Abstract

Software Product Line can be effectively tested with a framework utilization following Experimental Software Engineering concepts. During this research, a description of an Operational Maturity Structure for Software Testing Management was created for the Operational Domain of the Organizational Testing Management Maturity Model (OTM3) framework. The proposed structure addresses both the defects management and the measurement of software testing results, through a method using effective metrics. This structure allows to identify, to establish, and to keep the capabilities demanded by software testing maturity models. These goals are achieved according to standards, measures, controls, and continuous improvements. In order to verify and validate the application of the OTM3 framework, a case study was developed within a Research & Development Project at the Brazilian Aeronautics Institute of Technology (ITA).

Keywords: Software Testing, Software Testing Management, Software Product Line, Experimental Software Engineering, Process Management with Testing.

1. Introduction

The Organizational Testing Management Maturity Model (OTM3) framework [1] was developed by three doctoral degree students from the Computer and Electronic Engineering Graduate Program at the Brazilian Aeronautics Institute of Technology (Instituto Tecnológico de Aeronáutica - ITA). It is focused on the improvement of Software Testing (ST) process, based upon Software Process Engineering Meta-model (SPEM) [2]. It provides information for an organization and a method to identify, to establish, and to keep the main capabilities demanded by test maturity models for ST through: patterns, measures, controls, and also the best practices for software engineering improvements. The OTM3 is structured by: i) direct experiences on issues related to the measurement of ST results, with continuous improvements of the ST process; and ii) specifications and design techniques using the method of goal-question-(indicator)-measure (GQ(I)M) [3]. It allows the development of qualitative and quantitative metrics by using an iterative and incremental approach. The OTM3 is based on efficiency and effectiveness quality indicators and metrics for quality process evolution. In order to verify and validate the application of OTM3 framework, some Experimental Software Engineering (ESE) concepts were followed [1] and a case study was developed within a Research & Development (R & D) Project at ITA.

This paper is organized as follows. Section 2 describes the framework OTM3 Operational Domain. Section 3 deals with Effectively Testing for a Software Product Line (SPL). Section 4 describes ST Metrics. In Subsection 4.1 it is discussed the OTM3 Primary Metrics for SPL1/2. In Subsections 4.2 and 4.3, the OTM3 Secondary Metrics, respectively from TestLink and Mantis tools, are discussed. Finally, Section 5 highlights conclusions.

2. OTM3 Operational Domain

According to Pressman [4], if the ST is conducted ad hoc, time is wasted, unnecessary effort is expended and, still worse, faults can occur undiscovered. Thus, it is advisable to establish a systematic strategy for the ST that begins at the moment when the development starts and lasts until the end of the development lifecycle. With the importance given to the ST activity, it is necessary to have a clear-cut process of ST, organized in such a way as to guarantee process or product quality.

The OTM3 framework is composed by a set of processes maps that constitute the main phases of ST named [1]: Planning, Construction, Test Execution, Test Finalization, and Approval. The OTM3 is also composed of three categorizations: i) Columns for three ST processes domains: Operational, Collaborative, and Decision Support; ii) ST Processes improvements in horizontal’s Stages: Standards, Measures, Controls, and Continuous Improvements sub-plans; and iii) for each sub-plan, a
plan-do-check-act cycle (PDCA cycle) [5], as its best-known representation. The three OTM3 domains are: i) The Operational Testing Management - defining a set of operational ST processes allowing to follow its workflow [6]; ii) The Collaborative Testing Management - defining a set of ST processes for its integration and inter-operation, allowing its tasks consolidations and multi-programming activities; and iii) The Decision Support Testing Management - defining a set of processes for ST Management, where strategic patterns were extracted for decision making.

An Operational Maturity Structure addressing ST Management was created for the OTM3 Operational Domain framework [6], as shown in Figure 1. The proposed structure addresses the defects management, as well as the measurement of ST results through a method using effective metrics.

Figure 1. Operational Maturity Structure for the OTM3 framework [6]

3. Effectively Testing for a Software Product Line

The strategy adopted in the R & D Project at ITA was to provide software development in a Model Driven Architecture (MDA) approach. It was integrated with a Unified Process based on Software Product Lines (SPL) components for each subsystem, according to Linden et al. [7].

The SPL approach [8] is a concept used in the computing environment for software development projects. It focuses on engineering techniques usages. This encourages the creation of groups with similar software systems from a common set of specifications, by using common means of production. The SPL represents a software development technique transposed from other areas of production used in classical engineering.

The ST process occurs, in all Phases of the OTM3, according to each SPLn, where “n” refers to each line [9].

For the R & D Project at ITA, eight SPL were planned, while only three SPL were performed for this research. At this point, data were collected from three different experiments and placed in the following ST artifacts: Test Plan, Test Procedures, Test Cases, and Test Scripts, by using Functional and Structural techniques and Defect Prevention Analysis [9].

In the OTM3’s Planning Phase [9], the Test Plan was developed and integrated for each SPLn, before SPLn Planning Phase changes occur. At this point, the verification and validation steps become part of
the Planning Phase. Measurement and Analysis should be performed on the testing requirements, to minimize inconsistencies, faults, and inaccuracies.

In the Construction Phase [1], designing and testing implementations for all SPLn were performed, in order to prepare the environment for the next Test Execution Phase.

In the Test Execution Phase [1], the main purpose on Test Cases Scenarios measurements has been elaborated to produce capability in terms of failure detection effectiveness. Another purpose is to control the existence of open and closed defects.

In the Test Finalization Phase [9], the main activities' stereotypes are: to update the Test Plan, to register defects, and to implement the package for each SPLn. Within this phase, the ST process will be finalized. Measurement and Analysis will be executed to review SPLn draft changes.

The Approval Phase also performs Measurement and Analysis. At this phase occurs the preparation for the project approval. As shown in Figure 2, at this point, the Load, the Performance, and the Regression Testing, will be fired, by simulating situations that evaluates performance in critical processing. Therefore, it ensures the minimum requirements of quality, reliability, and compatibility for operation. Metrics will be used to update the Beta Test Approval and to register a Final Product Software Baseline.

ST management processes can be modeled by using SPEM. The abbreviations shown in Figure 2, and in each other OTM3’s process maps, are structured [9] by: (1) the initial letter of each Phase (Planning, Construction, Execution, Finalization, or Approval); (2) the initial letter of the specific SPEM stereotype; and (3) a sequential number.

Processes that are created with SPEM allow the creation of complex contextual models. It also allows for presenting decision-oriented processes through the usage of activity diagrams decision points [9].

This research intends to step up maturity degree information and implementation adherence for ST Operational Domain in the case study at the ITA’s Software Engineering Research Group (Grupo de Pesquisa em Engenharia de Software - GPES) [10].

![Figure 2. OTM3’s Approval Phase [9]](image)

4. Software Testing Metrics

Metrics are gaining importance and acceptance to improve ST quality. Primary and Secondary Metrics will be used for this research result analysis. Therefore, it is necessary to differentiate them. The primary metrics, in the process or software product context, have a relationship with the actual measure facts, represented by its action on the observed reality. On the other hand, secondary metrics have a relationship with indicators, which are obtained as a result of a relationship between the primary or secondary metric divided by a factor [6].

4.1. OTM3 Primary Metrics for SPL1/2

The Open Source TestLink [11] tool enables tracking Test Cases (TC) numbers, in percentage
terms, for status. In Figure 3, it can be seen, the TestLink SPL1/2 outcome for primary metrics, from ST activities phases, Integration and Regression Testing. They were classified from those TC which have passed (green), failed (red), were blocked (blue), or not yet executed (black). Notice that the defects on the 3rd Regression Testing are new, because the defects on the 2nd Regression Testing had already been removed [6].

Figure 3. OTM3 Primary Metrics for SPL1/2 [6].

4.2. TestLink OTM3 Secondary Metrics

The GQ(I)M [3] is a method for selecting appropriate metrics to meet the determined goals. These metrics provide indicators that will be used to calibrate the rest of the process. The TestLink is a tool for executing and tracking TC and organizing them into Test Plans. By using GQ(I)M method (Table 1) and TC from the TestLink tool, two metrics and indicators are formulated [6].

<table>
<thead>
<tr>
<th>GOAL</th>
<th>QUESTION</th>
<th>MEASURE</th>
<th>INDICATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>By the end of the project, increase the percentage of coverage of TC exercised</td>
<td>What level of ST coverage?</td>
<td>Percent Coverage Test (PCT) for each SPLn (PCT for SPL1/2, see Figure 4). PCT is a metric for efficiency.</td>
<td>The required TC prepared and exercised in units of time (months, weeks, hours) by SPLn.</td>
</tr>
<tr>
<td>By the end of the project, determine the TC rate with failed or blocked status</td>
<td>What is the TC percentage through ST that detects software failure?</td>
<td>Defects Found by ST for each SPLn.</td>
<td>Defects Found per Week (DFW) by SPLn (DFW for SPL1/2, see Figure 5). DFW is an indicator for effectiveness.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GOAL</th>
<th>QUESTION</th>
<th>MEASURE</th>
<th>INDICATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. GQ(I)M method.
4.3. The Mantis OTM3 Secondary Metrics

The Open Source Mantis [12] tool is used to manage defects in the ST process. The Mantis tool achieves integration with the TestLink tool. The existence of these two integrated tools adds project-level reporting, analysis, and management capabilities, especially when traceability information is used with requirements. It also allows full control on system defect management [1].

ST Manager must measure the ST Process effectiveness, since ST Process Measurement is essential for managing and evaluating the effectiveness of a ST Process. Through the defects recorded in the Mantis tool by Project team, it becomes possible to calculate the metric Effectiveness in Fault Detection (EFD) and other variables Defects Found by ST (DFST) and Defects Found by Other (DEO), formulated by Correa [13]:

\[
EFD = \frac{DFST}{DFST + DEO} \times 100
\]
Notice that, DEO refers to R & D Project persons from outside of the ST team [6].

4.3.1. Effectiveness in Fault Detection in each SPLn

4.3.1.1 SPL1/2.
EFD = (DFST / (DFST + DEO)) * 100
EFD = (80 / (80 + 53)) * 100 = 60%

4.3.1.2 SPL3.
EFD = (DFST / (DFST + DEO)) * 100
EFD = (114 / (114 + 27)) * 100 = 81%

The ST team, using the Operational Maturity Structure for OTM3 and following ESE concepts, have obtained a value of 60% within SPL1/2 trial and 81% within SPL3 trial, for Effectiveness in Fault Detection in relation to Project team, enabling the development of a product with more quality [6].

It is important to mention that most of the SPL3 TCs were developed before the actual system code has been generated. This way, certain defects are detected very early in the process.

5. Conclusion

The objective of this research paper was to present an Effective Testing for a Software Product Line with OTM3 in an Operational Domain framework. It presents the defects management, as well as measurement of ST process outcome, by using measures of effectiveness, structured in the GQ(I)M method or measurements based on Correa formulae, applied in an actual case study.

The ST team increased from SPL1/2 (60%) to SPL3 (81%) the Effectiveness in Fault Detection (EFD) in relation to Project team.

The OTM3 framework implementation in a case study developed within an R & D Project at ITA has improved the quality of organizational processes and the management of ST processes. It has allowed initial prioritizations of its Operational Domain. This paper major contribution can be seen in Figure 1.

In the near future, an iterative and incremental application to support the other two Domains (Collaborative, and Decision Support Testing Management) from other projects will be also developed by the GPES at ITA.

6. References


