Metrics for which data can be captured directly.
Derived metrics: Derive from base metrics.
b) After identifying the metrics, the source of data for each base metrics must be identified.
c) Characterize the common outline for capturing all base metrics.
d) Obtaining feedback from the team which captures the data.
Step4: Communication:
a) Communicate the need for metrics to all pretentious teams.
b) Obtain feedback from stakeholders.
c) Prepare the testing team concerning the data points that needs to be captured for generating the metrics.
Step 5: Capture and Verify data.
a) Ensure that the data capturing mechanism is set up and rationalized.
b) Proper guidelines are to be given to the team members on the data that is essential.
c) Set up verification points to ensure that all data is required.
d) Confine the data that is easily accessible to all team members.
Step 6: Analyze and Process the data:
a) Once the data is being captured, the data needs to be analyzed for completeness.
b) Authenticate whether the data is filled accurate and up-to-date.
c) Calculate all the metrics based on the base metrics.
d) Verify whether the metrics are conveying the right information.
Step 7: Reporting Metrics:
a) To obtain feedback from stakeholders.
b) Metrics should be presented based on the audience and in a steady format.
c) Reports should contain the review of annotations.
d) It should clearly point out all the issues.
e) Based on request, user should be able to access the data.
Step 8: Improving the metrics:
a) Continuous upgrading is the key to success for any process.
b) Frequently accumulate feedback from stakeholders.
c) Metrics report must be accessible to everyone.
d) Evaluate new metrics to capture.
e) Refine report template.

II. Types of Testing Metrics

There are various types of testing metrics:
1. Base Testing Metrics
2. Calculated Testing Metrics
3. Organizational Testing Metrics
4. Project Testing Metrics
5. Process Testing Metrics
6. Product Testing Metrics

BASE TESTING METRICS

It is also recognized as direct measure. A direct metric is a metric that does not depend upon a measure of any other attribute. It constitutes the unrefined data gathered by the test engineers all through the testing effort. They also provide project status reports to the test lead and to the project manager and also provide the input data to feed into the formulas used to derive calculated metrics. Example: test cases.

CALCULATED TESTING METRICS

It is also recognized as indirect measure. They convert the base metrics data into useful information. Calculated testing metrics are equipped by the test lead and is used to trail the progress of the development at different levels like at module level, tester level. They also provide some valuable information that when used and implemented often times lead to considerable improvements in overall SDLC.

ORGANIZATIONAL TESTING METRICS

Metrics at the level of organization are useful in overall project scheduling and management. Some of these metrics are obtained by aggregating attuned metrics across multiple projects. Organizational level test plan needs to integrate

metrics to make the testing activities discernible obtainable to process improvements.

PROJECT TESTING METRICS

Project test metrics relates to precise project. These are useful in monitoring and control of a explicit project. The ratio of actual to intended system test effort is one project metric.

The metrics process steps are explained as:

1. Identify metric patrons.
2.Target goals.
3, Ask questions allied to the metrics.
4. Select metrics.
5.Standardize definitions.
6.Choose a model.
7.Estbalish counting criteria.
8. Decide on decision criteria.
9.Define the reporting mechanisms.
10. Establish additional qualifiers.
11. Accumulate data.
12. Consider human factors.

At the start of the system test phase, for example, the project manager estimates the total system test effort. The ratio of actual to estimated effort is zero prior to the system test phase. This ratio builds up over time. Tracking the ratio assists the project manager in allocating testing resources.

Project testing metrics have do's and don't . They are stated as:

Do's :

1. Provide feedback.
2.Select metric based on goals.
3. Focus on processes, products and services.

Don't:

1. Ignore the data.
2. Use only one metric.
3. Measure individuals.

PROCESS TESTING METRICS

Reliable process metrics need to quantify aspects of the process/output of the process that are allied to business strategy and corporate goals. The big-bang approach is one process sometimes used in relatively small single-person projects. Process metrics by themselves can provide for:

1. Establishment of a baseline of performance.
2. Tracking of performance.
3. Communication about performance.
4. Identifying areas for improvement.

There are various key steps in developing process metrics:

1. Define the boundaries of the process.
2. Define the output of the process- what explicitly is produced?
3. Who is the stakeholder of the process and what are the specific requirements?
4. Who are the suppliers to the process and what inputs they deliver?
5. Develop process metrics based on stakeholder requirements.
6. Ensure that data for the process metrics is obtainable and consistent.
7. Design a process that will create an output for the stakeholders and is linked to organizational goals.

PRODUCT TESTING METRICS

Product metrics relates to specific product such as compiler for programming language. These are useful in making decisions related to the product. Product metrics illustrate the uniqueness of the product such as size, complexity, design features, performance, and quality level. Product metrics characterize some aspect of the structure of a software product, such as a requirements specification, a design, or source code. They are also commonly known as complexity metrics.

To develop good product metrics:

1. Define test coverage and quality related objectives for the product.
2. Consider questions about the effectiveness, efficiency with which the product is achieving those objectives.
3. Formulate quantifiable metrics, for each effectiveness and efficiency.
4. Determine rational goals for each metric, so that there is high level of confidence in the quality.
5. Monitor evolution towards goals and formative product status.

The product testing metric is categorized into:

Static Metric: Static metrics are those computed without having to execute the product. Example: Number of testable entities in an application.

Dynamic Metric: It requires code execution. Example: The average number of testers working on a project is a static project metric. Number of defects remaining to be fixed could be treated as dynamic metric as it can be computed accurately only after a code change has been made and the product retested.

III. COMMON METRICS FOR ALL TYPES OF TESTING

1. Effort Variance (EV)

This metric gives the variance in the estimated effort.

$$EV=\left[\left(\frac{actual\ effort-estimated\ effort}{estimated\ effort}\right)*100\right]\%$$

2. Schedule Variance (SV)

This metric gives the variance in the estimated schedule i.e. number of days.

$$SV=[\left(\frac{actual\ no.of\ days-estimated\ no.of\ days}{estimated\ no.of\ days}\right)*100]\%$$

3. Scope Change(SC)

This metric indicates how stable the scope of testing is.

$$SC=\left[\left(\frac{Total\ scope-previous\ scope}{previous\ scope}\right)*100\right]\%$$

IV. LIMITATIONS OF METRICS

1. Many company avoid metrics because its time consuming and expensive.
2. Metrics has to be equipped with full indulgence about the concepts of metrics and details of projects. Metrics has direct impact on the project as it tells the user about the progress in project.
3. It's very difficult to sustain and keep track of the metrics.

V. CONCLUSION

In this paper, we have attempted to give an overture concerning software metrics and software testing metric types. Metric is the keystone in evaluation and establishment for any business enhancement. It is a Measurement Based Technique which is applied to processes, products and services to supply engineering and

management information. This paper introduces the most commonly used software metrics anticipated and reviews their use in constructing models of the software development process. It indicates level of patron contentment, easy for organization to assimilate number and drill down, whenever required and act as monitor when the process is going out-of-control.

VI. REFERENCES

[1] Vikas Verma and Sonal Malhotra: "Applications of software testing metrics in constructing models of software development process", Journal of global research in computer science, May 2011.

[2] Mr. Premal B. Nirpal, Dr. K. V. Kale: "A brief overview of software testing metrics", International Journal on Computer Science and Engineering (IJCSE), Jan 2011.

[3] Arvind Gopu :"Software metrics and associated issues", Dec 2003.

[4] Mark Crowther: "Software test metrics", Empirical Pragmatic Testing.

[5] Vishal Kotekar: "Basics of software testing metrics", Software testing magazine, 2013.