JOURNEY TOWARDS ZERO DEFECTS: CHALLENGES, BEST PRACTICES AND TESTING MODELS

Organizations face different challenges at different levels of testing maturity. Testing Maturity Model (TMM) and Test Process Improvement (TPI) are two of the contemporary testing maturity models. Organizations that apply them would reap benefits and migrate to a higher level of maturity and better quality.

This white paper gives pointers to some of the best practices which can be followed by organizations for their testing activities and introduces TMM and TPI. It discusses the journey of a group from a stage when testing was not considered a separate activity to a stage where a number of activities take place under the scope of testing. It will highlight the challenges/issues faced at each stage and what were the incremental improvements made to overcome each hurdle till the vision of high quality was achieved. It would also help an organization on how to evaluate their group on where they stand with regard to testing i.e. what are their strong and weak points and also understand how to improve the quality and productivity of their group.

WHITE PAPER

ARTHI VENKATARAMAN
# Table of Contents

- **INTRODUCTION** ................................................................. 3
- **CHALLENGES FACED DURING DIFFERENT STAGES OF THE JOURNEY** ................................................................. 3
  - STAGE 1: NO DEDICATED TEST ENGINEER ................................................................. 3
  - STAGE 2: DEDICATED TEST ENGINEER INTRODUCED LATE INTO PROJECT ................................................................. 3
  - STAGE 3: DEDICATED TEST ENGINEER FROM START OF PROJECT ................................................................. 4
  - STAGE 4: DEDICATED TEST MANAGER AND CENTRALIZED TEST TEAM ................................................................. 4
  - STAGE 5: CHALLENGES BEING FACED ................................................................. 6
- **TMM** ......................................................................................... 8
  - WHAT IS TMM? ......................................................................................... 8
  - WHY TMM WAS RELEVANT? ......................................................................................... 8
  - MAPPING OF TMM WITH THE STAGES IN OUR JOURNEY ................................................................. 8
  - CONCLUSIONS ON TMM ......................................................................................... 9
- **TPI** ......................................................................................... 9
  - WHAT IS TPI? ......................................................................................... 9
  - HOW TPI WAS APPLIED BY US TO IMPROVE THE EFFICIENCIES IN OUR GROUP ................................................................. 11
  - CONCLUSION ON TPI ......................................................................................... 11
- **APPENDIX** ............................................................................. 12
  - SOME METRICS DATA SHOWING THE PROGRESS ......................................................................................... 12
  - HOW TPI MODEL LOOKS ......................................................................................... 13
  - CHECKLIST QUESTIONS TO QUALIFY FOR LEVEL A OF TPI FOR TEST ......................................................................................... 14
  - IMPROVEMENT SUGGESTIONS TO QUALIFY FOR LEVEL B FOR TEST ......................................................................................... 14
  - KEY AREA DESCRIPTION IN TPI ......................................................................................... 15
- **REFERENCES** .......................................................................... 18
- **ABOUT THE AUTHOR** .................................................................. 18
- **ABOUT WIPRO TECHNOLOGIES** .................................................. 19
INTRODUCTION

Testing is an important activity which normally accounts for more than thirty percent of the project life cycle. Many organizations do not give sufficient focus for testing. Different organizations are at different levels of maturity with respect to their testing processes. Test Maturity Model (TMM) and Test Process Improvement Model (TPI) are two of the contemporary models in the testing domain.

However, initially testing was not an important activity. It has been a long journey from a stage when testing was not considered a separate activity to a stage where a number of activities take place under the scope of testing. The group faced various issues and challenges at each stage and made incremental improvements to overcome each hurdle till the vision of high quality was achieved. Using TMM at the beginning of the journey could have shortened the cycle time and reduced the effort to reach improved quality. The TPI model could be applied for process improvement. TPI also has been used in the project for the purpose of improving efficiencies.

CHALLENGES FACED DURING DIFFERENT STAGES OF THE JOURNEY

The journey towards zero defects has gone through different stages, faced challenges in each stage and even overcome those. Each stage in this journey produces products at a higher quality level than the previous stage.

Stage 1: No dedicated test engineer

In the initial stages of the project there was neither a separate person identified as a tester nor was there any specific focus on testing. In such cases either the developer of the module itself did the testing or there was a cross validation approach followed, wherein the developer of one module become the tester for the other module. Further, there was no test plan or test cases. The purpose of testing was minimal and was to just prove that the software worked. The main disadvantage of this approach was that quality of the software was very poor. There were numerous field errors (more that 300 in one case). And in one case bugs were reported up to 2 years after the first delivery.

Stage 2: Dedicated test engineer introduced late into project

At this stage the management decided to exclusively identify a dedicated test engineer for the role of testing. However they decided to place a test engineer in the project five months after the project started since testing would start only by then. After this there was an improvement in the quality of product due to the independence of the tester from the developer. The test engineer focused on coming up with a test plan and test cases. Separate machines under test were reserved for the test engineer on which he could exclusively perform his testing activity. The organization had a clearly documented test approach which was used. At the organization level the generic template of test plan and how to update it was already defined and it was used to prepare the test plans. The defects though comparatively less than before were still on the higher side. Though there were many issues with this approach the most important issues which stood out with this approach were:

- Test engineers lack of sufficient understanding of all the requirements of the product under test as he is involved much later in the project lifecycle.
- Lack of sufficient time for planning and design. This in turn affected the kind of test cases the test engineer developed and executed.
**Stage 3: Dedicated Test Engineer from start of project**

During this period the management took a decision to place the engineer to the project earlier in the schedule thus involving him right from the requirements phase. A detailed test plan and detailed list of test cases was prepared. The bidirectional traceability was just to ensure every requirement was covered. Every deliverable from the test engineer was reviewed.

Though there was a significant improvement in quality with this approach there were some issues like:

- Reporting of the test engineer to the project manager reduced the ability of test engineer to have a decision on releases.
- Also as project managers were responsible for testing they could arbitrarily cut test cycle time/test coverage to make an on time release even if the implementation phase slipped.
- Defects were tracked and reported but there was no standard way of reporting. At different times different approaches were followed which included mails, excel sheets and word documents. There was no dedicated defect tracking tool.
- There was no focus on automation.
- Project managers were already tied up with day to day execution activities had no time to focus on testing related activities.
- As there was no central pool of testers there was no consolidated repository of knowledge through which sharing of test engineers could be done.
- There was a lack of planned training for the test engineers.
- No clear career path was visible for testers.

**Stage 4: Dedicated test manager and centralized test team**

The management wanted to further improve on the results and handle some of the issues. A separate productization test team lead by a test manager was formed. The test engineers reported to the test manager and had only a dotted line reporting relationship with the project manager. This team was responsible for testing all products that came out of the group. The manager studied the issues and identified over a period of time things that needed his attention. The following table describes some of the issues and the action taken to address these issues.

<table>
<thead>
<tr>
<th>Issue/Challenge</th>
<th>Action taken by test manager</th>
</tr>
</thead>
</table>
| Identify an ideal tool for defect tracking and installation of the same. | a) Evaluation was done on all standard available defect tracking tools.  
 b) Choice narrowed down to BugZilla due to its scalability, robustness as well as the fact that is was free.  
 c) Test team procured a dedicated server for itself.  
 d) BugZilla was installed on the same.  
 e) For every project separate entries were created. |
| Narrow down on the relevant metrics that would help in tracking the testing process. | Metrics tracked included Test Coverage, Test Case Efficiency, Fixed/Found Trend, Test Case Productivity, etc. |
| Come up with an efficient way of collecting and analyzing the identified metrics. | Designed an excel based Metrics Tracking tool which was easily customizable for all projects. |
| Training and Re-skilling of the test team members | a) Identified the relevant skill areas. Arranged for training on those areas.  
 b) Made team members prepare white papers on areas they have worked.  
 c) Ensured these white papers were checked into both the local as well organization repository. |
d) Arranged for regular presentations by team members. The above three points were made as part of the team member’s objectives.

| Improving the motivation and morale of team members. | a) Understood individual career aspirations. Provided sufficient learning opportunities to achieve the same.  
b) Regular recognition of good performance through on the spot awards/recognition.  
c) Showed a clear career path for all team members. |
|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------|

| Improve the quality of deliverables from the group. (This includes test plans, test cases and test code) | a) First level review of all deliverables coming from the group. Also coordinated review of critical modules with development team project manager.  
b) Instituted the practice of cross project reviews by team members. |
|---------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|

| Identify automation opportunities/relevant automation tools. | a) Did a study of available tools in market. Invested in one tool,  
b) Identified cases where lots of effort could be saved by writing shell scripts. (especially in the areas of build and installation) |
|-------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|

| Efficient Tracking of the Testing activity. | a) A standard weekly status reporting template was evolved.  
b) Team members submitted this report every Monday.  
c) Consolidated report sent to all project managers, team members and technical manager.  
d) Weekly team meetings held to track all activities and understand any issues. |
|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|

| Develop reusable repository of test case. | a) All test cases developed by group were reviewed at end of project to make it project independent.  
b) A repository was created in the test team’s server.  
c) These cases were checked into this repository. |
|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|

| Recruitment of new team members. | a) The manager created a clear set of criteria for recruitment. A standard set of pre-requisites in technical as well behavioral skills were laid down.  
b) All recruitment to the group ensured that the minimal requirements were satisfied. |
|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------|

<table>
<thead>
<tr>
<th>No clarity on when the code was ready for customer delivery.</th>
<th>a) Established clear release criteria</th>
</tr>
</thead>
</table>
Results

- A significant improvement in quality.
- An improvement in team morale.
- More engineers got interested in taking up a testing role.
- Reusable repository of test cases developed
- Cross-project help in terms of easy deploy ability of test engineers to projects needing more testers for a period of time.

However, despite all the good work there were still issues to be resolved and a need for a constant endeavor to improve the way of working.

Stage 5: Challenges being faced

After reaching this stage in the journey, the dream of zero defects though closer in sight was still not achieved. There were still many challenges at this level.

1. Repeated crashing of schedules for testing.
   a. With testing being the last activity in project life cycle and an end date having being committed to the customer, any slippage in other phases of the project directly reduces time for testing.
   b. Many defects found during testing further reduces uninterrupted testing time.
2. Cyclical Nature of Work
3. Consistently having to prove the advantages and cost benefits to some managers and upper management in having a centralized test team.
4. Looking for process improvements.

Repeated crashing of schedules for testing

One way to work around this issue is to follow an agile methodology/feature based mode for testing. Another approach is to involve the test team right from integration testing. If needed they could be involved for unit testing phase also.

Different Approaches to testing

1) Conventional approach/Big Bang approach to testing
   Here system testing is considered an end of phase activity. The release is delivered to the test team once all integration is completed. The advantage in this case is that since testing is done only once the time taken for testing is less.

However, there are numerous disadvantages which include:
   - Many hidden surprises can come out during testing which will increase the schedule pressure on the development team.
   - Any slippage from development team will shorten the testing life cycle. This can compromise the quality of testing.

2) Incremental approach/Agile Methodologies
   Here numerous planned releases are made to test team which are planned in advance. The features delivered as part of each release as well as the schedule for each release is decided in advance in coordination of the test team. Detailed schedules with the number of releases to test team and features of each release are also shared in advance with test team. Based on this a detailed test plan and test design can be prepared for each stage.
3) Involvement of test team in integration as well as unit testing

In this case test engineer is involved early in the project life cycle. His/her responsibilities include integration as well as unit testing. The advantages of this approach are:

- An independent evaluation of the product happens early in the life cycle of the project. This gives sufficient time for the developers to fix the reported issues if any.
- The test engineer has a detailed understanding of the system under test and can tune the test cases to test all the possible limitations.

The main disadvantage of this approach is that the test engineer needs more detailed understanding of the interfaces under test and their behavior. This would require more detailed interaction with the developers and also more effort to be put in by the test engineers.

Cyclical Nature of work

There would be phases during the project in which there would be hardly any work for the test engineers, especially if the project manager has opted not to use the services of the test team for the unit as well as integration testing phases. At other times just before the release the same engineer would be overloaded and would be forced to work many late hours.

It is a continuous challenge as to how to use the idle time productively/effectively. The other challenge is to ensure that test engineers are not stretched too much during release times. Some solutions which can be followed are:

- Ensure sufficient cross training for all team members so that they could contribute to the other projects which are currently under pressure.
- If no such opportunity exists the vacant time period could be used for coming up with expertise notes, reusable test cases, test automation frameworks as well as training materials.

Proving the effectiveness of centralized test team to upper management

The best way to handle this would be to share the success stories. A metrics based approach to this would yield good results. The baseline metrics data as well as the metrics data in the new environment can be shared. Associating a dollar value to these improvements would drive home the point much better. Qualitative data can also be shared with them.

Looking for Process Improvements

At this stage we simultaneously did an active effort to improve the way of our working. While investigating we came across a number of models for testing which aided in assessing as well improving the testing processes.

Available test models included:

- Software Testing Maturity Model
- Test Process Improvement (TPI)
- Test Organization Maturity TM
- Testing Assessment Program
- Proposed Evaluation and Test SW-CMM Key Process Areas (SW-CMM KPA)

Of all the models we narrowed down on the Software Testing Maturity Model (TMM) and Test Process Improvement (TPI) Model for further evaluation as we found them to be most appropriate. TMM and TPI and how they could be used for improving the performance/quality will be further discussed in detail.
Testing Maturity Model (TMM)

WHAT IS TMM?

TMM is a Testing Maturity Model developed by the Illinois Institute of Technology as a guideline for test process improvement. It has five levels.

The five maturity levels and related process areas of the TMM are:
- **Level 1: Initial**
  - No process areas are identified at this level
- **Level 2: Definition**
  - Process areas: Test Policy and Goals, Test Planning, Test Techniques and Methods and Test Environment
- **Level 3: Integration**
  - Process areas: Test Organization, Test Training Program, Test Life Cycle and Integration, and Control and Monitor
- **Level 4: Management and Measurement**
  - Process areas: Peer Reviews, Test Measurement and Software Quality Evaluation
- **Level 5: Optimization**
  - Process areas: Defect Prevention, Quality Control and Test Process Optimization

When we analyzed the TMM model it was found that there is a close map between the journey of improvement undertaken by the group and the different levels in TMM with a few exceptions.

WHY TMM WAS RELEVANT?

TMM was relevant because our organization was already compliant to the SEI-CMM and the TMM is complementary to CMM. The other advantages of using this model included its simplicity and also the ability of doing a self-assessment.

Despite being compliant to CMM Level 5 if we had done a self-evaluation of our group using TMM during the initial phase of our journey we would have been only at Level 1 of TMM. The need for a separate model for testing arose because in CMM there was no concept of testing maturity and there was no sufficient focus on testing practices, testing and quality related issues.

MAPPING OF TMM WITH THE STAGES IN OUR JOURNEY

Most of the questions in TMM refer to the entire organization. However if we tailor TMM questions so that frame of reference is only at a group level then we can map TMM levels with our different stages in the journey.

- Level 1 (initial) of TMM can be exactly mapped with the ‘no dedicated test engineer’ stage.
- Level 2 (definition) can be mapped with ‘dedicated test engineer late in the project life cycle’.
- Level 3 (integration) cannot directly be mapped to stage 3 (dedicated test engineer from start of project) in our journey. Most parts of the processes which refer to test training program, test organization, control and monitor were implemented in the next stage of our journey. The main commonality was the association of the engineer from the early stages of the project which was also required by TMM.
- Level 4 (management and measurement) can be mapped to the ‘dedicated test manager and centralized test team’.
- Level 5 (optimization) Here the focus is on improvements to the testing process and can be mapped to the final stage of our journey.
CONCLUSIONS ON TMM

Based on the above analysis we can see that even though the journey towards superior quality proved to be fruitful in the end (See Appendix - Some metrics data showing the progress, for diagrammatic representation of the improvements made on different parameters) the same could have been achieved in an easier manner by using the TMM model as a guiding framework. Using the TMM model would have prevented the effort we spent in reinventing the wheel. The advantage of TMM model is its easy integration with existing processes. As it is based on CMM, an organization already following CMM processes can easily adopt the TMM processes. The expected actions of the 3 views in TMM that is users, managers and testers for each of the levels serve as a good guide for expected behavior from these different entities.

However there are some drawbacks with the TMM model, for e.g.

- Though TMM states what is expected at each of the process levels it does not state clearly how a given baseline process improvement could be achieved. Although this model would have been ideal if used at the start of the journey it was not very clear on how we could further improve our process in certain selected areas after reaching the final stage of our journey.

- Many questions required changes at the organization level and hence for the model to be applicable to a group level some tailoring is required.

Test Process Improvement (TPI)

WHAT IS TPI?

TPI is a Test Process Improvement model. It classifies the entire testing process into 20 key areas. It gives an approach wherein an organization can self assess the level at which it is in each of the 20 key areas. There are a maximum of 4 levels (A, B, C, D) for each of the key areas with A being the least and D the highest. The manager can define a target required level for each of the areas. All key areas need not be at the same level. TPI also offers a set of improvement guidelines on how one could improve one’s process to a higher level for each of the areas. To reach a particular level there is a checkpoint which contains the set of requirements that the particular key area must satisfy to achieve that level.

There is a final maturity matrix which captures the levels and interdependencies between each of the key areas. Each of the rows of the matrix is a key area. The columns of the matrix are the maturity levels. There are 13 levels of maturity. Not all key areas at a given level map to a given level of maturity. For example if Test Strategy is at level A then the maturity for it is only 1 while if key area metrics is at level A the maturity is level 6. This is because all key areas do not have the same priority. In addition the 13 maturity stages are mapped into 3 categories: 1-5 is controlled, 6-10 is Efficient and 11-13 is Optimizing. (See Appendix - How TPI model looks, for a graphical representation of TPI components)
The base table is as follows.

<table>
<thead>
<tr>
<th>Key area</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Test strategy</td>
<td>A</td>
</tr>
<tr>
<td>Life-cycle model</td>
<td>A</td>
</tr>
<tr>
<td>Moment of involvement</td>
<td>A</td>
</tr>
<tr>
<td>Estimating and planning</td>
<td>A</td>
</tr>
<tr>
<td>Test specification techniques</td>
<td>A</td>
</tr>
<tr>
<td>Static test techniques</td>
<td>A</td>
</tr>
<tr>
<td>Metrics</td>
<td>A</td>
</tr>
<tr>
<td>Test automation</td>
<td>A</td>
</tr>
<tr>
<td>Test environment</td>
<td>A</td>
</tr>
<tr>
<td>Office environment</td>
<td>A</td>
</tr>
<tr>
<td>Commitment and motivation</td>
<td>A</td>
</tr>
<tr>
<td>Test functions and training</td>
<td>A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Key area</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Scope of methodology</td>
<td>A</td>
</tr>
<tr>
<td>Communication</td>
<td>A</td>
</tr>
<tr>
<td>Reporting</td>
<td>A</td>
</tr>
<tr>
<td>Defect management</td>
<td>A</td>
</tr>
<tr>
<td>Test ware management</td>
<td>A</td>
</tr>
<tr>
<td>Test process management</td>
<td>A</td>
</tr>
<tr>
<td>Evaluation</td>
<td>A</td>
</tr>
<tr>
<td>Low-level testing</td>
<td>A</td>
</tr>
</tbody>
</table>

Once the process has been evaluated we can map the actual levels for each of the key areas in this map. Based on our needs we could specify the target levels for each of the key areas. The model also offers improvement suggestions as a guideline. We can then bring in the necessary changes to the process to reach the target level. (See Appendix - Key area Description in TPI, for more details on what each key area stands for)
HOW TPI WAS APPLIED BY US TO IMPROVE THE EFFICIENCIES IN OUR GROUP

Once we reached stage 5 of our journey we were looking for ways we could improve and TPI seemed a right fit. There was also a readily available tool which aided us in our improvement. We did an initial evaluation and found our maturity levels for each of the 20 key areas. In around 5 areas we failed to meet the base A level itself. There were another 8 areas where we had an A rating level and in all the rest we had a B rating level. Discussed below are the details of how TPI was applied for process improvement in two key areas Moment of Testing and Test Automation.

Test automation

The base level for test automation key area was A. This involved answering Yes or No for a series of 3 questions. (See the Appendix - Checklist questions to qualify for Level A of TPI for Test Automation key area, to get the list of questions which needed to be answered) We wanted to reach a maturity level of B on this. For Base level B checkpoint questions at the initial level we got a score of 3 of 6. These questions were a trigger for us to improve our processes to the next level. There were also a set of improvement suggestions which gave us the direction on how to achieve a level.

Based on the improvement suggestions we came up with the need/basis of automation tools. (See Appendix - Improvement suggestions to qualify for Level B for Test Automation using TPI for a detailed list of suggestions) In our case there was a lot of manual testing involved where the engineer had to sit for hours in front of the monitor to ensure that the television outputted sufficient quality of service and there were no breaks and jitters. This led to lot of person dependence and frustration and hence the need for automation.

A structured study was then undertaken of the market and we narrowed down on few tools. After further study 2 tools stood out - the T1 tool and the T2 tool (names changed to maintain generality. As the second tool was inexpensive we went ahead and procured the same after a round of review with experts. For the T1 tool we applied the formal procedure as listed by TPI.

First training on T1 was arranged. Then a pilot project topic was chosen to check the applicability of Testquest and is currently in the process of implementation.

Once this pilot project is completed we would be in a better position to do the cost benefit analysis and conclude with a decision on the tool.

CONCLUSIONS ON TPI

This model is an excellent fit when there is a process for testing in place already and we want to improve the process on certain key parameters. This model gave us a good indication on the weakness in our process and what we need to focus on to improve. The improvement suggestions given in the TPI model give us direct directions to make the improvements. The model is also applicable when there is no base process and general improvement is needed. It is good as it allows us to focus on specific areas and gain improvement on the same.
CONCLUSION

A continuous focus would need to be provided to the processes being followed in testing and kind of techniques managers can employ to take us towards our goal of zero defects in products. Some of the points for improved performance include:

- A strong commitment/support from the senior management for ensuring the initiatives to improve the way of testing is successful.
- A dedicated/centralized testing group led by an independent test manager would go a long way in improving the quality of products coming from the division/organization.
- Testing activities need their own independent metrics which needs to be collected and analyzed.
- Testing professionals should be involved early in the project life cycle. Use of agile methodologies will also greatly improve product quality.

Any organization wishing to reach zero defects/superior quality will benefit a lot by applying the testing models in their group. Some conclusions on the test models:

- Test process models serve as a good pointer on what processes need to be improved and how they can be improved.
- An organization can self assess itself against TMM and TPI.
- There is no need to apply a model across the whole company and it can be initially piloted at a division/project level. If the results are good it can be applied across more projects in the organization.
- Organizations can also take a selective approach, tailor the model and try to apply only some parts of the model which they find necessary. The key to using these models is not about reaching some level but to see a relevant quantitative and qualitative improvement in the day to day performance.
- There is a need for careful and regular tracking the results of applying any changes in the process after applying suggestions from the models.

Application of the above points has helped in improved performance on many parameters. (See Appendix - Some metrics data showing the progress)

As managers we would need to find newer and radical ways as well as adapt prior knowledge and ideas to continuously improve. A focus on testing would go a long way in improving the performance of organizations.

APPENDIX

Some metrics data showing the progress

(Note: The numbers used here are not actuals and are used for illustration purpose only)
How TPI model looks

The model is visualised as follows:
Checklist questions to qualify for Level A of TPI for Test Automation key area

A decision has been taken to automate certain activities in the planning and/or execution phases. The test management and the party who pays for the investment in the tools (generally the line management or project management) are involved in this decision;

Use is made of automated tools that support certain activities in the planning and execution phases (such as a scheduling tool, a defects registration tool and/or home-built stubs and drivers);

The test management and the party paying for the investment in the tools acknowledge that the tools being used provide more advantages than disadvantages.

Improvement Suggestions to qualify for Level B for Test Automation using TPI

Make an inventory and find a basis for the need for and the necessity of tools. Do not restrict the search to commercially available packages. Even very small, personally created tools such as stubs, drivers and displays in the system can be very useful. Builders can often makes such tools within a short space of time.

Carry out a structured selection and implementation process. Requirements (restrictions) and wishes are possible with respect to various aspects:

* functionality (for example programmable, recognition of GUI objects)
* service level of the supplier
* quality
* costs
* (environment) hardware and software (very important: does the tool work in the specific environment?)
* number of users, knowledge level, quality of documentation.

Arrange training and support for a tool that is to be purchased.

Carry out a pilot project.

Ensure that expert knowledge about the tool is present within the team (this often concerns a person with a technical background, who may also have programming skills).

Describe the structure of the tool.

Make a well-founded cost/profit analysis prior to purchasing the tool. To get an impression of the differences in costs between the manual (M) and automated tests in the case of a Capture & Playback tool (A), see the following:

1. Determine which test effort qualifies for automation Suppose that a regression test is carried out four times a year and involves four fulltime testers for three weeks: 4 x 4 x 3 x 5 = 240 person-days a year.

2. Estimate the ‘pure execution time’ The ‘pure execution time’ is the time that can be automated. It is the time that someone spends in front of a monitor carrying out the application test cases, plus the time spent determining the differences (calculation yields 10 instead of 9). Analyzing the differences and searching for their causes do not belong to pure execution time (calculation yields 10 instead of 9, because in function X a certain percentage is not taken into account). In the example of 240 person-days a year, we estimate the pure execution time to be a quarter, i.e. 60 person-days a year.
3. Make estimates for the following propositions. The development of automated tests, on average costs $X$ times the amount of time as that of a manual test execution (in the example we use $X = 2$, therefore $A = M \times 2$). Automated retesting is $Y$ times faster than manual retesting (in the example, we use $Y = 4$, $A = M/4$).

4. Calculate the possible time gain. Manual = 60 person-days a year, or 15 person-days per regression test. Automated = (development of the test costs twice as much: $15 \times 2$) + (retests are four times faster: $3 \times 15/4$) = 41 person-days in the first year and $(4 \times 15/4 =) 15$ person-days for each year following. Profits = the difference, i.e. 19 person-days in the first year and 45 person-days in each following year. The above-mentioned information can also be displayed in a diagram, for example to determine the break-even point: after how many tests will the tool start to pay itself.

5. Estimate the following factors: (‘-’ stands for the costs and ‘+’ for the profits associated with a tool)
   - purchase of tool;
   - training;
   - setting up the tool;
   - maintenance of scripts in case of changes (maintenance of automated scripts is much more labor intensive than maintenance of manual scripts)
   + Higher quality of automated test (assuming that the attention of human testers decreases when carrying out the $X$th regression test);
   + Greater motivation and productivity of personnel (a tool provides a new dimension to testing, it is often ‘fun’);
   + Faster lead time. These factors must be estimated, where maintenance on the automated scripts in particular may require a great deal of effort, but is difficult to predict.

6. Now make the full costs/profits comparison. The comparison is based on a large number of assumptions, but provides a basis for further argumentation. Often the expectations appear to be far too high. It is, however, also possible to make a comparison for ‘normal’ tests instead of regression tests. In that case, it should be taken into consideration that the first test execution as a rule takes twice as long as a retest and that not all tests result in a round of retests.

Check the costs and profits periodically to see if the earn-back time has already been reached. Calculate budget costs, such as setting up the tool and training, separately and calculate a certain overhead or as the case may be, time gain for the use of the tool.

### Key area description in TPI

**Test strategy:** The test strategy has to be focused on detecting the most important defects as early and as cheaply as possible. The test strategy defines which requirements and (quality) risks are covered by what tests. The better each test level defines its own strategy and the more the different test level strategies are adjusted to each other, the higher the quality of the overall test strategy.

**Life-cycle model:** Within the test process a number of phases can be defined, such as planning, preparation, specification, execution and completion. In each phase several activities are performed. For each activity the following aspects should be defined: purpose, input, process, output, dependencies, applicable techniques and tools, required facilities, documentation, etc. The importance of using a life-cycle model is an improved predictability and controllability of the test process, because the different activities can be planned and monitored in mutual cohesion.

**Moment of involvement:** Although the actual execution of the test normally begins after the realization of the software, the test process must and can start much earlier. An earlier involvement of testing in the system development path helps to find defects as soon and easy as possible and perhaps even to prevent errors. A better adjustment between the different tests can be done and the time that testing is on the critical path of the project can be kept as short as possible.
Estimating and planning: Test planning and estimating indicate which activities have to be carried out when, and the necessary resources (people). Good estimating and planning are very important, because they are the basis of, for example, allocating resources for a certain time frame.

Test specification techniques: It is defined as “a standardized way of deriving test cases from source information”. Applying these techniques gives insight into the quality and depth of the tests and increases the reusability of the test.

Static test techniques: Not everything can and should be tested dynamically, that is, by running programs. Inspection of products without running programs, or the evaluation of measures which must lead to a certain quality level, is called static tests. Checklists are very useful for this.

Metrics are quantified observations of the characteristics of a product or process. For the test process, metrics of the progress of the process and the quality of the tested system are very important. They are used to control the test process, to substantiate the test advice and also to make it possible to compare systems or processes - why has one system far fewer failures in operation than another system or why is one test process faster and more thorough than another? Specifically for improving the test process, metrics are important by evaluating consequences of certain improvement actions, by comparing data before and after performing the action.

Test automation: Automation within the test process can take place in many ways and has in general one or more of the following aims:
- fewer hours needed
- shorter lead time
- more test depth
- increased test flexibility
- more and/or faster insight in test process status
- better motivation of the testers

Testing environment: The test execution takes place in a so-called test environment. This environment mainly comprises the following components:
- hardware
- software
- means of communication
- facilities for building and using databases and files;
- procedures.

The environment should be composed and set up in such a way that by means of the test results it can be optimally determined to what extent the test object meets the requirements. The environment has a large influence on the quality, lead time, and cost of the test process. Important aspects of the environment are responsibilities, management, on-time and sufficient availability, representativeness, and flexibility.

Office environment: The test staff need rooms, desks, chairs, PCs, word-processing facilities, printers, telephones, and so on. A good and timely organization of the office environment has a positive influence on the motivation of the test staff, on communication in- and outside the team, and on the efficiency of the work.

Commitment and motivation: The commitment and the motivation of the persons involved in testing are important prerequisites for a smoothly running test process. The persons involved are not only the testers, but also, for example, the project management and the line management personnel. The latter are mainly important in the sense of creating good conditions. The test process thus receives enough time, money, and resources (quantitatively and qualitatively) to perform a good test, in which cooperation and good communication with the rest of the project results in a total process with optimum efficiency.
Testing functions and training: In a test process the correct composition of a test team is very important. A mix of different disciplines, functions, knowledge, and skills is required. Besides specific test expertise, knowledge of the subject matter, knowledge of the organization and general IT knowledge is required. Social skills are also important. For acquiring this mix, training etc. is required.

Scope of methodology: For each test process in the organization a certain methodology or working method is used, comprising activities, procedures, regulations, techniques etc. When these methodologies are different each time or when the methodology is so generic that many parts have to be drawn up again each time, it has a negative effect on the test process efficiency. The aim is that the organization uses a methodology which is sufficiently generic to be applicable in every situation, but which contains enough detail so that it is not necessary to rethink the same items again each time.

Communication: In a test process, communication with the people involved must take place in several ways, within the test team as well as with parties such as the developer, the user, the customer, etc. These communication forms are important for a smoothly running test process, not only to create good conditions and to optimize the test strategy, but also to communicate about the progress and the quality. Reporting Testing is not so much “defect detection” as about giving insight in the quality level of the product. Reporting should be aimed at giving well-founded advice to the customer concerning the product and even the system development process.

Defect management: Although managing defects is in fact a project matter and not specifically of the testers, the testers are mainly involved in it. Good management should be able to track the life-cycle of a defect and also to support the analysis of quality trends in the detected defects. Such analysis is used, for example, to give well-founded quality advice.

Testware management: The products of testing should be maintainable and reusable and so they must be managed. Besides the products of the testing themselves, such as test plans, specifications, databases and files, it is important that the products of previous processes such as functional design and realization are managed well, because the test process can be disrupted if the wrong program versions, etc. are delivered. If testers make demands upon version management of these products, a positive influence is exerted and the testability of the product is increased.

Test process management: For managing each process and activity, the four steps from the Deming circle are essential: plan, do, check and act. Process management is of vital importance for the realization of an optimal test in an often turbulent test process.

Evaluation: Evaluation means inspecting intermediate products such as the requirements and the functional design. The importance of evaluation is that the defects are found at a much earlier stage in the development process than with testing. This makes the rework costs much lower. Also, evaluation can be set up more easily because there is no need to run programs or to set up an environment etc.

Low-level testing: The low-level tests are almost exclusively carried out by the developers. Well-known low-level tests are the unit test and the integration test. Just as evaluation, the tests find defects at an earlier stage of the system development path than the high-level tests. Low-level testing is efficient, because it requires little communication and because often the finder is both the error producer as well as the one who corrects the defect.
REFERENCES


ABOUT THE AUTHOR

Arthi Venkataraman is currently a Manager in the Consumer Electronics domain of Wipro Technologies. Her current role involves managing the quality of all products which comes out of the set top box group. She has over 8 years of experience in the design, development and testing of projects in the embedded domain. She has a B.E Degree in Computer Science from University Visveshvariah College of Engineering, Bangalore and an MBA (PGDSM) from IIM, Bangalore.