



Submission: Consultation on the Standing Offer Agreement for Quality Professional Service 2 (SOA – QPS2)

Office of the Government Chief Information Officer, Government of Hong Kong Special Administrative Region

April 3rd, 2012

Contents

Submission: Consultation on the Standing Offer Agreement for Quality Professional Service 2 (SOA – QPS2)

Submission: Consultation on the Standing Offer Agreement for Quality Professional Service 2 (SOA – QPS2)1

1.	Introduction to Independent Testing Services	3
1.1.	Background	3
1.2.	Why Test?	3
1.3.	Why Test Throughout the SDLC?.....	4
2.	Proposal for Independent Testing Service Category	5
2.1.	Why the Need for Independent Testing?	5
2.2.	Quality Management Standards for Independent Testing	6
2.3.	Independent Testing Principles	7
2.4.	Qualifications for Independent Testing Organisation	16
2.4.1.	Independent Testing Service Provider	16
2.4.2.	Qualifications for Human Resources.....	16
2.4.2.1.	Principal Test Consultant.....	16
2.4.2.2.	Test Manager	17
2.4.2.3.	Test Team Lead	17
2.4.2.4.	Senior Test Consultant.....	18
2.4.2.5.	Test Consultant/Engineer	18

1. Introduction to Independent Testing Services

1.1. Background

The Office of the Government Chief Information Office has sought the views from industry on certain aspects of the Standing Offer Agreement for Quality Professional Services 2 (SOA-QPS2).

This Submission addresses the issue of the desirability of introducing an Independent Testing Services category under the new SOA in particular the following paragraphs of the Consultation document issued on 15 March 2012:

Paragraph 22: “The increasing demand of high quality IT systems from Hong Kong Government Bureau and Departments in response to public’s demand generated the needs for Independent Testing services. The Independent Testing service may include designing and executing various kinds of testing e.g. unit test, functional test, regression test, integration test, load test, accessibility test, and usability testing, etc. Hong Kong Government is positively considering this suggestion and would like to know the views from the industry.” and

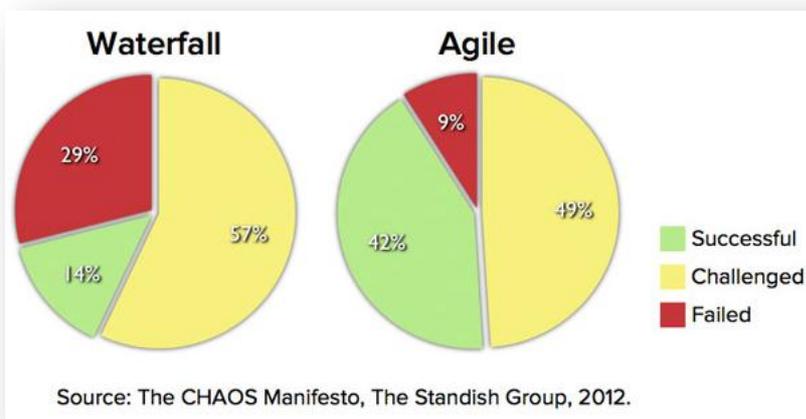
Paragraph 38: “ additional Staff Categories may need to cater for the specialist IT services like Independent Testing ... we welcome suggestions on ways to improve ... the human resource categorisation structure and the supplier-specific staff categories in the new arrangement.”

1.2. Why Test?

When technology is your business or central to it, testing your software or systems before implementing them is crucial. It protects a business from damaging inefficiencies such as loss of product functionality, poor usability or expensive system downtime.

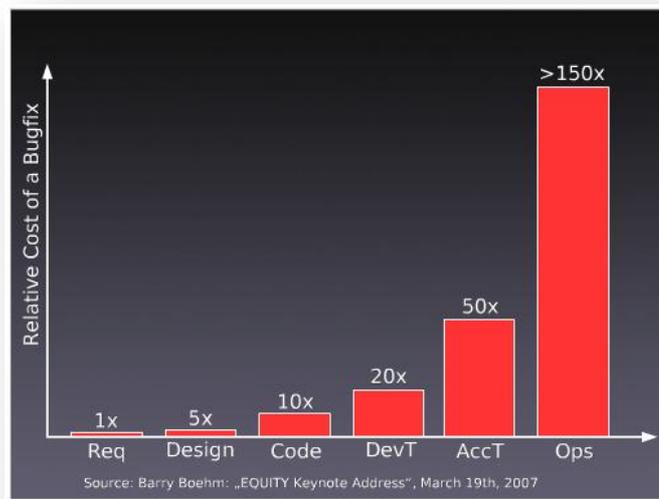
Of course if software applications and systems can be built “error free” there is no compelling reason to test. However the evidence overwhelmingly supports the case that this rarely – if ever - happens and the majority of software projects fail or are “challenged” ie they are delivered late, over budget or with less specifications than originally sought.

Annual reviews of software projects carried out by the Standish Group ([www.http://blog.standishgroup.com/](http://blog.standishgroup.com/)) over the past ten years reveal a constant trend irrespective of the methodology used.



1.3. Why Test Throughout the SDLC?

Research into the cost of fixing software defects throughout the software development life cycle (SDLC) shows that it is much less expensive to detect and fix defects as early as possible.



Independent Testing at different stages of the SDLC can significantly reduce errors and provide greater coverage than simply leaving testing to late in the SDLC as shown in the table below:-

SDLC Stage	Benefits from Independent Testing
Requirements	Minimize the GAP between actual business needs and IT deliverables at the later stage of SDLC.
Design	Translate the correct requirements into the design stage to ensure system design best matches business and real user requirements.
Code	Verify that the code functions as expected.
Development	Verify that the code works in the application.
Acceptance	Validate that the application / systems works as expected.
Operations and Maintenance	Confirm that the customer can use the application as expected.

2. Proposal for Independent Testing Service Category

2.1. Why the Need for Independent Testing?

As technology continues to change at a rapid rate with more sophisticated applications and devices capable of receiving information, the business case for independent testing becomes stronger for a number of reasons including:-

- Software developed by vendors will be tested as part of the software development life cycle (SDLC) – as time permits. Often testing is left to the last minute and products are frequently shipped with undetected errors (bugs) in them. Moreover vendors are likely to see their products in the best light possible and give an overrated assessment of the level of its quality – similar to a student writing an essay and then giving it a fail/pass/credit/high distinction mark.
- Rectifying bugs late in the SDLC is significantly more expensive than rectifying bugs at the early stages of development as described in 1.3 above.
- In many cases an independent software testing company can perform the role of “honest broker” between clients and vendors – especially when there are issues over whether the products or service are what the client wanted.
- Legitimate independent testing companies employ professional testers which results in a more professional outcome than the current practice of assigning developers to testing (who may have built the code being tested) or others who do not understand the need for comprehensive testing and do not have internationally recognised standards and methodologies to ensure that this happens every time. With the best will in the world it is difficult to pick up your own mistakes never mind the fact that the mindset of a developer is fundamentally different from that of a tester.

As Gartner pointed out recently:-

"The shift toward packages, service-oriented architecture (SOA) and business process management (BPM) also drives greater business analyst involvement in overall quality efforts, along with a shift in focus from finding defects in validation to ensuring that business objectives are met. Testing software can be an expensive process, but poor software quality leads to user dissatisfaction, as well as increased development and maintenance" – (Reference: [Gartner: Magic Quadrant for Integrated Software Quality Suites](#))

2.2. Quality Management Standards for Independent Testing

As a discipline, Independent Testing involves a set of complex processes and each one relies on a different set of tools and techniques. To be successful the workflow needs to be organised and managed within a structured quality assurance framework or quality management system.

A quality management system should be derived from internationally accepted standards such as the ISO/IEC 17025: General Requirements for The Competence Of Testing And Calibration Laboratories supplemented by other standards such as:

- ISO/IEC 14598-5: Information Technology – Software Evaluation. Part 5: Process for Evaluators, and
- IEEE 829 – 1998: Software Test Documentation

Related test methods provided by the independent testing service provider should also be based on international standards for software quality. These include:

- Quality of Product (technical aspects of the system and its performance against specifications and standards, interface specifications and interactive properties ISO9241 parts 10 and 12-17),
- Quality in Use (through usability ISO9241-11),
- Quality of Process (through human-centred processes ISO13407 and through user capability ISO TR 18529).

2.3. Independent Testing Principles

Having understood and implemented a quality management system Independent Testing organisations should also adopt internationally accepted testing principles as developed by organisations such as the International Software Testing Qualifications Board ISTQB® which was founded in November 2002 as a not-for-profit association based on volunteer work by hundreds of international testing experts.

A comprehensive set of testing principles is set out below:

Test Topic	Principles
Initiation	Ensure that the Quality Plan, Project Initiation Document (PID) or equivalent includes information regarding the test engagement.
Test Governance	<p>Ensure that test procedures are consistent with the overarching standards for governance mandated for the programme, project, or operational unit.</p> <p>Ensure that the test procedures and test activities are compliant with the commercial terms by which the test work stream is governed.</p>
Vocabulary	<p>Determine existing project vocabulary in use by stakeholders, then enhance as required to add any test specific terms that are required. Document new test terms in the project glossary.</p> <p>Establish and maintain a common vocabulary for the test team specific to assignment and document this in a test glossary in the test documentation. If the client has no existing test vocabulary refer to ISTQB Standard Glossary.</p> <p>Share test glossary terms with stakeholders at earliest opportunity to get terms in common use.</p> <p>Reference the location of the test glossary in the Test Strategy.</p>
Organisation	<p>Clearly document, and confirm with the client, the roles that:</p> <ul style="list-style-type: none"> • Are fulfilled by the test team, • Deliver artifacts (specifications, code, user guides, etc.) to the test team (e.g. software development teams), • Deliver services (e.g. configuration management, change control, infrastructure support, etc.) to the test team (e.g. configuration management), • Receive artifacts (e.g. defect reports, final test report from the test team (e.g. Development Team, Delivery Team, etc.). • Identify the stakeholders in the test process and get their engagement, and commitment to the test process. <p>Align and confirm the responsibilities and deliverables from the testing work stream to each stakeholder.</p> <p>Align and confirm the responsibilities and deliverables <u>to</u> the test work stream from each of the stakeholder work stream interfacing to testing.</p> <p>Confirm that test reports are communicated to the body responsible for arbitrating between cost, delivery time, and quality of the final deliverable to the end client.</p> <p>Build effective communication between developers and testers to ensure all</p>

	<p>staff are fully engaged in achieving the test objectives.</p> <p>Ensure key test specialists are certified by industry recognised professional test authorities.</p> <p>Organise test responsibilities and organisation taking into account:</p> <ul style="list-style-type: none"> • The need for end user interaction during test process, • Geographic spread of test resources. • Communication efficiency and effectiveness between testers, test site, and stakeholders. • Build the test team to give collective knowledge of test techniques, business awareness, client's marketplace, and the implementation technology. • Build a test team with a diverse range of aptitudes for addressing the test assignment. Successful test teams often have a mix of staff that collectively are pragmatic, analytical, cooperative, innovative, articulate, inquisitive, organised, meticulous, persistent, assertive, and energetic.
Scope	<p>Define and agree with the client the quality characteristics included in the test scope. For example ISO9126 defines 6 prime characteristics :</p> <ul style="list-style-type: none"> • Functionality • Reliability • Usability • Efficiency • Maintainability • Portability <p>Identify any known issues (e.g. from risks that are known to have been realised, past issue logs). Define and agree with the client the test strategy, test scope and test approach for the known issues.</p> <p>Define and agree with the client:</p> <ul style="list-style-type: none"> • the application of static, dynamic, and automated testing, • the test levels (e.g. component, integration, system and acceptance testing) to be performed in the assignment, • the boundary of the system under test in each of the agreed test levels, <ul style="list-style-type: none"> ➤ what testing is <u>included</u> in the assignment, ➤ what testing is <u>excluded</u> from the assignment, ➤ the entry and exit conditions for each test level. <p>Define and agree with the client the test specific deliverables, including but not limited to:</p> <ul style="list-style-type: none"> • Test Strategy • Test Plans • Test Specifications • Test Cases • Test Schedules • Test Input Files and Test Outputs • Automated Test Scripts <p>Define and agree with the client the other documents received or compiled by</p>

	<p>the test team, including but not limited to:</p> <ul style="list-style-type: none"> • Project Plans • Meeting Reports (including actions) • Correspondence • Memos • Standards and Guidelines • Reports on Progress and Quality • Incident (defect) Reports • Review Feedback Reports <p>Define and agree with the client the element of reuse of past test assets, including but not limited to:</p> <ul style="list-style-type: none"> • Test Strategy. • Test Plans. • Test Specifications. • Test Cases • Test Schedules • Test Input Files and Test Outputs. • Automated Test Scripts. • Test Environments. • Test Tools. • Test Tool Licences. <p>Define and agree with the client the supply of data from the business for:</p> <p>Volumetric data for performance and load tests.</p> <p>‘Anonomised’ operational data compliant with applicable data protection legislation.</p>
Risk Analysis	<p>Get agreement from client on the use and application of risk based test approach.</p> <p>Involve all relevant stakeholders in determining the relative importance of the quality risks. Stakeholders should include: developers (technology risk); client (business risk); end user (operational risk).</p> <p>Involve all relevant stakeholders in determining the relative importance of the ‘sub-system’. For this purpose the sub-system can be considered in terms of:</p> <ul style="list-style-type: none"> • Business transactions • User functionality • System components • Product or Service items. <p>Agree with the client and stakeholders the measures used for ‘Impact of Failure’ and ‘Probability of Failure’ and the algorithm used for translating these into the risk prioritisation. Make the measures and the ranking algorithm simple, practical and understandable to the client and stakeholders.</p> <p>Involve client stakeholders in assessing and agreeing “Impact of failure” for each quality risk.</p> <p>Review and reassess the “Impact of failure” whenever software, system, business process, service or product changes.</p>

	<p>Assess the “Probability of failure” initially on prior historical experience, and thereafter based on empirical data collected during testing.</p> <p>Agree risk analysis with client stakeholders prior to the start of test preparation.</p> <p>Communicate the risks identified in the risk analysis into the programme risk management system to ensure ownership and mitigation of each risk by the most appropriate party.</p>
<p>Estimation</p>	<p>Estimate test effort using at least two of the methods below:</p> <ul style="list-style-type: none"> • Prior test history (past project data). • Top-down (use broad metrics, usually percentages of development effort). • Bottom-up (define each test activity in detail). • Re-estimating by test level (estimating is a continuous process, re-estimate the test level as it is about to start, based on historic experience). • Estimation around the elements defined in the Access Testing Estimation Technique document – i.e. component driven, process driven, hierarchal driven, Decomposition, Complexity • Consensus of experts (otherwise known as Delphi Oracle technique). • Cost Averaging (three estimates made, most optimistic, probable, and most pessimistic, these are then combined in a weighted average $\{[a+4b+c]/6\}$ to give the estimate). • SWAG (Scientific Wild Ass Guess. Sometimes backed up by some simple formulae and usually quoted as a range). • Formulae and models (e.g. Test Point Analysis, Functional Point Count, effort per 1000 lines of code). • Estimating tools (use proprietary estimating models based on software effort/duration algorithms and a database of past projects). <p>If the client, or project, mandate specific test deadlines or cap test effort then make the implications of this approach clear to the client in terms of any constraints this imposes on the ability of testing to mitigate the risks to the business.</p> <p>Record the references, assumptions and constraints implicit in the effort estimation. Document the estimate so that it can be audited against actual experience, updated and reused in successive releases.</p>
<p>Strategy and Planning</p>	<p>Document the test strategy and test plan. Note: This may be embodied in one or more documents. Small or simple projects might be best served by a single document. Large or complex projects might be best served by a separate test strategy document and one or more test plan documents for an assignment covering different test levels (unit, component, integration, system, acceptance) and/or different types of testing (functional, performance, load, security, configuration). This is for the discretion of the test manager in discussion with the client.</p> <p>Agree the format of the test strategy, and the test plan documentation with the client.</p> <p>Define and document the test strategy and the test plan and agree with the client.</p> <p>Include the key milestones in the test plan.</p> <p>Include the key progress measures in the test plan.</p>

	<p>Define in the test plan the dependencies of the test activity and the stakeholders responsible for satisfying those dependencies.</p> <p>Clearly state entry and exit criteria for each test level.</p> <p>Agree and approve Acceptance Test Plan with the acceptance authority and any relevant stakeholders.</p> <p>Plan for test improvement. Include test activities in the plan for the improvement of the test assets and the test process during the life of the test team.</p> <p>Include order, procurement, installation, and commissioning lead-times in planning for the automated test tools and test environments.</p> <p>Put Test Strategy and Test Plans into the project configuration management system when they are approved.</p>
Engagement with suppliers	<p>Agree with stakeholders and client the involvement of the Access Testing Services managed test team in the test activities of suppliers of assets to the Access Testing Services managed test levels. Agree Access Testing Services test team involvement in test activities prior to Access Testing Services managed test levels, including but not limited to:</p> <p>Reviewing requirements, technical specifications and other project documents defining the system under test.</p> <p>Reviewing test specifications of the prior test activities.</p> <p>Witnessing of prior tests (e.g. unit test and component test).</p> <p>Reviewing prior test results and test evidence.</p> <p>Agree pre-delivery quality assurance activities with each supplier to the Access Testing Services managed test levels (e.g. test reviews, test witnessing, and test results review).</p> <p>Agree handover process with each supplier to the Access Testing Services managed test levels.</p> <p>Agree the format, content and detail in the documentation (e.g. release note) accompanying each suppliers' delivery into the Access Testing Services managed test levels.</p> <p>Clearly specify and agree the entry criteria for each deliverable from each supplier to the Access Testing Services managed test levels.</p> <p>Document the agreements made with the suppliers to the Access Testing Services managed test levels, and reference these in the Test Strategy.</p> <p>Mitigate quality risks early, the test approach for prior test activities should consider:</p> <p>Documentation reviews</p> <p>Test coverage required of branches and statements in the item under test.</p> <p>Test coverage of error trapping functionality for user interface and system generated errors.</p>

	<p>Test coverage of code implicated in high risk business functionality.</p> <p>Static and dynamic testing to identify memory leaks</p> <p>Test coverage of the interfaces of the item under test (e.g. subroutine calls, method invocations, service calls, messages, batch process calls and events)</p> <p>User interface verification for:</p> <ul style="list-style-type: none"> • Appearance including human factors, usability, design standards compliance, corporate image compliance. • Single field validation • Cross field validation • Drop down lists and menus • Data representation • Navigation • Formatted output and templates • Batch process interfaces
Dependencies	<p>Define and agree with the client and stakeholders their commitment to supply the major dependencies the test teams have on the activities that supply <u>test artifacts</u> (e.g. documentation, test environments, testable software).</p> <p>Define and agree with the client and stakeholders their commitment to supply the major dependencies the test team have on the activities that supply <u>services</u> to the test team (e.g. configuration management, change control, defect management, database administration, operational support of test environments, third party product support, training).</p> <p>Define and agree with the client how unfulfilled dependencies are to be managed. For example, should the test team budget for and then resource the unfulfilled dependencies by making them internal to the test team.</p> <p>In the event that test team dependencies cannot be planned to be met by staff/teams outside the test team, define and agree with the client the extra resources, training, equipment, facilities, budget, and constraints for any of the unfilled test team dependencies to be met by the test team itself.</p> <p>Raise defaulting dependencies to the test team as risks in the programme risk management system. Ensure there is an owner, mitigation plan and a new updated plan agreed with programme management.</p>
KPIs and SLAs	<p>Define and agree with the client and stakeholders their commitment to practical SLAs for support services supplied to the test team, in particular:</p> <ul style="list-style-type: none"> • review comment turnaround, • defect analysis responses to test incidence reports (in-house and third party product suppliers), • test environment support and configuration (e.g. database admin and operational support), • test environment availability. • test data provision. <p>Document and confirm with the client the KPIs used to measure testing progress, efficiency and effectiveness.</p>

Support Activities	<p>Document all the support activities (in test plan or separate document) required by the test team and receive commitment for the support from each provider. As a minimum consider:</p> <ul style="list-style-type: none"> • Change Control. • Configuration Management. • Defect Management. • Procurement (test tools and test hardware) • Human Resources (recruitment of client staff and external specialists to the test team) • Training on system under test • Test environment software installation • Test environment operational support. • Operational support of systems interfacing to the system under test. • Test data preparation. • Database management. • Test witnessing. • Test signoff.
Test Environment / Environmental V	<p>Scope the test environment regarding:</p> <ul style="list-style-type: none"> • hardware platforms and operating systems, • applications software, • physical environment (e.g. test rooms for testers, computer rooms, office facilities, simulated or example user environment), • data storage capacity, • network connectivity, • computer peripherals (e.g. printers, scanners, modems, etc), • specialised peripherals (e.g. phones, mobiles, PDAs, call centre desk, test rigs, other interfaced electronic equipment). <p>Define and agree overall ownership of the test environment and all its constituent parts with the client and relevant stakeholders.</p> <p>Define and agree overall ownership for all the support activities (installation, configuration, maintenance, operational support, fault rectification) relating to the test environment.</p> <p>Confirm and agree SLAs with each owner of the required support activities for the test environment.</p> <p>Agree with the owner and stakeholders of each test environment the schedule for delivery, use, and support of the test environment.</p> <p>Define Environmental tests that will be carried out based on the environment requirements and build.</p> <p>Execute a defined set of repeatable environmental, configuration and software installation tests that will be carried out.</p> <p>Report on the testing that has been performed – treat the environmental tests as if they were part of the software testing activity.</p> <p>Provide defect reports on the environmental testing.</p> <p>Perform non-functional / performance testing activities on the environment, as defined by the scope and requirements of the environment and the requirements within the SLA's / KPI's that will have been agreed.</p> <p>Set a number of acceptance criteria on the environment (high level) that will</p>

	<p>ensure that the environment is in a fit state for deployment of code, and that we are aware of the 'starting position' of the environment.</p> <p>Assure that any Unit testing that is carried out in the development environment has a subset of activities carried out in the testing environment, prior to acceptance of the developed code, and commencement of the software 'V' activities.</p>
Test Tools	<p>Only use test tools where the investment in software tools, automation design, automation scripting, and automation maintenance either:</p> <ul style="list-style-type: none"> • Yields a clear cost/benefit over manual testing, or • Can only be accomplished through tool use (e.g. load test, bespoke test rigs). • Clearly state the assumptions and constraints necessary to make the return on investment in the test tool. Factors for consideration include: <ul style="list-style-type: none"> • Practical lifetime of the tool. • Future technology upgrades of system under test. • Training in tool usage and administration. • Hardware environment (e.g. CPUs, disk storage, network, firewalls). • Ancillary software (e.g. database, web server, email, test data generators). • Ongoing maintenance costs on hardware and software. • Cost of migrating legacy test data and test scripts from past test tools. • Cost of migrating test data and test scripts in future upgrades of the chosen test tool. <p>Scope the relevant application areas for test tools in the programme, consider:</p> <ul style="list-style-type: none"> • Test management. • Assisted manual testing • Automated functional testing. • Performance testing. • Load testing. • Security testing. • Test data generation. • Collation and anonymisation of operational and volumetric data from the client. • Data migration tools and scripts. <p>Monitoring tools for application software, operating system, hardware, network, or other test environment entities.</p> <p>Any special purpose hardware or software for the test environment (e.g. simulators, test rigs, instrumented hardware).</p> <p>State the evaluation criteria to be used for selecting each of the test tools.</p> <p>Trial test tools in the most representative environment possible prior to</p>

	<p>commitment to purchase.</p> <p>Invite multiple tool vendors to a test trial as part of the tool procurement decision.</p> <p>Specify a realistic trial objective and setup a representative test environment for vendors to demonstrate their products against.</p> <p>Get commitments from vendors for order lead-times, installation, and commissioning times.</p> <p>Prepare budget for test tools then discuss and agree with client.</p> <p>Establish clear ownership for each of the test tools including the related environment, configuration, automated test scripts and test data.</p> <p>Document the use and application of automated test tools in the Test Strategy.</p>
<p>Non Functional Readiness</p>	<p>Scope the relevant application areas for testing in the programme, consider (notwithstanding the Performance Testing or Customer Experience Testing activities that will also be planned):</p> <ul style="list-style-type: none"> • Automated testing. • Security testing. • Test data generation • Compliance Testing • Auditability • Configuration • Disaster Recovery • Capacity Planning • Cut Over, Fallback and Restartability • Resiliency <p>Assess the readiness of the client, the organisation and the</p>
<p>Readiness Reviews</p>	<p>An activity that is carried out to ascertain whether the client is ready to move to a particular phase of testing activity, or indeed a particular concept that is new to them – such as</p> <ul style="list-style-type: none"> • Automation Readiness • Tool Readiness • Test Phase Readiness • Internal Testing Team Readiness <p>This provides an assurance that the client will be receiving the most positive ROI for their investment, and they are just not moving to a phase or activity that they may not be able to support in the medium to longer term.</p>

2.4. Qualifications for Independent Testing Organisation

2.4.1. Independent Testing Service Provider

Independent Testing providers should have the following as a minimum:

- Experience – at least three to five years in local or overseas markets.
- Proven track record – ability to provide at least three customer testimonials as a service supplier for Independent Testing services.
- Competency – methodologies that comply with international standards such as the Test Maturity Model or the Test Process Improvement model.
- Capacity – in-house qualified resources with the skill sets as outlined below.

2.4.2. Qualifications for Human Resources

2.4.2.1. Principal Test Consultant

- 10 years plus experience in software testing plus at least 5 years in a leadership role
- Degree in Computer Science, Software Engineering or equivalent
- **ISTQB*** Certified
- At least 5 years of software development lifecycle (SDLC) experience
- Minimum 3-5 years of Project Management experience
- Good understanding of business processes
- Knowledge of software testing methodologies
- Knowledge of both functional and non-functional testing activities
- Knowledge of all testing types (e.g. Black Box, Regression, System Integration)
- Thorough knowledge of defect tracking tools
- Good knowledge of test automation tools
- Knowledge of specific technologies may be required for certain roles
- Good knowledge of change management processes
- Thorough knowledge of version control processes
- Appreciation and knowledge of ITIL and other service management methodologies

***ISTQB** stands for International Software Testing Qualifications Board, is a software testing qualification and certification organisation that operates internationally. The certification enables people to have sufficient testing skills and knowledge to provide quality testing service, which such testing skills and knowledge are important elements to the success of Independent Testing Service. (For more details, please visit <http://www.istqb.org/>)

2.4.2.2. Test Manager

- 7 years plus experience in software testing plus at least 3 years in a leadership role
- Undergraduate degree in Computer Science, Software Engineering or equivalent
- ISTQB Foundation Level Certified
- ISTQB Practitioner Level Certified (desirable)
- Thorough knowledge of the software development lifecycle
- Thorough knowledge of software testing methodologies
- Thorough knowledge of both functional and non-functional testing activities
- Thorough knowledge of defect tracking tools
- Good knowledge of test automation tools
- Knowledge of specific technologies may be required for certain roles
- Good knowledge of change management processes
- Certification in Project Management
- Appreciation and knowledge of ITIL and other service management methodologies

2.4.2.3. Test Team Lead

- 5 years plus experience in a software testing
- Undergraduate degree in Computer Science, Software Engineering or equivalent
- ISTQB Foundation Level Certified
- Thorough knowledge of the software development lifecycle
- Thorough knowledge of software testing methodologies
- Thorough knowledge of both functional and non-functional testing activities
- Thorough knowledge of defect tracking tools
- Good knowledge of and exposure to test automation tools
- Knowledge of specific technologies may be required for certain roles
- Good knowledge of change management processes
- Good knowledge of project management processes
- Knowledge of estimation techniques and processes
- Ability to develop long term strategic enhancements

2.4.2.4. Senior Test Consultant

- A minimum of 3-5 years' experience in a software testing
- Undergraduate degree in Computer Science, Software Engineering or equivalent
- ISTQB Foundation Level Certified as a minimum
- Thorough knowledge of the software development lifecycle
- Thorough knowledge of software testing methodologies
- Good knowledge of both functional and non-functional testing activities
- Thorough knowledge of defect tracking tools
- Knowledge of, and exposure to test automation tools
- Knowledge of specific technologies may be required for certain roles
- Knowledge of change management processes
- Knowledge of project management and service management processes and approaches

2.4.2.5. Test Consultant/Engineer

- A minimum of 2-3 years' experience in software testing
- Undergraduate degree in Computer Science, Software Engineering or equivalent
- ISTQB Foundation Level Certified
- Thorough knowledge of the software development lifecycle
- Thorough knowledge of software testing methodologies
- Experience in both functional and non-functional testing activities
- Knowledge of all testing types e.g. black box, regression, systems integration etc
- Thorough knowledge of defect tracking tools
- Knowledge of, and exposure to test automation tools
- Knowledge of specific technologies may be required for certain roles
- Knowledge of change management processes
- Knowledge of project management methodologies

END