1.0 Purpose

At every point in time, test manager/test lead thinks in terms of efficiency of his test team. How is the team doing? How good are the efforts put in with time? This paper presents one approach that can be used for measuring the efficiency of the test team. This approach can give the efficiency in terms of units per hours in normalized form for a test project at any phase of the project.

2.0 Theory

The theory behind the approach is to normalize the efforts/work done to a basic unit of measurement for the various ‘work items’ of all phases of the test project i.e. defining the requirements and test ware into work items and then normalizing them to a basic unit.

2.1 Unit for normalization for various work items:

The following is the suggested normalization of various work items. Although the list below does not details all the work items but it does list the ones which are noticeable observed in a test project.

Work items 1 – Requirement documentation
Measure: Requirement docs can be given weightage of 1 to 5, 5 being maximum

Work items 2 – User cases
Measure: Use cases can be given weightage of 1 to 5, 5 being maximum

Work items 3 – test case creation
Measure: test case creation given weightage of 1

Work items 4 – test case execution
Measure: test case execution given weightage of 1

Work items 5 – defect reported
Measure: Each defect reported given weightage of 1

Work items 6 – defect managed
Measure: Each defect managed in defect meeting given weightage of 1

3.0 Sample Calculations

To provide an example of how the efficiency is calculated, consider the following:

3.1 The project phase

Test execution and defect reporting in progress but at the same time development working on various other modules. As the efforts of the new requirement/code is not done, the calculations are based on existing efforts done by the team.
3.2 Baseline for unit of normalization for the sample

<table>
<thead>
<tr>
<th>Work Item</th>
<th>Normalized unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case</td>
<td>5 units</td>
</tr>
<tr>
<td>Requirement docs</td>
<td>3 units</td>
</tr>
<tr>
<td>Test cases created</td>
<td>1 unit</td>
</tr>
<tr>
<td>Test cases executed</td>
<td>1 unit</td>
</tr>
<tr>
<td>Defect Reported</td>
<td>1 unit</td>
</tr>
<tr>
<td>Defect managed</td>
<td>1 unit</td>
</tr>
</tbody>
</table>

3.3 The statistics for the project are as under for the various work items

<table>
<thead>
<tr>
<th>Work Item</th>
<th>No of work items</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Case</td>
<td>34</td>
</tr>
<tr>
<td>Requirement docs</td>
<td>27</td>
</tr>
<tr>
<td>Test cases created</td>
<td>7243</td>
</tr>
<tr>
<td>Test cases executed</td>
<td>1712</td>
</tr>
<tr>
<td>Defect Reported</td>
<td>960</td>
</tr>
<tr>
<td>Defect managed</td>
<td>960</td>
</tr>
</tbody>
</table>

3.4 Based on this the efficiency is calculated as

\[
\text{Efficiency} = \frac{\left( \sum (\text{Normalized units}_1...\text{n}) \times (\text{work items}_1...\text{n}) \right)}{\text{Total efforts in man hours}} \times 100
\]

With total man hours around 14080, the efficiency is:

Efficiency = 79% (units/ hour)

4.0 Pros and Cons

1. This is a great way of finding the test team efficiency based on statistical data,
2. As no standard exists to baseline and measuring the complexity of work items and work units, the approach can show varying results for test managers/ leads across teams or organizations,
3. The baselining of normalization units can vary based on project and person who is performing the exercise,
4. Testware has to be managed using proper tools that assist collection of test statistics,
5. Statistics can be misleading, BEWARE!

5.0 Suggestions

Any other observations and/ or pros & cons, please mail jcrvs@hotmail.com