The Personal Test Maturity Matrix

Stuart Reid
Cranfield University
Dept. of Information Systems
Shrivenham, Wilts SN6 8LA, UK
+44 (0)1793 785490
s.c.reid@cranfield.ac.uk

ABSTRACT

This paper introduces a new framework called the Personal Test Maturity Matrix (PTMM), the purpose of which is to provide career guidance for software test practitioners. The PTMM is a new idea, and, as such, has not yet been tested in practice – the purpose of this paper is to gain feedback on the idea.

The main concept behind the PTMM is that testers, especially those new to the discipline, could benefit from using a framework that describes the different roles that software testers perform and the skills required to do this. The PTMM thus comprises roles and skills, which are categorized into four areas: test skills, IT skills, soft skills and domain knowledge. The idea is that an individual tester could identify their own specific skillspace, which describes their capabilities in these four areas, and then use the PTMM to provide them with suggestions on which skills they might wish to acquire next.

When considering the jobs within software testing, the use of motivating potential score (MPS) as a measure of job satisfaction was investigated and applied to a number of example software testing roles. The initial results suggest that different software testing roles vary greatly in providing motivation and satisfaction to the tester. A number of the roles scored such low scores for MPS that it appears that they would be good candidates to be combined to create a satisfying job rather than considered as a job in their own right.

Keywords

Personal Test Maturity Matrix (PTMM), career guidance, test roles, test skills, Motivating Potential Score (MPS).

1. INTRODUCTION

This paper introduces some new ideas aimed at career improvement for individual testers. It proposes a new framework, the Personal Test Maturity Matrix (PTMM), which could be used by testers to help decide which skills they set out to acquire next on their career path as a professional tester.

The PTMM is primarily intended for practicing software testers, and is expected to be of most use to new entrants to the discipline up to mid-level testers. It should also be useful to their managers as it highlights some of the difficulties (and opportunities) when designing a motivating and satisfying job for a tester.

The framework presented here is by no means the finished article and has not yet been tested in practice. Rather, it is a 'straw man' with the objective of both generating discussion of its disadvantages and provoking the generation of new and better proposals. The PTMM was created jointly by Stuart Reid and Julian Harty, both of whom are happy to receive feedback on it.

Despite its acronym of PTMM, there is no intention to imply a 'special' link with the Test Maturity Model [3]. There is, however, a connection between the two as both tackle the topic of improvement within the testing industry.

2. IMPROVEMENT AND QUALITY

Dr. W. E. Deming stated that "Quality is the continuous improvement of all processes". Process improvement has become increasingly important over the last ten to fifteen years, with many organisations trying to reduce their production costs by improving the efficiency of their development processes. Software testing, despite being initially sidelined to a minor role, can now boast a number of approaches that have been specifically developed for this purpose, such as TMM [3], TPI [10] and TMMi [13]. These test process improvement approaches first became available in the mid-1990s and are aimed at supplementing the more development-oriented approaches such as CMMI [5], ISO 15504 (SPICE) [9], and Bootstrap [2].

But, how have these initiatives helped the individual tester? It is generally recognised that when applying these top-down approaches to 'process improvement' we should also consider the supporting areas of the environment, tools, and management and people skills. To date, however, test process improvement in the people skills area has tended to limit itself to recommending the optimal mix of personality types in a test team.

The PTMM introduces the idea that individual testers can also 'improve' themselves by following a systematic (yet personalized) approach to their career development. Obviously many testers are already adept at managing their careers, but it appears there are also a large number who are not realizing their full potential. For these testers, awareness of the PTMM may simply provide the spur to re-evaluate their position, while some may find that the framework of skills raises questions pertinent to their future career path. As an aid to test management, the PTMM may help identify where testers are not being used to their full capability, opening the opportunity for better return on investment and, hopefully, happier, more motivated testers. The PTMM should be complementary to the currently available top-down test process improvement models.

2.1 Process Improvement and Personal Improvement

A high level overview of process improvement is shown in Figure 1. Initially the process improvement goals are decided and an assessment of the current process is performed against the goals. This process is analysed to identify what improvements should be made and these changes are subsequently implemented. The affects of the changes are measured (normally after some time) to determine the success, or not, of the attempted improvements. The information derived from this activity will help in subsequent iterations by identifying the next goals to be achieved and the process continues.

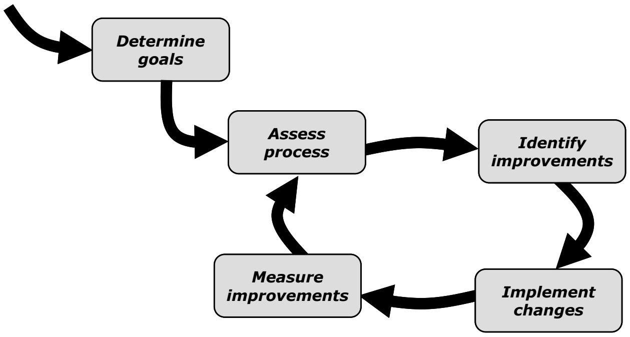

Figure 1. Process improvement.

The process shown in Figure 1 can also be applied to personal development. Personal goals may vary in timescale and scope. They may be short (current project), medium (next few projects) and long term (career) and concerned with the tester's position within their current organisation and the industry as a whole. Ideas for improvements and changes can be found in a number of places. The closest and easiest source of ideas is the current workplace, which is often overlooked. Otherwise conferences, training courses (both taught and self-paced), the internet, books and magazines are potential sources. For personal improvement the measurement of success (or not) will be on an ongoing basis, but will also benefit from periodic planned sessions of self-assessment.

3. TESTER SKILLSPACE

Each tester has a unique set of skills that, grouped together, can be considered as their individual 'tester skillspace'. It is this set of skills that defines the current potential capability of a tester – potential because testers tend to only use some of their skills at any given time or in any particular job. This is because most tasks require only a subset of their skills, and in many situations testers are not given the flexibility to use their full range of skills. The aim of the PTMM is to guide the tester in how best to expand their personal skillspace.

3.1 Characterization of the Tester Skillspace

The skills required by a tester have been categorized in a number of ways by different authors. For instance AT&T identifies two core competency categories: general engineering skills and tester-specific skills [14]. Within AT&T, knowledge of networks and telecommunications is clearly important and is included in the general engineering skills category. This domain knowledge is particular to AT&T's industry area, but a more generic categorization is useful for a more general scheme and would also explicitly cover knowledge of the application domain. Rex Black's scheme has three categories, including a specific category of 'Domain Knowledge', alongside those of 'Technical Expertise' and 'Testing Skills' [1]. Many would agree that it is also possible to discriminate between testers using a fourth category – that of their soft skills, which would include communication, interpersonal skills, conflict management and negotiation, discipline, work ethic, etc. Isabel Evans [7] covers this area in some depth. Jos van Rooyen includes 'social skills' in his 'knowledge quadrant' of required knowledge for testers alongside 'Business Knowledge', 'ICT Knowledge' and, of course, 'Test Knowledge' [12].

We chose a categorization that matches the knowledge quadrant categorization most closely, but with the slightly re-named skill areas of domain knowledge, soft skills, IT skills and test skills. To perform any testing job it is possible that you will need skills in each of the four areas, but each role you take on will require a different balance of these skills. For instance, if you perform the role of reviewing requirements then it has been reported that good domain knowledge significantly improves the review's effectiveness [4]. You need to be able to understand the requirements specification, which could be presented in a notation requiring knowledge of a particular development methodology. You will need reviewing skills from the testing area and soft skills in order to communicate the issues to the relevant stakeholders. A tester carrying out test automation will use a different mix of skills while they perform that role.

3.2 Required Skills

So, what skills do professional testers need? Ideally, they would 'carry' a set of transferable skills from job to job and industry to industry. Test skills will be used in all testing jobs, and a level of IT skills that provides an understanding of the underlying IT technology will be of benefit in many testing jobs. More system-specific and in-depth IT skills will be absolutely necessary in some roles, such as performance testing, and a significant number of long-term professional testers find that programming skills are something they cannot do without. It should be remembered, however, that specialised IT skills are probably the least enduring of the skills in the skillspace. Domain skills are, by their nature, less transferable, but can, nevertheless, be highly valuable. For instance, intimate knowledge of the banking domain will make you highly popular when a banking system needs testing, but is of little value when the next job is testing an avionics system. Soft skills are the most transferable, being of value not only if you decide to move industry area, but also if you decide to leave the testing industry altogether.

In the same way that testers can carry useful skills into other industries, the same can be said of new entrants into the software testing field. Former developers and users bring with them their

IT skills and domain knowledge respectively. A number of testers make the opposite journey, and it should be an accepted career path for software professionals to move back and forth between development and testing as our experience shows that some of the most effective testers are those who have 'feet in both camps'.

Each individual should be able to identify their strengths (and weaknesses) based on the skillspace categories – producing an individual skillspace 'signature' showing their current balance of capabilities. It should also be possible to look at the four categories and decide both what their potential capability is for each category (i.e. the best their skillspace signature could possibly be) and their ideal balance of capabilities that they wish to achieve in the future. It should then be possible to start planning their route to their ideal skillspace signature. To support this planning a tester needs to know what is covered by each of the skill areas, one of which, test skills, is described in more detail in section 4.

3.3 Qualifications

My experience is that at present most graduates from IT degrees would have difficulty recognising some of the test skills, let alone practicing them. Industry needs to make a major effort to encourage colleges and universities to cover the topic in a serious manner as at present most graduates who become testers enter the industry with very few test skills. There is also currently no single professional qualification that is recognised industry-wide, and there is a continuing debate as to whether the field is mature enough to support such schemes.

The most popular software testing qualifications (in terms of reported numbers) are those provided by ISEB/ISTQB at Foundation level, which were originally UK-based but are now available far more widely. These qualifications are generic and so provide no domain skills as they are designed to be suitable for anyone with an interest in testing from any industry area. In our opinion holding these qualifications does not demonstrate the core skills required by a professional tester as according to their syllabi they include practically no coverage of soft or IT skills and the coverage of test skills is at an introductory level with no practical element to the training. These qualifications should however provide an understanding of the basic concepts and, in an ideal world, promote a common terminology that will aid communication. As such the author sees these or similar qualifications as a necessary starting point for a professional tester, and also as a recommended foundation for a software developer, who should have a basic grounding in software testing in the same way that a software tester should have a basic grounding in IT development.

The more advanced versions of these qualifications, which are normally labelled 'practitioner', are supposed to include a minimum coverage of practical skills. Many argue, however, that this coverage of practical skills is diluted by exercises aimed at improving examination performance rather than providing experience of good industry practice.

4. TEST SKILLS

In our work on PTMM, we have initially concentrated on identifying those skills that make up the test skills part of the tester skillspace. We intend to cover the IT and soft skills categories at a later stage, leaving the domain knowledge to domain specialists.

The test skills form part of the framework known as the Personal Test Maturity Matrix (PTMM), the structure of which is shown in Figure 2.

4.1 Skills in context - Roles & Jobs

For the purpose of discussion of the PTMM we consider a testing job to require a tester to perform one or more test roles (see Table 1 for example test roles). For instance, a tester may be reviewing one afternoon, and then performing black box testing the next morning (so taking on the roles of reviewer and black box tester, respectively). From a test management perspective, the ideal tester will be able to fill a wide number of roles. A tester able to fill a larger number of different roles will generally find that they are more in demand. Although there are clearly situations where specialists are highly sought after, care must be taken not to specialize in an area that eventually becomes redundant without having a contingency plan.

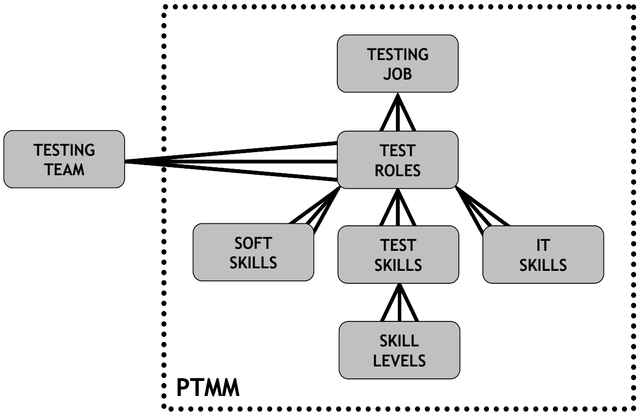

Figure 2. PTMM structure.

4.2 Skills and Roles

As shown in Figure 2, a number of skills are needed to perform a single role within the testing team. See Table 1 for *example* test skills for each test role and Table 3 for *example* descriptions of each test skill. To successfully perform in these roles the tester will often have to apply a number of different skills on a day-by-day basis, although sometimes the same skills will be useful in different test roles. An example of different skills being used to perform a single role might be that a tester has to understand code to perform white box testing and also has to be able to specify the test cases in a form suitable for test execution, possibly by someone else.

A top-down approach was use d to identify the example test skills. First, a range of example test roles were selected and then the test skills required to fulfill the roles were identified. One of the criteria used to 'break-up' roles into skills was to identify commonality between test roles to find skills that could be applied to more than one role. Commonality of skills between roles is useful in the PTMM as these links can often provide insight into how a tester's current skills can be used to help them move more easily into new areas.

Table 1: Example test roles to test skills mapping.

Exploratory Testing	Model-Based Testing	Black Box Test Design	White Box Test Design	Test Execution	Test Reporting	Automated Test Support	Test Env't Support	Reviewing
•Ability to create and/or reason about models (L) •Knowledge of test case design/test coverage (M) •Ability to select test cases •Ability to identify and record anomalies •Ability to present test results	•Ability to create and/or reason about models (H) •Knowledge of test case design/test coverage (H) •Ability to identify and record anomalies •Ability to present test results	•Ability to create and/or reason about models (M) •Knowledge of test case design/test coverage (H) •Ability to select test cases •Ability to specify test cases	•Ability to create and/or reason about models (M) •Knowledge of test case design/test coverage (H) •Coding ability (L) •Ability to select test cases •Ability to specify test cases	•Ability to follow test script	•Ability to identify and record anomalies •Knowledge of test case design/test coverage (L) •Ability to present test results	•Ability to set up test env't •Coding ability (H) •Knowledge of test automation software	•Ability to set up test env't	•Ability to check conformance with specifications •Ability to check adherence to standards

Key

Each column corresponds to a test role.

The test role's name is shown at the top of the column, with the test skills required to perform the test role listed below it.

Where test skills are followed by a (H), (M) or (L) this denotes a relatively high, medium or low level of skill is required.

4.3 Skill Levels

Generally the test skills in the PTMM are not simply acquired as a whole all at once. In reality skills are honed and testers become better both at applying skills as they gain practice and, where the skill comprises a number of distinct areas, as they acquire experience in more of the areas. For this reason a number of skill levels can be defined for a particular test skill. An example might be 'coding ability'. As we have already seen, this skill is required as part of the white box testing role, but typically only at a level where the tester has to understand the code well enough to follow control flow through it. Coding ability is also a skill needed to perform the automated test support role, which may require the tester to create bespoke testing tools, an application of the same general skill, but requiring a far higher level of capability. As an example, a possible range of skill levels within the 'coding ability' test skill are shown in Table 2.

4.4 The 'Next' Test Skill

The main purpose of the PTMM is to provide guidance to individual testers on what skill(s) they might want to acquire next. The PTMM is intended to work on two levels; at a high level it shows the skills needed to perform a test role, while at a more detailed level it shows the skill levels that must be progressed through in order to fully master a skill. It should be noted, however, that it is not always necessary to fully master a skill (i.e. achieved all the skill levels) before it is possible to use that skill in support of a testing role. For instance, to return to the white box testing role as an example, we do not need to progress beyond skill level 1 (shown in Table 2) to be able to perform white box testing (for code written in a 3GL).

The skill levels give an indication of a typical order in which a skill can be gradually extended from a basic knowledge to a more advanced level. The PTMM also provides guidelines on the level of capability required to acquire each skill. This creates a second possible order for the acquisition of knowledge by following the range of necessary skills to perform a test role (normally from the lowest capability level to the highest).

Table 2: Example skills for 'coding ability'.

Skill Level	Skill
1	Able to understand source code written in 3GLs.
2	Understanding of automated test scripts.
3	Able to generalize a recorded test script into a reusable script.
4	Able to write small utilities in languages good at handling expressions, such as Perl, Tcl, Python, etc.
5	Ability to write automated test scripts to handle special situations.
6	Ability to design and code substantial (test automation) applications.

Table 3: Descriptions of example PTMM test skills.

Test Skill (in alphabetic order)	**Description**
Ability to check adherence to standards	Be able to identify when the product under review does not comply with the relevant standards and guidelines.
Ability to check conformance with specifications	Be able to identify when the product under review does not conform to the source specification. For instance, checking whether the design specification is consistent with the corresponding requirements specification.
Ability to create and/or reason about models	Be able to create and/or reason about models of the application under test. These models may reflect different aspects of the application such as its inputs, outputs, control flows, data structures, and behaviour over time, etc. The models may use formal standard notations, such as UML, or any other suitable notations. Mental and informal models as used by Exploratory Testing (ET) (L). Models used by test designers, such as control flow graphs, syntax graphs (M). Formal models that include oracles as used by Model-Based Testing (MBT) (H).
Ability to follow test script	Be able to execute a well-specified test script, assuming that the application under test has already been set up on the test environment.
Ability to identify and record anomalies	Be able to compare actual and expected results to identify anomalies and record and classify the anomaly in sufficient detail for subsequent action.
Ability to present test results	Be able to present the results of the testing (tests passed/failed, severity/scope of faults found, test coverage achieved, etc.) both verbally and in writing.
Ability to select test cases	Be able to select suitable test cases to exercise a particular attribute of the application under test. Requires the tester to have some knowledge of the application under test and knowledge of test case design techniques.
Ability to set up test env't	Be able to set up the test environment to a level that enables a test executor (or MBT or ET) to run their tests. The skills required will be highly dependent on the application under test (e.g. sometimes it may be necessary to set up supporting databases and networks, etc.).
Ability to specify test cases	Be able to specify test cases in sufficient detail so that a test executor can execute them (normally when test cases are part of a test script).
Coding ability	Concerned with the understanding and writing of code. Being able to understand code such as Java, C++, Ada, VB, and Fortran (L). Being able to work with automated test scripts and write code such as Java, C++, Ada, VB, Fortran, Perl, Tcl, Python etc. (H).
Knowledge of test automation software	Concerned with testing tools, ranging from the use of capture/playback tools, to the selection of suitable tools, and to the customization of tools to extend them beyond their typical use.
Knowledge of test case design/test coverage	Know which test case design techniques and corresponding test coverage criteria are most suitable for different situations. The choice may be based purely on the application under test, but may also be based on the available application model and test environment (such as is the case with MBT) or may also be based on results from previous tests and the test environment (such as is the case with ET). Level of knowledge for documentation purposes (L). Level of knowledge for ET (M). Level of knowledge for black and white box test case designers (H).

4.5 PTMM Capability Levels

Having identified the test skills required to perform a given role (example are shown in Table 1), the next step was to assign a rough estimate of capability to each of the test skills. This corresponds to the required aptitude for acquiring and performing the skill along with the amount of supporting knowledge required (and so it is often also related to the amount of training needed).

As an example, seven levels of capability were assigned to the skills and these can be seen in Table 4 with lowest capability level skills (level 1) at the left and highest (level 7) at the right.

A number of conclusions can be drawn from the structure of this table. First, some roles require different levels of capability than others, with some roles requiring a relatively low level. This means that a typical new entrant to the field of software testing can quickly become competent in such roles. A number of roles include skills with relatively high capability levels. This suggests that only more capable testers would be able to perform these roles successfully and that they would probably also require far more training for these roles.

As there will inevitably be fewer testers at the highest capability levels then the rules of supply and demand should mean that those roles including skills at the higher capability levels will command better rewards.

4.6 Commonality of Skills

Table 4 is set out so that each column contains only one test skill. Thus by reading up and down the columns it is possible to see where there is commonality of test skills across test roles. For instance, the level 2 capability skill of 'ability to present test results' in the test reporting role is also found in the both the model-based testing and exploratory testing roles, as is the test skill of 'ability to identify and record anomalies'. Where there is such commonality then it is possible to identify paths from one test role to another.

Table 4: Example PTMM capability levels.

		2 Ability to present test results				2 Ability to identify and record anomalies	3 Ability to create and reason about models (L)		4 Ability to select test cases	4 Knowledge of test case design/test coverage (M)				Exploratory Testing		
		2 Ability to present test results				2 Ability to identify and record anomalies					Model-Based Testing				6 Knowledge of test case design/test coverage (H)	7 Ability to create and/or reason about models (H)
	White Box Test Design				2 Ability to specify test cases			3 Coding ability (L)	4 Ability to select test cases		5 Ability to create and/or reason about models (M)				6 Knowledge of test case design/test coverage (H)	
	Black Box Test Design				2 Ability to specify test cases				4 Ability to select test cases		5 Ability to create and reason about models (M)				6 Knowledge of test case design/test coverage (H)	
						Automated Test Support						5 Ability to set up test env't	6 Knowledge of test automation software	6 Coding ability (H)		
	2 Ability to check adherence to standards			2 Ability to check conformance with specifications		Reviewing										
		2 Ability to present test results	2 Knowledge of test case design/test coverage (L)			2 Ability to identify and record anomalies	Test Reporting									
						Test Env't Support						5 Ability to set up test env't				
1 Ability to follow test script	Test Execution															

Key

Columns show test skills in increasing levels (1 to 7) of capability.

Rows correspond to test roles (as indicated).

The grey shaded area in a row shows the range of capability levels required to perform the test role.

The number of skills in a row (test role) corresponds to the 'Skill Variety' score used for that role in the calculation of MPS.

4.7 Core Test Skills

Given the format of Table 4 it is tempting to identify the set of skills that could be considered the core skills of a professional tester. As this paper is intended to elicit discussion the author suggests that the core set of test skills (not including soft, IT and domain-specific skills) could be up to and including capability level 3. This is a gross over-generalization, and there are also some roles that appear to require few if any of the lower level test skills; this is because the table does not show implicit dependencies between skills that are necessary to perform most of the higher level roles. For instance, not shown for the test environment support role is the need to understand the tasks to be performed on the environment in order to know how to set it up correctly.

The idea is that this set of 'foundation' skills would help create a 'well-rounded' professional who was able to communicate on general testing matters with their peers, understand how their work affected others and be adaptable to changes in other parts of the project.

4.8 Acquiring Skills

Although it is not currently part of the PTMM, the intention is that when the skill definitions are stable, we shall identify guidelines on ways to acquire the different skills.

5. MATCHING JOBS AND PEOPLE

The PTMM as shown in Table 4 describes a number of testing roles where the range of skills needed to perform a role varies dramatically. For example, the test execution role requires a single skill at the lowest level of capability, while others require up to five different skills over a wide range of levels. It thus appears that for many people the test execution role alone is not going to keep them happy in their job for long as it will be monotonous and will not challenge them intellectually. This raises the question of who should perform these less demanding roles.

5.1 Motivating Potential Score

The Job Characteristics Model [8] has been around for over thirty years and is a well-respected approach to the design/re-design of jobs to achieve both motivation and satisfaction for the employee and higher retention and productivity rates for the employer. The model is based on the idea that five job attributes can be used to provide a measure of how satisfying a job is. Scores are recorded for each of the five attributes and combined to produce what is known as the motivating potential score (MPS) for the job.

The five attributes, scored in the range 1 (low) to 7 (high) are:

Skill Variety (V)	-	the range of different skills needed;
Task Identity (I)	-	the degree of completing a whole job;
Task Significance (S)	-	the importance of the job;
Autonomy (A)	-	the level of control of their own time;
Feedback (F)	-	the degree of supervisory and results-based feedback on performance.

MPS can then be calculated using the following formula:

$$\text{MPS} = \frac{(V+I+S)}{3} * A * F$$

It can be seen from the above that no explicit account is taken of the level of capability required to perform the job.

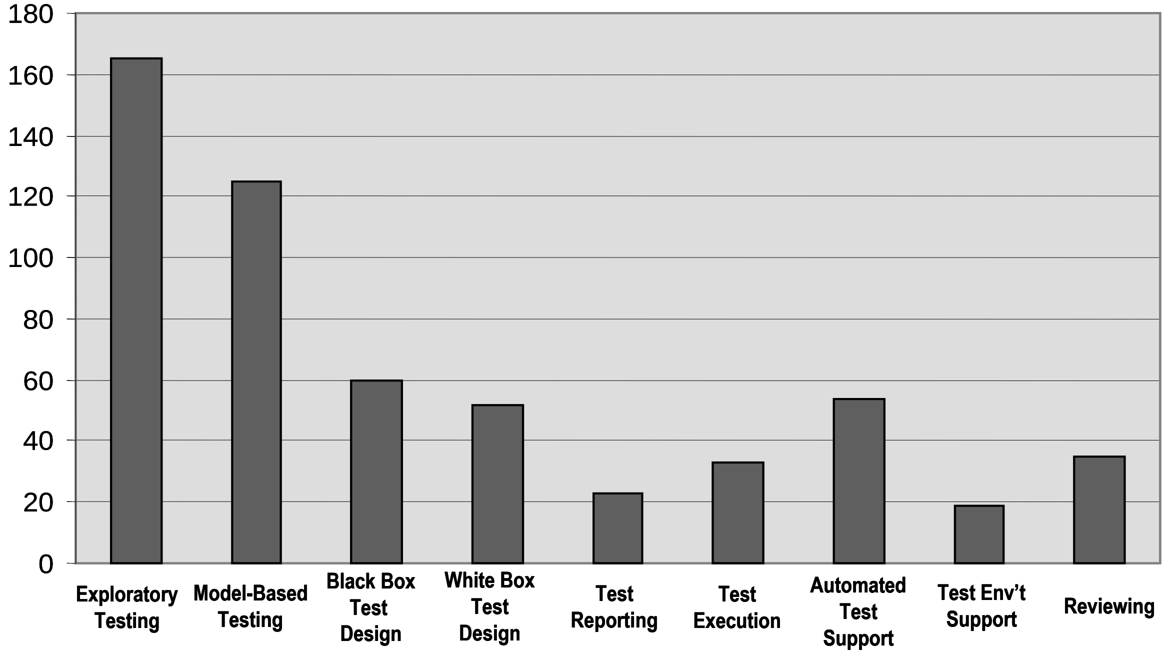

Figure 3. MPS for test roles.

5.2 Test Roles & MPS

Motivating potential scores for each of the example test roles in the PTMM have been calculated and are shown in Figure 3. It should be noted that the scores assigned to each of the five attributes are necessarily subjective and so only provide a crude measure of MPS that would be expected to change from one individual to another. For instance, if we consider the reviewing role, then the score assigned for skill variety may change based on the range of document types reviewed as these may well require different IT skills. The task significance may also vary depending on the importance of what is being reviewed. Ideally the scores would be generated by surveying a large number of practitioners and taking the average score.

As can be seen, unsurprisingly, many of the test roles have low values for MPS. So, does that mean that these roles are not high in job satisfaction? The MIP Report, a motivational survey of IT staff [11], states: "How could anyone believe that a job below MPS=60 could keep the simplest human being occupied?"

5.3 Test Jobs & MPS

Given the calculated values for MPS, it seems clear that most of the example testing roles are not suitable as actual jobs in their own right, although the relatively high motivating potential scores for both exploratory testing and model-based testing suggest that these two roles are suitable as stand-alone jobs. The MIP Report shows software testing as having an average MPS of below 120 (lower than the MPS of developers, maintainers, infrastructure staff and quality practitioners, which appear to average about 130). A separate study of analyst programmers came up with scores averaging 140 [6].

In practice many testers in industry are known as test analysts and their job generally comprises the three roles of 'black box test design', 'test execution' and 'test reporting'. When combined, the MPS of the test analyst job is calculated as 156, which suggests a motivating job with high job satisfaction.

5.4 Lessons Learned from MPS

Apparently many of the test roles (those with MPS less than 100) may not be suitable as jobs in their own right and any tester who finds themselves in this situation should consider their position. From the manager's perspective, then only unusual circumstances would appear to justify using staff in this manner. Exploratory testing scored the highest, which was not surprising given what proponents of this approach say. It should be remembered, however, that the figures used here are subjective, and further research needs to be performed to include feedback from a wider population of practitioners.

Those testers who feel lacking in motivation and job satisfaction could consider their particular circumstances in terms of the five attributes that contribute to the MPS. For instance, by acquiring new test skills they may be able to take responsibility for a larger variety of activities and thereby increase their MPS. Perhaps unconsciously, this may be the reason why some testers like to write their own test utilities. The ability to perform more tasks may also increase a tester's MPS as it increases the likelihood of job completion. Of course, opportunities for testers to expand their responsibilities are also highly dependent on the organisation in which they work.

6. FURTHER WORK

The current version of the PTMM is an incomplete first draft. We (Stuart Reid and Julian Harty) welcome any feedback on the framework as it is presented here as we recognise that creating something useful will involve a process of continual refinement. We also welcome any offers of help in populating those parts of the PTMM that cover the other skill areas (soft skills and IT skills).

7. CONCLUSIONS

In a competitive world, what is best for the project or the organisation does not always coincide with what is best for the individual tester. The PTMM is suggested as one way of providing guidance and ideas that are primarily aimed at improving the individual tester.

Each tester should recognise that their capability as a practicing tester is dependent not solely on their test skills but on a wider skillspace that also includes IT skills, soft skills and domain knowledge. Each tester will have a different balance of these skills – their own tester skillspace. The secret is to achieve the best balance based on your capabilities, interests and motivation - and the demands of the industry.

The PTMM is not advocating a single 'best practice' in career advancement. As we have already said, each tester brings with them their own unique set of skills, and each tester's career will be different. The PTMM is aimed at providing ideas for the next step in a tester's career - it is not saying that one path is better than another. If the tester is better informed of their choices they stand a better chance of making the right decision.

The PTMM will have succeeded if it can spur testers to seriously consider their next step in testing. It will also have succeeded if it provides testers with a new way of looking at their career that they had not previously considered.

The calculation of motivating potential score (MPS) for test roles confirmed what many testers already know – many roles are not in themselves enough for a satisfying job; a message that test managers should also be aware of. The MPS provides a useful way of evaluating your job and suggesting ways in which it can be improved, given the right environment. Interestingly the MPS for exploratory testing also confirmed why it has so many ardent supporters among testers.

Finally, to reiterate the point made in the introduction, the framework presented here is by no means the finished article and has not yet been tested in practice. Rather, it is a 'straw man' with the objective of both generating discussion of its disadvantages and provoking the generation of new and better proposals.

8. ACKNOWLEDGMENTS

I would like to express my thanks to Julian Harty of Commercetest, who has worked with me to create the PTMM and to the reviewers of this paper for their valuable feedback.

9. REFERENCES

[1] Black, R., *Critical Testing Processes*, ISBN 0-201-74868-1, Addison-Wesley, 2004.

[2] Boelter, P., Test Assessment and Improvement in BOOTSTRAP, ICSTEST-UK, London, Sept 2002.

[3] Burnstein, I., Suwanassart, and T., Carlson, C. R., Developing a Testing Maturity Model: part 1 and 2, *CrossTalk: Journal of Defense Software Engineering*, Vol. 9, No's 8 & 9, Aug/Sept 1996.

[4] Carver, J., Shull, F., and Basili V., *Investigating the Effect of Process Experience on Inspection Effectiveness*, Univ. of Maryland ESEG Technical Report, CS-TR-4441, March 2003.

[5] CMMI Website at: http://www.sei.cmu.edu/cmmi/cmmi.html (accessed February 2006).

[6] Couger, D. and Adelsberger, H., Comparing Motivation of Programmers and Analysts in different Socio/Political Environments: Austria compared to the United States, *Computer Personnel*, Sept 1988.

[7] Evans, I., *Achieving Software Quality Through Teamwork*, ISBN 1-58053-662-X, Artech House, 2004.

[8] Hackman J. and Oldham, G., *Work Redesign*, ISBN 0201027798, Addison-Wesley, 1980.

[9] ISO/IEC 15504:1998, Information technology -- Software process assessment.

[10] Koomen, T. and Pol, M., *Test Process Improvement: A Step-by-step Guide to Structured Testing*, ISBN: 0201596245, Addison Wesley, 1999.

[11] Nicholson, I. and Warden, R., *The MIP Report Volume 2: Motivational Survey of IT Staff*, Second edition, December 1996.

[12] Rooyen, J. v., Selecting Your Baggage For The Journey Through Testland, *TTN On-Line Edition for May 2003*, http://www.soft.com/News/QTN-Online/qtnmay03.html (accessed February 2006).

[13] TMM***i***™ Foundation web-site at: (http://www.tmmifoundation.org/) (accessed Feb 2006).

[14] Weyuker, E., Ostrand, T., Brophy, J. and Prasad, R., Clearing a Career Path for Software Testers, *IEEE Software*, Vol. 17, No. 2, March/April 2000.