BIO

PAPER

T12

November 7, 2002
1:30 PM

MAKING A DIFFERENCE WITH TEST ASSESSMENTS

Sigrid Eldh
Ericsson AB

International Conference On
Software Testing Analysis & Review
November 4-8, 2002
Anaheim, CA USA

Sigrid Eldh

Sigrid Eldh has over 20 years experience in the software industry working with a wide range of software in different companies. She is a co-founder of SAST, the Swedish Association for Software Testing and also started ASTA, the Australian Software Testing Association. She has a Masters degree in Computer Science and a degree in Psychotherapy and Organisational behaviour. Sigrid is also on the examination board for ISEB certification of testers, and the chair for the Swedish National Board for the of ISEB certification. She is an author of books in software, testing and management. She currently works as a Test Expert and Manager for Ericsson AB. Sigrid was the program chair of EuroStar'99. Sigrid is a frequent speaker at international testing conferences and has published many papers throughout the world on software testing and related subjects.

Making a Difference with Test Assessments

By Sigrid Eldh
Integration & Verification
Ericsson AB, www.ericsson.com
P.O. Box 1505, SE- 125 25 Älvsjö, Sweden
Tel: +46 8 727 35 16 Mobile +46 730 477 998
Email: Sigrid.Eldh@uab.ericsson.se

Abstract

Test assessments are a powerful way to understand the current status of your testing. These assessments provide an independent view of where you are and they guide you to where you're going. They also highlight what your team needs to do to reach its testing goals. From her experiences performing test assessments, Sigrid Eldh covers all aspects of her assessment approach including processes, management issues, automation, and test deliverables. You too can use assessments as a tool for your own success because they confront these contextual issues and give you valuable feedback that can be implemented immediately.

Get a step-by-step explanation of how to improve your testing using assessments and learn about the positive side benefits of using assessments. Find out how to get immediate benefits from a test assessment.

Introduction

The power of assessing lies in the getting a second opinion about how the testing is done. If you are doing a self-assessment, you need a model, questionnaire or some outside reliable resource to give that second opinion. The easiest is of course to get someone from the outside of your company, with great experience from testing and assessment to give you that independent view. This paper is hopefully supportive to you regardless of what you chose to do.

Why assess?

The reason for assessment is that you want to improve. There is almost always room for improvement, you can always be more cost efficient, do testing more effectively, find areas that makes the work easier and the test execution not the endless effort it often becomes (at the end of the life-cycle). Regardless in what stage you are in your project, organisation or position, there is always possible to do an assessment. The answer you get is a confirmation on what is good and bad in the way your test team works, and how it works in conjunction with other roles within your company. If the “model” is good, you will also be able to pinpoint the most important areas to improve your own testing – or where your get the most benefit for yourself and your organisation. You will basically see the possibilities to move forward.

An assessment has also other benefits, the well known “Hawthorn effect” will work, in short, by just doing the assessment, the assessment in it self will focus your personnel, make your management aware of testing issues, and bring improvements as a side effect result. What your highlight will be in focus. What you attend to will be done. Just the ambition of wanting to change things to the better will have an opportunity to emerge. You will earn valuable time, money and motivation with an assessment.

An assessment could work as a snapshot of how the verification works for you right now. It will give you an instant feedback on how your process works, if the tools are used or not and why, how your testing personnel view their work, and also you the others view the testing. You will know what status testing has within your organisation. You will know what is good and what is bad within your verification, and a great “map” with different directions to take. You will get confirmation about issues within your verification, and many possibilities to move forward.

The hardest part will be to decide which area that will give the best benefit to improve, a decision that I hope to make easier in this paper. By assessing your test you will clarify your goals that will give you that ”push” to move forward.

Why assess verification?

A part of this paper is about assessment, which of course is a general statement that is valid in all phases and areas of software development and maintenance. I think that assessment of verification is specially valuable since it is often not understood and fully utilized by the rest of the organisation, and can use extra support from the outside to make the organisation aware of the potential contribution verification can offer.

When to do an assessment

Of course an assessment gives more value in an ongoing project and organisation, when at least some things remains stable, which every organisation will do if it contains the same people to a larger part. There is never a "good" time, but there is always time to conduct the assessment, regardless of which phase you are in the project. If the projects differ a lot, there is of course always valuable to evaluate each project by itself and compare what is. Totally new projects and organisations can use the set of assessment question as a "checklist" for what to implement. ”Have you thought of this?" could be a good to ask you.

Self-Assessment vs. Independent Assessment

Self-assessment has definite drawbacks because we are "home-blind". It is a human skill to adjust and be flexible into an environment, but it is just that particular skill that makes us loose the "objective" and maybe introspective view of the organisation in which we function. Basically, we are biased, because there are people we like and dislike in the organisation, regardless of how professional we are. We are humans and our personal "likings" tend to influence our judgement and objectivity. The problem will be for you as the assessor to differentiate your views from other views, there is a time where your opinion has a place and there is a time when you are listening and are more a "recorder" of others views. Another drawback about self-assessment if you go around in your organisation is that people you interview might have pre-conceptual ideas about you (depending on things beyond you, such as organisation, history etc). An example can be that the mere fact that you come from "the verification department" can make a designer not telling you truthfully about what is going on and not.

There are advantages of doing self-assessment compared to an independent assessor. An inside person might easier see through a "façade" that some organisations tend to show when an outsider comes. You can actually use the assessment as an excuse to talk to "the others", and actually be invited to do so. The strength of assessing – regardless of which method or area you chose, will give you a good overview of the area. Your subjectivity can be an asset in the picking of improvements and choosing the persons to interview. Depending on who and how (and when) you can sometimes as a self-assessor get change must faster, especially if you are the manager or test leader for a team. You can just go ahead and implement some of the improvements on the list. The important thing to keep in mind is that you, when you do a self-assessment actually have a model or set of questions for support – so you have some distance from yourself.

Every model has also its drawback in the sense that it also includes an underlying judgement that might not be right in the context you work. A good "independent" assessment can therefore be very helpful, if you are trying to improve your testing, and a good model. The model provides you with a set of questions, and is the tool, that will support you. Every model has their totally different approaches but is often using the same underlying method.

What to select for improvement

It is important to select areas for the improvement, and to be successful; the best start is to choose Quick winners. It could be items of improvement that has a low cost, or is an easy action to complete, or is already ongoing. Examples could be creating a test plan template out of a test plan and put it on the web as well as an example. This is easy and low cost, but will save time for next project (also if template has been discussed). This could go for any test documentation, thus test reports, test environment specification, etc.

Also items with some sort of degree of acceptance should be chosen. An example is if there is a lot of attention on the unit testing in an object-oriented (C++) environment, that needs improvement, there might be a partly acceptance to introduce a memory-leek search tool to the developers, and also introduce inspections on headers and interfaces as a quick improvement. Maybe there is even acceptance enough to go for a full-fledged test harness tool usable and fitted for the developers unit testing. Maybe a short course on how to perform white-box testing, and a walk trough of the standards in component testing all that is needed.

Cost/benefits ratio is a selection item as well for improvement. In the above example, introducing a test harness tool might be more costly than the benefit in the time pan for a particular project allows, and maybe it is not worth it, compared to just do a workshop combined with course and inspections.

To do improvements that will reduce the highest risk, is of course beneficial, usually highest risk are – being not ready on time (with the right quality) and the remedy is of course to make sure that the quality of the program/system is up to standard, and that all test executions are automated or prepared by instructions before test executions starts. To reduce the risk of not being ready on time might be possible if you are aware of this "early on" in the process. Then the remedy is most often the need of more resources (trained testers and people). Sometimes the problem of always becoming late is not a resource issue, but a skill problem, that of really knowing testing and the system enough to be able to prepare beforehand. If you do not know enough about the system until it is visible for you, then preparing is of course difficult to do in advance. Remedy, is of course participating more in the design, and more education on testing.

Another selection criteria could be based on what is most visible for others, or some particular group. One area to improve could be the progress report from the test team that get sent to a large part of the organisation every week. Another could be improving websites, taking part in development and system development work in the early stages of the projects, which makes testing visible for others.

To work with what the organisation is already working with is also a successful strategy, thus fining improvements that are in synchronization with goals. An example could be shorten delivery times, where verification can put a lot of process and planning issues in to make sure they time on the critical path (the test execution) is as effective as possible, and that proactive activities take place to make quality improvement possible.

Conducting an assessment

The practicalities of conducting an assessment are straightforward. Therefore my focus here lies in things that one tends to forget when conducting an assessment. First, I think defining a goal for the assessment is a must, if you do not know what you want you are not likely to get it. At least the goal for the assessment should be: "I want to know how good and bad we are" or "I would like a list of improvements to do within my testing" or something similar.

When the goal is written the assessment must have a reasonable scope. One way to define what the assessment is to be done is going “backwards”. Examples of that is “Whatever I can assess within 2-3 days” or “Maximum total of 100 hours for entire assessment”. Other scope setters are “all verification units within organisation X” or “Entire company” , which both are defined as a scope “forward”, thus time limits are harder to estimate (and cost of assessment). The reason is of course the level of thoroughness; you could do a “quick overview” of a more in-depth analysis of problems and improvement areas. The first might be done as a starter, and the in-depth analysis might be valuable as a second step, or in an organisation that has a rather solid or stable development. Also you need to decide when, which is rather obvious, but result can depend greatly on the timing. I would advise against after a big re-organisation (if not people are stable as well as processes etc) or in the new company. Then you can use the assessment more as a checklist than a constant improvement tool. One “easy to forget” thing is to say to whom, or for whom the assessment is done. This is especially important if you come from the outside. You must know two things, one you are paid your fee and secondly, that you are asked to be “independent”, thus your goal should be that you are as honest as you can, even if it might be somewhat “painful” sometimes for the management or the person requesting your services. But there are always ways to say it in a non-critical fashion, as an improvement suggestion. My experience is that honest open feedback “how you received it” could be very important to the persons who are “ordering/paying for” the assessment.

Finding a Model that suits

Then you need to make sure that you have defined the structure and method, which means that you have a plan for the assessment in it self, but also that you have a model to use and base the assessment on, you need to choose the ”model” to use. One example is the book (TPI) Test Process Improvement by Martin Pol and Tim Koomen, (Addison-Wesley, ISBN 0-201-59624-5), which is rather helpful. Another more “general” assessment (not particularly focused on testing) is CMM or the new CMM Integrated (by Carnegie Mellons Software Engineering Institute (SEI) see http://www.sei.cmu.edu/activities/cmm/cmm.sum.html), which the latter has a process area called verification and validation (but which is mostly focused on inspections). There is also a method called TMM from Illinois Institue of Technology, (TMM = Test Maturity Model), see http://www.stsc.hill.af.mil/crosstalk/1996/aug/developi.asp, or a power point presentation http://www.cs.umn.edu/crisys/spin/talks/hedger-jan-00.htm , which to my knowledge is only published “in paper” and not as a ready available book. Several models have been circulating on conferences, for instance TIM (Test Improvement Model) see EuroStar 1996 Proceedings. “Any” goes, because all have positives and drawbacks, though I prefer the TPI model, much because it is published, and because it also has a questionnaire attached to its website, see http://www.iquip.nl/tpi. Even if it is one of the better and thorough models it has drawbacks. Its underlying definitions of in what order things are to be done, and the comparisons of levels, are sometimes questionable. I will discuss more what I mean with these levels later in this document. Basically, it is good to be soundly questionable to any model existing compared to your own experience. When browsing the Internet for more models, I found too, that the StarWest Program chair Lee Copeland in his “The Seven Habits of Highly Effective Testing Organizations” in a Sticky Minds publication on the web, http://www.stickyminds.com/sitewide.asp?ObjectId=2919&Function=DETAILBROWSE&ObjectType=COL, has pretty much drawn the same conclusions, that CMM and TPI are really two basic documents to rely on. Then he added the much needed conclusions of areas important to improve, which I fully support.

The Structure of an Assessment

First you should hold a ”kick-off meeting” where the purposes are the following:

- Defining the main questions to be answered
- Define scope thus describe limits and setting
- Answer on when and how the assessment is to take place
- Defining material to be reviewed.
- Defining Goal of assessment

Then the preparations start by checking available documents. A suggestive list of these test documents is:

- Test Process and/or Development process
- Test Strategy and/or Development strategy
- Test Plans (several levels if exists)
- Test Records
- Test Report
- Test Progress Reports
- Any Test Automation documentation (strategy, directive etc)
- Test Specifications from all levels (not all but defining test type, level priority)
- Test Instruction(s) from all levels
- Any directive document, for example process on how to prioritise tests, RCA (Root Cause Analysis) etc.
- Project Specification(s)
- System documents (that might define Test design!)
- Test Environment description (where content is the hardware/software environment where the tests have been conducted)
- Any other important document

The idea is for the assessor to prepare questions to the interview personnel, partly related to the documentation, and what to look for is:

- Absence of document: If the above document does not exist, what exists instead? Where is the information stored? Is this a sign on immaturity on the testing, or is it just not a problem?
- Completeness; is the document “complete” enough to support its intention? For example: Test instruction, Test specification or Test process.
- Usability; Do people use this document regularly, or use it to its purpose? For example: Test plan or Test instruction, Test Automation document.
- How well known is this document? For example test strategy.
- Are others (design, project managers etc) aware of this document and do they use it? Thus take decisions based upon it? For example: test progress report, RCA or prioritisation
- Presence and absence of testing in documentation (a general question, but visible in project specifications, development process and systems documentation) – this will tell you if testing is an isolated area or an integrated part of the development.

- Level of thoroughness: Does the documentation show “maturity”? Is the selected test specification “good tests” to your knowledge? Why not? How has the process of selecting test cases been conducted?

I’m sure the list can be much longer, but this is to start your own creative thinking. The end result should be a lot of questions that you “suspect” are the case, and already here, you are starting to form an opinion of the organisation. Here it is important to separate your own conclusions and make sure that you really ask “openly” to confirm or negate your conclusions. My opinion here is that probably your “idea” of how things are done is partly right, but it is very hard to base only on documentation. I have visited an excellent test site; where the documentation had deteriorated due to no other (outside test) person ever read them. All the basic documents were in place of course, and the only real problem that happened, was the dependency to the personnel (which always is the case if the documentation is not up to standard). With this I mean it is important to be careful to pass judgment. The conclusion is though basically that the general rule is “poorly written – poorly thought”, thus a bad documentation does usually witness of the test process is not conducted well enough. Sometimes you find institutions that have “great documentation” but the content and the actual test execution, test selection and test process is really weak. This comes usually from a lack of knowledge of what is done right, and can be quite tricky to puncture, since it from the outside looks fine. Usually this gets revealed when you ask people questions about more “in-depth” methods of verification and the approach they have to their testing.

Before you start your interviews, you should yourself fill in the assessment questionnaire, with your own opinions. Now you will have an opportunity to see what insight and influence you have when you compare the result later, and also you will have time to focus on those questions that you have absolutely no idea what to answer. This will be something you need to find out in the interviews. It is also a good “self- rehearsal” of the model before entering the interviews.

Interviews

Now you are prepared for interviews with a lot of questions. If you are lucky you have an idea of what persons you are to talk to. I think you should use combined interviews, to make sure that you talk also to non-testers (project managers, designers, etc) and also if there is a receiving or dependent organisation to the organisation you are assessing. This could be good to get a perspective of the testing, since testing always work in a context. A positive side effect of this is also that a lot of other organisations and people around testing, gets aware that the testers are taking their job seriously, and really want support to improve. Many issues also bounces back to other parts of the organisation, and this becomes clearer if it is a result of the assessment.

There is a whole science in how to interview people, and I’m sure you can pick up a book in the subject if it is your first time (and anyhow to get some good hints!). The most basic idea I have to get as much out of an interview as possible is that first you must “make a contract”, thus, “Welcome, my name is x, my purpose is to assess the verification in this department, we have this long time, and what ever you say will be Quoted Anonymously or disguised in my report. I think it is of greatest importance to have some sort of “protection” for the persons to be interviewed, and also that you respect this code of ethics. You need to create a secure environment, where the interviewed person feels secure. After contract negotiation (note, goes both ways!), you should always ask open-ended questions first (before you have conveyed your ideas to the person). So for each person ask: “What do you think needs improvement within testing from your perspective?" You will be amazed of the insight persons have to what is going on, and the answers to these first questions might be the most important and valuable improvements you can suggest. Also, it comes from the persons itself, thus fulfils a wish from them to become better. You will definitely be a winner writing that down.

> HINT to use this method for goals:
> A "super advice": If you are a manager of a group of tester, this is what you really can ask each and everyone of them in a short "post-it" note 10 minute session "What do you think needs improvement within testing from your perspective?" Name 3 short improvements for each person: Combine that and goal process for this year is done!

After the open-ended improvement question, you will start leading the interview person into the areas where you have prepared questions based on documentation, but still try to keep them as open as possible: Thus "How do you know how long testing will take?" could aim to check if they are aware of the test process, if they are doing some test analysis and design before they commit to a time plan for the tests, and when and if they are writing it down and updating the test plan when new information comes? You could also – when you feel the discussion is open and honest, ask directly – Do you update your test plan? When etc…there might be reasonable answers and reasons. Then you also need to check your "model" with some questions, thus is TPI model one great question is when test gets involved in the project. The area is called "moment of involvement", and is a good sign of maturity in the organisation, and also an area you definitely should improve quickly to answer, when the project starts. The problem could be if you only stick to the question in the model, you might get a very narrow view, depending on the model. Also, you do not want longer interviews than 1-1.5 hours, and there is never time to go through "everything". So, my suggestion is that you ask questions that seem to fit the person, thus no idea to talk planning figures with a tester, since it is the test leader that plans, etc.

Self Assessment for several roles

Then you could hand out the questionnaire to the persons you think knows enough about the testing and the process (or should know!) and ask them to do a self assessment. It is good to select self-assessors out of several persons with preferable different roles.

Another area of your attention and focus should be what improvements that already exist within the organisation. These should also be noted, if they actually contain a direct connection with testing & verification, or if the relation is somewhat unclear. This could be helpful for you to know how to "sell" the improvements in your report. If you see improvements that support the company goals, it is much easier to get them done.

Compare Questionnaires – "Assess"

Now comes the most difficult task. You will need to combine all information you have at hand into one "opinion" or suggestion. I do this by basically writing the report.

I as an assessor also take the actual assessment process into place, thus, I usually write down my impressions along the way. Thus I do my report in the same steps as described above.

My headings are the same as the above steps including my reflections: General, Kick-off Meeting (which includes goals etc) and my "personal reflections" at this stage. Then the heading is the documentation feedback (where I clearly state the questions I feel has come as a result, and where I comment each document with good and bad. In the interviews headings I make sure that the areas where disagreements of how things work are shown. Most I really relate to each area based on the model, thus the combined self-assessment, the interviews will be the result in the model (and questions raised from the documentation is answered). My goal is to be has honest as I can on how I perceive things. (This makes you as an assessor vulnerable, but it also gives the organisation an opportunity to disagree – which is actually a good thing… it is not about who is right, it is about what is going to be done about it!).

It sounds harder than it is to assemble the results. For each area in the model, I tell the assessed result, where I suggest improvements and comment on what is good and bad, and if there was someone interviewed that had opinion about that particular area. I will write as model true answer as possible, then adding clearly what my opinion is, for the receiver of the report to be able to differentiate the model and my expertise advice.

After each area has been covered, a summary statement is good to close the report.

Result Presentation and Report

In addition to the report a presentation should be held in immediate or time wise very close to the assessment. This will not be the full report, but the first "collective" feedback of "obvious improvements". Also here is the chance to push/sell the report for details, since the largest problem is sometimes to get people to actually read the improvement suggestions and take them to their heart and do something with them. Many participate in the interview, and then that is that. Therefore a "first result" presentation will have a chance to give a taste of what the report has for content, a great time to push improvements and make the organisation responsible for what they are doing. I also suggest that the people "ordering" the assessment gets to read your report through before you publish it, and you can check their reaction and also get real help in reviewing it (if you do not have that service). There might be things you totally misunderstood, or things they totally misunderstand, so you might want to rephrase things.

This concludes the step-by step assessment method practically, and I will now work into more details of the assessment method.

What to investigate

The hardest problem is of course to understand the testing model and really understand what is a "better testing" than not. I will try to explain. There are four main areas in the TPI. I think a good assessment should address all of them, but definitely there are some things that are more important. The four areas are:

- Life-cycle model
- Techniques
- Infra-structure
- Organisation

The Life-cycle model is how testing is related to the entire development and maintenance. The actual techniques used are a particular testing skill level (and maturity). The infrastructure is of course about the actual context in which the testing works? Is it even possible to conduct a good testing? Finally the organisation, which has to do if the testing & verification is independent or not, and how the organisation treats, utilise and use the verification & testing as well as their testers. The status of testing will be obvious. Within this area there are several particular topics as presented in the TPI model, to which I have attached my personal comments! (Please use the book as a "cross reference", I'm aware that this might be rather difficult to encompass in this paper).

In the TPI model (see page 35-40 in book ref 1 for description) the following areas have been highlighted as key areas:

Test Strategy: Focuses on detecting the most important defect as early as possible. The better the strategy, the better it adapts the test depending on risks covered by the test. It is important that there is a strategy that encompasses both low- and high-level testing, because it is the only way to really improve.

Life-cycle model: Within the test process is a number of phases, and defined activities (to a set of rules). The importance of the life-cycle model gives controllability and predictability of the test process. Here the maturity lies in having a clear test process, with all the phases planning, preparation specification, execution and completion. Of course the details of what activities are done within each phase is the difference of good and bad, and any "process" has these activities. The TPI model differentiates that the better level does also prepare and conclude, which I think is obvious for doing a good job in any process. If you don't prepare, how are you to execute, and if you are not concluding (in lets say a test report) how are you to learn anything for the future (and convey the result achieved?) Thus, this is rather basic, thus going in and specifying details of activities can mean a tremendously large difference in the result – i.e. visible signs of HOW things are done.

Moment of involvement: Basically, this means when the testers starts actively in the development process (when do the people get involved with the testing activities!) Earlier the better, thus I think that starting at project initiation is very important for project success.

Estimating and planning: This area differentiates between substantial and statistically based estimation and planning. I personally feel this is more general project know-how than particular testing specific, even if the artefacts are testing items, thus a CMM model is as useful here as TPI.

Test Specification Techniques: Here is the purpose to derive test cases from source information in a standardized way. Applying these techniques is valued from how "formal" they are used, thus if you can apply coverage as a result. This is a limited view to my point, thus many black-box techniques lack the coverage aspect, and thus some techniques, which of course should be known and used when appropriate, is not always the most effective way of creating the test cases. In low-level testing (unit testing), test techniques are much more useful and should be used (and known) by all programmers/developers, thus of course is this a very important key area. If you are not familiar and use the available techniques, how can you state that the testing is appropriate?

Static Test Techniques: Inspection and using checklists is all that is mentioned. Again, this is a good software engineering practice, and on the borderline of particular test technique, but fits in here because there are no other efficient test way at non-executable items. I think using inspections are extremely valuable, and checklist to support and improve the technique is good and ambitious, but sometimes (not TPI) over exaggerate the value of the formalism in the technique.

Metrics: Metrics are about quantifying observations of product and process. The value system in TPI is that A-level (lowest level) is Project metrics for a product. The Project metrics for the process is level B. Then Level C is system metrics, and the highest-level D if of course organisation metrics (more than 1 system). Have worked extensively with metrics, my viewpoint is that an organisation can have great metrics system and still repeatedly to severe mistakes. Thus, being "good" has unfortunately little to do with the fact of collecting the measurements and keeping metrics. It has more to do with how the metrics is used, what decisions are based upon them, and if the metrics collected really has the validity they claim. Some metrics definitely have not, so I think this has little to do specifically with testing. There are some few metrics that is of course important: Cost and time, thus faults/defects found in different phases of the product, and correction rate. This is an area I do not initially spend much time in with an assessment. I check that the above exists properly, since it is a research subject in itself.

Test Automation: The area focuses on the tool support of testing (planning control, specification, test databases, execution and analysis). Here I totally disagree with the TPI model in their ranking of tool usage and insight. I do not think that the ladder is feasible in all products and in all testing, but I can see that their suggestion is true for some types of systems. The most important question to ask in an assessment is, what tools are used and not, where could tools be beneficial, and if the tool selection has been done properly. Usually (but definitely not always) tool support for managing the test cases and test documentation could be beneficial for most systems (to support reuse) but again, this depends on size of system, its reuse etc, and should not be generalised.

Test Environment: Having a proper available test environment has to do with if it works sufficiently or not. The importance is that the testers feel it is sufficient in an assessment, and improvements here should be taken care of. Again I do not agree with the TPI authors, thus the "levels" severely depend on the system developed, and is not generally applicable to my mind. Having a flexible test environment must be a business decision, when there is a need for a flexible environment, otherwise one can never reach the "highest level" if you are in a stable product test environment.

Office Environment: That this area of TPI is mentioned in the test improvement is to me so basic, that it really is misplaced. It is a basic that the workplace is sufficient, and is definitely not something I focus on in an assessment. I can of course think of some areas in the world when this has a place, but sorry, I will not spend time on that.

Commitment and motivation: Here is on the other hand an attempt to grasp an important leadership issues. First level is that there is budget and time assigned to testing (which is many times a starting point!). Second level (B) is that testing is integrated in the project organisation, which I consider being a “non” question. I think the goal is to say that testers are an important part, and I couldn’t agree more on the importance of that, how otherwise are testing at all going to work? The next motivation level is Test Engineering (Level C), basically that there is an important skill and knowledge in test (the question is really: Does the rest of the company think so, or is it just the testers themselves?). But again, even if this is an important assessment area, it has less to do with testing, and more to do with the status and dependence the company has to an efficient testing. Thus there is never an argue in a safety-critical software company, how important testing really is! Thus, it definitely depends again on the product, what respect and understanding testing is going to have (and implicit, how professional you are in your testing!).

Test functions and training: Here is again a “basic” role description, where the level are first having appointed testers and test managers. Next level (B) is the (Formal) methodical, technical and functional support management of the test process, testware and infrastructure. Which again, I do not agree is a necessity to spend time on. The next and highest level here (C) is formal internal quality assurance, which also are “good”, CMM- basics, and good software engineering behaviour, but except from having testing roles (which also could be questioned in some companies), I suggest checking proper training in testing is done is the most important check.

Scope of methodology: Here the goal is to have general applicable test process, to be used anywhere. Again, the levels goes from project specific to organisation generic and then optimising. This is to me not as important and definitely depends on the size and culture of the company. It can be both a god and bad sign that the process is different, so the assessment must take into consideration what is really useful. In large companies, a generic process could be more a hindrance (or so shallow it is hardly useful), than adapting to particular needs. I can see it as a sound sign, if the project (based from the generic) can really put the interest and effort in and make a better particular process improvement and evaluate that in a project. But the role is in itself only “one” sign to assess the maturity of the company.

Communication: This is of greatest importance in a company, but is it particular to testing? No. Thus the three levels in TPI seem very artificial in its measurement. Are you unsure of how good communication is done and what the signs are, you can get support from the TPI book, otherwise I suggest that you use your common sense. The most important feature I always look for is “what is NOT talked about”. And, can the group discuss the most difficult problems (in a sensible respectful way?). Is it an “us and them”-situation, is “blaming” a common feature? How is people’s personal responsibility for solving issues and improving?

Reporting: Well, particular good testing reports, regular with right information is definitely a “good” behaviour, if the result is used, and decisions based upon the information, which is asked for. TPI gives some support in what to report Level A- Defects, Level B Progress (status of tests and products) and activities (Cost, time milestone) and also defects with priorities. Level C adds risks and recommendations, (substantiated with metrics) and Level D also adds that the recommendations have a software process improvement character in addition. This area is again “basic” software engineering practice, but the TPI support of what is a great contribution. What goes where level differentiation might not be the most focus, but level A & B should be a must to any company working seriously with testing.

Defect Management: Defect Management is an obvious area to glance at, since it is simple if it works or not (and is usually easy to improve quickly) by introducing a defect reporting tool.

Testware Management: Reusability has been a big work for a decade at least. I think that the basics are that testware should be treated like any other software and hardware and the problem is solved (it is not a side product!). The problem is that reuse is overestimated and very much dependent on what product that is developed and in what company (size), and assessment must fit appropriately.

Test Process Management: I wouldn’t spend so much energy on the management of the test process. As long as it is continually kept up to date and is appropriate it is good. Common sense. But here I suspect a lot of companies waste money on work that means rather little in the long run. It can be good occasionally to have a process overview, but I would be suspicious if a group spent any longer time in this area other than for research purposes. Of course the constant improvement is a part of any developing company, thus doing regular assessments can be a sanity check and booster for improving.

Evaluation: This area is as I view it bound with the former, but has a wider scope. Again, any formal or informal overview, self-check or assessment is an evaluation, and evaluations are good. I think creating levels and having “evaluation strategy” is creating work, is a rather formal approach compared to being open to new insights and methods of assessments and evaluation. I’m not sure that having an evaluation strategy makes the company in any better shape than a company that have not.

Low-level Testing: Finally this comes at the end, and should not be separated from other testing in my viewpoint. I think the purpose is that the TPI model clearly differs “testers” from developers/Programmers, thus there is a need to talk about high and low level testing – thus there are different people (groups, roles) involved. I think encapsulating the low-level testing suggestions in all the above, thus include this (sometimes) separated group in the assessment is a better approach. The area of course focus on “all the above” but particular for unit/component/module testing.

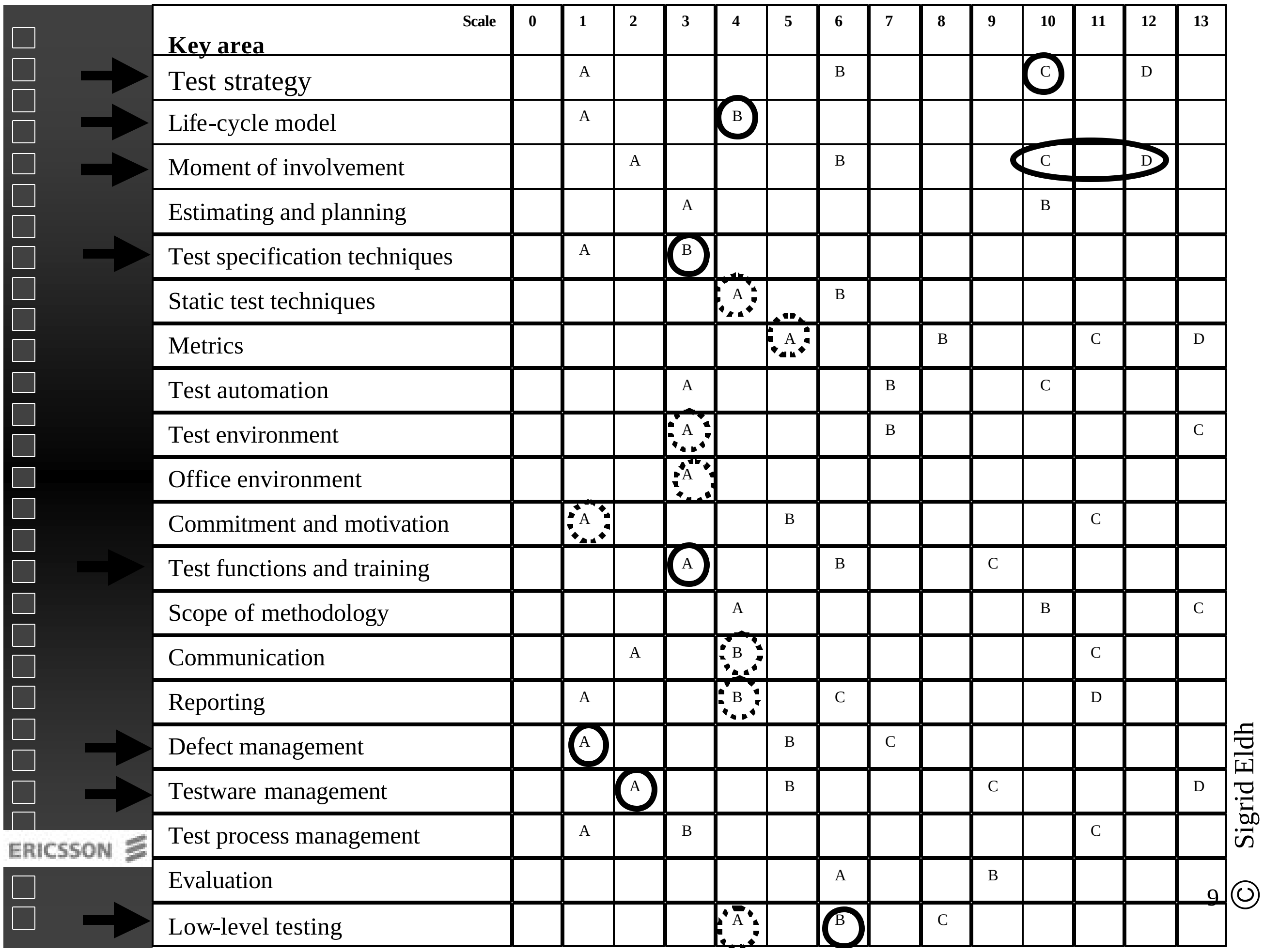
This was the “walkthrough” of the key areas in the TPI model above. Of course there are many other “models” to use, and you can yourself use the TPI as a “guide” and create your questions based and modified on the above.

Your role as test assessor

After the first kick-off meeting, and contact (if you are not doing it for your own company), you need to do the questionnaire yourself, with your first impressions. You will have a lot of question marks and unknown areas that you can use when you are interviewing to clear out. I assume that you have read yourself on some documentation. Your own “questionnaire” should of course be updated after the interviews.

Your first task and goal during kick-off and initial read-in of documentation is to understand what the organisations you are assessing really do. And you are at liberation from the start to ask questions about everything, and also ask about the documentation (control-check). One example could be “I saw in the documentation that you only have a preliminary version on the process, is it a new process in this project? What is it based on? Who was involved?” and so on with questions based on the documentation. Another example could be “ I see you are using the tools “such and such”, how did you choose them? Have people got training in them? Are they used? Are they considered valuable for you (time saving) or not?

You can of course ask a few selected persons after the interview is done to fill in your questionnaire (maybe you can be available to clarify the questions if there are questions on the questions.   If you are using TPI you will get support in giving value to “what is better” and “what is development”, but personally, I think it is much more valuable to look into the organisations will and together with your experience really improve what is obviously not working well. The assessment is a guide, a support of what should work and again, I put emphasis on the areas (and where you should be) with this drawing: The full circles are the TPI key area that you should definitely aim for, and the dashed circles are the areas that you should expect to be.



	Scale **Key area**	0	1	2	3	4	5	6	7	8	9	10	11	12	13
➡	Test strategy		A					B				C		D	
➡	Life-cycle model		A			B									
➡	Moment of involvement			A				B				C		D	
	Estimating and planning				A							B			
➡	Test specification techniques		A		B										
	Static test techniques					A		B							
	Metrics						A			B			C		D
	Test automation				A				B			C			
	Test environment				A				B						C
	Office environment				A										
	Commitment and motivation		A				B						C		
➡	Test functions and training				A			B			C				
	Scope of methodology					A						B			C
	Communication			A		B							C		
	Reporting		A			B		C					D		
➡	Defect management		A				B		C						
➡	Testware management			A			B				C				D
	Test process management		A		B								C		
	Evaluation							A			B				
➡	Low-level testing					A		B		C					

Figure 1. Selecting important areas and levels for your assessment with TPI.

What is the more difficult task as an assessor is that some customers are very familiar with the model, and will answer “correct” and that is why you need a mix of people (and control questions of how - which usually reveals any fuzzy truth). Another focus of yours is to understand what is NOT talked about - If you interview open-ended (as I have proposed is the best approach), you will also see what people focus on. Sometimes easy things (that we do well) are not talked about. I think it is as important to reveal and enforce the confidence in the good practices as well. Also there might be “avoidance” (aware or unaware) in peoples mind, where they are not focusing on all issues (a very natural behaviour!). The task for you is that you must keep a good overview and remind and ask about “other things”, thus you must keep the Model your are using in the back of your head at all times. In other words, where are they, where do they want to go (and why), and where could they go if the knew about an improvement there?

I think a really good trick, since this is about improving testing is to really do the obvious, i.e. ask about what improvements each person sees, and let them motivate why this is a good idea. Usually you will have a ton of improvements to use and implement and write about in your report, both small and large, but also try and find out why the person feel this improvement is important, and also why (and why not) they have not pursued it themselves. The pursuing bit might be “little” controversial for some persons, and remember that your role as an assessor is also to use the moment to motivate and make people aware about their inner strength to actually do better. This is of course a much easier task if you are from the outside, and can be rather problematic coming from the inside.

Asking Questions

Another approach, as the assessor in the interview is to talk about areas in a general fashion: “but what about xxx?” How do you do that? This statement should be targeted as relevant to your opinion of the current level in that area. Then you will get a good open approach about it. Your goal during the interviews is to understand more about what the people know and is aware about in a particular area. It might be that it does not even exist and that people never hear of either standards applicable or about a certain test technique. So it is a quest in itself to find out what knowledge there is about a certain area with people. Also deciding if it is general or just individual. I usually bear in mind that the people selected are of course representing the “general” view and take the liberty to generalise a bit from the statements, and especially if they are collaborated (second) by some other person.

There is to my mind a ladder of better that goes like this:

- Non-existent, ignored, avoided
- Awareness (of existence) but have not yet done anything
- Tried (but failed) to put practice in place (limited knowledge or not fitted corrected)
- Practice or area exists and is used. Basic knowledge in place, just by some projects and people

- Improved area, there are small things that have been adjusted, and improvements are made More than one project is using it
- There is quality in the practise or area, and other alternatives are evaluated, many projects and people are using it.
- The area or practice is optimised and used by all projects (and people)

To find out where you are is to rephrase this into questions; I call them “X-Questions” where x could be a key area, practice etc.

- What is the quality of X? (this shouldn’t be directly, but rephrased into many questions.
- Do people use X?
- What is lacking for X to become better, more useful etc?

So lets explain with an example: Thus, let X be the area of Test Strategy! The questions could then look like this:

- Do you have a Test strategy? Written down? Show me!
- Do you believe the Test strategy is useful?
- What levels of testing do you have? Do you think that is sufficient? (Does it include all levels of testing?)
- Are all within organisation familiar with the strategy (ex. Designers for Unit Test, project managers) (check with different persons/roles).
- What is lacking for the test strategy to become better?

You are looking for the signs of the following in your answers:

- Lack of management support?
- Lack of cooperation between organisations
- Lack of quality in the result?
- Lack of competence in testing?
- Areas of training (Particular persons?)
- Lack of coherence?
- Efficiency?
- Cost effectiveness vs. Over-administration?

In conclusion, the hardest problem for the assessor is to keep “all this” in mind when asking questions, but still be in contact with the interviewee and still not lead the interview or ask closed questions.

Important Findings

During the interviews you are to write down (or if you have help) the answers. Unfortunately taping interviews is an excellent tool, but people usually get uncomfortable with it, and it is also in reality hard to re-listen long interviews, and also, the security in destroying the tapes afterwards is also extra work, so basically, I just do my best to put down the notes I have. I really "trust" myself, and only write down things that are "odd" but never write down what I would expect to be answered.

Basically every straight answer is a "finding". Thus in strict large scale assessment (which I'm not advocating for here) there would be many assessors writing down "everything". Thus collaborative evidence is when two statements agree (have the same basic meaning), and non-collaborative evidence is when two findings says the opposite or are disagreeing. Thus if two people in the different roles or departments or even projects have a different opinion about the test strategies they are using, is a non-collaborative answer. The problem lies in if there are two collaborative and one non-collaborative answer in a model. I guess it basically depends but in general one rules this non-collaborative. It can be rather philosophical if the non-collaborative is in one project and the collaborative is in another, what is then true is that at least not all project fulfil the criteria.

Also, during the interview, if you follow my advice on line of questioning you will get a summary of improvements, which you can use in your summary report. This is exactly what an assessment means, that you will put together all info in the report. It is of course good to structure improvements based on area they are applicable in, but also a summary of improvements to be done could also be a valuable outcome of the report.

In the report you need to explain where "they are", thus what level, and then also what is missing in particular to reach next level. Here both your suggested improvements and the people's improvements, thus also the improvement that is needed to reach the next level is important. I usually take concrete examples from the interview (and taken care to hide the identity of the person) by stating important findings. This could for example be that several persons stated that the priorities were misused and had too much impact on steering what got tested and not. There is a need to improve the prioritisation process for the test cases, and also have a better adaptive test strategy around the test cases priorities.

One thing that keep coming back to me is that often in organisation many persons know very well what is important and what should be done but yet they don't do it: WHY? This is really something to ask about, what is the hindrance to get things done. It is more usual that these obstacles are the most interesting to focus on in an assessment to get things working better. Sometimes (quite often) it is personally related, and sometimes it is leadership or management related. I think in an open trustful environment these difficult matters should be able to open up for discussion, and if done properly amazing improvements occur. To summarise this I could state that the assessment role is really about to find out what ”stops them” from improvement.

Something about Assessment "Ethics"

As an assessor I cannot highlight enough the difference between actually improving testing versus the situation or context to test is in. Very often the true reasons are totally other reasons like organisational issues that has more to do with personal development, leadership and hidden organisational culture. Examples are ”blaming”, ”non-listening” ”management hostage”

”resistance to organisation...”, “imposed upon from management = non-trusted management!” etc The problem with seeing and identifying this in an assessment is that you must be from the ”outside” to get a grip and see this and the second advantages is that if you are from the outside you might also dare say something about it. Then, on the other hand, many organisations are not prepared or open enough to receive advice from outside in these matters, so I’m just warning on how thin line this is to walk. On the other hand, I think if we cannot get “the truth” from an assessor when we ask for it, then we should really reconsider our aims and improvement ideas from a company point of view. It is of course necessary that confidentiality agreements are signed, and also that that there is a mutual trust that there is no malice behind the assessment, but really good intentions to try and look for what improvements there are and why they are not done.

Assessment - Real Feedback

Another area of your feedback is to give feedback both on Technical issues but also clearly state when it is a judgemental feedback. This is what you see, understand and have an opinion about. My experience is that if it is a clear opinion, then it is easier for the company to digest some of the “bad news”. One can always say, but that was the assessor’s point of view. Which is not a bad thing, because the assessor might not have the full picture, since the assessment is a sample of reality, and it is possible the sample was not correct.

What could be an advantage and disadvantage with your assessment is your own experience with domain. You can more easily compare similarities and are of course more biased if the domain is the same, but remember that if the domain is different, you should be careful to judge thing in perspective. A new young business cannot expect to have the same approach and experience as a more mature company. Also the product or the domain to be tested differs a lot between different systems. Some you can easily get tool support for and others is all home made, and have no available off the shelf tools to buy. So what is good and bad should of course have a perspective if you comment on that level. Which I think is your job to do. Honest feedback is what the assessed want.

What can also surprise during an assessor are the very different views you can get from the same company. For example one person can state “Yes we do it” and the next person had no idea about it. So the question is, is this a real problem? Or is it just a lack of internal Knowledge and information that is missing for the persons you are speaking with?

In any case, you should clearly differ YOUR opinions and suggestions clearly from statements and suggestions from the organisation, in the report, so it is clear that what the organisation thinks of it and you assess and think of it. The most interesting part of value judgement can be who “owns” it. What is really “right”? Is the model deciding? Is it the assessor? Or is it the organisation in itself? I think that these matters should be discussed beforehand (in the kick-off meeting), so it is somewhat clear. Either you follow a model strictly or you are asked to do the assessment based on your experience.

Follow-up and going for the next step

When you have done your assessment and written the report, you might want to think about with the organisation (and discuss this beforehand) what will be done with this information.
If I have a chance I will personally ado a short “first impressions” summary in a seminar for the people and some managers involved in the assessment, to create an interest to read the report.

This could also be delayed to promote and hand out the report. In either of these cases, you should consider and answer the following questions:

- What will be done with this information?
- Why were you doing it in the first place?
- How is the organisation to proceed?
- Who will be the owner of the next step?

When this is clear, you are available to support the organisation, sometimes you are asked to do the improvement, but to my mind, that diminishes the organisations internal strength of moving ahead. It is a sort of responsibility. Also, I think many in the "improvement business" tend to forget the economical background to which improvements are made. It must be economical beneficial to become better, otherwise there will be hard to create a driving force behind it.

Another work for you is to return afterwards, after a few months, and see how the report was received, and also how the next step is going. It is often very nice to hear the positive news of improvements that took place on your recommendation. Also, you might want to discuss after 6-9 month if the company are looking for a Re-assessment? Or if they feel they still are working after your improvement suggestions.

Using assessment to reach your testing goals

Obviously, short term testing goals, like "I will finish my 142 test cases on time" are maybe not what we are talking about. I'm talking about the vision you have for your testing. Basically, you can invent the goals, as you want them. In any case the key to achieve goals is that they must be committed. Thus, if a person "own" it as a personal responsibility – they will do it, especially if someone cares about them being done, for instance, the organisation cares (read manager!). Then there will be a point in doing them. An even stronger motivation than that someone else cares, is if you directly see that your work is getting better, easier, faster, more effective or you really gain something doing it, then you will feel very motivated, but even so, that determination is rare and hard, so try and get support from others. To my point of view, the follow-up of the goals is a second key. You as the leader of the improvement need to check how things are going, and if the person needs support etc. Showing interest goes far. Also, if it is a goal that the persons themselves will suggest, own and "decide", then the improvements has a much more likelihood of succeeding. A third important key is that you must make the results visible. People forget very quickly when they have learned new skills how it was before, and thus they require more. Remembering all that has been done already, and that everyone sees that both encourage the test group and others around. It will clearly show that something has happened, and that makes people proud, and wanting to continue.

I have also learned the practical way that many small goals are better and easier to achieve than too big goals, the reason is of course that it is human. So big goals can be achieved, but make them into many small and then it want feel so difficult. In that way you will support your ultimate goal by constantly improve.

It is sometimes easy to give up, and I think that what is often missing is the basic question for the testers and the test team: “Is this a goal that will make testing work better?” If you can clearly state yes to that question, then time is NOT an issue, because a good goal will save time and money soon for you, and then you regain your strength again. Setbacks can happened to any test team (or a person for that matter), but it can always be turned into a learning experience.

Special attention areas

In the “ad” for this paper it was promised to deliver areas that needs special attention. Thus I have collected the areas I think needs a bit extra. Having the knowledge of TPI model the figure 1 (above in paper), the arrows on the side really points to these areas. But in any case I will explain them more in detail below.

First you need to focus that you get a Process that fits your organisation, complexity of the code (conceptually), the levels and your test team. I do think testing should be done both internally by developers and with an independent team. The process must be written down, understood and under constant improvement. I would not stress the importance of having “one” process for a large company, because what is developed in a large company can differ greatly, and thus the process must be adaptable for that. The process must handle all levels and all stages

- Planning and Preparation phases: Strategy, Analysis and Design
- Execution phases: Unit, Integration and System
- After Release: Acceptance, Beta and Maintenance
- Conclusion phase

The moment of involvement should of course be at the start of the project, and nothing less should be acceptable, thus this really shows the ”status and respect for test”, are testing something that is considered during design, and that the system is really designed for test.

To have a clear and useful test documents and use tools when appropriate, and also that the handling of the documents and tools are treated like software (configuration managed, stored properly, secure etc) that is real testware management, and something that just should be done.

Having a proper defect management, and a good test configuration management will give you control and order, which is necessary. Some basic measurements should be in place, thus, some planning figures, progress data (often measured in test cases), defects (with priority), and code coverage... are some to have a look into. Code coverage should be carefully considered is my advice, since it is easy to fulfil some code coverage as a testing goal, but not necessarily does the testing become much better. Always test where there are bugs!

Then you need do know what test techniques you can use and where (and when)? You need to know how you create and select test cases depending on your domain and your goal. You also need to look into automation (in what level are the tool to be applied, to what, how much, should you use integrated tools, or separated tools, and all testware issues and management about the tools, such as getting proper education, understanding maintainability and support of the tool.)

Having reviews in place, good communications etc I consider basic for a good organisation, but not particularly a test issue (well, maybe reviews could fit in!).

Powerful feedback – and what to do with it

The assessment report has now given you where you are and what your goals for improvement could be. Now you need to make something happen with this powerful feedback of the assessment. First you need to get all persons in the test team to take their part of responsibility!

A good suggestion is to start going through the result with everyone. Then have a walkthrough of your test process (and pinpoint improvements) that you know and also see.
Clarify which areas are important for you, for instance by voting on importance of improvements among the team. Also, I good exercise is to mark each improvement by what is easy to do, and cheap to do (and let everyone in the team) agree on most of the work. Then select improvement from that list. I think it is important that management for instance show their support by economically funding at least one of the more “expensive” improvements. For instance, letting the group get a good education in the test tool they are using, or certifying the testers to a scheme, or likewise. Write what you do down in a small document and get going.

Benifits and Side Benifits

There are many benefits with doing an assessment – most of all, getting started with improvement that saves money for your company, and make you successful.
An interesting side-effect is what is called the “Hawthorne effect”, which means that the result can improve by just investigating it, thus paying attention to the area. By focusing on testing through an assessment, this will have a positive effect in itself. Persons will reflect on improvements and why they are doing certain tasks and not. Also the focus will bring interest to the subject.

To my mind, what qualifies an improvement is that it makes things better. What is then better? Cheaper, faster, easier, more fun etc, thus for me any improvement is worth doing. Why do boring repetitive tasks when there are more intelligent, fun things to do? Testing is all about learning more and learning about the system in itself (the domain) the testing, and how to be more effective during the test. But to really learn fast, a good outside support (TPI, Assessor, books, courses, mentor systems, conferences…) is necessary. Why repeat mistakes? Avoid the most common pitfalls and improve. The best benefit is that a real good process of “Take a good hard look at you” will be an eye-opener than makes all small pieces come to place. It is the details that you miss, and the whole you usually do not see. Also, things change, and there might be a development in all directions. An assessment will definitely keep you on the improvement path, just by the fact that you are conscious about what is a better way to do things than other. You will know where you are and then you can decide where to go, thus the assessment will show you the way. Do the assessment and the result will be your success.

Summary

Will you be successful with your assessment? Well, the model and the way you conduct and follow up the assessment will definitely be the two important drivers. So if you choose TPI model, is it worth the effort? Yes, TPI has a great overview and you need support from "outside" to get things in order in your assessment. I remind you to never loose your common sense. What is really important? Maybe there are another model that suits you better. Having a "base" is good, and then use the advice in this paper, and you will get a successful assessment.

Many also ask me if they understand TPI (or any model), does the work really get better, i.e. saving money, being more efficient etc? I think that understanding testing is all its aspects, and in its different contexts, domain and organisation is what really improves the way we handle testing, not necessarily a particular model. The goal of TPI is somewhat questionable in some areas, and I think that we need to understand what is really good for the system and environment we work in. Especially in large companies I have lately started to question why organisational wide processes and approaches really is better. If systems are different, they should be handled different. Improvements are if they come from the internal wish of the people working with it always to the better, but some imposed improvements from outside are not always to the better. The problem is to see the difference, and judge when one or the other is right in the situation.

I usually state that every new bug I found before it reaches the customer, saves money, but that is very difficult to prove, thus many parts of the computer systems that are out are never used or looked at.

I would like to end this paper with a question to you, the reader. What is your goal? Maybe the total perspective in which you work also needs a glance, maybe you need to look at where are "the real troubles".

I would also like to thank you for your interest in reading this paper, and I'm sure that my cultural and language differences (and lack of knowledge) might have made my statements unclear at times, but do not hesitate to send me an email with any questions you might have, and I will do my best to answer them. Any information on how you succeed would be a wonderful feedback (both good and bad) for me. I'm sure you will be successful doing assessments in the future and I wish you the best of luck!