BALANCED TESTING SCORECARD
A Model for Evaluating and Improving the Performance of Software Testing Houses

Renata Alchorne[1,2], Rafael Nóbrega[1,2], Gibeon Aquino[1], Silvio Meira[1] and Alexandre Vasconcelos[1]
[1] *Informatics Center of Federal University of Pernambuco, Recife, PE, Brazil*
[2] *Inmetrics, São Paulo, SP, Brazil*

Keywords: Performance measurement, Balanced scorecard, Software testing.

Abstract: Many companies have invested in the process of testing to improve the test team's performance. Although Testing Maturity Models aim at tackling this issue, they are unpopular among the many of the highly competitive and innovative companies because they encourage displacing the true goals of the mission of "achieving a higher maturity level". Moreover, they generally have the effect of blinding an organization to the most effective use of its resources, as they focus only on satisfying the requirements of the model. This article defines the Balanced Testing Scorecard (BTSC) model. This aims to evaluate and improve the test team's performance. The model, based on Balanced Scorecard and Testing Maturity Models, is capable of aligning clients' and financial objectives with testing maturity goals in order to improve the test team's performance and client's performance. The model was based on and developed from the specialized literature and was applied in a software testing house as a case study.

1 INTRODUCTION

According to Hutcheson (2003), the test effort has to provide not only the proof of performance, but also to show that it adds enough value to the product to justify its budget. Testing Maturity Models are used to evaluate and improve the performance of test teams. However, this approach focuses on measures, analysis and actions to improve the process of testing (Sogeti, 2008, Burnstein, Suwanassart & Carlson, 1996), without aligning them with the mission or future vision of the organization.

According to James Bach (1994), Maturity Models encourage the displacement of goals from the true mission to the artificial one of achieving a higher maturity level, which generally has the effect of blinding an organization to the most effective use of its resources. According to Kaplan & Norton (2004), it is common to find indicators of internal processes, such as those derived from testing maturity models, which are not related to the value that internal and external customers have been advised to expect.

By using a model for evaluating and improving performance that occasions the alignment of goals from different perspectives (those of finance, customers, and processes), the management of test teams can be optimized. This is because it allows an the organization to improve its processes to meet its clients' needs and its own strategic planning, thereby increase its overall performance.

This article defines the Balanced Testing Scorecard (BTSC) model, which aims to evaluate and improve a test team's performance. The model, based on Balanced Scorecard and Testing Maturity Models, enables clients' and financial objectives to be aligned with testing maturity goals so as to improve the test team's performance.

2 BALANCED TESTING SCORECARD

Balanced Testing Scorecard (BTSC) was based on the Generic Strategy Map put forward by Kaplan and Norton (2004).

BTSC consists of two main components: a map for the strategic plan and a procedure for customization of the processes.

Alchorne R., Nóbrega R., Aquino G., Meira S. and Vasconcelos A. (2010).
BALANCED TESTING SCORECARD - A Model for Evaluating and Improving the Performance of Software Testing Houses.
In *Proceedings of the 12th International Conference on Enterprise Information Systems - Databases and Information Systems Integration*, pages 370-378
DOI: 10.5220/0002899703700378

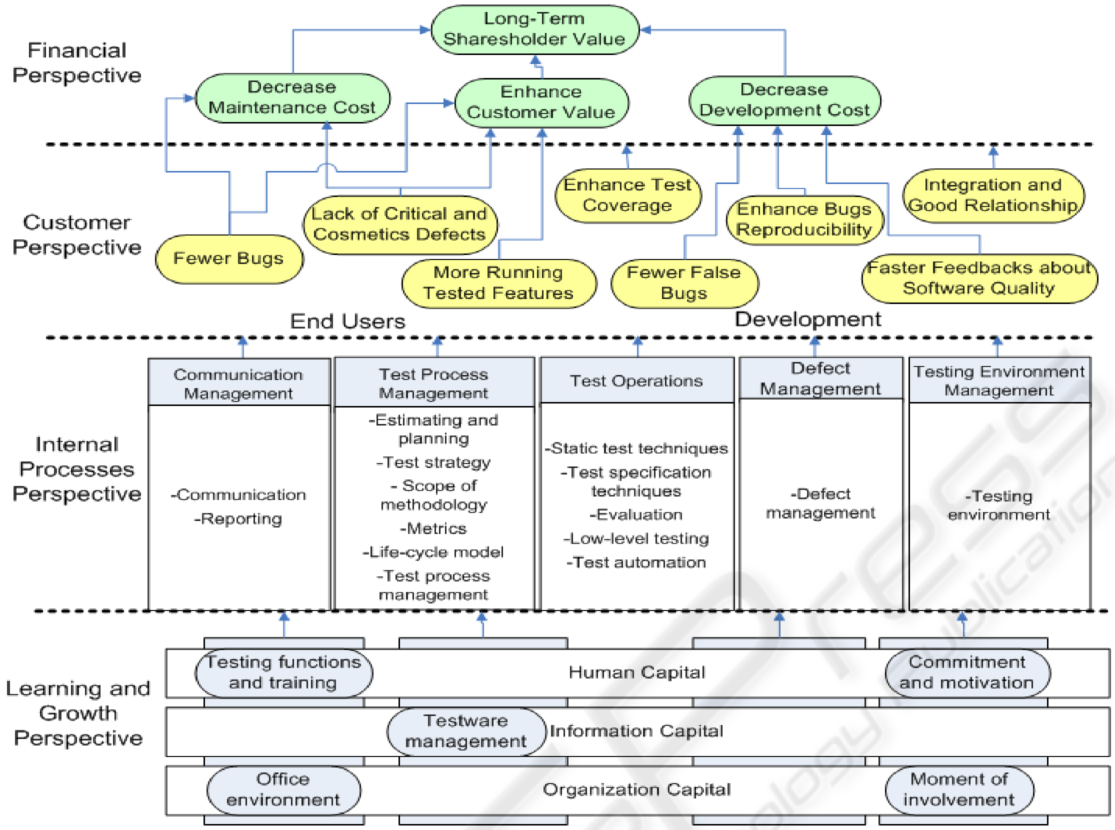

Figure 1: Overview of BTSC perspectives.

The BTSC Strategy Map records objectives, which will provided indicators, covering different perspectives (Finance, Customer, Internal Processes and, Learning and Growth) for the managers of test teams.

For the management of internal activities (Internal Processes and, Learning and Growth), a testing maturity model is used. In this study, the TPI - Test Process Improvement model was chosen (Sogeti, 2008), but other models, such as TIM or TMM, for instance, may be used as all of them have a similar structure with key-areas and checkpoints.

The Strategy Map was developed based on the literature and was refined by 21 experts in software testing, each having at least 2 years' experience in this area, besides their having the profile of a manager or leader of test teams. These refinements are not explained in this article, for which see (Nobrega, 2008).

As the BTSC is a generic model, a customization process is needed to tailor it into a specific Strategy Map. This process, based on (Amaratunga, Haigh, Sarshar & Baldry, 2002) and (Kaplan & Norton, 1997) is explained in Section 2.1.

2.1 Strategy Map

Within each perspective of the BTSC Strategy Map, goals are set to guide the construction of a specific strategy map for a specific testing organization or project.

2.1.1 Overview

Figure 1 gives an overview of the BTSC Strategy Map. Each perspective is explained below.

2.1.2 Financial Perspective

The Financial Perspective indicates how the software testing process and initiatives to improve them will add more value to the organization.

To achieve the main objective of the financial perspective of the BTSC, we have how to demonstrate how **Long-Term Value for Shareholders** will be achieved.

To achieve this goal, three other goals must be achieved. The first is to **Decrease Maintenance Costs**, the second is to **Decrease Development Costs** and the third to **Enhance Value for Clients**.

The objectives of **Long-Term Value for Shareholders** and **Enhance Value for Clients** were based on the Generic Strategy Map (Kaplan & Norton, 2004). The other two goals were defined based on the financial impacts of customers' goals which are described below.

2.1.3 Customer Perspective

Under the Customer Perspective, BTSC enables one to identify what internal and external clients expect from the test team to be identified.

As the first activity, it is necessary to define who the clients of the test team are.

According to Kaner *et al* (2001), a test team has several internal and external clients. They are: Project Manager, Programmers, Technical Writers, Technical Support, Senior Management, Marketing; and End Users, respectively.

In order to simplify the problem, the Project Managers, Programmers, Technical Writers, Technical Support and Senior Management were grouped as the **Development Team**. As Kaner *et al* (2001) say the marketing area needs to know when an issue will affect a feature that is vital to end users. Who may, for example, be companies, public sector, bodies or individuals. They were grouped as a single client: **End Users**.

After having defined the test team’s client, their expectations need to be determined. According to Gupta and Aggarwal (2005), the test team must pursue the goal of delivering software with as **Fewer Bugs** as possible. According to them, the number of defects in a production process is a very important metric for determining the effectiveness of the test process. Thus, the fewer bugs there are in production, the greater will be the possibility of reducing the cost of maintenance software, a goal of the Financial Perspective of BTSC.

In the same article, Gupta *et al* (2005) define that there must be **Fewer False Bugs** identified by Test Team to the Development Team. Consequently, this will reduce the cost of software development.

According to Kaner *et al* (2001), another want Development Teams express is to have a **Faster Feedback about Software Quality**. When software is changed, the test team should test it quickly so that the tests do not create a "bottleneck" in the project.

The Financial Perspective indicates how the software testing process and initiatives for improving it will add more value to the organization.

Finally, the objective that both developers and end users of software set is that the software has increasingly **More Running Tested Features**. According to Jeffries (2004), Running Tested Features can be explained as:

(i) **Running** means that the features are shipped in a single integrated product.

(ii) **Tested** means that the features are continuously undergoing tests provided by the requirements givers – the customers in XP parlance.

(iii) **Features** means real end-user features, pieces of the given client requirements, not techno-features like "Install the Database" or "Get Web Server Running".

2.1.4 Internal Process Perspective

This Perspective is concerned with identifying the most critical activities that achieve the goals of the test team's clients and the financial goals of the organization.

This Perspective's objectives are represented by key areas of the Test Process Improvement (TPI) related to the internal processes of a test team. The key areas were organized into five main areas so to have the same representation as the Strategy Map.

The content of the main areas are showed in the Table below.

Table 1: Main Areas of BTSC and Key Areas of TPI.

Main Areas	Key Areas of TPI
Communication Management	Communication
	Reporting
Test Process Management	Test strategy
	Life-cycle model
	Estimating and planning
	Metrics
	Scope of methodology
	Test process management
Test Operations	Test specification techniques
	Static test techniques
	Test automation
	Evaluation
	Low-level testing
Defect Management	Defect management
Test Environment Management	Test environment

2.1.5 Learning and Growth Perspective

The last Perspective of BTSC, Learning and Growth, identifies the infrastructure the test team must build

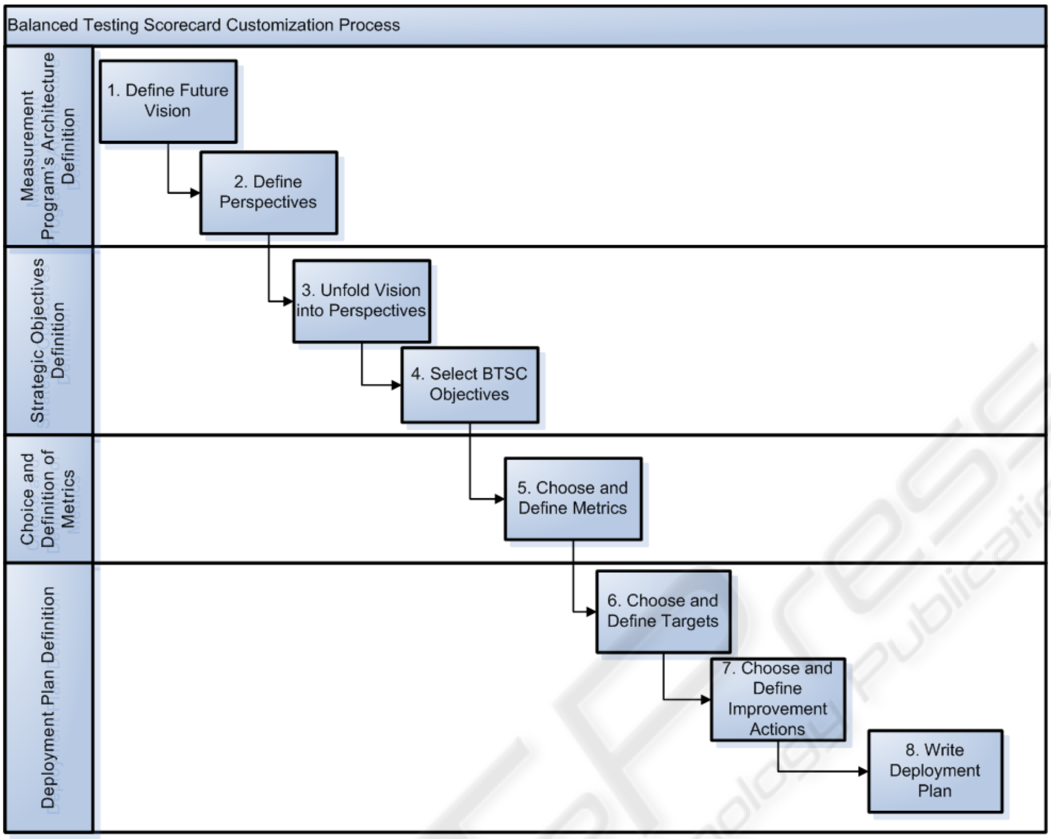
Balanced Testing Scorecard Customization Process
Measurement Program's Architecture Definition
Strategic Objectives Definition
Choice and Definition of Metrics
Deployment Plan Definition
1. Define Future Vision
2. Define Perspectives
3. Unfold Vision into Perspectives
4. Select BTSC Objectives
5. Choose and Define Metrics
6. Choose and Define Targets
7. Choose and Define Improvement Actions
8. Write Deployment Plan

Figure 2: BTSC's Customization Process.

to generate growth and improvements in the long term.

To define this perspective several key areas of TPI were used, related to the Learning and Growth of the test team. They are listed in the Table 2 below.

Table 2: Key Areas of TPI related to Learning and Growth.

Key Areas of TPI
Moment of Involvement
Office Environment
Commitment and Motivation
Testing Functions and Training
Testware Management

2.2 Customization Process

As BTSC is a generic model, it must be customized in order to be used. Thus, a process of customization has been set up, which is detailed as follows.

The customization process of BTSC was based on the customization process of BSC, presented in (Kaplan & Norton, 1997) and (Amaratunga *et al*, 2002) is divided into the following steps:

- **Step 1:** Defining the Architecture of the Measurement Program;
- **Step 2:** Defining Strategic Objectives Definition;
- **Step 3:** Choice and Definition of Metrics;
- **Step 4:** Defining the Deployment Plan.

Each step has its own specific activities as can be seen in Figure 2.

These steps as well as their activities are detailed as follows.

Step 1 – Defining the Architecture of the Measurement Program:

The major aim of this step is to promote an understanding and a critical analysis of the future vision of business and the test team.

This step will be divided into the following activities:

(i) **Defining the Future Vision of the Test Team:** The first activity to start the strategic planning is to define the future vision of the test team.

(ii) **Defining Perspectives:** After defining the future vision, it is necessary to define which perspectives should represent the global management of test team's activities. The BTSC suggests 4 perspectives: Finance, Customer, Internal Processes and Learning and Growth. However, these perspectives can be removed, altered or extended.

Step 2 – Defining the Strategic Objectives:
The activities of this step will allocate the strategic objectives under the perspectives of the BTSC. To perform this step, the following activities should be undertaken:

(i) **Examine the future vision within each perspective so as to set general objectives:** To define the strategic objectives, a first interview can be carried out to define the strategic objectives of the software testing team.

(ii) **Select the BTSC objectives that need to be achieved within each perspective:** After general objectives are set for each perspective, secondary objectives based on BTSC may be added for each perspective.

Step 3 - Choice and Definition of Metrics:
In this step, the metrics that will be used to measure whether a particular strategic objective has been achieved are identified.

This stage includes the following activity:

(i) **Choose and Define Metrics:** For each strategic objective, a metric or a set of metrics must be defined that best captures and communicates its intentions. For each proposed metric, the sources of information and actions necessary to make such information available it should be identified and detailed. And for every perspective, identify the critical relationships between its metrics, and between this perspective and all others.

Step 4 – Defining the Deployment Plan:
After having defined the metrics associated with different strategic objectives, defining targets, plans of action and who is responsible for guiding the implementation of the strategy should be undertaken.

(i) **Choose and Define Targets:** Targets should be set for each metric. The organization needs to verify whether a goal has been reached or if it is necessary to take corrective action (Improvement Action).

(ii) **Choose and Define Improvement Actions:** To help the improvement process, the BTSC has several suggestions for improvements that can be used; however, new suggestions may be added. Based on the objectives chosen for each perspective, improvement actions must be documented in order to facilitate its initiatives.

(iii) **Write Deployment Plan:** This activity will develop a deployment plan for the BTSC. The difficulties of implementing this new management model, which aims at evaluating and improving the performance of software testing teams should be taken into account.

3 CASE STUDY

This section shows the implementation of a case study in a Factory Test organization type. This case study concerns the first phase of the deployment project BTSC within the organization studied. Due to the phase that the project is, will present the partial results concerning the implementation of some objectives mapped in Balanced Testing Scorecard. Below is showed the plan of the case study, its implementation and obtained results so far.

3.1 Scenarios and Objectives

This case study was conducted with the main objective to assess the adoption of BTSC - Balanced Testing Scorecard to evaluate the performance of both the Factory Test where the case study was used as the customer's suppliers seeking increasing of efficiency.

The organization where was applied the BTSC is Inmetrics, a company that has in its staff of about 100 employees, including researchers, test architects, testers, performance analysts and quality analysts.

The idea of deploying a methodology for assessing organizational performance appeared from the service format of the company studied. Its largest customer (accounting for 70% of revenues) is an organization that has a Software Factory as part of their IT infrastructure. The Software Factory, in turn, hires suppliers to develop modules of their systems, performing functional tests and acceptance tests of the developed systems. Because it is a client of international magnitude, it has a consolidated methodology of software development that extends to their suppliers, including. Thus, all

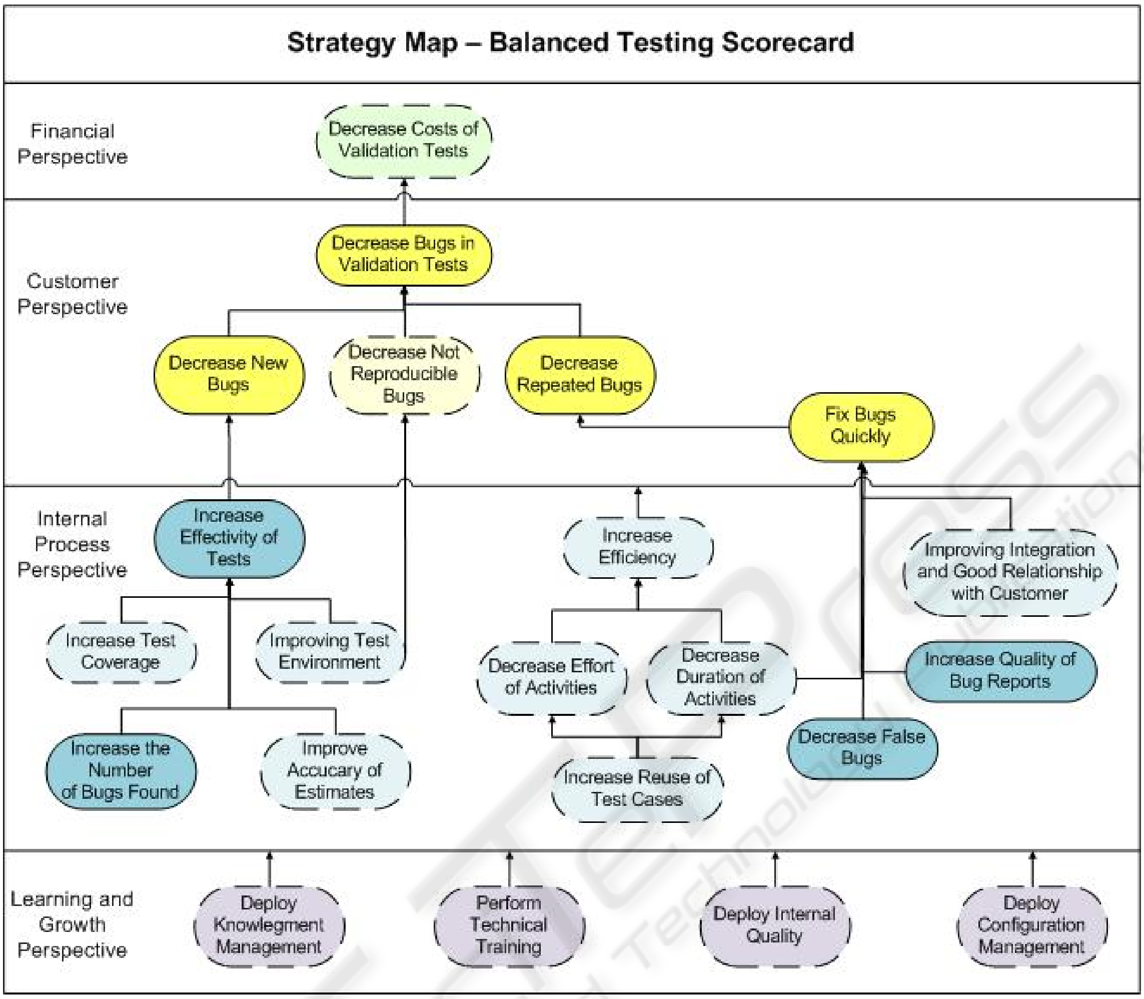
Strategy Map – Balanced Testing Scorecard
Financial Perspective
Decrease Costs of Validation Tests
Customer Perspective
Decrease Bugs in Validation Tests
Decrease New Bugs
Decrease Not Reproducible Bugs
Decrease Repeated Bugs
Fix Bugs Quickly
Internal Process Perspective
Increase Effectivity of Tests
Increase Efficiency
Improving Integration and Good Relationship with Customer
Increase Test Coverage
Improving Test Environment
Decrease Effort of Activities
Decrease Duration of Activities
Increase Quality of Bug Reports
Increase the Number of Bugs Found
Improve Accuracy of Estimates
Increase Reuse of Test Cases
Decrease False Bugs
Learning and Growth Perspective
Deploy Knowlegment Management
Perform Technical Training
Deploy Internal Quality
Deploy Configuration Management

Figure 3: Strategy Map of Inmetrics.

service providers should adopt a standard software development methodology. In addition, suppliers will be evaluated periodically in agreement with Guide of Performance Measurement-defined by the processes core of the customer. Thus, all suppliers should work following the same pattern, creating the evidences in the same format and providing the metrics established by the customer to evaluate the performance of its suppliers.

At this time Inmetrics observed that only way to differentiate themselves from other suppliers would be through increased efficiency in service delivery of functional tests keeping quality standard and offers to customer value-added information for your self-knowledge and allowing decision making for the customer in concerning of their suppliers.

To this end, was created a BTSC specific for Inmetrics, but with the customer's perspective aligned. Another important point that was taken into consideration is that this is aligned to BTSC metrics Guide of Performance Measurement defined by customer.

3.1.1 BTSC Deployment Planning

The implementation of BTSC has become a project at Inmetrics and was divided into phases. As a first step, drew up a strategy map for the BTSC Inmetrics, as shown in Figure 3.

Since the creation of strategy map, some goals were prioritized to be implemented in the first phase of the project. The goals are represented in the full line ellipses located in the blue perspective – Internal Process. The implementation of these goals allows an action started internally under control of Inmetrics, allowing the effects be reflected in the customer.

For the prioritization of goals, were adopted two criteria: speed in the visibility of results and ease of data extraction without having significant changes for the structure of the tools used. This last factor was called "Impediments". Basically, the goals without impediments were pre-selected.

To help this work were drawn a table with all the objectives in the map, where for each goal, were defined metrics aligned to Guide of Performance Measurement, the priority and the impediments to measure the goal. The table below shows four goals of full line in blue, located in the Internal Process perspective.

Table 3: Prioritization of Goals.

ID	Goal	Metric	Priority	Impediment
01	Increase Number of Bugs Found	Number of Bugs Found	1	No
02	Increase Test Effective ness	% of Test Effectivenes s	1	We do not have access to bug details.
03	Decrease False Bugs	% of False Bugs	1	It needs active participation of Client to validate the bug reported.
04	Improve Quality of Bug Reports	% of Bugs Reported Incorrectly	1	No

Among the objectives listed in the table, was observed that, according to criteria adopted, two of the four goals have no impediments. That is, they are ready to be worked out. As the priorities of all of them were set to 1, i.e., they have a quick visibility to the customer. Were chosen those two that have no impediments. They are: “Increase Number of Bugs Found” and “Improving Quality of Bug Reports”.

3.1.2 Collected Data

In this first phase of the BTSC project deployment, the main objective is to know the numbers of Inmetrics and his customer. So, will be measure “Number of Bugs Found” and “% of Bugs Reported Incorrectly”, related to selected objectives.

The metric “Number of Bugs Found” derived from the goal “Increase Number of Bugs Found” measures the amount of problems identified by the Factory Test at the phase called “Integrated Test” which is the stage prior to the Validation Test phase.

The amount of problems is obtained from the extraction of the data reported in Mantis tool, “a popular free web-based bug tracking system”. In a first measurement, were detected the following problems:

(i) The customer’s suppliers who develop modules of the system are not properly identified at the time of bug report;

(ii) There is not a definition of the cause of bug identified, ie, whether it was caused by an environmental issue, problem of access to the system, problem of mass data, the lack of documentation available for testing or if it really is a system failure.

Based on these two issues identified by the first data extraction performed, two actions were taken allowing obtain more accurate data that provide information for customer decision-making:

(i) Was created a custom field in Mantis, to indicate the supplier responsible for developing the system in which the bug was detected;

(ii) Was created a field to categorize the causes of bugs detected. This field is "validated" by the supplier responsible for the bug reported as a way to wake up the problem detected.

Table 4: Number of Bugs by Supplier.

Name	Number of Bugs
Supplier 1	165
Supplier 2	137
Supplier 3	101
Supplier 4	69

The table above allows not only identify the amount of bugs found, but indicate to the customer the level of bugs generated by its suppliers of software development.

Table 5: Causes of Bugs.

Name	Access Proble ms	Doubt s	Data Mass	Failur es
Supplier 1	0	11	0	154
Supplier 2	4	2	11	120
Supplier 3	1	1	0	99
Supplier 4	0	2	0	67

The table above shows a categorization of the causes of bugs found by the supplier, allowing identifying the main causes of failure in tests, giving visibility to the customer about what aspects he should pay more attention.

The metric “% of Bugs Reported Incorrectly”, derived from the objective “Improving Quality of Bug Reports” measures the amount of bugs that were reported without being bugs, in fact. That is, bugs were reported by misunderstanding of the business, problems occurred only in the test environment, but that does not occur in the production environment, among others. Currently, each project team reports bugs differently.

As a first action to mitigate this problem, was created a standard reporting format that includes key information of the bug detected in order to avoid misconceptions, and additional effort to clarify the bug between Software Factory and Test Factory.

Following is showed the standard format adopted:

Summary:
Write a <summary> of the request that will appear under the same;
Standard adopted:

(i) **Location:** Where the error occurred.
Example: Screen Name, System, Program, Job, etc.
(ii) **Error:** Error Message, Error Code, etc.
(iii) **Error Summary:** Describe the problem occurred.

Description:
Describe in more detail the problem ocurred.
Example: “The system has an error when trying to delete parameter in the purge screen XPTO”.
“There was no description corresponding to the message identifier MSG”.

Steps to reproduce: Inform the steps to reproduce the problem.
Example:
1 – Access menu X → Y → Z;
2 – Fill "Product" and "Sub product" fields with valid data;
3 – Click the "Search" button;
4 – Select the product listed and click "Delete".

Expected Result: Describe the output that was expected.
Example: The system deletes the parameter successfully.

Obtained Result: Describe the obtained result.
Example: Error trying deleting parameter.
“There was no description corresponding to the message identifier MSG”.

With respect to the metric "% of Bugs Reported Incorrectly" has not been generated numbers that showed the effects caused by this action because it was recently established through training of testers and architects and publication of the handbook for employees.

3.1.3 Obtained Results

The results presented in the previous section show that through simple actions on the existing historical basis and with the support of a tool its possible generates information for decision-making.

The BTSC deployment, besides serving as an evaluation tool for internal performance of Inmetrics, it focuses on increase of efficiency and provides metrics to evaluate the customer’s software development suppliers.

Note that this is the first step in implementing the project of BTSC deployment and therefore the metrics generated so far should be considered in careful when taking into account other factors that may explain the behavior of the numbers presented.

4 CONCLUSIONS

This paper presented the Balanced Testing Scorecard - BTSC, based on the methodology for assessing organizational performance BSC. By implementing this methodology, was possible conduct a case study to demonstrate simple actions possible to be implemented to evaluate and improve organizational performance.

4.1 Future Works

Continuing this work, was carried out a planning for implementation of all goals full line placed on the strategy map. For each of them will be specify the activities necessary for the extraction and measurement of metrics related to the goals and structure of the measurement process.

As an immediate activity to be presented, will be measure the metric “% of Bugs Reported Incorrectly”, it was implemented the standard format for reporting bugs.

After that, the other objectives of this first group will be implemented through a new project that will cover the goals in dashed line. This work has an effect in the medium and long-term involving impacts related to changes in culture primarily on the customer and its suppliers of software development, further structural changes in the tools used.

REFERENCES

Amaratunga, D, Haigh, R, Sarshar, M & Baldry, D, 2002, ‘Application of the balanced score-card concept to develop a conceptual framework to measure facilities management performance within NHS facilities', *International Journal of Health Care Quality Assurance*, vol. 15, issue 4, p. 141-151.

Bach, J, 1994, 'The Immaturity of CMM', *American Programmer*. vol. 7(9), p. 13-18.

Burnstein, I, Suwanassart, T, Carlson, C, R, 1996a, 'Developing a Testing Maturity Model', *Part I, CrossTalk.*

Burnstein, I, Suwanassart, T, Carlson, C, R, 1996a, 'Developing a Testing Maturity Model', *Part I, CrossTalk.*

Gupta, V, K, K, Aggarwal, Y, S, 2005, 'Objectively Managing Software Testing Projects', *Journal of Conceptual Modeling.*

Hutcheson, M, L, 2003, 'Software Testing Fundamentals: Methods and Metrics', *John Wiley & Sons*, p.408.

Jeffries, R, 2004, *A Metric Leading to Agility*, viewed 6 May 2008, < http://www.xprogramming.com/xpmag/jatRtsMetric.htm.>

Kaner, C, Bach, J, Pettichord, B, 2001, 'Lessons Learned in Software Testing', *Wiley*.

Kaplan, R, S, Norton, D, P, 1997, 'Strategy in Action: Balanced Scorecard', *Rio de Janeiro: Campus*, 20.ed.

Kaplan, R, S, Norton, D, P, 2004, 'Strategic Maps: Converting intangible assets into tangible results', *Rio de Janeiro*, 6.ed.

Nobrega, R, O, 2008, 'Balanced Testing Scorecard: A Model for Assessment and Improvement of Software Testing Teams' Performance', *MSc Dissertation*.

Sogeti, 2008, 'TPI – Testing Process Improvement', viewed 29 April 2008, <http://www.sogeti.nl/Home/Expertise/Testen/TPI.jsp>