Terms of Reference

For

Design, develop, deploy and maintenance of the Integrated Land Management System (Stage 1)

Terms of Reference

For

Design, develop, deploy and maintenance of the Integrated Land Management System

(Stage 1)

(ICTA/SG2/GOSL/CON/QCBS/2018/004)

1. Introduction

Lands are one of the key economical grounds in the context of national economy. It is the base to major operation of the country. Information related to land splits in to various avenues according to the different dimensions of use. Lands owned by public and private sector are usually monitored by different government authorities of the country. As an administrative purpose different government bodies were established to fulfilment of the requirement. The Ministry of Land of the central government, Department of Land Commissioner General, Registrar General's Department, Survey Department, Department of Land Title Settlement, Valuation Department, Land and Agriculture Ministries of Provincial Councils, Provincial Land Commissioner's Department, Local Government Institutions and Divisional Secretariats are the prominent organizations which handle crown land matters.

The Government of Sri Lanka (GoSL) has recognized ICT as a tool, which the social integration, peace and growth can be fostered and the poverty could be reduced by the means of improving the reach and responsiveness of public services, to reduce transaction costs of businesses, making government more transparent, accountable and addressing the urgent needs of poverty - stricken communities and isolated regions. With the collaboration of the authorised bodies related to the Ministries and the ICTA launched the project to integrate the land related information under a single umbrella called Integrated Land Management System (ILMS). Thus, the proposed ILMS is a centrally located hub system which integrates the individual developed systems. As a national objective provide better land information for the citizens of Sri Lanka and international community who interested in lands.

2. Background

Mainly the departments and organizations related to different Ministries are marching towards the early dawn of fully computerized environment. Most of these institutions are in the migration period from manual to fully computer environment. Some of them are computerized provide service to the general public. Systems which have been developed categorized in to in-house developments to outsourced developments.

As a country, strategic development of ICTA is a one stop solution for given domain. These individual systems which developed work together to provide a quality service to the general public. Integration of the developed sub system to a central hub would be the solution. These are the central government organization with the authority relating to land matters in Sri Lanka. Further, Land Commissioners General Department, Survey Department, Land Title Settlement Department and Registrar General's Department are central government institutions which are

mandated to work on private and public lands Sri Lanka. Ministries have implemented a re-engineering study to increase the efficiency of the citizen service delivery mechanism related to land activities in various services. In this endeavour head Ministries are working with a number of government organizations as stated above. In order to overcome many issues related to lands, these sub systems are developed by depth and re-engineering study by covering all the organizations and a set of re-engineered processes have been established at this moment. Furthermore a software requirement study has been conducted based on the re-engineered processes and a detailed requirement specification has been finalized with the participation of all the institutions. Few of the sub systems are in the middle of the development, rest of them are up and running in their respective institutions. Developed systems are namely; Land Information Systems developed by the survey department which include the preparation of complete information embedded parcel fabric. eLR system related to the land registration process handles by registrar department. Land valuation system to automate the calculation related to the land valuation process. Online payment gateway to pay the land related stamp duty and taxes. Finally, integrate the sub systems to the local authorities such as Municipal Councils and "Pradeshiyasabas", pilot run to be implemented with the Colombo Municipal Council (CMC).

Proposed ILMS consist of systems of existing systems developed and some are under development stage. Current Systems are : eLand Registry System , Land Information System Land Use Planning system , eSLIMS , Integrated Land Asset Valuation & Revenue System.

3. Concise statement of the objectives

This project envisages to hire a firm for acquiring a robust and scalable Integration system to achieve the below stated brief scope of services within a period of Seven Months (7) , conforming to the industry standards; implementing, integrating with internal applications and providing support and maintenance for a period given under the scope of work. The proposed system will integrate the mainly the stakeholders related to the land domain supporting service base organizations. This will increase the quality of life and information availability of the land domain to the general public. Integrated Land Management System which is integrated system span to multiple stages. Stage one (1) of the project focused on primarily of development of integrating engine for the existing systems and open standard for the future integrations

Objective of the Project

The key objectives of Integrated Land Management System Stage 1

- To develop a common integration platform for the land domain
- To improve the base land information and transactional services to the government and public

4. Scope of Work

4.1 Proposed system consist of 2 main stages, where this project will only cover the stage 1 of the Integrated Land Management System. Design of the system should be expandable to the stage 2 development.

4.2 Study, design, develop, deploy and maintenance of Integration System based on the key land domain stakeholders.

4.3 Conduct detail study of the processes of the systems been deployed, under development and future planned systems to be integrated to the ILMS.

4.4 Prepare the Detail Software Requirement Specifications (DSRS) for the proposed system.

4.5 ILMS shall interface the systems data bases including Document Management Systems (DMS) of the stakeholder organizations and provide data and information.

4.6 The ILMS shall facilitate transactional data which are in forms of Text , Images , Documents and Geographical Data.

4.7 The ILMS shall integrate with multiple payment gateways and bank wallets to facilitate online payments

4.8 ILMS shall be hosted at Lanka Government Cloud 2.0 (LGC), consultant to provide the requirements and deployment architecture. LGC offers IaaS

4.9 The consultant shall provide all required licences, subscriptions for the 3rd party software solutions

4.10 ILMS shall be deployed as high availability on the cloud environment and setup server requirements including system certificates / security - (Eg: SSL, HTTPS, etc)

4.11 Shall create the staging and production environments for the system.

4.12 Adherence to Web 2.0 concepts, open standards and Service Oriented Architecture (SOA) principles

4.13 The system architecture shall have proper disaster recovery mechanism, equipped with high availability and fault tolerance.

4.14 Shall carry out iterative design and development related activities in relation to each iteration.

4.15 The proposed services/modules offer to public (interfaces) shall made available in tri-languages (Sinhala, Tamil and English)

4.16 ICTA or its designated entity (end user) shall have the ownership rights to client specific components arising from the requirements specified and not limited to it.

4.17 ICTA or its designated entity (end user) should have the right to access, modify, further develop and enhance the system at no cost to ICTA or its designated entity (end user)

4.18 If any commercial version of the software need to be used in the proposed solution, the consultant need to inform in advance with proper justification of the requirement.

4.19 The consultant should follow the proper coding standard.

4.20 Maintain project source code and documents in the ICTA's Concurrent Versions System (SVN) and Source Code Management (SCM).

4.21 Maintain all issues in the issue tracking system by ICTA.

4.22 Shall adopt a proper application release procedures to release the ILMS during staging and production environments iteratively.

4.23 Shall develop proper alerting mechanism to monitor the system performance issues, exception and downtimes via notifications: SMS, email, etc.

4.24 Security assessments shall be carried out by SLCERT and shall fix reported issues identified. This cover during maintenance period.
4.25 Implement the system in collaboration with the 'Software Process Audit (SPA)' team at ICTA (Refer Annex C).
4.26 Shall derive UAT for test cases and obtain User Acceptance for the implemented processes iteratively
4.27 The application shall be compatible with latest technological components and best practices
4.28 Provide support and maintenance service and change requests for a period three years.
4.29 Shall adhere to the Service Level Agreement (Annex B)
4.30 Provide training to the system users of the organizations
4.31 Participate at the Project Review Committee meetings as a member
4.32 Work collaboratively with ICTA and related project stakeholder organizations
4.33 Refer following Annexes which form a part and parcel of the "Terms of Reference"

5. Technical Information for Integrated Land Management

Integrated Land Management System will be consisting of integration of existing systems using an Application Programming Interface (API) Core of the proposed system is the integration engine which will be used as the core to connect the existing systems and the future systems. Apart from the integration it is essential to develop new modules to the core system. The details of the following provide a guide to identify the scope of the project.

Integration Application:

During the development following are the expected new modules / tools;

- Integration core system module
    - Integration hub
- Application Integration Interfaces (API's to with individual systems)
    - Revenue department integration
    - e Citizen app (eLR) integration
    - Parcel fabric integration
- Mobile Apps
    - Development of land information citizen App
    - Development of Land related stakeholder App
- Colombo Municipal Council application window
- Professionals registrations
    - Registration of professionals eg : lawyers , surveyors , etc
- Deed preparation editor
    - Development of deed preparation editor
- Customizable reporting engine
    - Customizable reporting engine
- Dash board & visual Information tool
    - General dashboard for the public
    - Dashboard specifically develop for the Government organizations
    - Visual information tool link with the parcel fabric / cadastral map
    - Information visualization of the land domain
    - Visual analytics

- System administration module
    - General high end admin panel for overall management
    - Organizational panel for each organization
- Land transaction module
    - Capture the land transactions

Existing Systems;

1. eLand Registry : (Integrate with Stage 1)

System provide the core functions related to the land registry. It consist of land registry office system and online app for the general public to carryout transactions related to land registration and searching, etc

2. eSLIMS System : (Integrate with Stage 2)

State Land Information System provides activities related to government land domain. Main stakeholder of the system is Land Reform Commission, Land Commission , Mahaweli authority.

3. LIS System: (Integrate with Stage 2)

Land Information system function in the Surveyor General Department which consist of land related information.

Systems which are Under Development:

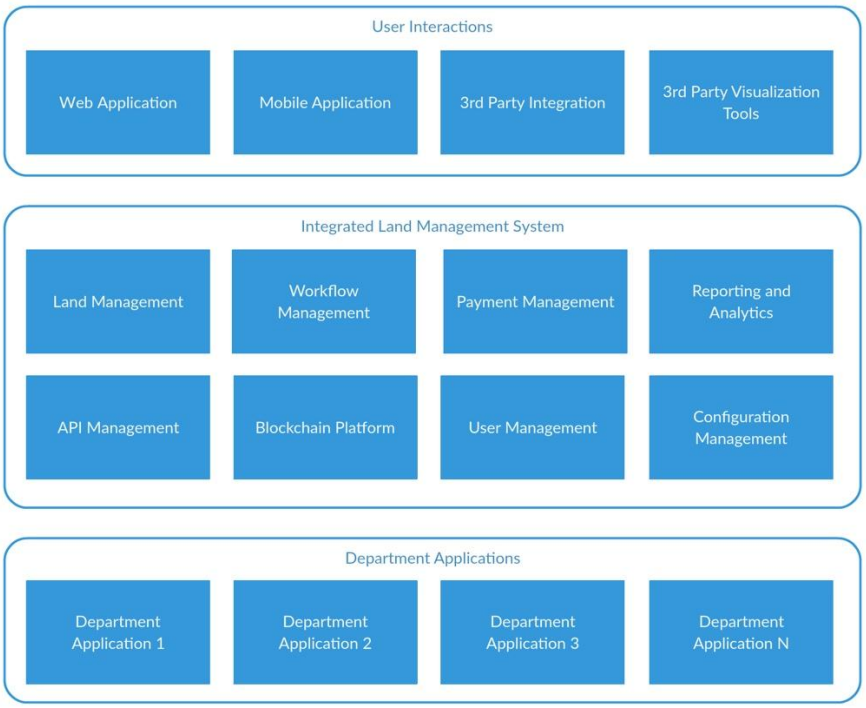
1. Land Asset Valuation System: (Integrate with Stage 2)

Objective of the proposed system is to provide a comprehensive solution for asset valuation system for National Valuation Department and Revenue Department of the Western Province.

2. Land Use Planning for LUPPD (Integrate with Stage 2)

Design of land use planning is the key output of the system which is currently under development.

PROPOSED HIGH-LEVEL ARCHITECTURE FOR INTEGRATED LAND MANGAEMENT SYSTEM



Introduction

The Integrated Land Management System will allow government departments to manage lands of Sri Lanka and will allow citizens to obtain land related services in a single window where multiple government departments will be integrated into a common system. Note that above diagram is drawn at an abstract level and detailed architecture diagram is expected in the RFP from the vendor.

Core components

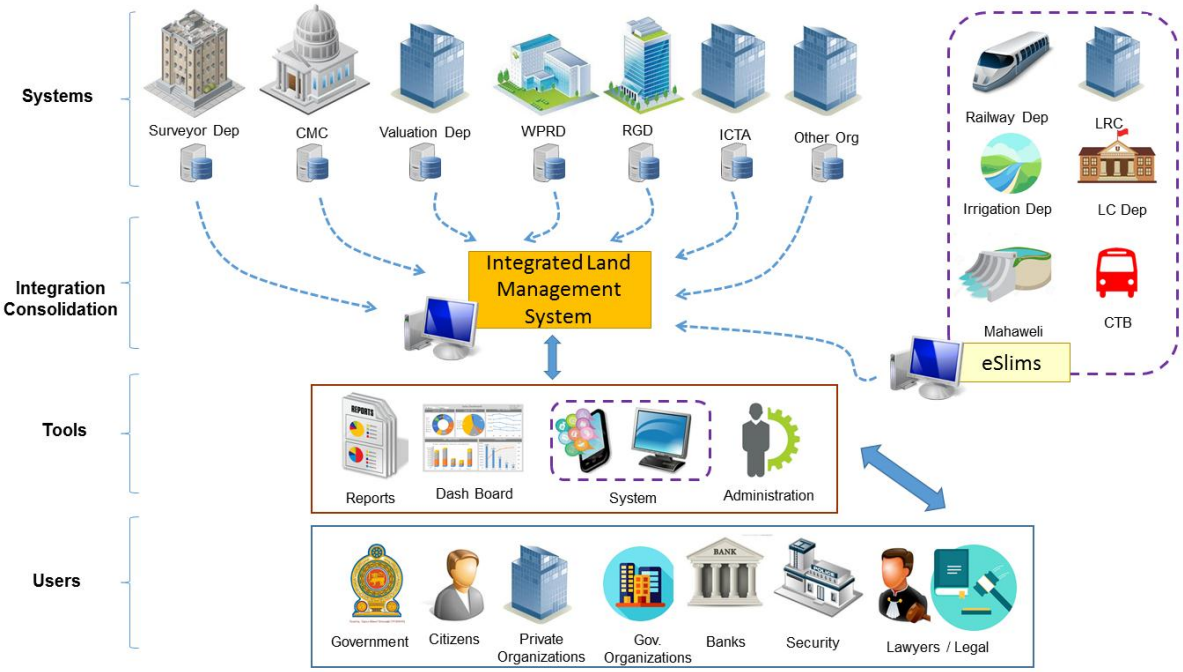
- Integrated Land Management System – The platform will consist of APIs in order to communicate with 3rd parties and UI where users interact. The departments' functionalities will be exposed via REST API into the platform to provide consolidated services.
    - Land Management – To facilitate the process of managing the use and development of land resources by aggregating multiple services.
    - Workflow Management – Allow the user to define different workflows for different types of jobs or processes.
    - Payment Management – Allow the management of payment across the system

- Reporting and Analytics – To facilitate the process of exploring data and reports in order to extract meaningful insights to it's stakeholders.
- API Management – To facilitate the process of creating and publishing web APIs, enforcing their usage policies, controlling access.
- Blockchain Platform – To preserve the integrity of the data of the system. Will be integrated in the phase 02.
- User Management – To facilitate identity and access management across the system.
- Configuration Management – To facilitate common interface to configure the system.

The vendor is allowed to proposed a cost effective, future-proof and a sustainable architecture. A similar nature architecture is expected and marks will be given specifically for adhering to the above.

Concept Diagram

Integrated Land Management



Function	Related System
Land owner registration	Citizen App
Lawyer registration	Citizen App
Surveyor registration	Citizen App
Land block registration	Citizen App
CMC ownership application	CMC
Online payment - CMC ownership	CMC
Building & street line application	CMC
Online payment - Building & street line	CMC
Non vesting application	CMC
Online payment - Non vesting	CMC
Title search application	RGD
Online payment - Title search	RGD
Deed preparation editor	Citizen App
Deed approval application	RGD
Online payment – Deed approval	RGD
Land valuation application	WPRD
Online payment - Land valuation	WPRD
Online payment – Stamp duty	WPRD
Customer application manager	Citizen App
Citizen complain mechanism	Citizen App
Online help desk (chat room)	Citizen App

Detailed requirements relating to above processes are documented in Business Process into detail levels such as main processes, sub processes, process components, steps and related forms.

Citizen System / App :

1. Land owner registration
2. Lawyer registration
3. Surveyor registration
4. Land block registration
5. CMC ownership application
6. Online payment - CMC ownership
7. Building & street line application
8. Online payment - Building & street line
9. Non vesting application
10. Online payment - Non vesting
11. Title search application
12. Online payment - Title search
13. Deed preparation editor
14. Deed approval application
15. Online payment – Deed approval
16. Land valuation application
17. Online payment - Land valuation
18. Online payment – Stamp duty
19. Customer application manager
20. Citizen complain mechanism
21. Online help desk (chat room)

Note :-

Each application link with a function of documents uploading facility

Reports :

1. Logins
2. Administration
3. Application status reports
4. User registration reports
5. Online payment reports
6. Transaction reports
7. Land blocks repots with different filtering options
8. Complain reports
9. + 25 Reports

Administration Panel :

1. User Logins
2. Parameter configuration (taxes , currency , etc)
3. Customizable report engine
4. Data base & traffic monitoring

Dash Board & visual Land Information :

1. Visual application status manager
2. Land parcels in visually
   - Commercial / residential
   - Land / building size
   - Road access
   - Location wise (archeological , forest , irrigation , etc)
3. User logins
4. Lawyers / Users / Surveyors / Land owner details (age , address , profession , etc)
5. Ownership
6. Building and street line
7. Non vesting
8. Title registration
9. Deed registration
10. Land valuation
11. Stamp duty payments
12. Land sales movements
13. Disaster areas
14. Colombo Smart city parameters

Annex A - Non-Functional Requirements
Annex B - Service Level Agreement (SLA) for Support and Maintenance Services
Annex C - Software Project Audit Process

6. Final outputs, Reporting Requirements, Time Schedule for Deliverables;

Project duration is Seven (07) months including requirement Verification, designing, developing and deploying the system.

Software firm is required to submit the following list of deliverables for the Design, develop, deploy and maintenance of the Integrated Land Management System by Agile Basis.

No	Deliverables	Phase
6.1	Implementation Proposal 6.1.1 Inception report 6.1.2 Implementing schedule / Project Plan 6.1.3 Detail Requirement study Report 6.1.4 Requirement verification report	Inception

No	Deliverables	Phase
6.2	6.2.1 Design and Architecture Document and to be updated iteratively (DSTD) 6.2.2 Detailed Requirement specification to be completed iteratively 6.2.3 Data migration and integration plan (if applicable) 6.2.4 Release Management plan (including staging, production and support and maintenance) 6.2.5 Proper maintenance of issues in the Issue tracking System 6.2.6 QA Test plan 6.2.7 Acceptance criteria for the UAT and test cases for all use cases 6.2.8 Report on any 3r party modules / tools use 6.2.9 Specification for hardware (if applicable)	Elaboration
6.3	**Iteration one** 6.3.1 Iteration one Prototype 6.3.2 Updated test plan for the iteration (Functional and Non-functional) 6.3.3 Release 1 – UAT Test cases , test data and UAT acceptance (Functional and Non-functional) 6.3.4 Proper maintenance of source code SVN 6.3.5 Proper maintenance of issues in the issue tracking System 6.3.6 Developer and QA release notes 6.3.7 Staging , Production deployment and confirmation report 6.3.8 Detailed requirement specification for iteration 2 6.3.9 Detailed design for iteration 2 **Iteration two** 6.3.10 Iteration Two Prototype 6.3.11 Updated test plan for the iteration (Functional and Non-functional) 6.3.12 Release 2 – UAT Test cases , test data and UAT acceptance (Functional and Non-functional) 6.3.13 Proper maintenance of source code SVN 6.3.14 Proper maintenance of issues in the issue tracking System 6.3.15 Developer and QA release notes 6.3.16 Staging , Production deployment and confirmation report 6.3.17 Release 2 – User training 6.3.18 Release 2 – User / Administration manual 6.3.19 Release 2– Deployment guide Proper maintenance of source code in SCM for all three iterations	Construction
6.4	6.4.1 Solutions deployment and installation guide 6.4.2 Online help and the User manual for back office application 6.4.3 Administrator Manual 6.4.4 Proper maintenance of issues in the Issue tracking System 6.4.5 Successful UAT acceptance of the system 6.4.6 Production deployment confirmation report	Transition

Refer http://en.wikipedia.org/wiki/IBM_Rational_Unified_Process for more information about RUP (Rational Unified Process) phases.

7. Preferred Qualifications of the Key Consultants

Preferable Minimum Qualifications;

- **Key team**

Key Professional Staff	**Academic Qualification**	**Experience in the PROPOSED ROLE**	**Experience in working in SOA / web services / integration projects**	**Exposure SQA Process**
Project Manager	B.Sc or equivalent	5 years	2 years	2 years
Software Architect	B.Sc or equivalent	3 years	2 years	2 years
Technical Lead	B.Sc or equivalent	3 years	1 years	2 years
Business Analyst	B.Sc or equivalent	3 years	1 years	2 years
Quality Assurance Lead	B.Sc or equivalent	2 years	1 years	2 years
UI / UX Lead	B.Sc or equivalent	2 years	1 years	1 years

- **Support and Maintenance Team**

Key Professional Staff	**Academic Qualification**	**Experience in the PROPOSED ROLE**	**Experience in working in SOA / web services / integration projects**	**Exposure SQA Process**
Technical Lead	B.Sc or equivalent	2 years	1 years	2 years
Support Engineer	B.Sc or equivalent	2 years	1 year	1 year

8. Services and Facilities Provided by ICTA

7.1 Web-based access to the ICTA SCM system
7.2 Designs of the existing system
7.3 Access to staging/ production servers
7.4 Issue Tracking System
7.5 SQA dashboard

9. References

[1] e-Government Policy

https://www.gov.lk/elaws/wordpress/wp-content/uploads/2015/03/eGov-Policy-structured-v4-0.pdf

[2] Lanka Interoperability Framework

http://www.life.gov.lk/

10. Review Committees and Review Procedures

The Software Development Service Provider is required to work closely with the ICTA Technology Team and the Software Process Audit (SPA) consultants or the review committees such as SAGE – Software Architecture Group of Experts.

All versions of deliverables will be reviewed by/either the SPA consultants, SAGE, or ICTA Technology Team.

All the deliverables must be verified and confirmed to be accurate and complete by the Project Implementation Committee (PIC) or the Project Management Committee (PMC). Deliverables must be formally endorsed by the PIC or PMC or CTO or Head of Technology Team.

Annex (A)

NON-FUNCTIONAL REQUIREMENTS

1. SECURITY

I. User authentication and authorization

All applications should be able to access via ICTA's common infrastructure/application itself and independently via respective department's web site if required. Any authorization requirements should be implemented within the specific web/mobile application.

However, the solution should have the provision to integrate with the ICTA's proposed Identity Management solution in future.

An administrative application need to be developed wherever applicable.

Wherever applicable internal small applications need to be developed to capture and store relevant data.

II. Confidentiality and Integrity

All developed web/mobile applications should ensure "confidentiality" and "integrity" whenever required by adhering to transport and message level security standards. (i.e.: HTTPS, WS-Security)

III. Authentication

The web/mobile application should be able to verify the users.

IV. Authorization

The web/mobile application should be able to verify that allowed users have access to resources.

V. Non-repudiation

All Web/mobile applications should ensure non-repudiation by having standard audit-trails and provisions to have WS-Security using digital signatures.

VI. OWASP Guidelines

All web/mobile applications should ensure that the OWASP guidelines for security are followed when designing, developing and deploying the web/mobile application.

2. AUDIT FACILITIES

Wherever applicable, an audit trail of all activities must be maintained. On a service or operation being initiated, the system should log the event, creating a basic 'audit log entry'. It should not be possible for the operation to be executed without the log entry being made.

The information recorded in the audit trail depends on the type of activity which takes place. Each service would be responsible for logging detailed information. The different types of operations are -

- Data Capture & Maintenance
- Creation of an entry / item
- Modification an item
- Deletion
- Control (or status change)
- Process execution
- Data synchronization
- Print (only selected item)
- Retrieval
- Monitor

Detail logging may be enabled or disabled for each type of operation, and/or for each business object. It should be possible to configure which attributes of a data item should be traced at the detail level. Tracing of some attributes may be considered mandatory, and they should not be turned off.

3. BACKUP AND CONTINGENCY PLANNING

The main contingencies that should be considered and the training with regards to these shall be given to the relevant staff -

- Equipment failure
- Physical / natural Disaster
- Messaging or communication facilities
- Changes in operations and policy
- Sudden absence of key personnel
- Breach in Security

Automatic Backups daily, weekly *and* monthly should be taken. All the backup procedures and backups needs to be tested regularly for restoration.

4. Performance Testing

Please find the below index as a guide to determine the benchmark values for the Application under the test.

Following performance criteria is provided as a guideline only. If the actual performance is falling below the stipulated figures, the consultant is to justify the reasons. However, the performance level must be accepted by the technical evaluation committee appointed by the client. The bandwidth is assumed at 1mbps (shared) with 1,000 concurrent users (50% load factor) in total.

Item	**Performance**
Screen Navigation: field-to-field	< 5 milliseconds
Screen Navigation: screen-to-screen	< 3 seconds
Screen Refresh	< 3 seconds
Screen list box, combo box	< 2 seconds
Screen grid – 25 rows, 10 columns	<3 seconds
Report preview – (all reports) – initial page view (if asynchronous)	< 40 seconds in most instances. It is understood that complicated / large volume reports may require a longer period
Simple inquiry – single table, 5 fields, 3 conditions – without screen rendering	< 4 seconds for 100,000 rows
Complex enquiry – multiple joined table (5), 10 fields, 3 conditions – without screen rendering	< 6 seconds for 100,000 rows
Server side validations / computations	< 10 milliseconds
Client side validations / computations	< 1 millisecond
Batch processing (if any) per 100 records	< 120 seconds
Login, authentication, and verification	< 3 seconds
Daily backups (@Dept.) – max duration	1 hour (on-line preferred)
Total Restore (@Dept.) – max duration	4 hours

4.1 Performance Test Process Outputs

- Performance Test Scripts
- Performance Test Results

5. USABILITY

The web/mobile application should be extremely usable, even a greenhorn user should be able to handle the system and incorporate all the functionality of the system in a simple and user friendly interface. The web/mobile application should be internationalized and localized if needed. The web/mobile application should be responsive where it should be viewable on any computing device.

6. INTEROPERABILITY

The web application should be able to view in standard compatible web browsers.

7. AVAILABILITY

The web/mobile application should be performed as follows,

- 99.99% available unless the web/mobile application is designed with expected downtime for activities such as database upgrades and backups.
- Hence to have high availability, the web/mobile application must have low downtime and low recovery time.

8. ROBUSTNESS

The web/mobile application should be able to handle error conditions gracefully, without failure. This includes a tolerance of invalid data, software defects, and unexpected operating conditions.

- Failure Detection
    - Once deployed, there should be appropriate tools to discover anomalies and failures of the system
- Fault Tolerance
    - Web/mobile application developer should anticipate exceptional conditions and develop the system to cope with them. The web/mobile application must be able to use reversion to fall back to a safe mode, meaning, the application should continue its intended functions, possibly at a reduced level, rather than falling completely.

9. MAINTAINABILITY

The code of web/mobile application should be properly documented with appropriate comments and no complex codes (highly cohesive and loosely coupled) to do modifications such as corrections, improvements or adaption.

10. COMPLIENCE TO STANDARDS

The code of web/mobile application should be standardized by following web/mobile standards like W3C and ECMA – European Computer Manufactures Association, to save time, augment the extensibility of the code, increase web/mobile traffic and improve the accessibly and load time of your application.

11. REUSABILITY

The web/mobile application should be able to use of existing assets in some form with the software product development process. Assets are products and by-products of the software development life cycle and include code, software components, test suites, design and documentation.

12. INTERNATIONALIZATION

The web/mobile application should be able to access in Sinhalese, English and Tamil. The web/mobile application should be able to view in a usable manner in all three languages in any computing device.

13. API MANAGEMENT

13.1. API Standards and Best Practices

API standards and best practices that *should be adhered* to the code.

13.2 API Documentation

- Swagger documentation should be provided.

13.3. API Security

The web/mobile application should be used appropriate API security protocol mentioned below.

- Basic API authentication
    - Basic authentication should never be used without TLS (formally known as SSL) encryption as user name and password combination can be easily decoded otherwise.
- OAuth1.0a
    - Uses cryptographic signature value that combines the token secret, nonce, and other request based information. Can be safely used without SSL.
    - Recommend for sensitive data applications

- OAuth2
    - No need to use cryptographic algorithms to create, generate and validate signatures as all the encryption handled by TLS.
    - Recommend for less sensitive data applications
- JWT (JSON Web/mobile Tokens)

14. SCALABILITY

The web/mobile application should be both scalable and resilient. A well-designed application should be able to scale seamlessly as demand increases and decreases. It should be resilient enough to withstand the loss of one or more hardware resource.

15. LEGAL AND LICENSING

The web/mobile application should comply the national law.

17. EXTENSIBILITY

The web/mobile application should be designed and developed in a way that it can cater to future business needs.

17. TESTABILITY

The web/mobile application should be designed and developed in a way that testability is high, meaning, the ease of testing a piece of code or functionality, or a provision added in software so that test plans and scripts can be systematically executed. In simple terms, the software should be tested easily with most famous 5 testing categories,

- Unit test
- Integration test
- System test
- Safety test
- Experience test

Refer Aden (2016)'s view on semantic testing for more information.

The web application should be working according to the given criteria in the latest version and 5 versions before in web browsers such as Mozilla Firefox, Google Chrome, Opera, and Apple Safari and the latest version and 2 versions before in Internet Explorer.

18. NOTES

- Some of the none-functional requirements shall be excluded based on the project requirement with the approval of the ICTA Technology Team.
- The vendor can propose similar standards/requirements for the above-mentioned standards/requirements with the approval of the ICTA Technology Team.
- The design documents should be based on 4+1 architecture model or the template provided by ICTA.
- If APIs are available (Service Layer) 'API Documentation' should be provided as an annexure to the design document as stated in 'Section 13'.

BIBLIOGRAPHY

1. The White House. *White House Web/mobile API Standards*. Washington, D.C.: git hub.com, 2015. Print.

2. Aden, S. (2016). Semantic Testing. Retrieved August 30, 2017, from https://semantictesting.org/

Annex (B)

SERVICE LEVEL AGREEMENT *for*
SUPPORT AND MAINTENANCE SERVICES

1. Introduction

The aim of this agreement is to provide a basis for close co-operation between the Client and the Consultant for support and maintenance services to be provided by the Consultant, thereby ensuring a timely and efficient support service is available. The objectives of this agreement are detailed in Section 1.1.

This agreement is contingent upon each party knowing and fulfilling their responsibilities and generating an environment conducive to the achievement and maintenance of targeted service levels.

1.1 Objectives of Service Level Agreements

1. To create an environment conducive to a co-operative relationship between Client, Consultant and Client's representatives (government organizations) to ensure the effective support of all end users.
2. To document the responsibilities of all parties taking part in the Agreement.
3. To define the commencement of the agreement, its initial term and the provision for reviews.
4. To define in detail the service to be delivered by each party and the level of service expected, thereby reducing the risk of misunderstandings.
5. To institute a formal system of objective service level monitoring ensuring that reviews of the agreement is based on factual data.
6. To provide a common understanding of service requirements/capabilities and of the principles involved in the measurement of service levels.
7. To provide for all parties to the Service Level Agreement a single, easily referenced document which caters for all objectives as listed above.

1.2 Service Level Monitoring

The success of Service Level Agreements (SLA) depends fundamentally on the ability to measure performance comprehensively and accurately so that credible and reliable information can be provided to customers and support areas on the service provided.

Service factors must be meaningful, measurable and monitored constantly. Actual levels of service are to be compared with agreed target levels on a regular basis by both Client and Consultant. In the event of a discrepancy between actual and targeted service levels both Client and Consultant are expected to identify and resolve the reason(s) for any discrepancies in close co-operation.

Service level monitoring will be performed by Client. Reports will be produced as and when required and forwarded to the Consultant.

1.3 Support Levels

The consultant must provide support and maintenance services during Support Levels mentioned below;

Support Level: **High**

Component/ Service	Core Components of Lanka Gate
Support Hours	24 hours a day, all days in the week (including public and mercantile holidays)

Support Level: **Medium**

Component/ Service 1	Government Interface and related backend services (deployed at Government organization site)
Support Hours	From 08:00 AM to 05:00 PM Monday to Friday (excluding public holidays)
Component/ Service 2	For front-end portlets and supporting back-end applications (web services, etc.. deployed at Lanka Government Cloud (LGC))
Support Hours	From 08:00 AM to 09:00 PM, all days in the week (including public and mercantile holidays)

1.4 On-Call Services Requirements

Consultant MUST make at least ONE qualified personnel available to the Client by telephone and email for the reporting and resolution of non-conformities or other issues, defects or problems. Dedicated telephone numbers and emails should be available for reporting issues. Client will nominate the personnel who are authorized to report non-conformities or other problems with the system from the departments. Reporting of non-conformities includes requests by the Client to apply critical software updates or patches.

Table-1 shows the response priority assigned to faults according to the perceived importance of the reported situation and the required initial telephone response times for the individual priority ratings. All times indicated represent telephone response time during specified Support Levels. The indicated telephone response time represents the maximum delay between a fault/request being reported and a Consultant's representative contacting the Client by telephone. The purpose of this telephone contact is to notify the Client of the receipt of the fault/request and provide the Client with details of the proposed action to be taken in respect of the particular fault/request.

Support Level	**Business Critical**	**Business Critical**	**Non-Business Critical**	**Non-Business Critical**
	Fatal	**Impaired**	**Fatal**	**Impaired**
High	60 minutes within Support Hours	90 minutes within Support Hours	90 minutes within Support Hours	120 minutes within Support Hours
Medium	120 minutes within Support Hours	150 minutes within Support Hours	150 minutes within Support Hours	180 minutes within Support Hours

Table-1: Response Priority

Note:
Fatal - Total system inoperability
Impaired - Partial system inoperability
Business Critical - Unable to perform core business functions
Non-Business Critical - Able to perform limited core business functions

Consultant notification can occur outside Support Level time, and thus the response may occur after the next Support Level begins. Furthermore, "Time to Arrive On-Site (Table-3)" starts from Support Level starting time and "Time to Resolve the Problem" is Support Level time starting from the actual time of arrival on site.

1.5 Problem Resolution and Penalties

If problems have not been corrected within two (2) hours of the initial contact, the Consultant shall send qualified maintenance personnel to the respective Client's site to take necessary actions to correct the issue reported (defect, problem or non-conformity).

If faults are not corrected within the time limits specified in the Table-2, the Client shall be entitled to a penalty payment for each hour that the Consultant fails to resolve the fault.

Maximum ceiling of penalty for a given month is 10% of the monthly support and maintenance price.

The time to arrive on-site is specified in the Table-3.

Support Level	**Business Critical**	**Business Critical**	**Non-Business Critical**	**Non-Business Critical**
	Fatal	**Impaired**	**Fatal**	**Impaired**
High	6 Hours LKR 15,000.00 per hour	10 Hours LKR 8,000.00 per hour	10 Hours LKR 8,000.00 per hour	15 Hours LKR 5,000.00 per hour
Medium	8 Hours LKR 10,000.00 per hour	12 Hours LKR 8,000.00 per hour	12 Hours LKR 8,000.00 per hour	20 Hours LKR 5,000.00 per hour

Table-2: Resolution Time and Penalties

Support Level	**Business Critical**	**Business Critical**	**Non-Business Critical**	**Non-Business Critical**
	Fatal	**Impaired**	**Fatal**	**Impaired**
High	Not applicable	Not applicable	Not applicable	Not applicable
Medium	2 Hours	3 Hours	3 Hours	5 Hours

Table-3: Time to arrive on-site

Annex C

Software Project Audit Process

Version 1.2

Information and Communication Technology Agency of Sri Lanka

1. Introduction

1.1 Purpose

Purpose of this document is to describe the Software Project Audit Process which is capable of auditing and ensuring the quality of different activities carried out throughout a software project life-cycle. The main purpose of this process is to provide much higher level of confidence in the quality of the deliverables received by the client from the developer. The quality level of the audited activity is presented using a measurement technique called metrics.

The process should be followed by both the development team and the Software Project Audit team to derive their own metrics to measure the quality status of a software product in its life cycle. Eventually, the trend analysis of such metrics can be used to identify any potential project issues or failures and to come up with solutions.

This document explains several guidelines which can be used within the audit process for project progress calculation and mapping payment milestones with project deliverables or and project artifact reviews to effectively manage the project.

Further, the document contrasts the Software Project Audit process from typical software development life cycle and illustrates how it has been automated by integrating several testing tools and testing methodologies as well as embedding best industry standards.

1.2 Scope

Scope of this document is to provide an insight about the Software Project Audit Process, importance of metrics, analysis of metrics, automated process of metric generation, skills required to generate certain metrics, guideline for project progress calculation, guideline for mapping payment milestones with deliverables and guideline for Review of Project artifacts.

1.3 Definitions, Acronyms and Abbreviations

Acronym	Definition
AQI	Architecture Quality Index
AD	Architectural Design
CQI	Code Quality Index
DD	Defect Density
DQI	Design Quality Index
DSI	Defect Severity Index
ISI	Issue Severity Index
PERI	Project Execution Readiness Index
RCI	Requirement Clarity Index
SPA	Software Project Audit
SR	Software Requirement
TTEI	Tasks Tracking Efficiency Index
TR	Transfer
UAT	User Acceptance Test
OAT	Operational Acceptance Test

2. Process Overview

It is often said that if something cannot be measured, it cannot be managed or improved. There is immense value in measurement, but you should always make sure that you get some value out of any measurement that you are doing.

What is a Metric?

It is a standard of measurement which can be used to measure the software quality. It gives a confidence in the software product. They are typically the providers of the visibility of the software product you need.

Why Measure?

When used appropriately, metrics can aid in software development process improvement by providing pragmatic, objective evidence of process change initiatives. Although metrics are gathered during the test effort, they can provide measurements of many different activities performed throughout a project. In conjunction with root cause analysis, test metrics can be used to quantitatively track issues from points of occurrence throughout the development process. In addition, when metrics information is accumulated, updated and reported on a consistent and regular basis, it ensures that trends can be promptly captured and evaluated.

What to Measure?

When considering the metric driven process, it can be divided into two parts. The first part is to collect data, and the second is to prepare metrics/charts and analyze them to get the valuable insight which might help in decision making. Information collected during the software development process can help in:

- Finding the relation between data points
- Correlating cause and effect
- Input for future planning

Normally, the metric driven process involves certain steps which are repeated over a period of time. It starts with identifying what to measure. After the purpose is known, data can be collected and converted into the metrics. Based on the analysis of these metrics appropriate action can be taken, and if necessary metrics can be refined and measurement goals can be adjusted for the better. Data presented by Development/testing team, together with their opinion, normally decides whether a product will go into client or not. So it becomes very important for Development team/test teams to present data and opinion in such a way that data looks meaningful

to everyone, and decision can be taken based on the data presented. Every software project should be measured for its schedule and the quality requirement for its release. There are lots of charts and metrics that we can use to track progress and measure the quality requirements of the release. In Figure 1.0 shows some of main metrics which can be derived at specific level of the software development life-cycle.

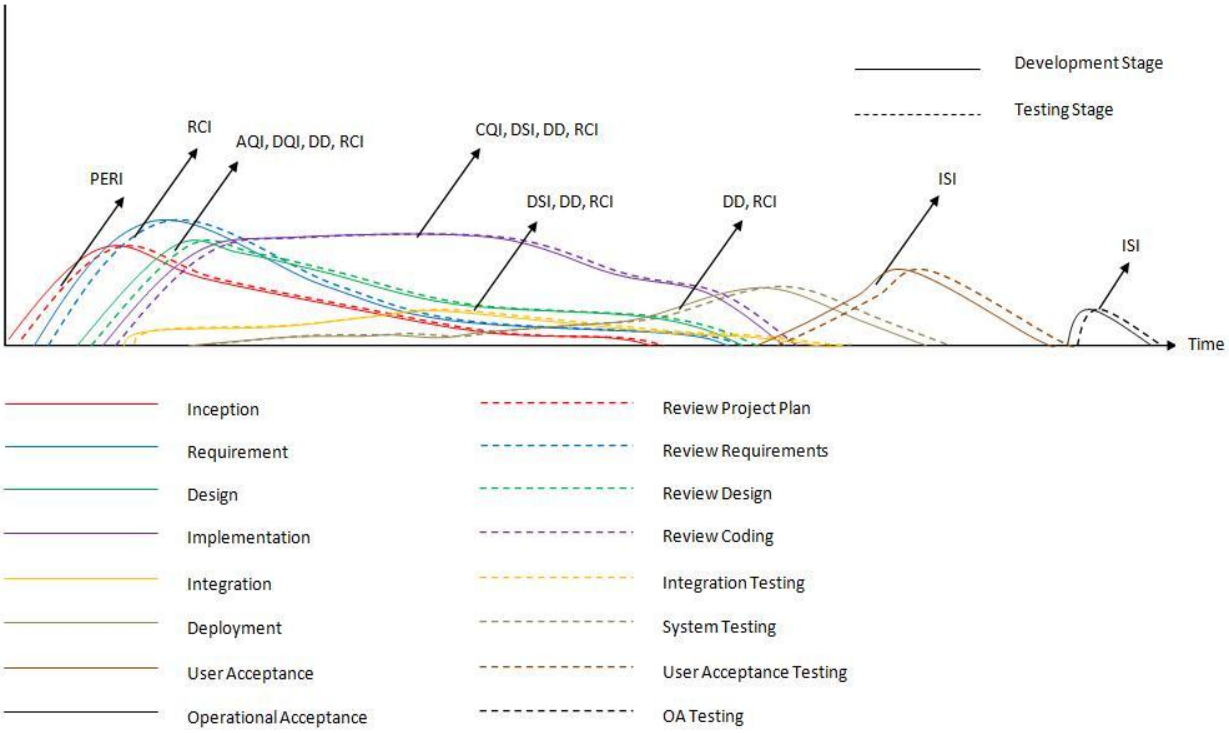


Figure 1.0 - Various Metrics derived at different levels of SD process

2.1 Metrics in Brief

Metric	Purpose
Project Execution Readiness Index (PERI)	This Proposed index at requirements stage is derived based on quality of the documents involve with this phase. The main Documents involve in this phase are; * User Requirements Document * Acceptance test plans * Project management plan for the SR phase * Configuration management plan for the SR phase * Verification and validation plan for the SR phase * Quality assurance plan for the SR phase When reviewing , reviewers can verify the document by checking its content with a checklist. Each of thesecontent in a checklist is categorized under their Severity to the System. All defects in those contents should be logged in a defect tracking system. Finally, index can be derived as; Weighted average of the total number of Open Issues in the product detected till date against all categories (Blocker (B), Critical (C), Major (Ma), Normal (N), Minor (Mi), Trivial(T)). Metric: $\frac{(B*162 + C*54 + Ma*18 + N*6 + Mi*2+ T)*10}{}$ Total weight (162+54+18+6+2+1) Note: Can be calculated based on the review cycles
Requirements Clarity/Change Index(RCI)	This index measures following two criteria relevant to requirements 1. Requirements Clarity This is the proposed index is at Specification Stage which should indicate how well each member of the Software development team comprehend the requirements and also indicates How well the requirements are cleared for Software Development Team. 2. Requirement Changes Requirement changes may be arisen at any stage of a project. Therefore, this index should be continued till UAT phase of a project and all the requirement changes arisen during that period should be captured under this index. The index indicates, weighted average of the total number of Open Issues

	in the product detected till date against all categories (Blocker (B), Critical (C), Major (Ma), Normal (N), Minor (Mi), Trivial(T)). Metric: $\frac{(B*162 + C*54 + Ma*18 + N*6 + Mi*2+ T)*10}{\text{Total weight } (162+54+18+6+2+1)}$ Note: Can be calculated based on the review cycles.
Architectural Quality Index (AQI)	Testing indicator for Architectural design level. The main documents of the AD phase are; *Architectural Design Document (ADD); *Software Project Management Plan for the DD phase (SPMP/DD) *Software Configuration Management Plan for the DD phase (SCMP/DD) *Software Verification and Validation Plan for the DD Phase (SVVP/DD) *Software Quality Assurance Plan for the DD phase (SQAP/DD) *Integration Test Plan (SVVP/IT) When reviewing , reviewers can verify the document by checking its content with a checklist. Each of these content in a checklist is categorized under their Severity to the System. All defects in those contents should be logged in a defect tracking system. Finally, index can be derived as; Weighted average of the total number of Open Issues in the product detected till date against all categories (Blocker (B), Critical (C), Major (Ma), Normal (N), Minor (Mi), Trivial(T)). Metric: $\frac{(B*162 + C*54 + Ma*18 + N*6 + Mi*2+ T)*10}{\text{Total weight } (162+54+18+6+2+1)}$ Note: Can be calculated based on the review cycles
Design Quality Index (DQI)	This is the Index proposed at Detailed Design Level. Should define a quality index (DQI) to measure and evaluate the quality of the Detailed Design based on the quality of the documents involve with the Detailed Design phase. The main documents of the AD phase are the; *Detailed Design Document (DDD) *Software User Manual (SUM) *Software Project Management Plan for the TR phase (SPMP/TR) *Software Configuration Management Plan for the TR phase (SCMP/TR) *Software Quality Assurance Plan for the TR phase (SQAP/TR)

	*Acceptance Test specification (SVVP/AT) When reviewing , reviewers can verify the document by checking its content with a checklist. Each of these content in a checklist is categorized under their Severity to the System. All defects in those contents should be logged in a defect tracking system. Finally, index can be derived as; Weighted average of the total number of Open Issues in the product detected till date against all categories (Blocker (B), Critical (C), Major (Ma), Normal (N), Minor (Mi), Trivial(T)). Metric: $\frac{(B*162 + C*54 + Ma*18 + N*6 + Mi*2+ T)*10}{\text{Total weight } (162+54+18+6+2+1)}$ Note: Can be calculated based on the review cycles
Code Quality index (CQI)	- Indicates how well the software codes are written and maintained. - To be derived using considering multiple aspects. This will be decided in project execution. - Index can be derived as; Weighted average of the total number of Open Issues in the product detected till date against all categories (Blocker (B), Critical (C), Major (Ma), Normal (N), Minor (Mi), Trivial(T)). Metric: $\frac{(B*162 + C*54 + Ma*18 + N*6 + Mi*2+ T)*10}{\text{Total weight } (162+54+18+6+2+1)}$
Defect Density (DD)	- Number of defects per unit size of the application (KLOC) - Calculated end of each drop cycle. - The Number of Known Defects is the count of total defects identified against a particular software entity, during a particular time period - Size is a normalizer that allows comparisons between different software entities (i.e modules, releases, products). Size is typically counted either in Lines of Code or Function Points.
Defect Severity Index (DSI)	- Indicates application stability - Weighted average of the total number of Open Defects in the product detected till date against all categories (Blocker (B), Critical (C), Major (Ma), Normal (N), Minor (Mi), Trivial(T)). Metric: $\frac{(B*162 + C*54 + Ma*18 + N*6 + Mi*2+ T)*10}{\text{Total weight } (162+54+18+6+2+1)}$

	Note: Calculated weekly and delivered by drop
Issue Severity Index (ISI)	During the User Acceptance Test(UAT) time issues can be arisen. All those issues should be logged in UAT documentation as well as in the bug tracking System. - Weighted average of the total number of Open issues in the product arisen during the UAT period against all categories (Blocker (B), Critical (C), Major (Ma), Normal (N), Minor (Mi), Trivial(T)). Metric: $\frac{(B*162 + C*54 + Ma*18 + N*6 + Mi*2+ T)*10}{\text{Total weight } (162+54+18+6+2+1)}$
Defect Category	ttribute of the defect in relation to the quality attributes of the product. Quality attributes of a product include functionality, usability, documentation, performance, installation, stability ,compatibility , internationalization etc. This metric can provide insight into the different quality attributes of the product. This metric can be computed by dividing the defects that belong to a particular category by the total number of defects.
Defect Cause Distribution Chart	This chart gives information on the cause of defects.
Defect Distribution Across Components	This chart gives information on how defects are distributed across various components of the system.
Defect Finding Rate	This chart gives information on how many defects are found across a given period. This can be tracked on a daily or weekly basis.
Defect Removal Efficiency	The number of defects that are removed per time unit (hours/days/weeks). Indicates the efficiency of defect removal methods, as well as indirect measurement of the quality of the product. Computed by dividing the effort required for defect detection, defect resolution time and retesting time by the number of defects. This is calculated per test type, during and across test phases.
Effort Adherence	As % of what is committed in contract. Provides a measure of what was estimated at the beginning of the project vs. the actual effort taken. Useful to understand the variance (if any) and for estimating future similar projects.
Number of Defects	The total number of defects found in a given time period/phase/test type that resulted in software or documentation modifications. Only accepted

	defects that resulted in modifying the software or the documentation are counted.
Review Efficiency	# of defects detected /LOC or pages reviewed per day
Test Case Effectiveness	The extent to which test cases are able to find defects. This metric provides an indication of the effectiveness of the test cases and the stability of the software. Ratio of the number of test cases that resulted in logging defects vs. the total number of test cases.
Test Case Execution Statistics	This metric provides an overall summary of test execution activities. This can be categorized by build or release, module, by platform (OS, browser, locale etc.).
Test Coverage	Defined as the extent to which testing covers the product's complete functionality. This metric is an indication of the completeness of the testing. It does not indicate any thing about the effectiveness of the testing. This can be used as a criterion to stop testing. Coverage could be with respect to requirements, functional topic list, business flows, use cases, etc. It can be calculated based on the number of items that were covered vs. the total number of items.
Test Effort Percentage	The effort spent in testing, in relation to the effort spent in the development activities, will give us an indication of the level of investment in testing. This information can also be used to estimate similar projects in the future. This metric can be computed by dividing the overall test effort by the total project effort.
Traceability Metric	Traceability is the ability to determine that each feature has a source in requirements and each requirement has a corresponding implemented feature. This is useful in assessing the test coverage details.
Scope Changes	The number of changes that were made to the test scope (scope creep). indicates requirements stability or volatility, as well as process stability. Ratio of the number of changed items in the test scope to the total number of items
Task Tracking Efficiency Index (TTEI)	This index indicates the average time taken to attend to general project tasks. TTEI = $\frac{\Sigma \text{ Time taken to attend task}}{\Sigma \text{ open task}}$

Table 1.0 – Metrics

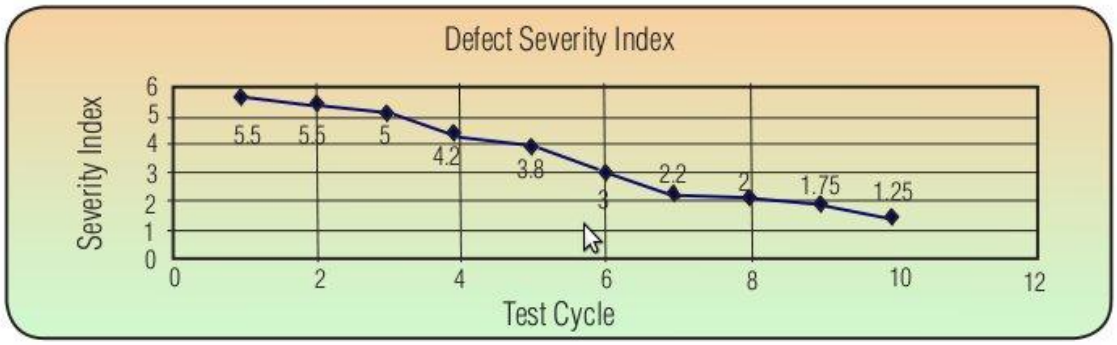
2.2 Metrics Analysis

Much as the time is spent gathering or maintaining metrics, enough time should be spent to review and interpret on a regular basis throughout the test effort, particularly after the application is released into production. During review meetings, the project team should closely examine all available data and use that information to determine the root cause of identified problems. It is important to look at several metrics, as this will allow the project team to have a more complete picture of what took place during a test.

Let's assume that as part of the SPA Process, the following metrics are collected by the SPA team.

Metric	Purpose
Defect Severity Index	Weighted average index of the Severity of defects. A higher severity defect gets a higher weight. S1 is a show stopper, S2 is high severity, S3 is medium & S4 is low. Ideally, this should slope down as test cycles progress.

For instance, if the test team has generated the following metrics:



a

t the graphs one can safely deduce the followings;

Defect Severity Index Trend:

What does the graph indicate? The defect severity index is sloping down consistently. This indicates an increasingly favorable trend. As the test cycle progresses (from cycle 1 to cycle 10), the severity index is sloping which suggests increasing quality of the application (as lesser number of critical and high severity defects are being reported).

This is what it could mean: While a fall in the defect severity index is definitely a good trend, looking at this index in isolation could be misleading. Following factors need to be considered in order to have a meaningful analysis.

Number of defects logged - let us consider an example where the test team executed two cycles of testing (assuming other things as constant). The number of defects logged against each of these cycles along with the calculated severity index is shown below.

Number of Defects

Defect Severity	Cycle 1(# of defects)	Cycle 2(# of defects)
s1	5	5
s2	10	15
s3	50	30
s4	100	100
Severity Index	1.52	1.50

At first thoughts, when we compare cycle 1's Severity Index with cycle 2's Severity Index, cycle 2 looks to be favorable (as the severity index is lower). If you go into the details of the number of defects logged and their severity, the picture turns out to be the opposite. While the total number of Severity 1 and Severity 2 defects for cycle 1 is 15, the number of Severity 1 and Severity 2 defects for cycle 2 is 20. In terms of quality, cycle 1 is better than cycle 2 as cycle 1 has lesser number of high severity defects (though the total number of defects logged in cycle 1 is more than cycle 2 defects and the severity index is greater than cycle 2 severity index). Test coverage has a similar impact. A lower test coverage coupled with reducing severity index would not be a healthy trend.

Severity of Defects

Defect Severity	Cycle 1(# of defects)	Cycle 2(# of defects)
s1	4	0
s2	4	0
s3	42	75
s4	27	2
Severity Index	1.81	2.03

- Defect Severity - let's consider another example where the test team executed two cycles of testing (assuming other things as constant). The severity of defects logged

against each of these cycles along with the calculated severity index is shown below.

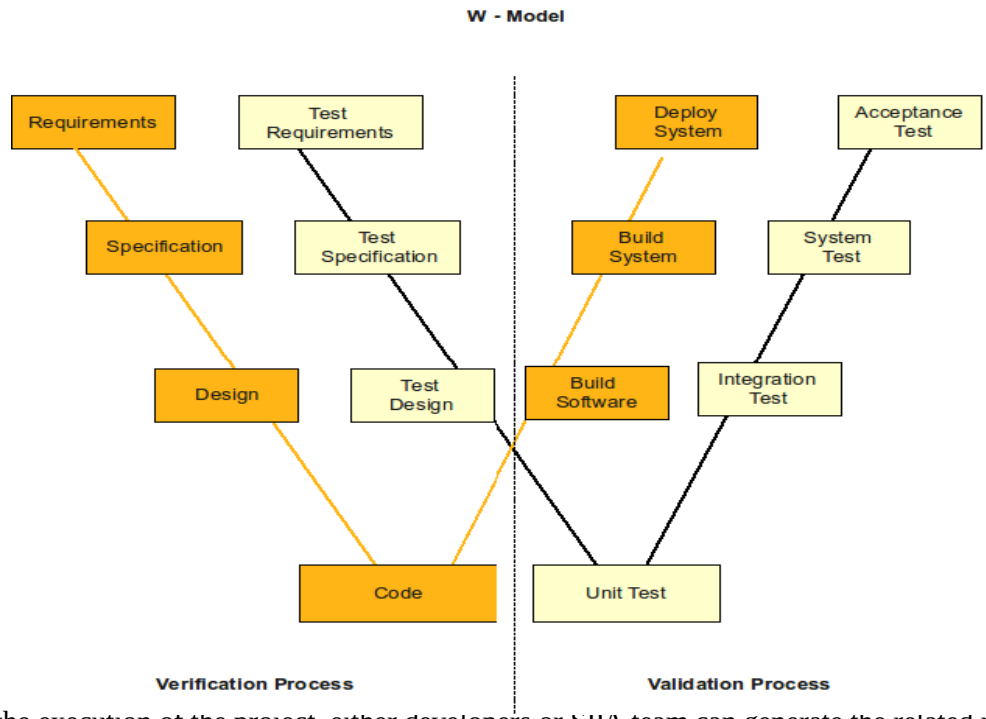
Looking at the severity index, it looks like cycle 1 is better than cycle 2 (as the severity index is low for cycle 1 compared to cycle 2). However, cycle 2 is better than cycle 1 as total number of Severity 1 and Severity 2 defects is zero compared to a total of 8 severity 1 and severity 2 defects of cycle 1. Just because the severity index is low, do not believe the quality of the application is better than the earlier cycle.

3. Process Automation

In following section describes about the testing methodologies, process and tools to be used while automating the typical software development life-cycle in order to deriving the metrics.

3.1 Testing Methodology

According to the Automated testing process, every development activity is mirrored by a test activity. The testing process follows a well-proven testing methodology called W-model. Following Figure-2.0 explains, the way of testing activities of W-model involve with the standard software development life-cycle.



While the execution of the project, either developers or SPA team can generate the related metrics.

3.2 Skills required to generate Metrics

During the different stages of a software project, several roles and parties will be involve with development, reviewing and testing activities. In Figure 3.0 shows the different stages of a software project, the main activities which should perform during those stages, the roles/parties should involve and the metrics which derive and maintain in those stages.

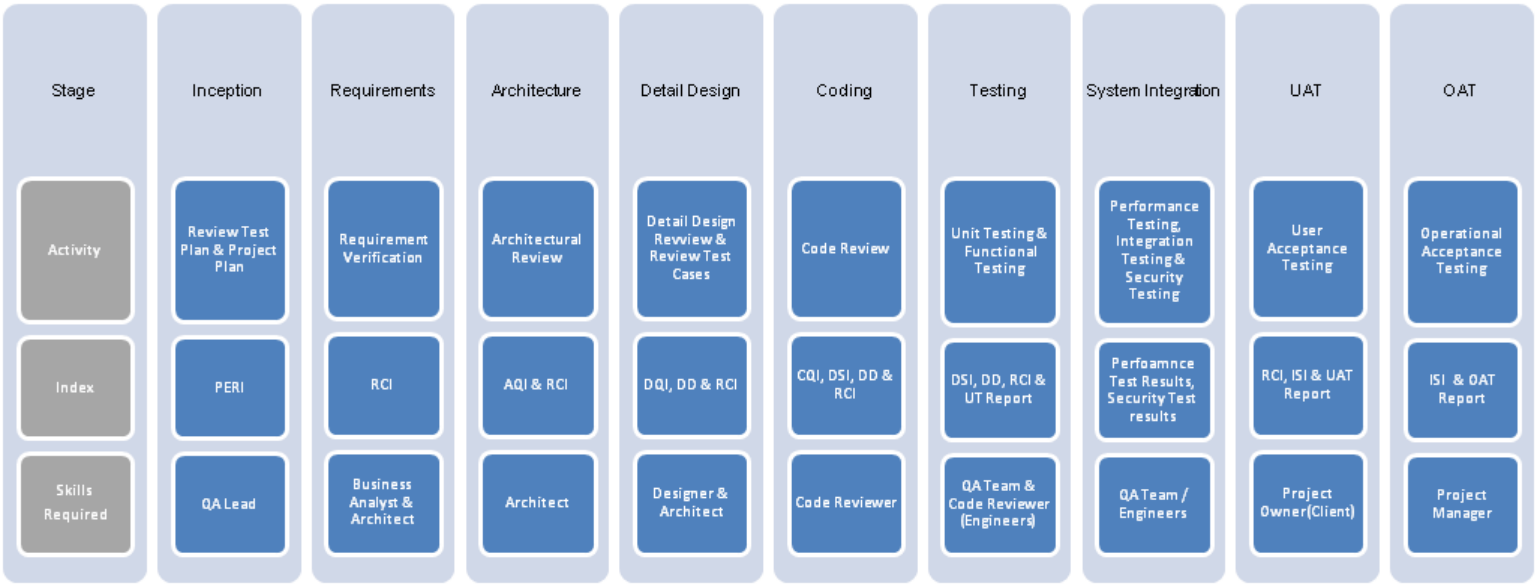
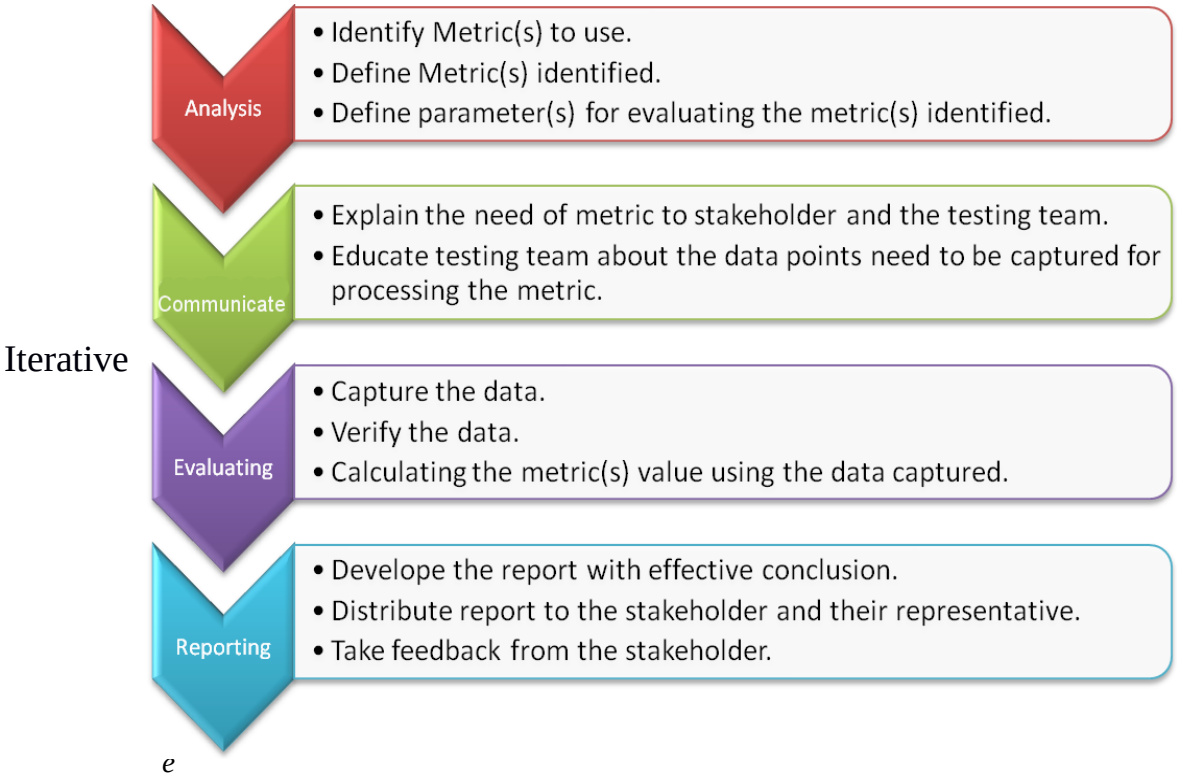


Figure 3.0 - Skills required to generate Metrics

3.3 Process of Setting-up a Metric

The Figure-4.0 explains the life-cycle of a Metric or the process involved in setting up the metrics:



e

trics Life-Cycle

When implementing this process, several testing tools and techniques will be used along with the automated testing process in order to generating, maintaining and evaluating the metrics derived at specific level of the Software development life-cycle.

3.4 Integration of testing tools/process

Below you find a list of tools /process which will be used when automating the typical SD life-cycle suits to the Software Project Audit Process.

- **Fagan inspection -** Fagan Inspection defines a process as a certain activity with a pre-specified entry and exit criteria. Activities for which Fagan Inspection can be used are:

1. Requirement specification
2. Software/Information System architecture (for example DYA)
3. Programming (for example for iterations in XP or DSDM)
4. Software testing (for example when creating test scripts)

- **Cruise Control –** It is both a continuous integration tool and an extensible framework for creating a custom continuous build process. It includes dozens of plug-ins for a variety of source controls, build technologies, and notifications schemes including email and instant messaging. A web interface provides details of the current and previous builds.

- **Bug-zilla -** It is a Web-based general-purpose defect tracking and testing tool.

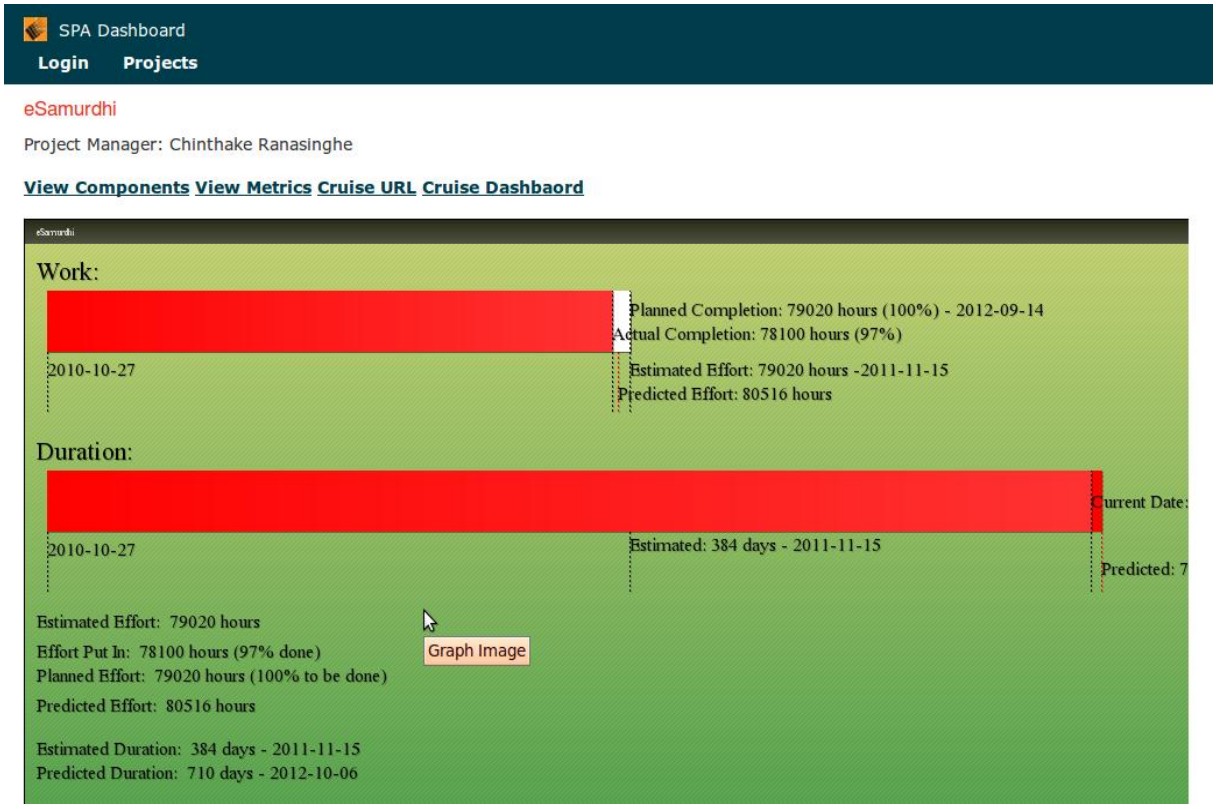
- **SVN -** It is a revision control system which use Subversion to maintain current and historical versions of files such as source code, web pages, and documentation.

- **Git -** Git is a **free & open source, distributed version control system** designed to handle everything from small to very large projects with speed and efficiency.

- **SCM -** For Configuration identification and Identifying configurations, configuration items and baselines. Also for Configuration control ,Configuration status accounting and Configuration auditing

3.5 Displaying Metrics –The Dashboard

The Dashboard is the interface to help project teams to visualize their project statuses by several indexes. And also it could be used to displaying the test results of specific tests carried by the SPA team who responsible for the given project. As an example; in Figure 5.0 displays the current status of the project with its estimated effort against



the predicted effort.

Figure 5.0 – SPA Dashboard view