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

EFFECTIVE SYSTEMS ANALYSIS AND DESIGN GUIDE
[G61]

Version: 1.2

Jul 2020

Distribution	
Copy No.	**Holder**
1	Office of the Government Chief Information Officer

Amendment History				
Change Number	**Revision Description**	**Section Affected**	**Rev. Number**	**Date**
1	As detailed in 1.01 to 1.14		1.1	December 2016
1.01	Describe the various development approaches (including Agile) and their impacts to various deliverables	Preface - (c); 3.1(d), 4.2(d), 5.2(d), 5.3(b), 5.4(c)		
1.02	Add description of appendices	Preface - Structure of the Guide (b)(iii) *(New)*		
1.03	Revise legends in the figure of "Summary of recommended SA&D Phase and Deliverables"	Figure 1		
1.04	Revise the legends of key task and deliverables figures	Figures 2, 3, 4, 5		
1.05	Separate Appendix B into Appendix B and Appendix C; Under sections of "Tools and Templates", include references of related template of Appendix C.	2(v); 3.5(b), 4.2.4(b), 4.3.4(b), 5.2.4(b), 5.3.4(b), 5.4.4(b), 6.2.4(b), 6.3.4(b), 6.4.4(b), 6.5.4(b), 6.6.4(b)		
1.06	Rename "Current Business Model" to "Current Business Process"	Figure 2; 3(d)(ii); Figure 6; 3.2.2, 3.2.2(a); Table 3 - 1(d), 3(ii), 4(d); Table 4 – 3, 3(a), 3(b)(i), 3(c), 5(a); Table 7 - 1(b), 1(c)(i); Table 31 - 1(b)(iii)		

Amendment History				
Change Number	**Revision Description**	**Section Affected**	**Rev. Number**	**Date**
1.07	Revise the icons of “Cannot be Simplified” from  to 	Figures 6, 7, 8		
1.08	Define when to identify the current business process	Table 5 - (c)		
1.09	Include the State Diagram in “Tools & Templates“ column of relevant tables of “Key Tasks and Deliverables”	Table 7 - 1; Table 19 – 1, 2		
1.10	Include the State Diagram in relevant sections of “Tools and Templates”	4.2.4(a)(x); 5.4.4(a)(viii)		
1.11	Define the task of updating the Requirements Traceability Matrix (RTM) in each deliverable	Table 7 - 2(g), 3(c); Table 10 - 1(f)		
1.12	Add the common timeframe estimation methods for reference	Table 34 - 1(c)		
1.13	Add the “Reference” section	Reference *(New)*		
1.14	Rename the section name of “Exhibit” to “RACI Matrix” and move the whole section to Appendix B	Remove the “Exhibit” of v1.0		
2	Update the URL of “Practice Guide for Developing Mobile Apps” in Reference section	Updated the URL	1.2	Jul 2020

Table of Contents

List of Figures & Tables

CONVENTIONS

Table 1 - List of Acronyms used throughout the Effective Systems Analysis and Design Guide

Abbreviation	Full Name
API	Application Programming Interface
B/D(s)	Bureaux and Departments
BSO	Business System Option
CED	Current Environment Description
CIG	Central Internet Gateway
CMS	Content Management System
CPU	Central Processing Unit
DAS	Direct Attached Storage
DBMS	Database Management System
DiD	Defense-in-Depth
DFD	Data Flow Diagram
DITP	Departmental Information Technology Plan
DR	Disaster Recovery
EGIS	e-Government Infrastructure Service
FPA	Function Point Analysis
FS	Feasibility Study
GCN	Government Communication Network
GNET	The Government Backbone Network
GUI	Graphical User Interface
HKSAR	Hong Kong Special Administrative Region
IAM	Identity and Access Management
I/O	Input / Output
ISSS	Information Systems Strategy Study
IT	Information Technology
ITMU	Information Technology Management Unit
MPLS	Multiprotocol Label Switching
N/A	Not Applicable
NAS	Network Attached Storage

Abbreviation	Full Name
OGCIO	The Office of the Government Chief Information Officer
OS	Operating System
PAT	Project Assurance Team
PM	Project Manager
PMP	Project Management Plan
PSC	Project Steering Committee
RACI	R- Responsible; A - Accountable; C - Consulted; I – Informed
RTM	Requirements Traceability Matrix
SAN	Storage Area Network
SA&D	Systems Analysis and Design
SDLC	System Development Life Cycle
SME	Subject Matter Expert
TSO	Technical System Option
UAT	User Acceptance Testing
UI	User Interface
UML	Unified Modelling Language
URD	User Requirements Document
VPN	Virtual Private Network
WAN	Wide Area Network
WSDL	Web Services Description Language

PREFACE

(a) Since the 1990s, the Government has progressively adopted a number of methodologies for delivering IT systems. These methodologies have been adopted for more than 10 years and parts of them have over time become less capable of coping with the current technological and business environment. Difficulties are often encountered by IT project teams in applying these methodologies effectively in view of rapidly evolving technologies and the dynamic business environment.

(b) The Effective Systems Analysis and Design Guide ("the Guide") serves as a self-contained document providing a set of comprehensive steps to enable the effective conduct of Systems Analysis and Design (SA&D). It makes reference to local and international industry best practices and is tailored in accordance with HKSAR Government's environment and merits of the methodologies that have been adopted in Government IT projects.

(c) The Guide is suitable for adoption for different project natures and development approaches such as waterfall, iterative, incremental, Agile or a combination of approaches. The project manager should select the appropriate development approach according to the characteristics of the project such as user requirements, familiarity with the proposed technology, project complexity, timeframe and so on. Agile software development practices for conducting SA&D are also considered and included into the Guide wherever appropriate. More information for the use of Agile development approach in SDLC can be found in the document "*Practice Guide for Agile Software Development*[1]" published by the Office of the Government Chief Information Officer (OGCIO).

Scope and Target Readers of the Guide

(a) The Guide is to provide guidance for SA&D stage of the System Development Life Cycle (SDLC). The target readers of the Guide include the following parties who participate in and contribute to IT system development projects, but are not limited to:

i) Business Analysts

ii) Systems Analysts

iii) Systems Architects

iv) Internal Project Managers (Internal PM)

v) Project Steering Committee (PSC) chairman and members

vi) Project Assurance Team (PAT) chairman and members

vii) Project Owner

(b) For a small project team, the responsibilities of the Systems Architects may be fulfilled by the Systems Analysts.

(c) This Guide only covers the SA&D phase of the SDLC. Nevertheless, readers may make reference to the relevant parts of this Guide as appropriate in conducting Feasibility Study.

Structure of the Guide

(a) The Guide is structured according to reference cases from local and international best practices, adapted from existing system development methodologies of delivering IT systems and incorporating relevant elements of Agile methodology.

(b) The Guide is divided into two parts and three appendices :

i) Part I - The SA&D Methodologies

Gives an overview of and describes the activities involved in each key process, and highlights the key task flow and deliverables of that process.

ii) Part II - The SA&D Deliverables

Describes in greater detail the key steps and tasks required in order to achieve the deliverables of that process. The Part is designed in such a way that enables readers to identify easily what to do and deliver in each process.

iii) Appendices

Tools, templates, checklists and sample documents are provided in the following appendices for reference -

- Appendix A – Tools
  - Provides a list of tools, techniques, checklists and diagrams.
    Examples include prototyping techniques, cost / benefit evaluation, action checklist, use case diagram, etc.
  - The tools provided are based on commonly used industry standards (e.g. UML, TOGAF) and industry best practices.
- Appendix B – RACI Matrix, checklists and RTM
  - Provides a RACI matrix, decision checklist and RTM sample.
- Appendix C – Templates
  - Provides a sample template of the SA&D Report which includes templates for each process (i.e. Current Environment Description, Requirements Specification, System Specification, and Technical System Option).

.

PART I - THE SA&D METHODOLOGIES

1 THE SA&D PHASE DESCRIPTION

(a) Part I will describe the Requirements and the Analysis & Design stages as illustrated in ***Figure 1***, highlighting the key processes within them.

(b) The process flow presented in the figures is not necessarily sequential and processes can be recursive.

(c) The objective of the SA&D phase is to capture requirements, model business processes, define system architecture and design the solution. Additionally, the strategy to test and deploy the solution is created during the SA&D phase. Technical feasibility of the project is assessed against the business goals and objectives, before the end of the SA&D phase.

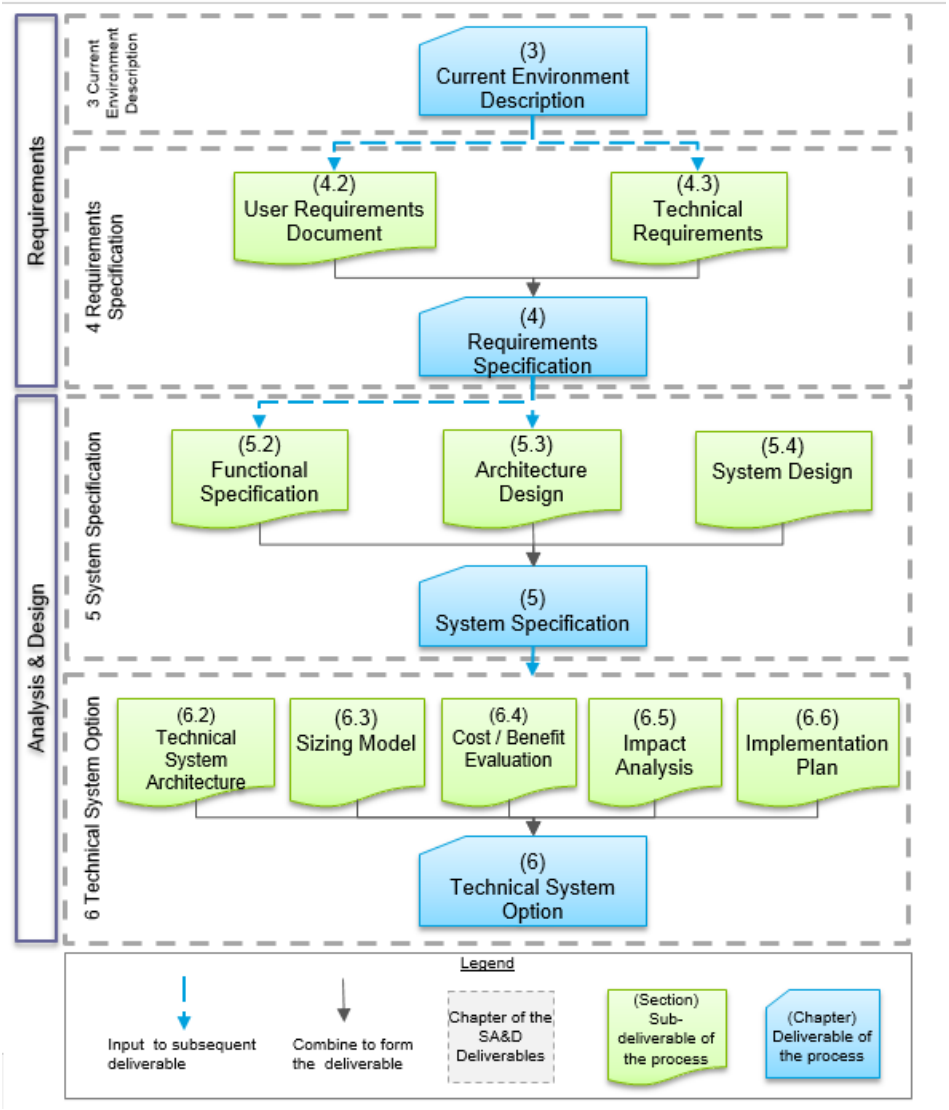

Figure 1 - Summary of recommended SA&D Phase and Deliverables

1.1 REQUIREMENTS STAGE

The Requirements stage includes the modelling of current and future business processes (***Figure 2***), as well as capturing and eliciting functional and non-functional requirements (***Figure 3***). The objective of the Requirements stage is to identify the business function areas and corresponding business process models to understand the client's current state of the people, processes, and technology (systems) interactions and then analyse aforementioned processes and requirements as well as evaluate assumptions, critical success factors, inefficiencies and problems. As some of the tasks in the Requirements stage are to be performed or assisted by Business Analysts, more information on how Business Analysts will participate and be involved in IT system development projects can be found in the document "*Best Practices for Business Analyst*[2]" published by OGCIO.

Key Activities

The following ***Figures 2 and 3*** illustrate the key tasks and deliverables performed during the Requirements stage:

i) Current Environment Description (CED)

ii) Requirements Specification

- User Requirements Document (URD)
  - Future Business Process
  - Functional Requirements
  - Non-functional Requirements
- Technical Requirements

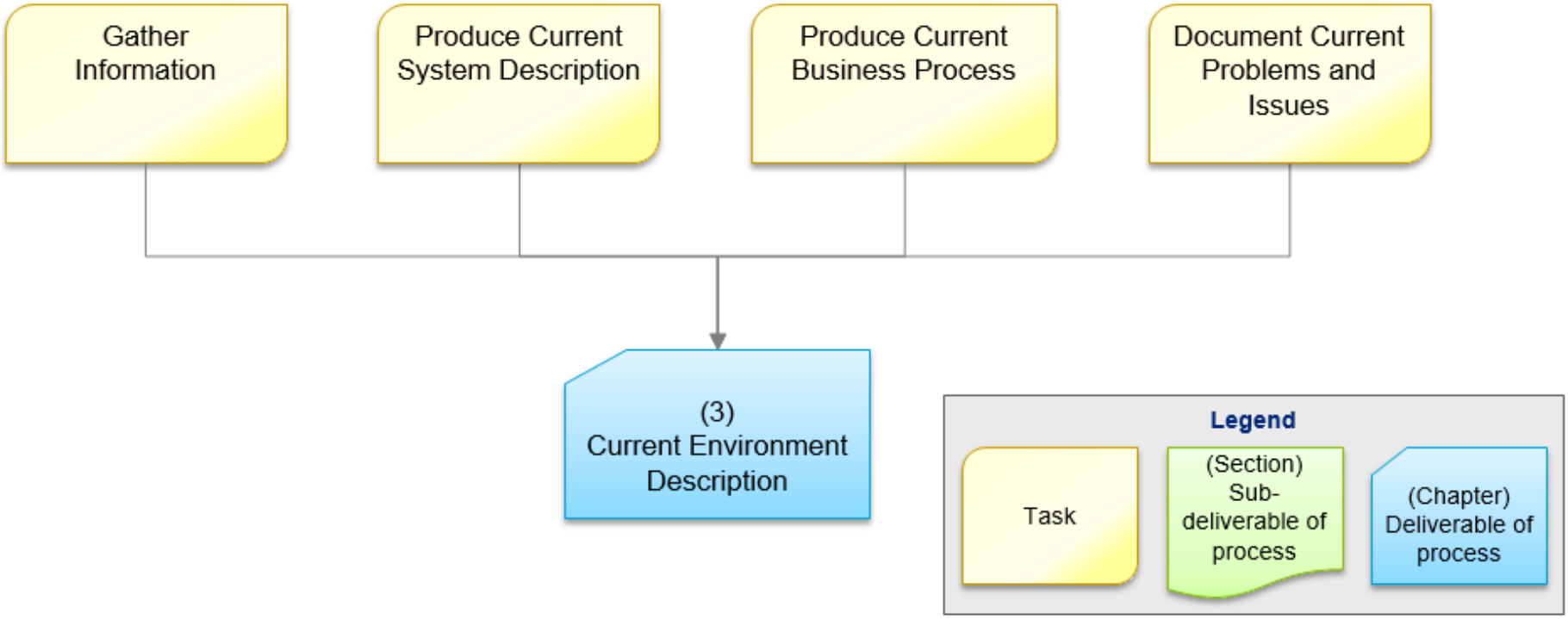

Figure 2 - Summary of CED key tasks and deliverables

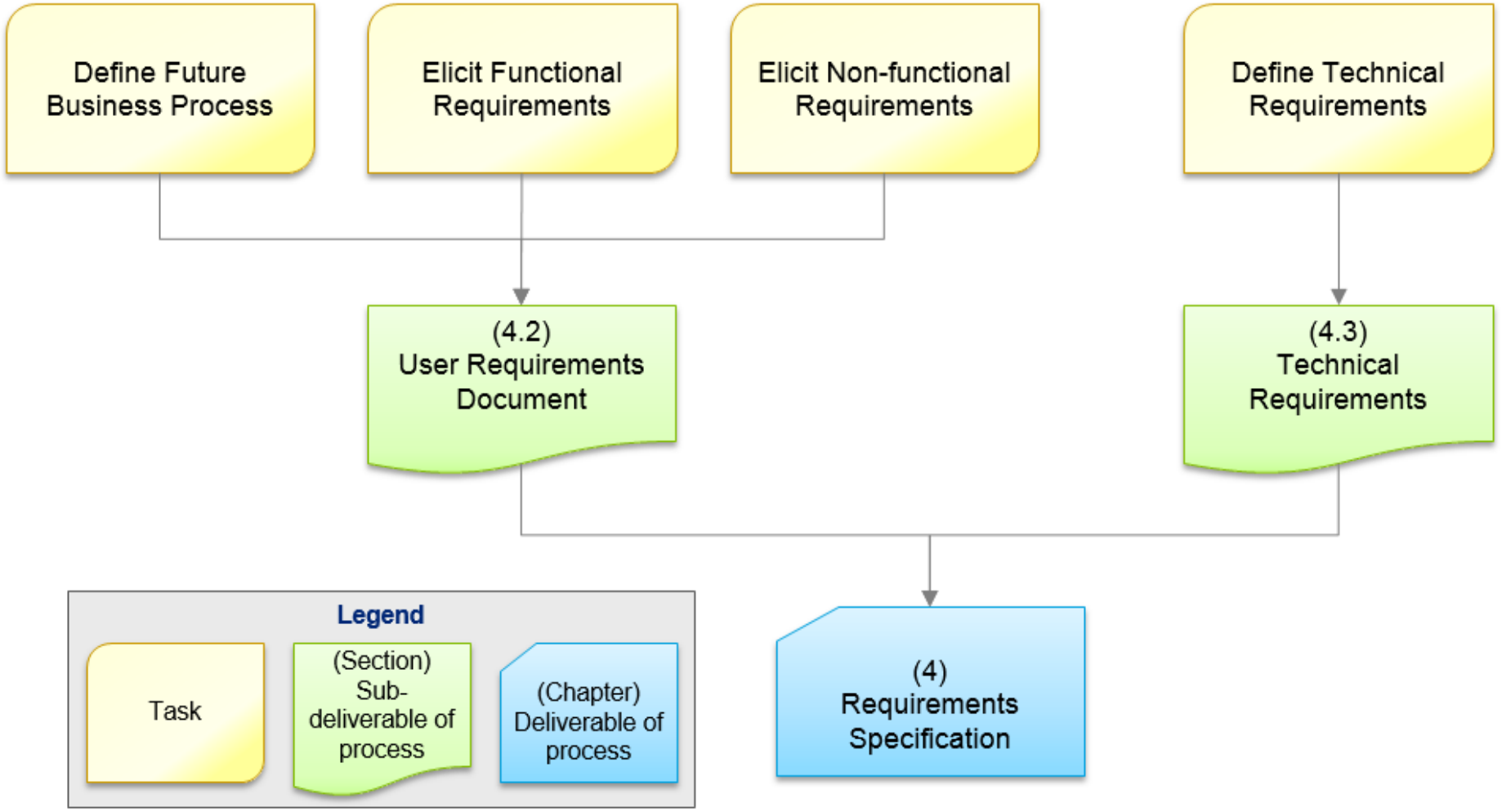

Figure 3 - Summary of Requirements Specification key tasks and deliverables

1.2 ANALYSIS & DESIGN STAGE

(a) The Analysis & Design stage includes a description of the overall high level system architecture and the Application Design.

(b) To perform these processes, architectural decisions must be made with an understanding of the whole system: its scope, major functionality and non-functional requirements such as performance requirements and technical requirements. The activities in Analysis & Design stage help to ensure that the architecture, requirements and implementation plans are accurately recorded and the risks are sufficiently mitigated. They also determine the cost and schedule for the completion of the development.

Key activities

The following ***Figures 4 and 5*** illustrate the key tasks and deliverables performed during the Analysis & Design stage:

i) System Specification

- Functional Specification
- Architecture Design
- System Design

ii) Technical System Option (TSO)

- Technical System Architecture
- Sizing Model
- Cost / Benefit Evaluation
- Impact Analysis
- Implementation Plan

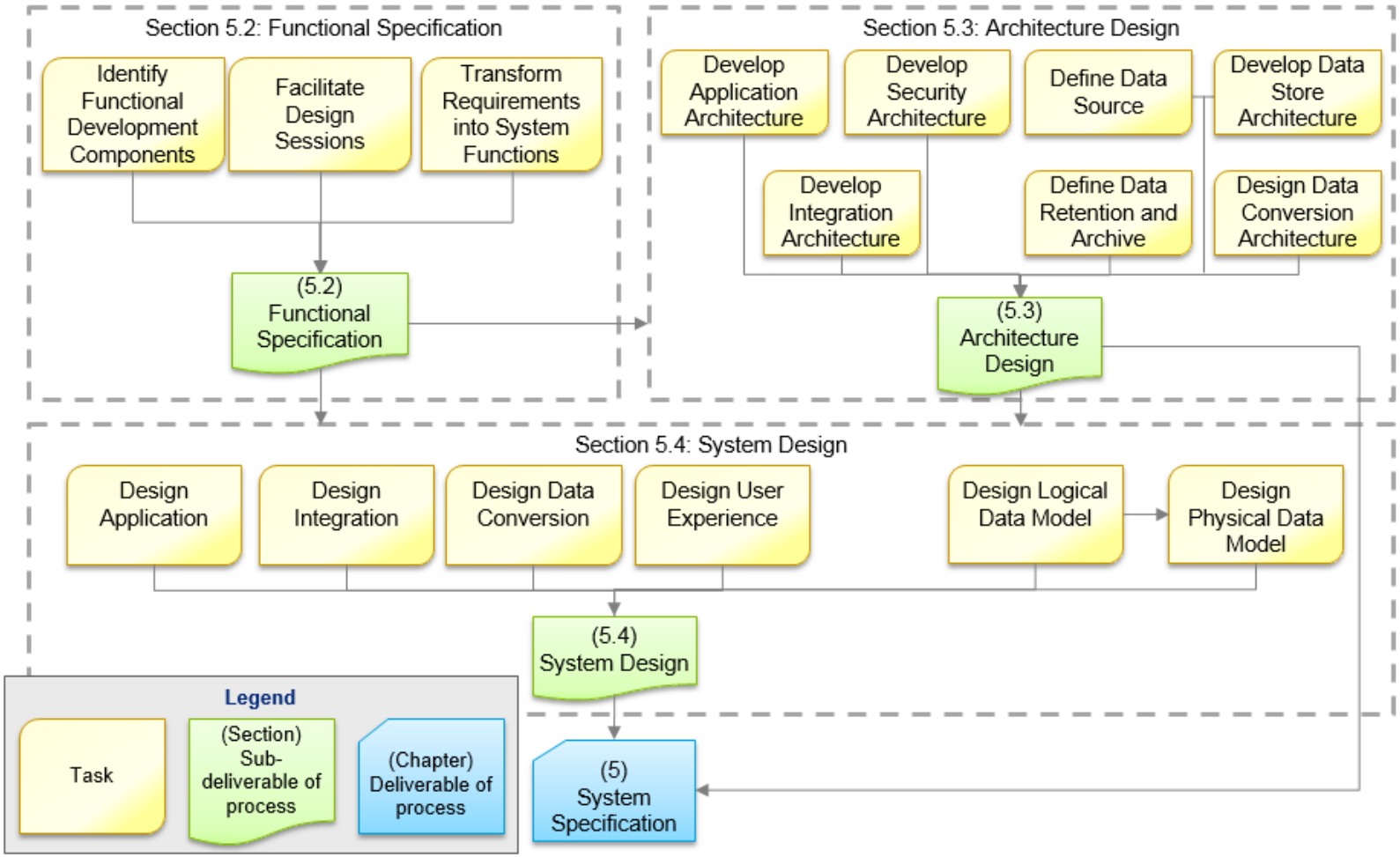

Figure 4 - Summary of System Specification key tasks and deliverables

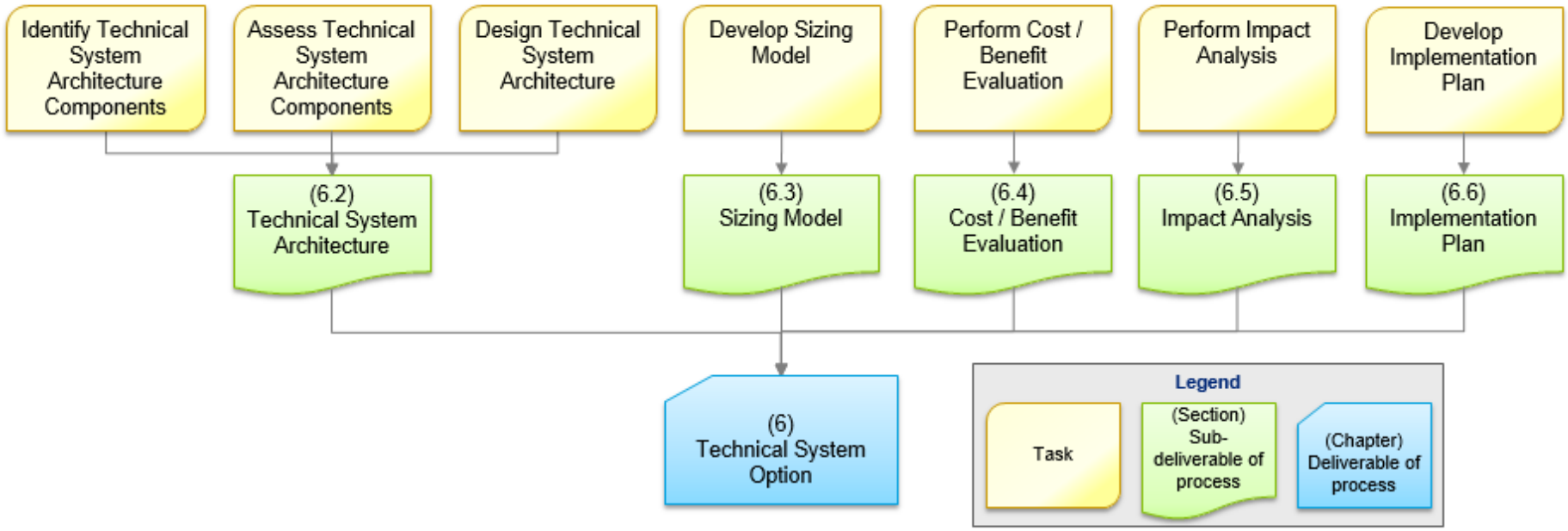

Figure 5 - Summary of TSO key tasks and deliverables

PART II - THE SA&D DELIVERABLES

2 HOW TO USE THIS PART

The Part is designed in such a way that enables readers to identify easily what to do and deliver in each process. For each process under this Part, the following information is documented, as applicable:

i) Flexible Adoption of processes

- Some of the SA&D documentations (such as Current Environment Description and the processes within System Specification) can be suitably adopted and simplified. With reference to the prevailing industry practices globally and an analysis of the existing SA&D practices in the HKSAR Government, a set of factors for consideration were derived for the users of this Guide on their adoption of individual components of the SA&D practices in their respective activities. The following consideration factors will be captured under “Flexible Adoption of processes” and “Hints & Tips” headers.

Table 2 - List of Key Consideration Factors

Factors to be considered	Description	Conditions affecting the decision for adopting deliverables
Project Scale and Complexity[a]	Whether the project is considered as large/small and its level of complexity	(a) For large scale / highly complex projects, all tasks have to be conducted and documented in detail. (b) For small scale / less complex projects, Current Environment Description can be simplified under certain conditions. Other conditions will be required for consideration for System Specification and Technical System Option[b].
Project Type	Whether the project involves: - Building a new system - System enhancement - System replacement	(a) For the development of a new system, Current Environment Description can be simplified under certain conditions, while Systems Specification and Technical System Option should normally be required except under special conditions. (b) New system includes system for new business/operation. (c) For system enhancement, level of detail per task can be flexibly adapted according to certain conditions. (d) System replacement includes replacing the current IT system or replacing the current manual system or semi-automated

[a] Please refer to the "*Practice Guide for Scoping and Planning of Large-scale IT System Development Projects*[3]" for project scale and complexity.
[b] Specific conditions shall be further elaborated in the Hints & Tips sections of the Effective SA&D Guide.

Factors to be considered	Description	Conditions affecting the decision for adopting deliverables
		system, e.g., spreadsheets software, where all tasks have to be conducted thoroughly.
System Dependencies	Whether there are multiple system interfaces within or across the department	Affects the technical design, where the tasks on integration within Application Architecture and System Design can be simplified if there is no or few system interfaces required.
Stakeholder Complexity	Whether the project involves multiple groups or large number of stakeholders	(a) Generally, stakeholder complexity does not directly affect the way the SA&D practices are to be adopted. (b) However, if the project involves only small groups or small number of stakeholders, it would be possible to adopt an iterative / Agile approach where it requires heavy stakeholder involvement.
Solution Approach	Whether the project is to deliver a packaged solution, custom-built system or Mobile application	(a) Affects the adoption of deliverables within Technical System Option. (b) If the project is to deliver a packaged software, Cost / Benefit Evaluation in Technical System Option processes can be optional.
Documents up-to-date	Whether there is existing documentation from past projects	Up-to-date information is available from existing system or Feasibility Study (FS)/ Information Systems Strategy Study (ISSS)/ Departmental Information Technology Plan (DITP) conducted within a reasonable timeframe.

Factors to be considered	Description	Conditions affecting the decision for adopting deliverables
Technical System Option Already Confirmed	Whether the technical system has already been selected	The Technical System Option was specified in the project tender and/or the contractor team has already recommended a Technical System Option in proposal.
Client App (UI Only)	Whether the project is only to build the user interface (UI) of mobile application without backend system or the project also includes the backend system	(a) Affects the adoption of deliverables within System Specification and Technical System Option. (b) If the project is only to build the UI of mobile application, System Specification and Technical System Option, except implementation plan, can be optional.

- This provides the users of the Guide with a set of key consideration factors on whether relevant processes can be simplified or not.
- A Decision Checklist is provided to facilitate this tailoring process for analysing the consideration factors (e.g., project scale, project type, etc.) and documenting the outputs (e.g., final tailoring decisions made) in a structured manner. Please refer to ***Appendix A - Tool 26*** for details and ***Appendix B – Decision Checklist*** for the sample template.
- A decision tree (where applicable to the process) is provided to facilitate the consideration of whether and which deliverables can be simplified.

ii) Overview & Approach

Provides a high level overview of the process and the objectives of performing it.

iii) Key Roles & Responsibilities

Provides the roles, for example Business Analyst and Systems Analyst, and associated responsibilities involved in the process. The responsibility assignment matrix to describe the participation by various roles in completing tasks or deliverables can be found in ***Appendix B – RACI Matrix***.

iv) Key Tasks & Deliverables

Provides a list of deliverables and associated responsibilities of the people who will be involved in the process. An example of deliverable would be the Current Environment Description and the associated tasks include identifying existing IT applications and interviewing business users to understand their current business processes.

v) Tools & Templates

- Provides a list of available tools, techniques, templates and checklists that would assist and speed up the process in developing the deliverable. Project team can flexibly adopt ALL or SOME of the Tools and Templates provided in the Guide, or adopt other tools such as text writing, which can fulfil the same purpose, depending on their skills, development tools, development language, software availability, users' preference, and format and manner of the existing documents, etc. For example, the Guide lists out Use Case Diagram and Activity Diagram tools for Current Environment Description. Users can choose to use the listed tools to convey the information as they see fit.
- The Tools & Templates are documented in ***Appendix A*** and ***Appendix C*** documents with further elaboration on the usage.

vi) Hints & Tips

Provides a list of useful hints and tips that could enhance the project delivery efficiency or quality of the deliverables based upon past project experiences and industry practices.

3 CURRENT ENVIRONMENT DESCRIPTION

(a) The CED process allows the project team to identify the current problems and issues, and understand the business activities and systems within the existing environment. This process would facilitate the project team in performing subsequent processes such as the System Specification and TSO.

(b) The information captured in the CED should be purely on the existing environment; any information/activities related to the future system should be performed at later stages.

(c) Please refer to ***Figure 2***, which summarises the key tasks and deliverables produced under CED process.

(d) The scope of the CED includes:

i) Current System Description

ii) Current Business Process

iii) Current Problems and Issues

3.1 FLEXIBLE ADOPTION OF PROCESSES WITHIN CED

(a) The decision tree as shown in ***Figure 6*** is a recommendation on under what consideration factors a process within the CED could be simplified.

(b) For the detailed explanation of the consideration factors, please refer to the "***Part II - How to Use this Part***"

(c) Note that the CED can be omitted if entirely new services are to be provided, such that the study of the current environment is not necessary.

(d) Other than the consideration factors described in the decision tree, the development approaches adopted such as iterative and Agile may also be the factors to be considered. For example, if Agile approach is adopted, the CED can be created at a high-level as long as it is sufficient to start user requirements capturing or even omitted depending on the project team requirements.

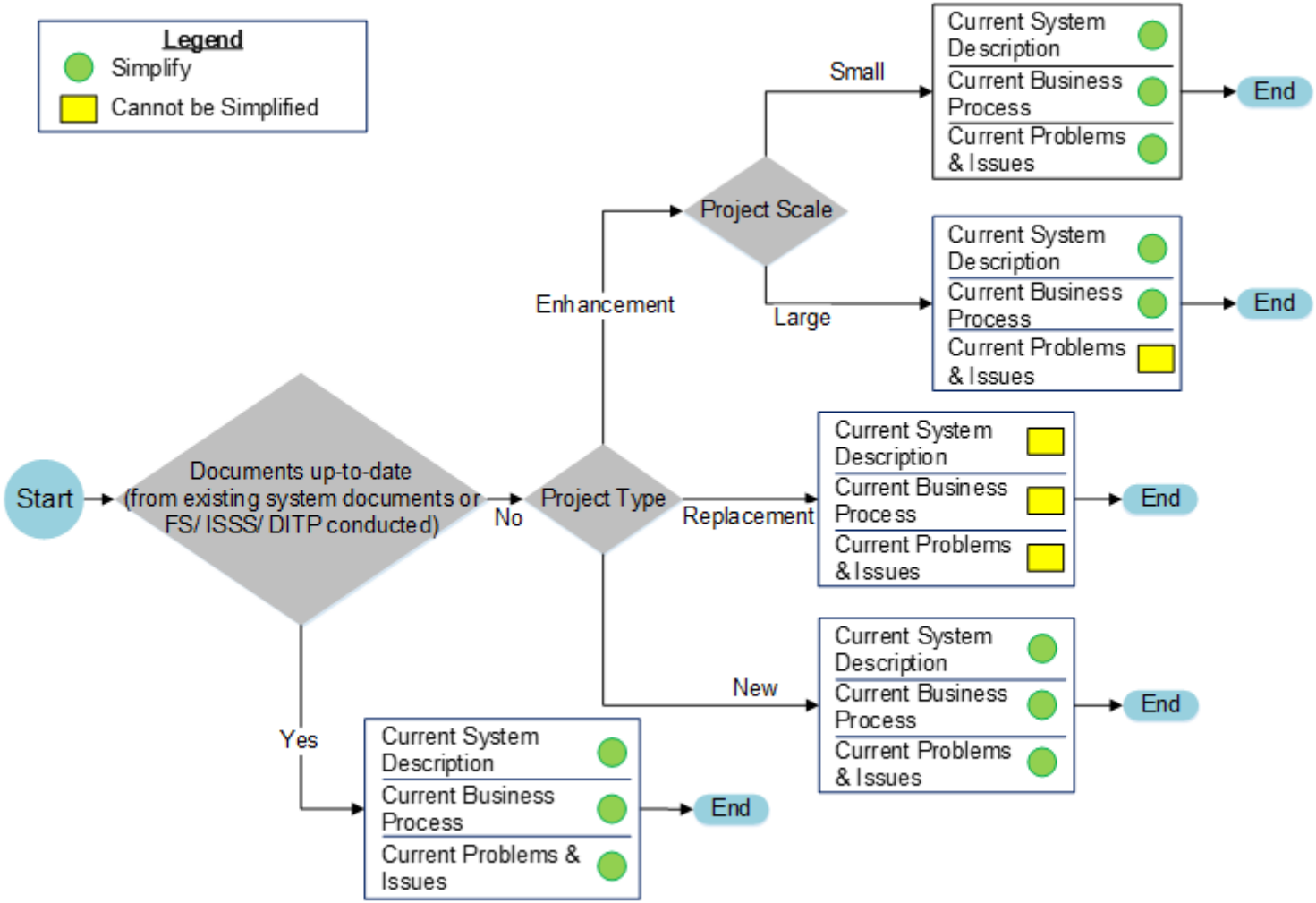

Figure 6 - Decision Tree for CED

3.2 OVERVIEW & APPROACH

3.2.1 Current System Description

(a) The Current System Description is to identify and document all the related current system information such that the project team has a clear picture of the current system settings and configurations.

(b) The following information is recommended to be documented for each system:

i) Current System Overview

- Show the interactions between the system and the users with the high level diagram. The high level diagram should also highlight any interfaces or interactions between the system and external systems/parties (please refer to ***Tools & Templates in section 3.5*** for details).
- Describe the current system name, current system description/objectives, business area served, users and current system functions.

ii) Current System Hardware, Software and Network

- Describe the current system's hardware and software configuration, if any.
- Document the Disaster Recovery (DR) configuration, if any.
- Describe and provide a high level system network diagram, if any.

iii) Volumes and Frequencies

- Summarise the data volumes and frequencies of processes of the current system.

iv) Interface with Other Systems

- Describe the current system interfaces.

(c) The Current System Description should also capture any related manual system and semi-automated system in use currently, e.g., spreadsheet software. Information about the manual system can be presented in the forms of "Current System Overview - High Level Diagram" and "Volumes and Frequencies".

3.2.2 Current Business Process

(a) The Current Business Process is to model all related current business activities. This process helps the project team to understand how the organisation currently functions. It is recommended to use Process Diagrams such as Flowchart Diagram, Activity Diagram and Use Case Diagram, to depict the current business processes.

(b) For complex business process, the process could be broken down into smaller processes, each with a separate diagram. There should also be a single diagram showing an overview of the whole business function.

(c) The major elements that need to be captured in the Process Diagrams are:

i) Event

- Describe the external work that triggers the project's business process (e.g., order received, end-of financial period).

ii) Activity

- Describe the kind of work that needs to be performed.

iii) Gateway

- Describe the decision factors and determines where the next step should go.

iv) Connections

- Describe how the process flow components are connected.

3.2.3 Current Problems and Issues

The Current Problems and Issues process is to capture and analyse the current problems and issues encountered by the key users, such as Business owners. It provides the project team with insights on some of the root causes of the issues and allows the project team to design the future system to tackle these problems and issues.

3.3 KEY ROLES & RESPONSIBILITIES

Table 3 - Key Roles & Responsibilities for CED

Who will be involved?	
Roles Involved	**Key Responsibilities**
1. Business Analyst	(a) Assist in eliciting and consolidating information/documents obtained from business users. (b) Seek input from or work jointly with the Internal PM to identify the appropriate stakeholders for interviewing and capturing of current environment information from a business perspective. (c) Arrange and conduct interviews/workshops with appropriate stakeholders to gather and understand current business environment and problems and issues. (d) Document all the current environment findings related to business through text descriptions and diagrams, i.e., related part of Current System Description, Current Business Process and Current Problems and Issues.
2. Business User	(a) Provide information and documents on current business processes and environment. (b) Participate in interviews/workshops to explain the current business processes, the interactions between business users with the current

Who will be involved?	
Roles Involved	**Key Responsibilities**
	systems and the major problems and issues encountered by the business users currently.
3. PAT / PSC	Review the CED: i) Current System Description (focus on Current System Overview); ii) Current Business Process; and iii) Current Problems and Issues.
4. Systems Analyst	(a) Provide information and documents on current technical environment, i.e., the system type and their components, hardware and software settings and configurations, network settings and configurations, data volume and frequencies, system interfaces. (b) Participate in interviews/workshops to explain the current system settings and functionalities as well as its usage. Highlight the major problems and issues encountered by IT project team currently. (c) Document all technical part of the current environment findings through text descriptions and diagrams, (d) Consolidate the technical part and business part from BA to provide all the sections of the CED i.e., Current System Description, Current Business Processes and Current Problems and Issues.

3.4 KEY TASKS & DELIVERABLES

Table 4 - Key Tasks & Deliverables for CED

What will be delivered?	
Key Tasks	**Tools & Templates**
1. Information Gathering (a) Obtain all relevant current environment information/documents from appropriate stakeholders. For examples, i) Current system's SA&D documents; ii) Current system's business process diagrams; iii) Relevant package solution materials (e.g., user, operations manual); and iv) ISSS and FS documents. (b) Conduct interviews/workshops to further capture and understand the current environment from identified stakeholders. i) The interviews/workshops should strictly focus on current	• Information Collection Techniques

What will be delivered?	
Key Tasks	**Tools & Templates**
environment and capture users' current problems and issues. ii) The interview/workshop for stakeholders could be separated as they provide information from different perspectives**.**	
2. Produce Current System Description (a) Review and analyse the information gathered on current technical (or manual) system to produce the Current System Description. (b) For each relevant system, document the following information: i) Current System Overview - Explaining the system's objective(s) and major functions. ii) Current System Hardware, Software and Network - Explaining the system's hardware specifications, software specifications and network infrastructure/configurations. iii) Volumes and Frequencies - Highlighting the transaction volumes and frequencies of each system component. iv) Interface with Other Systems - Highlighting all the current system's interfaces. (c) Sub-deliverable: Current System Description	• Flowchart Diagram *and/or* Use Case Diagram *and/or* Activity Diagram
3. Produce Current Business Process (a) Review and analyse the information gathered on current business processes to produce the Current Business Process. (b) For each major business process, produce a high level overview process diagram along with separate low level business process diagrams. i) Current Business Process could be presented in business process diagram with Flowchart Diagrams, Use Case Diagrams or Activity Diagrams (please refer to ***Tools & Templates in section 3.5*** for details). ii) The business process diagram should capture at least major key user interactions, business workflows, system activities and system process flows. (c) Sub-deliverable: Current Business Process	
4. Document Current Problems and Issues (a) Produce the Current Problems and Issues by analysing and consolidating the information captured during the interviews/workshops. i) Cover both business and technical issues.	N/A

What will be delivered?	
Key Tasks	**Tools & Templates**
ii) Focus on key issues only / business needs. iii) Problems and issues are potentially the ones that need to be addressed by the future system. (b) Sub-deliverable: Current Problems and Issues	
5. Review and Signoff (a) The CED (i.e., Current System Description, Current Business Process, and Current Problems and Issues) needs to be reviewed by PAT to confirm the correctness and completeness. (b) Once the contents are confirmed to be accurate and complete, the PSC can either signoff the CED or wait and signoff the whole SA&D report after the SA&D report is completed, subject to the decision of the project owner. (c) Deliverable: CED	• Action Checklist (CED)

3.5 TOOLS & TEMPLATES

(a) The list below includes the recommended tools, techniques, and checklist applicable to CED:

i) Information Collection Techniques: Please refer to ***Appendix A - Tool 1*** for details

ii) Flowchart Diagram: Please refer to ***Appendix A - Tool 3*** for details

iii) UML Diagram - Use Case Diagram: Please refer to ***Appendix A - Tool 4*** for details

iv) UML Diagram - Activity Diagram: Please refer to ***Appendix A - Tool 5*** for the sample of Activity Diagram

v) Action Checklist: Please refer to ***Appendix A - Tool 10*** for details

(b) The template can be found in ***Appendix C – Section 2: Current Environment Description***.

3.6 HINTS & TIPS

Table 5 - General Tips for CED

(a) Project team should identify what are relevant to the project that must be captured within the current environment. This will minimise reviewing and working on current environment information and documents that are not particularly useful for the project. (b) Current environment information should not be documented in an overly complex or detailed manner since the purpose of the CED is to allow the project team to obtain an

understanding of the current environment only. The project team should put the focus and time on designing and implementing the new system.

(c) The current business process should be identified after performing the gap analysis during preparation of business case.

(d) Project team could focus on the following three topics when trying to obtain the current problems and issues from the users:

i) What are the "common issues" faced by the users currently?

ii) What are the "frequently asked questions" by the users currently?

iii) What are the "time consuming" issues faced by the users currently?

4 REQUIREMENTS SPECIFICATION

(a) This process refers to the combination of User Requirements Document (URD) and Technical Specification as one deliverable, i.e., **Requirements Specification**.

(b) Please refer to ***Figure 3***, which summarises the key tasks and deliverables produced for Requirements Specification.

4.1 FLEXIBLE ADOPTION OF PROCESSES WITHIN REQUIREMENTS SPECIFICATION

Flexible adoption is not applicable to Requirements Specification since all its processes are mandatory, and therefore the decision tree is not necessary for this process. However, requirements are not necessarily documented in text format only; it could be documented using Prototyping techniques in some situations such as mobile application development.

4.2 USER REQUIREMENTS DOCUMENT

(a) URD consists of Future Business Process, Functional Requirements and Non-functional Requirements. Functional Requirements define the functions or features of a system that can be utilised by a user to fulfil business operation (i.e., what the system should do to provide business value when satisfied), while Non-functional Requirements specify criteria of how the system can perform and maintain these functions and features (i.e., how the system should work) from a business perspective.

(b) Deliverables from the Future Business Process, Functional Requirements and Non-functional Requirements will combine as one sub-deliverable for the URD process, i.e., the **User Requirements Document**.

(c) It is recommended to have a diagram to model high level overview of the whole system and define major system user profile showing major user roles involved in the system and supplement low level process diagram to model detailed process flow on a need basis.

(d) If Agile approach is adopted, high-level requirements are documented as “User Story” which are written by users in layman terms.

4.2.1 Overview & Approach

4.2.1.1 Future Business Process

(a) The Future Business Process definition captures the future business process models, and corresponding narratives and information that will be leveraged to elicit functional and

non-functional requirements. The definition process would facilitate the project team in performing subsequent activities such as the System Specification and TSO.

(b) The future business process model is to define the future business activities related to the new system to be implemented. The activity helps the project team and stakeholders to better visualise the future functions, processes and user interactions required in the new system. It is recommended to use Process Diagrams to present the future business processes.

(c) The major elements that need to be captured in the Process Diagrams are:

i) Event

- Events are of three types:
  - External inputs: inputs from outside the system boundary.
  - Decisions made in business activities within the system.
  - Scheduled points in time.

ii) Activity

- Describe the kind of work that needs to be performed.

iii) Gateway

- Describe the decision factors and determines where the next step should go.

iv) Connections

- Describe how the process flow components are connected.

4.2.1.2 Functional Requirements

(a) The objective of this process is to elicit, analyse, prioritise and validate the functional requirements that define the capabilities of the system.

(b) Functional requirements are specified by statements of functionality. It is highly recommended that Use Cases should be developed for all functional requirements, especially those that relate to user-system interaction and support a user-centric operation model.

(c) A complete set of functional requirements for the new system should be specified and stated in the form of a list. Requirements for the new system may include handling of existing functions, problem resolutions, provision of new facilities or functions etc.

(d) Functional requirements represent the functions or features that the system must have and the tasks (Use Cases) that the users must be able to perform. The major elements that need to be captured in the list of functional requirements are:

i) Requirement ID

- Specify a unique ID for each requirement entry.

ii) Requirement Title

- Title for the functional requirement.

iii) Priority

- Describe the priority of the requirement, i.e., "Must", “Should”, “Could”, “Won't” or other ranking scheme such as “High”, "Medium" or "Low". A must (high priority) requirement should be developed first followed by should (medium) and then could (low).

iv) Functional Requirement Description

- Describe the required functional features of the system, i.e., “what” the system should do. The functional requirement should include the reasons and objectives of the requirement and benefits expected when the new service is provided.

v) Frequency of use

- How frequent is the function used on average (e.g. 100 times daily).

vi) Acceptance criteria

- Describe how or what level of quality the feature should be provided to satisfy the users’ needs. It is important to ensure that, wherever possible, requirements are defined in terms that are quantifiable and measurable. The acceptance criteria (per requirement) from a business perspective includes:
  - Functional acceptance criteria
  - Constraints and special considerations

vii) Related Business Process

- State the related business process ID or reference for this functional requirement.

4.2.1.3 Non-functional Requirements

(a) Non-functional Requirements are requirements that specify criteria used in evaluating the operation of a system instead of specific behaviour as is the case with Functional Requirements.

(b) The objective of this process is to elicit, analyse, prioritise, and validate non-functional requirements that define the capabilities of the system from a business perspective.

(c) The Non-functional Requirements normally consists of different categories such as audit, control and security, global business rules, data requirements, usability requirements (e.g., user experience, user interface, report, etc.), service level targets, user volume and equipment requirements, data growth and retention requirements.

(d) The elements to be captured and documented for the Non-functional Requirements are:

i) Requirement ID

- Specify a unique ID for each requirement entry.

ii) Requirement Title

- Title for the non-functional requirement.

iii) Priority

- State the priority of the non-functional requirement, e.g., "Must", "Should", "Could" and "Won't" or other ranking scheme e.g., "essential", "beneficial if cost justified", "subsequent enhancement".
- Possibly have more level of classifications depending on the project situation.

iv) Non-functional Requirement Description

- Describe the non-functional requirement of the system, i.e., "how" the system should work.

v) Category

- Describe the category of the non-functional requirement. The category can be one of the followings or other required items:
  - Audit, Control and Security[c]
    - System Audit (e.g., audit trail and audit reports).
    - System Control (e.g., access to system, physical access control).
    - System Security (e.g., data protection during storage and transmission and application security).
    - System Backup and Recovery (e.g., backup frequency for system/data and time allowed for recovery of system/data).
    - Disaster Recovery Requirements (e.g., time required to recover data upon disaster and time required to switch to DR site).
    - Privacy Requirements (e.g., protection of personal data from unauthorised disclosure)[d].
  - Global Business Rules (e.g., policies, regulations, standard, etc. that affect the system globally).
  - Data Requirements (i.e., data requirements that affect the system globally or have significant impact to other systems).
  - General Usability Requirements (e.g., requirements for ease of use of system, language interface, user interface and report).
    - This section states the usability requirements i.e., ease of use for the proposed IT system.

---

[c] The security risk checklist can be referred to the "*Security Risk Assessment & Audit Guidelines*[4]"

[d] The Six Data Protection Principles (DPP) of the Personal Data (Privacy) Ordinance can be found in PCPD website: http://www.pcpd.org.hk/index.html

- Language requirements e.g., user interface, report etc. may also be stated.

- Service Level Targets
  - They refer to the levels of service provided by the system as expected by the users. They define the frequency or amount of time for a task to reach a certain condition such as system availability (i.e., how often the system can be accessed) and performance (i.e., the performance level that the system must satisfy like the response time for a data entry).
  - The purpose of this activity is to ensure that proper elements and commitments are in place to provide consistent and stable IT services in which the business requires. It provides a clear, concise and measurable description of service provision for the system.
- Data Growth and Retention Requirements (e.g., the annual growth rate of data and how long the data will be stored in the system or removed from the system).
- User Population and Equipment Requirements (e.g., the estimated number of users like concurrent users and maximum number of users and number of equipment required like workstations, printers, scanners, mobile devices, etc., required at various office sites).

4.2.2 Key Roles & Responsibilities

Table 6 - Key Roles & Responsibilities for URD

Who will be involved?	
Roles Involved	**Key Responsibilities**
1. Business Analyst	(a) Collaborate and communicate with stakeholders to elicit, capture and verify future business processes and functional and non-functional requirements from a business perspective. i) Identify the appropriate stakeholders to discuss, capture and verify the future business processes and requirements. ii) Arrange and conduct interviews/workshops with appropriate stakeholders to understand, capture and verify the future business processes as well as functional and non-functional requirements. iii) Prepare and document Future Business Process through text descriptions and diagrams. iv) Analyse, prioritise, refine, organise, document and verify functional and non-functional requirements. If required, assistance may be sought from the IT project team to help to develop user-system interaction diagrams (e.g., Use Case diagram,

Who will be involved?	
Roles Involved	**Key Responsibilities**
	Activity diagram) to facilitate business users to visualise and confirm the functional requirements. v) Provide input to the Systems Analyst for updating the RTM for functional and non-functional requirements. (b) Manage stakeholders' expectations and needs, and help to ensure the requirements are complete, unambiguous and map to real business needs.
2. Business User / Project Stakeholders	Verify and review the proposed Future Business Process, Functional and Non-functional Requirements, and support the PAT to review the URD.
3. Systems Architect	Review Non-functional Requirements and provide early feedback to business users if requirements are not feasible to implement within the project timeframe.
4. Systems Analyst	(a) Participate in interviews/workshops to provide supporting information (from a technical perspective) for the future business processes. (b) Highlight major concerns regarding the feasibility of the future business processes (from a technical perspective). (c) Document and update the RTM for Future Business Process, Functional Requirements and Non-functional Requirements.

4.2.3 Key Tasks & Deliverables

Table 7 - Key Tasks & Deliverables for URD

What will be delivered?	
Key Tasks	**Tools & Templates**
1. **Define Future Business Process** (a) Identify all business users and related project stakeholders (leverage the "*Practice Guide to Project Management for IT Projects under an Outsourced Environment*[5]" stakeholders list) that need to provide inputs and requirements to the Future Business Process. (b) Review Current Business Process while referencing other pertinent information such as current and projected future policies, procedures and legal/regulatory documentation that may impact business process. (c) Conduct interviews, workshops, etc. with business users and related	• Information Collection Techniques • Flowchart Diagram *and/or* Use Case Diagram

What will be delivered?	
Key Tasks	**Tools & Templates**
project stakeholders to discuss and capture the flow and requirements of the future business processes. i) Business Analyst should use the current business processes and current problems and issues identified during the development of business case and CED process to assist with the future business processes discussions/formulation. ii) Some initial drafts of the Future Business Process could be drawn prior to the interviews/workshops to enable discussions. (d) Review and analyse the information collected to create the future business process diagrams by seeking assistance from the IT project team, if required. (e) For each key future business process, a separate business process diagram should be produced. (f) A complex business process can be broken down into smaller processes, with each a separate diagram. (g) Future business processes could be presented by using Process Flowchart Diagrams, Use Case Diagrams, Activity Diagrams, Sequence Diagrams or State Diagram (please refer to ***Tools and Templates in section 4.2.4*** below for details). (h) Sub-deliverable's section: Future Business Process	*and/or* Activity Diagram *and/or* Sequence Diagram *and/or* State Diagram
2. Elicit Functional Requirements (a) Elicit and capture functional requirements via the same workshops, interviews and surveys, etc. organised and conducted for stakeholders for capturing the future business processes. (b) Analyse the findings from the interviews, workshops and surveys to derive the initial set of Functional Requirements. (c) Based on the interviews, workshops and survey findings, define the detailed acceptance criteria that determine the boundaries for the Functional Requirements. (d) Based on the initial set of Functional Requirements, produce diagrams such as Use Case diagrams and Activity diagrams or User Story for Agile approach project, to help facilitate business users to verify their understanding and confirm the Functional Requirements (and acceptance criteria) identified.	• Information Collection Techniques *and/or* Prototyping Techniques • Flowchart Diagram *and/or* Use Case Diagram *and/or*

What will be delivered?	
Key Tasks	**Tools & Templates**
(e) Different approaches of other techniques such as decision tables, decision trees, etc. can be found in Tools and Templates in section 4.2.4 below. (f) Requirements can also be documented using prototyping techniques supplemented with some text descriptions if needed, such as mobile application and system development using Agile approach. (g) Document and update RTM for functional requirements to establish and maintain traceability throughout the project lifecycle. (h) Sub-deliverable's section: Functional Requirements	Activity Diagram *and/or* User Story • Decision Tables *or* Decision Trees • RTM
3. Elicit Non-functional Requirements (a) Organise and conduct interviews with appropriate stakeholders to capture their non-functional requirements. (b) Elicit the non-functional requirements according to different categories, for example: i) Identify the audit, control and security requirements based on the sensitivity and classification of data managed and projected by the system and accessed by users. For example, accounting staff may be permitted to access to invoices while the engineering staff should not access. ii) Determine the service level targets such as system availability and response time requirements. These requirements may be needed in order to fulfil the business turnaround time or performance pledge. iii) Some global business rules are required to be met such as ordinances, policies and regulations. (c) Document and update the RTM for non-functional requirements. (d) Sub-deliverable's section: Non-functional Requirements	• RTM
4. Walkthrough Session (a) Conduct a structured walkthrough session with business users and related project stakeholders to verify and fine tune the URD for accuracy, consistency and applicability. During the walkthrough, it is useful to highlight how the identified gaps are resolved through future	• Action Checklist (Requirements Specification)

What will be delivered?	
Key Tasks	**Tools & Templates**
business processes, and which of the current business processes are to be retired.. (b) Sub-deliverable: URD	

4.2.4 Tools & Templates 

(a) The list below includes the recommended tools, techniques, and checklist applicable to URD:

i) Information Collection Techniques: Please refer to ***Appendix A - Tool 1*** for details

ii) Prototyping Techniques: Please refer to ***Appendix A - Tool 2*** for details

iii) Flowchart Diagram: Please refer to ***Appendix A - Tool 3*** for details

iv) UML Diagram - Use Case Diagram: Please refer to ***Appendix A - Tool 4*** for details

v) UML Diagram - Activity Diagram: Please refer to ***Appendix A - Tool 5*** for details

vi) UML Diagram - Sequence Diagram: Please refer to ***Appendix A - Tool 6*** for details

vii) User Story: Please refer to ***Appendix A - Tool 7*** for details

viii) Decision Table and Tree: Please refer to ***Appendix A - Tool 8 & 9*** for details

ix) Action Checklist: Please refer to ***Appendix A - Tool 10*** for details

x) State Diagram: Please refer to ***Appendix A - Tool 27*** for details

(b) The template can be found in ***Appendix C – Section 3.1: User Requirements Document***.

4.2.5 Hints & Tips 

Table 8 - General Tips for URD

(a) The gaps between current and future business process provide a starting point for eliciting functional requirements and non-functional requirements, because they help identify future functionalities of the desired system.
(b) Ensure all the necessary stakeholders are involved in the elicitation and approval processes of the Non-functional Requirements, which should often include IT project team in addition to Business Users.
(c) Functional requirements stakeholders may not always be the same stakeholders involved in the elicitation and approval of the non-functional requirements.

(d) Each functional requirement must be defined clearly such that the acceptance criteria can be established.

(e) Business Analyst should work closely with IT project team in particular the Systems Analyst to work out what are relevant to the project that must be highlighted and discussed at interviews and workshops. This will minimise reworks on the Future Business Process and the activities that follow.

(f) Prior to approaching stakeholders, Business Analyst and Systems Analyst could set some directions (e.g., start processes from user touch points) to guide stakeholders in coming up with the Future Business Process.

(g) Systems Analyst should focus on designing and implementing the actual system and the feasibility of these suggestions, whilst Business Analyst collects input from business users.

(h) Business Analyst should consider if stakeholders of the Future Business Process differ from those of the Current Business Process, and whether the former group is being involved in interviews, workshops or validation sessions

(i) For the List of Functional Requirements, Business Analyst should concentrate on the functionalities to be provided by the required system from end users' point of view; design or implementation details should be documented in other documents.

(j) Performance requirements are sometimes overlooked by requirements stakeholders during requirements elicitation. Keep in mind that by identifying performance requirements along with the other system requirements can minimise rework. Assistance from IT project team may be sought when eliciting performance requirements.

(k) Manage stakeholder involvement during requirements sessions. Failure to have all stakeholders involved early in the elicitation and approval of the requirements could result in change requests later or a system that does not fully meet business objectives or user expectations.

(l) There are various characteristics that constitute as a good requirement, which are listed below:

i) Concise - does not contain unnecessary information
ii) Design Independent - does not specify a particular solution or a portion of a particular solution.
iii) Identifiable - possesses identification that is used to uniquely identify the requirement.
iv) Complete - contains all necessary information including acceptance criteria
v) Consistent - does not conflict with other requirements, and has appropriate amount of detail
vi) Feasible - can be technically and legally accomplished within project constraints
vii) Testable - physically and functionally testable (some requirements can also be verified through inspection, analysis, or demonstration via prototypes)
viii) Modular - can be changed without impacting the overall system

ix) Traceable - each requirement is uniquely identified and can be traced forward and backward

(m) There are other characteristics that constitute as a bad requirement, which should be avoided:

i) Ambiguous - avoid descriptors such as “TBD”

ii) Vague - avoid using words such as “fast”, “efficient”, “reasonable”, “generally”, “user-friendly”, and “approximately”

iii) Suggestive - avoid using the words “might”, “may”, “ought”, “perhaps”

iv) Guarantees - avoid words such as “guarantee”, “totally safe”, “never fail”

v) Technical - do not describe requirements using technical details or pseudo-code that cannot be understood by the stakeholders

vi) Assumptions - avoid “missing” descriptions because they are assumed

(n) It is just as important to identify the out-of-scope requirements as it is to identify the in-scope requirements. All stakeholders must agree on out-of-scope requirements as part of the base-lining of the in-scope requirements.

(o) The content of a User Story should include a unique User Story ID, User Story description, acceptance criteria and assigned business value (if any).

4.3 TECHNICAL REQUIREMENTS

4.3.1 Overview & Approach

(a) The Technical Requirements definition supplements the Non-functional Requirements and is not raised by the Business Analyst, but from the IT project team.

(b) This process allows the IT project team to define, document and validate the detailed operational, performance and technical architecture requirements and all relevant technical factors which may impact the technical architecture.

(c) It is similar to Non-functional Requirements, but the Technical Requirements are prepared and provided by the IT project team.

(d) The elements to be captured and documented for the Technical Requirements are:

i) Requirement ID

- Specify a unique ID for each requirement entry.

ii) Requirement Title

- Title for the technical requirement.

iii) Priority

- State the priority of the non-functional requirement, e.g., “Must”, “Should”, “Could” and “Won’t” or other ranking scheme e.g., “essential”, “beneficial if cost justified”, “subsequent enhancement”.
- Possibly have more level of classifications depending on the project situation.

iv) Technical Requirements Description

- Describe the technical requirement of the system, i.e., “how” the system should work.

v) Category

- System Backup and Recovery Requirements
  - Backup arrangements
  - Recovery procedures requirement under various system failures
- Disaster Recovery Requirements
  - Minimum service level under disaster
  - Off-site backup arrangement
  - Recovery procedure
  - Time required to recover upon disaster
- Privacy Requirements
  - Protection of personal data from unauthorised disclosure e.g., protection on personal identification document number
- Technical Support Requirements
  - Software and hardware support levels

- Equipment maintenance and repair cycles
- Test/diagnostic equipment

- Interface Requirements
  - User groups
  - Content presentation
  - Application navigation
- Maintainability, Control and System Management Requirements
  - System failure(s)
  - Operational readiness and success
  - System effectiveness evaluation and improvement
- Testing
  - Design stage testing procedure
- Data Conversion
  - Data conversion process
  - Data cleansing
  - Verification program
- User Experience
  - Overall experience and satisfaction when a user is using a product or system
  - Details within the user interface functionality, behaviour, and design
  - Industry best practices as part of standard requirements
  - For mobile application, please refer to the "*Practice Guide on Developing Mobile App*[6]", which includes user interface guidelines on iOS, Android, Blackberry, Windows Phone, etc.

4.3.2 Key Roles & Responsibilities

Table 9 - Key Roles & Responsibilities for Technical Requirements

Who will be involved?	
Roles Involved	**Key Responsibilities**
1. Systems Analyst	(a) Identify and document the technical requirements and validate the technical requirements with appropriate technical and business users. (b) Document and update the RTM for Technical Requirements.
2. Systems Architect	Review Technical Requirements and provide early feedback to business users if requirements are not feasible to implement within the project timeframe.
3. PAT / PSC	Review the Requirements Specification: i) URD • Future Business Process

Who will be involved?	
Roles Involved	**Key Responsibilities**
	• Functional Requirements • Non-functional Requirements ii) Technical Requirements

4.3.3 Key Tasks & Deliverables

Table 10 - Key Tasks & Deliverables for Technical Requirements

What will be delivered?	
Key Tasks	**Tools & Templates**
1. Define Technical Requirements (a) Gain an understanding of the data entities, business processes and logical information flows. i) Identify the high volume and time critical transactions. ii) Understand the external interface requirements. (b) Detail the support requirements for the system. This will include software and hardware support levels, equipment spares, multipurpose test/diagnostic equipment, maintenance and repair cycles, minimum inventory levels for system consumables and any constraints on the supply of consumables (e.g., central purchasing). (c) Detail the constraints imposed by the existing environment or systems which must be interfaced with (e.g., web-services, Remote Procedure Call). This includes system software, communications protocols, special hardware devices, etc. (d) Determine maintainability, control and system management requirements. This will include all features needed to manage, control, administer, maintain and support the system (e.g., utilities, backups, operator involvement levels). It can also include special requirements relating to performance monitoring, utilisation measurement and operational analysis. (e) Review the technical requirements in the context of the expected functionality to identify potential incompatibilities and restrictions and draft the Technical Requirements. (f) Document and update the RTM for technical requirements. (g) Sub-deliverable: Technical Requirements	• RTM

What will be delivered?	
Key Tasks	**Tools & Templates**
2. Review and Signoff (a) As Technical Requirements is the final deliverable of Requirements Specification, the following task is to review and signoff the whole document of Requirements Specification. (b) Conduct a walkthrough session with IT project team to walkthrough and verify and fine tune the technical requirements for consistency, applicability and to determine if directionally accurate. (c) The Requirements Specification (i.e., URD and Technical Requirements) needs to be reviewed by PAT to confirm the correctness and completeness. (d) Once the contents are confirmed to be accurate and complete, the PSC can either signoff the Requirements Specification or wait and signoff the whole SA&D report after the SA&D report is completed, subject to the decision of the project owner. (e) Deliverable: Requirements Specification	• RTM • Action Checklist (Requirements Specification)

4.3.4 Tools & Templates

(a) The list below includes the recommended tools, techniques, and checklist applicable to Technical Requirements definition:

i) Action Checklist: Please refer to ***Appendix A - Tool 10*** for details

ii) Requirements Traceability Matrix: Please refer to ***Appendix A - Tool 11*** for details

(b) The template can be found in ***Appendix C – Section 3.2: Technical Requirements***.

4.3.5 Hints & Tips

Table 11 - General Tips for Technical Requirements

(a) The gaps between current and future business processes provide a starting point for eliciting technical requirements, because they help identify future functionalities of the desired system. (b) Ensure all the necessary stakeholders are involved in the elicitation and approval processes of the technical requirements, which should often include IT project team. (c) RTM must be prepared and updated by Systems Analyst when eliciting technical requirements so the requirements are captured and stored in the RTM properly. (d) There are various characteristics that constitute as a good or bad requirement which can be

found in the Hints & Tips for URD.

(e) It is just as important to identify the out-of-scope requirements as it is to identify the in-scope requirements. All stakeholders must agree on out-of-scope requirements as part of the base-lining of the in-scope requirements.

5 SYSTEM SPECIFICATION

(a) This process refers to the combination of several documentations, including Functional Specification, Architecture Design and System Design, as one deliverable, i.e., **System Specification**.

(b) Please refer to ***Figure 4***, which summarises the key tasks and deliverables produced under System Specification.

(c) If mobile application is to build, please cross check the "*Practice Guide on Developing Mobile App*[6]" for mobile application development.

5.1 FLEXIBLE ADOPTION OF PROCESSES WITHIN SYSTEM SPECIFICATION

(a) The decision tree below in ***Figure 7*** is a recommendation on under what consideration factors a process within the System Specification could be simplified.

(b) For the detail explanation of the consideration factors, please refer to the "***Part II - How to Use this Guide***".

(c) Note that Functional Specification can be omitted if the functionalities are provided by software package without customisation. For package approach with customisation, only the modified functions have to be documented here.

(d) Other than the consideration factors described in the decision tree, the development approaches adopted such as the iterative approach and the Agile approach may also be a factor to be considered when deciding which deliverables are to be simplified.

(e) For example, as the Agile approach is to create the high-level requirements for the core functions and critical system components to the extent that is sufficient to start system development, details will normally be worked out and refined iteratively in the System Implementation phase.

(f) However, the level of design detail is not dictated by the iterative approach. The Architecture Design and System Design may be further simplified by covering just high level of details, or may even be excluded depending on the project nature and project criticality. Once the system design has been stabilised, the detailed Architecture Design and System Design may be complemented by other documents in later phases of the SDLC.

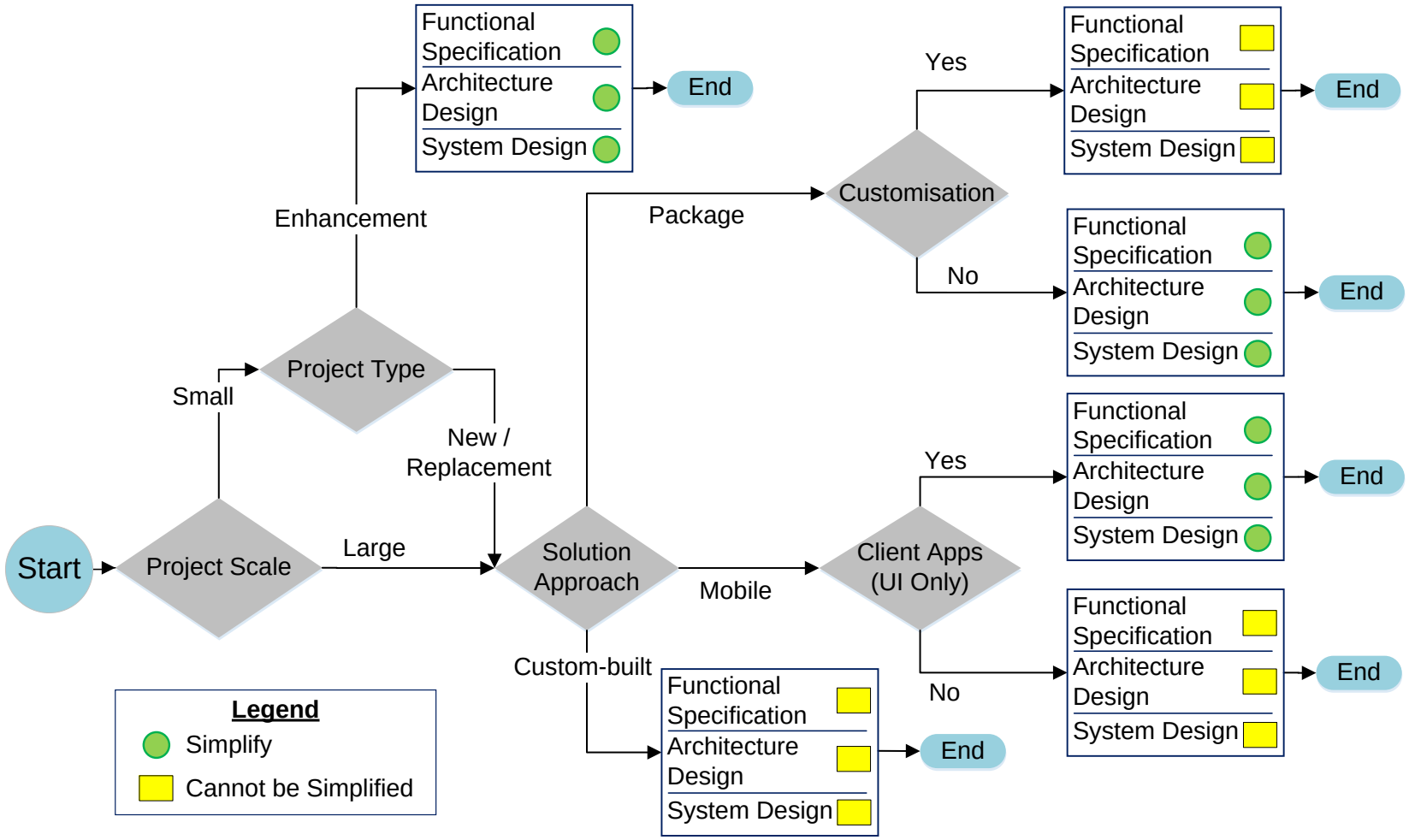

Figure 7 - Decision Tree for System Specification

5.2 FUNCTIONAL SPECIFICATION

(a) Functional Specification precisely states the functions and capabilities that the system must provide and the constraints that it must adhere to. Functional Specification is the basis for all subsequent project planning, design, and coding as well as the foundation for all system testing and user documentation. It should describe, as completely as necessary, the system's behaviours under various conditions.

(b) The Functional Specification is mainly used by the following project roles:

i) Systems Architects / Systems Analysts as a baseline to design the initial system architecture and identify potential constraints.

ii) Developers use the Functional Specification as a basis to develop the System Design and establish traceability between the requirements and the design components, including Logical Architecture Design and System Design.

iii) Internal PMs refer to the Functional Specification for input to control and monitor the project progress and scope.

iv) Testers use the Functional Specification to develop the test cases and establish traceability between the requirements and the test cases/scripts.

(c) Functional Specification consists of the following:

i) Required System Overview

ii) Function Definition

(d) If Agile approach is adopted, Prioritised Requirements List could be used as Functional Specification directly. If considered appropriate to facilitate system implementation, Required System Overview should be defined to provide a functional overview of the targeted system and Function Definition should be included to detail the expected behaviour of the system for the user stories.

5.2.1 Overview & Approach

5.2.1.1 Required System Overview

The Functional Specification should include components (the logic group of business functions) that provide a functional overview of the targeted system with respect to its features, its relationships and interactions with other systems and components, and its dependencies (if any) on other system functions, such as the following:

i) Business Needs

- Business needs of the project that are met in part or full by the functionality of the system.

ii) Major Features

- Key features about the system that address one or more system requirements should be documented.

iii) System Related Functions

System functions related in the overall project by this system's functionality should be listed with brief descriptions.

5.2.1.2 Function Definition

(a) The objective of the Function Definition is to detail the outputs created in the prior requirements activities - Functional Requirements (including user-system interaction diagram such as Use Cases) and Non-functional Requirements - to individual functionality, and to specify the functions provided by the required system. The outcome of this task is a Functional Specification document which serves as the agreement between the project team and business users regarding the system to be built at a specific time.

(b) The major elements that need to be captured in the Function Definition are:

i) Function ID

- Give a unique ID for each function. Project team should decide their own naming convention to suit their project situation (e.g., RFxxxx for retrieval functions, UFxxxx for update).

ii) Function Name

- Give the name of a function.

iii) Component Name

- Describe the logic group of the business function to be implemented.

iv) Category

- Categorise the function as presentation layer, operational design consideration, or exception and error handling as specified below ***5.2.1.2 (c)***.

v) Related Requirements

- List the requirements which are resolved by the function (e.g. Requirements Traceability Matrix).

vi) Function Description

- Describe briefly the purposes of the function and how it will be used by end users (e.g., the input parameters required).

vii) Mode

- Classify the function as Online/Batch, Enquiry/Update.

viii) Frequency

- Specify the minimum, average and maximum frequency in a specified period.

ix) Special Service Level Requirements

- Specify the service level required (e.g., response time < 10 seconds), if necessary.
- Note: General Service Level Requirements have been specified in the Non-functional Requirements.

x) Business rules (including data validations)

- Describe any business rules for the function, such as validation logic (i.e., acceptance criteria), if any.

xi) Reports

- Describe the layout and logic of the report, if any.

xii) User Input Screens and Forms

- Describe the layout of the screen, if any.

xiii) Security Requirements

- Describe the identity and access management, user registration; roles based access, privacy requirements, if any.

(c) In addition, it is also essential to group the system functions into their respective categories, namely presentation layer, operational design consideration, as well as exception and error handling.

i) Presentation Layer

- Categorise the functional design of the system features that are directly visible to the user and with graphical user-interface (GUI) component, such as screen view and field, form and table layout.

ii) Operational Design Consideration

- Categorise essential design considerations that should be kept in mind to ensure smooth operation of the system's functionality, including performance considerations, constraints and dependencies, security and control design, as well as initiation, frequency, and scheduling design.

iii) Exception and Error Handling

(d) Categorise the system’s behaviour when it encounters exception conditions or error scenarios, including exception and errors handled, log output, restart procedures.

5.2.2 Key Roles & Responsibilities

Table 12 - Key Roles & Responsibilities for Functional Specification

Who will be involved?	
Roles Involved	**Key Responsibilities**
1. Business Analyst	(a) Collaborate and communicate with stakeholders to manage their expectations, needs and changes in requirements. (b) Provide support to users and coordinate with IT project team to help review the design of the IT system from the business perspective, resolve issues/conflicts among stakeholders.
2. Systems Analyst	(a) Identify functional development components to facilitate design session. (b) Conduct design session with business users and Subject Matter Experts (SMEs). (c) Transform both functional and non-functional requirements into technical system functions. (d) Cross check the traceability and relevancy between requirements and system functions. (e) Document and update the RTM for Functional Specification.
3. Systems Architect	Review Functional Specification and produce the enabling recommendations on Architecture Design.
4. Business User	(a) Being informed and consulted for the system functions during design sessions. (b) Due to technical level of the system function, business user is not accountable for Functional Specification, but he/she should contribute to the User-Interface design.

5.2.3 Key Tasks & Deliverables

Table 13 - Key Tasks & Deliverables for Functional Specification

What will be delivered?	
Key Tasks	**Tools & Templates**
1. Identify Functional Development Components Review the Requirements Specification and identify functional development components and any gaps that may exist, which will help to conduct design session.	N/A

What will be delivered?	
Key Tasks	**Tools & Templates**
2. Facilitate Design Sessions (a) Facilitate design sessions/workshops with key business users, SMEs, to document Functional Specification inputs. (b) Managing user involvement during design sessions is a key success factor. Failure to have all stakeholders involved early in the elicitation and approval of the design results in change requests later or a system that does not fully meet business objectives or user expectations. (c) Sub-deliverable's section: Required System Overview	• Prototyping Techniques
3. Transform Requirements into System Functions (a) Create a clear vision of the end state of development. (b) Based on the functional and non-functional requirements defined in the URD process, transform each requirement into technical system functions. For the technical system functions: i) It is defined as a collection of processes to serve end-user and each technical system function may handle a number of transactions and/or enquiry. ii) One requirement may be transformed into one or more technical system function(s). iii) One function many also be applicable to several different requirements if it shares similar purpose (e.g., submit, delete). (c) Clearly document what functionality needs to be developed under the Function Definition Template. The level of details may vary depending on the development approach, for example, Agile approach may contain high level functions descriptions. (d) Document references to requirements for any custom development in the RTM, i.e., map related requirement into corresponding system function. (e) Present the end-user access control requirements as a matrix of user access roles, feature permissions, and data access rules, and document in the System Specification Template. (f) Sub-deliverable's section: Function Definition (g) Sub-deliverable: Functional Specification	• RTM

5.2.4 Tools & Templates 

(a) The list below includes the recommended tools and techniques applicable to Functional Specification:

i) Prototyping Techniques: Please refer to ***Appendix A - Tool 2*** for details

ii) Requirements Traceability Matrix: Please refer to ***Appendix A - Tool 11*** for details

(b) The template can be found in ***Appendix C – Section 4.1: Functional Specification***.

5.2.5 Hints & Tips 

Table 14 - General Tips for Functional Specification

(a) Project teams may consider simplifying the standard template to suit their own situation. For example, if there is a common Special Service Level Requirement for some functions, this information may be documented centrally at the beginning of these functions, rather than repeating them in every function.
(b) If Agile approach is adopted, the level of detail of technical system functions may be high.
(c) Functional Specification can also documented in Prototyping format for certain types of projects, such as mobile application

5.3 ARCHITECTURE DESIGN

(a) Architecture Design is related to designing the architecture of the different system components and their interfaces to establish the framework for the detailed design of the system. It covers the data and application aspects of the system based on the functions defined in the Functional Specification.

(b) If Agile approach is adopted, the architecture design needs only be defined at a high level with the focus on core functions and components.

(c) Architecture Design consists of the following:

i) Application Architecture

- Application
- Security
- Integration

ii) Data Architecture

- Data Source
- Data Store Architecture
- Data Retention and Archive
- Data Conversion Architecture

(d) Architecture Design will be produced as the final product of this process.

5.3.1 Overview & Approach

5.3.1.1 Application Architecture

The objective of the Application Architecture is to create the high level design for the application. This will include a complete architecture definition for the application excluding data, focusing on the core functions, security and integration.

i) Application

Describe the core functions of the application and divide the system into subsystems. This helps facilitate a common understanding of the application architecture and the quality and reliability of the system.

ii) Security

Define the high level security components of the core application solution. This helps integrate security components from the project start rather than retrofit security capabilities during downstream activities.

iii) Integration

Develop the architecture definition for the various integration components of the application such as web services, messaging and service orchestration.

5.3.1.2 Data Architecture

The objective of the Data Architecture is to design and document the data architecture of the system, i.e., how data is processed, stored, and utilised by the system. The elements that should be covered for Data include:

i) Data Source

- Provide a list of data sources required by the system which includes:
  - The type of data in the data source

ii) Data Store Architecture

- Describe the data stores, which encapsulates the core data entities, required by the system which includes the following elements:
  - The flow of data across the different data stores.
- Provide an overview of the core data entities and relationships which will be ingested and processed by the system
  - The conceptual data model can be illustrated by using a diagram such as Conceptual ER-Diagram or Conceptual Class Diagram.
  - For each of the core data entities and relationships identified, provide an overview of the functional processing to be performed on it

iii) Data Retention and Archive

- Describe which and how the data are retained and archived. The elements that should be captured include:
  - Data Element to be Retained and Archived
  - Archive Method and Frequency (e.g., Backup to offline tape storage twice a week)
  - Data Retention Policy

iv) Data Conversion Architecture

- Describe how data are extracted, transformed and loaded from the source data stores to the target data stores. If the goal is to evolve or replace a legacy system, data conversion will identify data from the old system to be converted into the new system's format. If a brand-new system is being implemented, data conversion may be necessary to leverage existing data stores or pre-populate parts of the target data stores.

5.3.2 Key Roles & Responsibilities

Table 15 - Key Roles & Responsibilities for Architecture Design

Who will be involved?	
Roles Involved	**Key Responsibilities**
1. Business Analyst	(a) Collaborate and communicate with stakeholders to manage their expectations, needs and changes in requirements. (b) Provide support to users and coordinate with IT project team to help review the design of the IT system from the business perspective, resolve issues/conflicts among stakeholders.
2. Systems Analyst / Systems Architect	(a) Design the data architecture that supports efficient delivery and business needs. (b) Design the application architecture by defining the architecture goals, components and interfaces that satisfy business needs.

5.3.3 Key Tasks & Deliverables

Table 16 - Key Tasks & Deliverables for Architecture Design

What will be delivered?	
Key Tasks	**Tools & Templates**
1. Develop Application Architecture * * This key task can be conducted in parallel with other key tasks with asterisk (*) (a) Identify architecture constraints which are limitations or restrictions that affect the solution design and technology selection. i) The constraints can be identified through analysis of the functional, non-functional and technical requirements. ii) For example, the application must be based on existing software, and the application must be integrated with security technology already available in the organisation. (b) Define the layout for application's high level technical solution. Divide the application into subsystems that implement the overall application functionality if necessary. i) At a minimum, identify and describe all the different layers in the application architecture, such as presentation layer, business layer and persistence layer, and detail the communication medium by	• Flowchart Diagram *and/or* Block Diagram *and/or* Deployment Diagram

What will be delivered?	
Key Tasks	**Tools & Templates**
which the different application layers interact. ii) A diagram may be included to illustrate the different layers in the application and their relationships.	
2. Develop Security Architecture * * This key task can be conducted in parallel with other key tasks with asterisk (*) Ensure that all critical security architectural requirements and constraints regarding the implementation have been identified. i) Security - define the tools and frameworks used to secure the application, with the design approach employed. The following security components should be documented: • Identity and access management - functions such as user identity management, access management and provisioning. • Audit logging - audit trail and correlation to detect and monitor user access (read/write/update) to data. • Data privacy - data privacy functions, such as data-handling controls and data leakage prevention solutions to protect personal identifiable information. • Network security - network security functions, such as encryption of data in transit, firewalls, an intrusion prevention system, and trust zone architecture. • Database and media security - database and media security functions, such as user activity monitoring and data encryption.	• Security Controls Implementation Overview Diagram
3. Develop Integration Architecture * * This key task can be conducted in parallel with other key tasks with asterisk (*) (a) Define the design approaches that will be taken for the various integration components of the application, such as web services, messaging, service orchestration and batch. i) Web services - describe the web services standards, interface contract design (e.g., Web Services Description Language (WSDL)), transaction and exception handling, and integration with underlying business services and identify the web services engine if applicable. ii) Messaging - describe the messaging server, messaging model (e.g.,	• Block Diagram *and/or* Application Communication Diagram

What will be delivered?	
Key Tasks	**Tools & Templates**
publish/subscribe, point-to-point), message payload format, API, transaction and exception handling. iii) Service orchestration - describe the workflow server and the tools to be used for workflow creation. iv) Batch - describe the scheduler, dependencies between various jobs, any batch processing framework used, failure and exception handling. (b) Define the necessary details about the deployment of the application, such as how the application is packaged into binaries, and the different server environments in which the application will be deployed. (c) Define the tools and software used for application development, security, integration, and other architecture components to develop the application. (d) Sub-deliverable's section: Application Architecture	
4. Define Data Source * * This key task can be conducted in parallel with other key tasks with asterisk (*) (a) Provide a list of data sources to be considered. Consider the sources of existing data such as from legacy system. Note that existing data may be in the form of documents or spreadsheets. (b) Describe the kind of data in these data sources and frequency of accessing this data, such as nightly batch or on every search query run.	N/A
5. Develop Data Store Architecture (a) Describe the data stores that will be built, including the data stores for staging data and describe the flow of data across different data stores. i) Data Dissemination Diagram, as complementary tool, may be developed to visualise how data gets disseminated within the application components or between different systems. (b) Develop a high level diagram, known as the Conceptual Data Model Diagram, to show only the core data entities that have a business meaning and the relationships between these core data entities. • Note: attributes or primary keys are not required to be included in the diagram. (c) Describe the core data entities in the form of a catalogue or table.	• Data Dissemination Diagram *and/or* Conceptual Data Model Diagram

What will be delivered?	
Key Tasks	**Tools & Templates**
(d) Provide an overview of the functional processing to be performed for each of the core data entities and relationships in the core data entities table. (e) Create a cross reference table between the core data entities described and data stores described. • Note: this step is optional and relevant to migration related projects. In the case of data migration, particular attention is needed to maintain the required relationships.	
6. Define Data Retention and Archive (a) Design data archiving approach based on the functional requirements of the system. The approach may include schedule of data archiving, strategy to ensure data quality, and strategy to address identified data quality problems. (b) Design the data retention policy based on the functional requirements, legal requirements and Privacy Impact Assessment requirements imposed on the system.	N/A
7. Design Data Conversion Architecture (a) Identify sources of data to be extracted from one or multiple data stores and loaded into one or multiple, target data stores and estimated/projected volume. (b) Consider the following: i) Sources of existing data to be migrated and their target location; ii) Systems that use the current data sources and the systems that will use the new data store; iii) Data not currently held electronically and how it should be entered into the new data store; and iv) Historical data that also needs to be migrated (Consider historical data that may not be converted to the new system's format, historical data from prior years that only need to be kept at a summary level). (c) Define an approach for any data conversion required for a system implementation. A Data Migration Diagram, as complementary information, may be created to illustrate the flow of data from the source to the target data stores. This includes: i) Interdependencies among conversion items to determine if any	• Data Migration Diagram

What will be delivered?	
Key Tasks	**Tools & Templates**
prerequisites or sequencing of conversions need to occur. ii) High-level mapping of the data from the data sources to the target data store(s). iii) Whether a manual conversion or an automated conversion is appropriate based on volume and data quality. iv) Whether the data will be cleaned up prior to conversion, during conversion, or after conversion. (d) Consider the security control requirement (i.e., encrypt the data, transfer the data in security channel etc.) for the data source and the converted data in data store. (e) Describe the approach to verify the converted data in high level detail. (f) Sub-deliverable's section: Data Architecture (g) Sub-deliverable: Architecture Design	

5.3.4 Tools & Templates

(a) The list below includes the recommended tools and techniques applicable to Architecture Design:

i) Application Architecture:

- Flowchart Diagram: Please refer to ***Appendix A - Tool 3*** for details
- Block Diagram: Please refer to ***Appendix A - Tool 15*** for details
- Deployment Diagram: Please refer to ***Appendix A - Tool 16*** for details
- Application Communication Diagram: Please refer to ***Appendix A - Tool 17*** for details
- Security Controls Implementation Overview Diagram: Please refer to ***Appendix A - Tool 20*** for details

ii) Data Architecture:

- Conceptual Data Model Diagram: Please refer to ***Appendix A - Tool 12*** for details
- Data Dissemination Diagram: Please refer to ***Appendix A - Tool 13*** for details
- Data Migration Diagram: Please refer to ***Appendix A - Tool 14*** for details

(b) The template can be found in ***Appendix C – Section 4.2: Architecture Design***.

5.3.5 Hints & Tips 

Table 17 - General Tips for Architecture Design

(a) Application:

i) Ideally all critical requirements have been documented in previous activities, in which the architecture goals and constraints can simply be referenced. However, if the goals and constraints have not been exhaustively documented before, that should be completed during architecture design.

ii) Pay attention to all Non-functional Requirements - security, service level targets, and so on, and consider their impact on architecture design.

(b) Data:

i) Identify appropriate stakeholders who understand the existing data model and possess the ability to determine how to transform the source data.

ii) Note any historical data from legacy systems that may not be able to be converted into the new system's format. Data structure changes in the new system may make old data unable to be converted. In addition, consideration should be given to certain historical data from the old system which may need to be kept only at summary level.

(c) The Application Architecture, design with infrastructure provisions in mind and/or early start of the Technical System Architecture design, may be performed in parallel if needed.

5.4 SYSTEM DESIGN

(a) System Design, also known as the detailed design of the system, includes detailed design of the various components of the system and the working relationships, to the extent that the design is sufficiently complete to begin system development, based on the functions defined in the Functional Specification. The detail design includes:

i) Application

- Application Design
- User Experience Design

ii) Data

- Logical Data Model
- Physical Data Model

(b) System Design will be produced as the final product of this process.

(c) If Agile approach is adopted, the high-level Logical Data Model and User Experience Design, based on the business rules and user stories are generally defined in System Design.

5.4.1 Overview & Approach

5.4.1.1 Application

The Application Design and User Experience Design should be developed and documented as described below:

i) Application Design

- Define the design of the application functionality provided by the system by decomposing the functionality into smaller subsystems and then presenting the subsystems in terms of collaborations of contained design elements such as business components, attributes, interfaces, workflows, etc. Detailed diagrams should be developed to communicate the Application Design.
- It consists of the following sub-tasks:
  - Design Application
  - Design Integration
  - Design Security
  - Design Data Conversion

ii) User Experience Design

- User Experience describes the overall experience and satisfaction when a user is using a product or system.

- The design includes the definition of details within the user interface functionality, behaviour, and design elements relating to the user experience. The User Experience Design should be validated with business user to confirm users are comfortable with the design.
- For mobile application, please refer to the "*Practice Guide on Developing Mobile App*[6]", which includes user interface guidelines on iOS, Android, Blackberry, Windows Phone etc.

5.4.1.2 Data

The Logical Data Model and Physical Data Model should be developed and documented as described below:

i) Logical Data Model

Depict the logical entity types, data attributes describing the entities, and the relationships and relationship cardinalities between the entities. It forms the basis for development of the Physical Data Model, which results in physical implementation.

ii) Physical Data Model

Include the model elements (e.g., tables, views and constraints) that represent the detailed physical structure of the database and model elements (e.g., schemas and table spaces) that represent the underlying data storage design of the database. For Agile approach, the Physical Data Model will be refined during implementation and therefore is unnecessary at the SA&D phase.

5.4.2 Key Roles & Responsibilities

Table 18 - Key Roles & Responsibilities for System Design

Who will be involved?	
Roles Involved	**Key Responsibilities**
1. Business Analyst	(a) Collaborate and communicate with stakeholders to manage their expectations, needs and changes in requirements. (b) Provide support to users and coordinate with IT project team to help review the design of the IT system from the business perspective, resolve issues/conflicts among stakeholders.
2. Systems Analyst / Systems Architect	(a) Design the logical and physical data models, application components and interfaces for a system to satisfy specified requirements. (b) Define the security design and controls to satisfy security policies, guidelines and requirements. (c) Develop the user interface architecture, wireframes and visual design.

Who will be involved?	
Roles Involved	**Key Responsibilities**
	(d) Conduct review for: i) Application Design ii) User Experience Design iii) Logical Data Model iv) Physical Data Model (e) Produce the System Design and System Specification. (f) Document and update the RTM.
3. PAT / PSC	Review the System Specification: i) Functional Specification • Required System Overview • Function Definition ii) Architecture Design • Application Architecture • Data Architecture iii) System Design • Application • Data

5.4.3 Key Tasks & Deliverables

Table 19 - Key Tasks & Deliverables for System Design

What will be delivered?	
Key Tasks	**Tools & Templates**
1. Design Application * * This key task can be conducted in parallel with other key tasks with asterisk (*) (a) Divide each subsystem into manageable components. i) For each subsystem, divide its implementation functionality into different logical components. Use modelling tools, such as class diagram to show the components of subsystems. Define the component's functionality and characteristics, such as operations, interface, attributes, preconditions/post conditions, and relationships. The components can be underlying services or custom software programs that form an independent and	• Class Diagram • Sequence Diagram *and/or* Activity Diagram *and/or* State Diagram

What will be delivered?	
Key Tasks	**Tools & Templates**
irreplaceable part of the subsystem and fulfil a clear function. (b) Model functionality flow between components. i) Use modelling tools such as Sequence or Activity diagrams to identify the internal flow of control between components. These tools help decompose the subsystem in further components or group the components together. (c) Define business rules. i) Analyse the business rules and capture the details about the rules including: ● Rule conditions and actions captured using inference rules such as if-then or decision trees or decision tables. ● Metadata information such as rule categorisation, rule hierarchy level, etc. ● Time period validity for rule. ● Rule constraints such as time based execution or overwritten by other rules. ● Rule priority. ● Business validation constraints. ● Preconditions. ● Rule dependencies and chaining. ● Rule execution sequences. ii) Design implementation of business rules or decision services to be separated from the application functionality. This separation helps develop, manage, and maintain the business rules without having to navigate through the implementation logic and code of the application to manage and maintain the business rules. iii) Review dependencies between the business rules. Minor changes to a rule may require affected dependent rules to be updated. (d) Design common frameworks. i) Design and review the common functions used throughout the application, such as error handling framework, logging framework, and persistence framework. List all such common frameworks, and model the services as a framework so that they can be leveraged consistently and uniformly across the application.	

What will be delivered?	
Key Tasks	**Tools & Templates**
ii) Identify system components. • Security - Assess sensitivities and define security controls. ♦ Review the functionality of each subsystem and the corresponding personally identifiable information (e.g., Hong Kong Identity card number) collected, used, stored and shared. ♦ For each subsystem, define security controls such as identity and access management, database security, network security and business continuity controls. ♦ Follow existing guidelines to help classify the subsystems and define security controls requirements. ♦ Develop and document a security controls implementation overview diagram which demonstrates where security controls will be implementation based on a user's view. This will help the user and system owners understand where controls are placed and how data will be protected. • Validation - determine how validations will be performed in the application. ♦ It is recommended that any non-trivial business validations be located in the business services layer instead of the presentation layer. This makes the validation logic available for use by other parts of the application which are not invoked through the User Interface (UI). • Transaction - The changes on a single entity occurrence as triggered by an event is "effect". The collection of all effects of an event is a transaction, which is either effective completely (i.e., "committed") or cancelled at all (i.e., "rollback"). • Logging - define the approach to record and retrieve diagnostic information. ♦ Diagnostic information may include the amount of time needed to execute a critical method, the number of transactions committed per second, or the number of users currently with active sessions. • Exception handling - define the approach to exception handling and how the exception will eventually be handled and logged and	

What will be delivered?	
Key Tasks	**Tools & Templates**
how it will be presented to the user. ♦ In general, the application should support unchecked exceptions. The guiding principle of exception handling in the application is that an exception should only be caught, and similarly a method should only declare a checked exception, if there is some valid response the invoker can make in response to that exception. ● Reference table - describe the approach to manage static look-up table or reference table information, such as country list, error messages. ● Internationalisation - describe the approach to handle internationalisation of the application. If internationalisation is not necessary, this should be stated. Considerations would include: ♦ Support for multi-lingual and other internationalised content. ♦ How database-backed internationalised content will be delivered. ♦ Support for different character sets.	
2. Design Integration * * This key task can be conducted in parallel with other key tasks with asterisk (*) (a) Analyse integration landscape. i) Analyse the integration flows with each end system, such as whether the messages are synchronous or asynchronous and any transactions that are supported between the interfaces. Use UML models such as class diagram to show the interfaces. ii) Analyse the type of message integration exposed by the interfaces with the end systems, such as Web service, point-to-point messaging, and queue-based messaging. (b) Create data mapping and transformation rules. i) Analyse the data types and data fields that will be exchanged between integration subsystem and end systems, and perform data element mappings. ii) Capture details including validation rules, transformation rules, cross-reference details, and mapping rules.	• Class Diagram *and/or* Sequence Diagram *and/or* State Diagram

What will be delivered?	
Key Tasks	**Tools & Templates**
(c) Design integration sub-system. i) Provide detail logic implemented within the integration subsystem for each interface. Use UML models such as sequence diagram to depict the logic. Document details such as how the main starter process calls the sub-processes, how the sub-processes are used, and what tasks are performed by the sub-processes. Provide details on sub-process names, input and output fields, and data types. Give adequate attention to the interface description and handshake between the interfaces as any change in the interface description can be costly. ii) Define details on handling of exceptions, usage of global variables, and implementation of logging and security frameworks. Document details on usage of any proprietary Application Programming Interface (API) names or design time libraries, field types, field names, and input/output details that will be used.	
3. Design Data Conversion * * This key task can be conducted in parallel with other key tasks with asterisk (*) (a) Design data conversion process flow and procedures. i) Determine the source data components and elements, their business and technical descriptions, existing logical and physical relationships among the components, cardinality of elements, and constraints such as minimum and maximum lengths, data type, and allowable values of data elements. ii) Determine the expected results of the data conversion process. iii) Determine any business rules that need to be executed during the conversion process. iv) Determine the data conversion tasks that must be carried out in advance or during the conversion runs, and create process flow and data flow diagrams to depict dependencies. v) Determine error handling requirements and include error handling design in process and data flows. (b) Design extraction programs. i) Define the subsystems that make up the data extraction programs. Design the programs that extract data from the different sources of	N/A

What will be delivered?	
Key Tasks	**Tools & Templates**
data stores. • Identify execution frequency of the extraction programs based on business and technical needs and limitations. • In case of multiple subsystems in an extraction program, identify the sequence of execution and dependencies among the subsystems. • Identify criteria for data eligible for extraction, depending on the business and technical requirements. • Identify data size limitation for extraction based on business and technical needs. • Identify error conditions, and design extraction programs to handle and report errors. (c) Design conversion, cleansing, and loading programs. i) Define detailed design to implement business rules required during the conversion and cleansing. ii) Design cleansing programs to resolve any inconsistencies in the data related to syntax, definitions and formats. iii) Design conversion programs to take the cleansed source data and convert it to the target data format. iv) Design programs to load the cleansed and converted source data into target data store. Load the data to maintain the referential integrity of the target data store throughout the process. (d) Sub-deliverable's sections: Application Design	
4. Design User Experience * * This key task can be conducted in parallel with other key tasks with asterisk (*) (a) Define UI information architecture. i) Define UI information architecture such as a site map to provide a top-down view of how the users will interact with various pages, functions, and content and identify the screens or pages that would be developed for the application and their relations. • When reviewing the site map, consider framing it in the context of a specific user scenario that depicts a day in the life to ensure completeness.	• Prototyping Techniques

What will be delivered?	
Key Tasks	**Tools & Templates**
(b) Finalise visual designs. i) Gather feedback from stakeholders and update the appropriate number of screen/report/form mock-ups needed to finalise the design. (c) Develop UI design. i) Identify and align the data entry and display elements by type (e.g., button, label) with the Logical Data Model. ii) Identify the validation and error messages that will be used. iii) Identify the necessary security roles that have access to the screen/form/report. This includes screen level (coarse grain) and field level (fine grain) access. (d) Sub-deliverable's sections: User Experience Design	
5. Design Logical Data Model * * This key task can be conducted in parallel with other key tasks with asterisk (*) (a) Define entities. i) Identify the candidate names for each business entity that represents a collection or grouping of similar data elements. Then, create a comprehensive list of data elements that will be included in the solution. (b) Define relationships. i) Define relationships between the various entities, expressed as the relationship name, relationship type (such as simple, aggregation, or composition), and the multiplicity of the relationship (one to one, one to many, or many to many). (c) Define entity attributes. i) Attributes are the specific logical data elements that require storage within the application. Specify a list of individual attributes comprising each entity in the model, along with a short description of the business information each attribute represents. (d) Determine data use and volumes. i) Determine the data use and volume for entities, the goal is to identify entities that might be candidates for an unusually large volume of data or might require heavy or unusual usage patterns. Take these candidates into account when developing the physical	• Entity Relationship Diagram *and/or* Class Diagram

What will be delivered?	
Key Tasks	**Tools & Templates**
data model as they might have a significant impact on how the physical data store is designed. (e) Normalise data model. i) Conduct normalisation which re-structures the entities to eliminate ambiguity and duplicity of data and to make the model more flexible. ii) Take an iterative approach when inspecting the model to ensure each entity contains only attributes that define that particular entity or its relationship to another distinct entity. If required, subdivide entities into one or more simpler, more cohesive entities that relate to one another.	
6. Design Physical Data Model (a) Create primary key and unique constraints. i) Define the one or more columns that uniquely identify a row in the table. Apply constraints on primary key columns that guarantee the uniqueness of the data or collection of data. (b) Create and verify referential integrity (Foreign Keys) enforcement rules. i) Define mechanisms that maintain the integrity of the data. Data integrity rules, also known as constraints, should be defined which verify that data values lie within defined ranges. (c) Normalise / Denormalise data model. i) Refine the data model using normalisation and denormalisation. Normalisation includes creating tables and establishing relationships between those tables, while denormalisation combines columns from two or more tables into the same table. (d) Create database access controls. i) Define database users and roles to maintain database security. Include additional database security measures, such as audit columns, to record information on transactions if necessary. (e) Define storage characteristics. i) Design the space allocation and disk page organisation of the database based on the data use and volumes. (f) Define database. i) Define tablespace, synonyms, views, triggers, store procedures, indexes, constraints, etc.	• Entity Relationship Diagram *and/or* Class Diagram

What will be delivered?	
Key Tasks	**Tools & Templates**
(g) Sub-deliverable's section: Data Model	
7. Produce System Specification (a) Include the contents in System Design produced in this process, as well as the Architecture Design and Functional Specification produced in previous process in one document. i) System Specification is just a collection of documents that provides the detailed functions, the designs of the system and architecture. The System Specification should not contain any requirement information. (b) Sub-deliverable: System Design	N/A
8. Review and Signoff (a) As System Design is the final sub-deliverable of System Specification, the following task is to review and signoff the whole document of System Specification. (b) Cross check the traceability and relevancy between system functions and requirements. (c) The System Specification (i.e., Functional Specification, Architecture Design and System Design) needs to be reviewed by PAT to confirm the correctness and completeness. (d) Once the contents are confirmed to be accurate and complete, the PSC can either signoff the System Specification or wait and signoff the whole SA&D report after the SA&D Report is completed, subject to the decision of the project owner. (e) Document and update RTM to establish and maintain traceability as it evolves throughout the project lifecycle. (f) Deliverable: System Specification	• Action Checklist (System Specification) • RTM

5.4.4 Tools & Templates

(a) The list below includes the recommended tools, techniques, and checklist applicable to System Design:

i) Prototyping Techniques: Please refer to ***Appendix A - Tool 2*** for details

ii) Activity Diagram: Please refer to ***Appendix A - Tool 5*** for details

iii) Sequence Diagram: Please refer to ***Appendix A - Tool 6*** for details

iv) Action Checklist: Please refer to ***Appendix A - Tool 10*** for details

v) Entity Relationship Diagram: Please refer to ***Appendix A - Tool 18*** for details

vi) Class Diagram: Please refer to ***Appendix A - Tool 19*** for details

vii) Security Controls Implementation Overview Diagram: Please refer to ***Appendix A - Tool 20*** for details

viii) State Diagram: Please refer to ***Appendix A - Tool 27*** for details

(b) The template can be found in ***Appendix C – Section 4.3: System Design***.

5.4.5 Hints & Tips 

Table 20 - General Tips for System Design

(a) Data:

i) Although the Logical Data Model can be high-level detailed for certain projects, it is still important to develop and document the model as it may form the basis for the development of the Physical Data Model and support design of the system

ii) Naming conventions for data model may be developed (or referred to if existing), which help maintain consistency and allow the designers and developers to understand the business meaning of an entity, its attributes, or the relationship between the attributes.

iii) De-normalising more than two tables is rare and it increases the cost of inserts and updates as well as the cost of non-join queries. Limiting de-normalisation to two tables is a best practice unless strong and convincing evidence can be produced regarding the benefits.

iv) If Agile approach is adopted, a high level logical / conceptual data model showing relationships of key entities should be sufficient in the SA&D phase and it is not necessary to design every single detail in the models. It is expected that the models are not yet finalised at this stage but will be gradually evolved during System Implementation phase.

(b) Application:

i) Integration requires both technical (infrastructure and applications) and financial (sharing integration costs) cooperation between the subsystems. Consensus from all the concerned parties during the analysis and design stage is recommended.

ii) There is no predetermined number of mock-ups that have to be created. If the primary purpose is only to finalise the look and feel, then only two or three key screens can be designed and the rest can be designed as needed.

iii) A best practice for developing wireframe is to use “lorem ipsum” (Latin gibberish) to depict text or data on a screen in order not to detract the reviewer from worrying about content over design.

(c) Keep the design as simple as possible and avoid including anything that you don't need.

(d) At times, it is very difficult to capture all information in one single diagram for large subsystems, therefore it is always recommended to breakdown large subsystems into smaller manageable subsystems.

(e) A prior long-term planning on the application framework, covering core subsystems of the different layers of the system, may be necessary for generalisation of common functions, performance consideration and better code maintenance.

6 TECHNICAL SYSTEM OPTION

(a) TSO process refers to the combination of several documentations, including Technical System Architecture, Sizing Model, Cost / Benefit Evaluation, Impact Analysis and Implementation Plan, which will form one deliverable, i.e., the **TSO**.

(b) Please refer to ***Figure 5***, which summarises the key tasks and deliverables produced under TSO.

6.1 FLEXIBLE ADOPTION OF PROCESS WITHIN TSO

(a) The decision tree below in ***Figure 8*** is a recommendation on under what consideration factors a process within the TSO could be simplified.

(b) For the detail explanation of the consideration factors, please refer to the "***Part II – How to Use this Guide***".

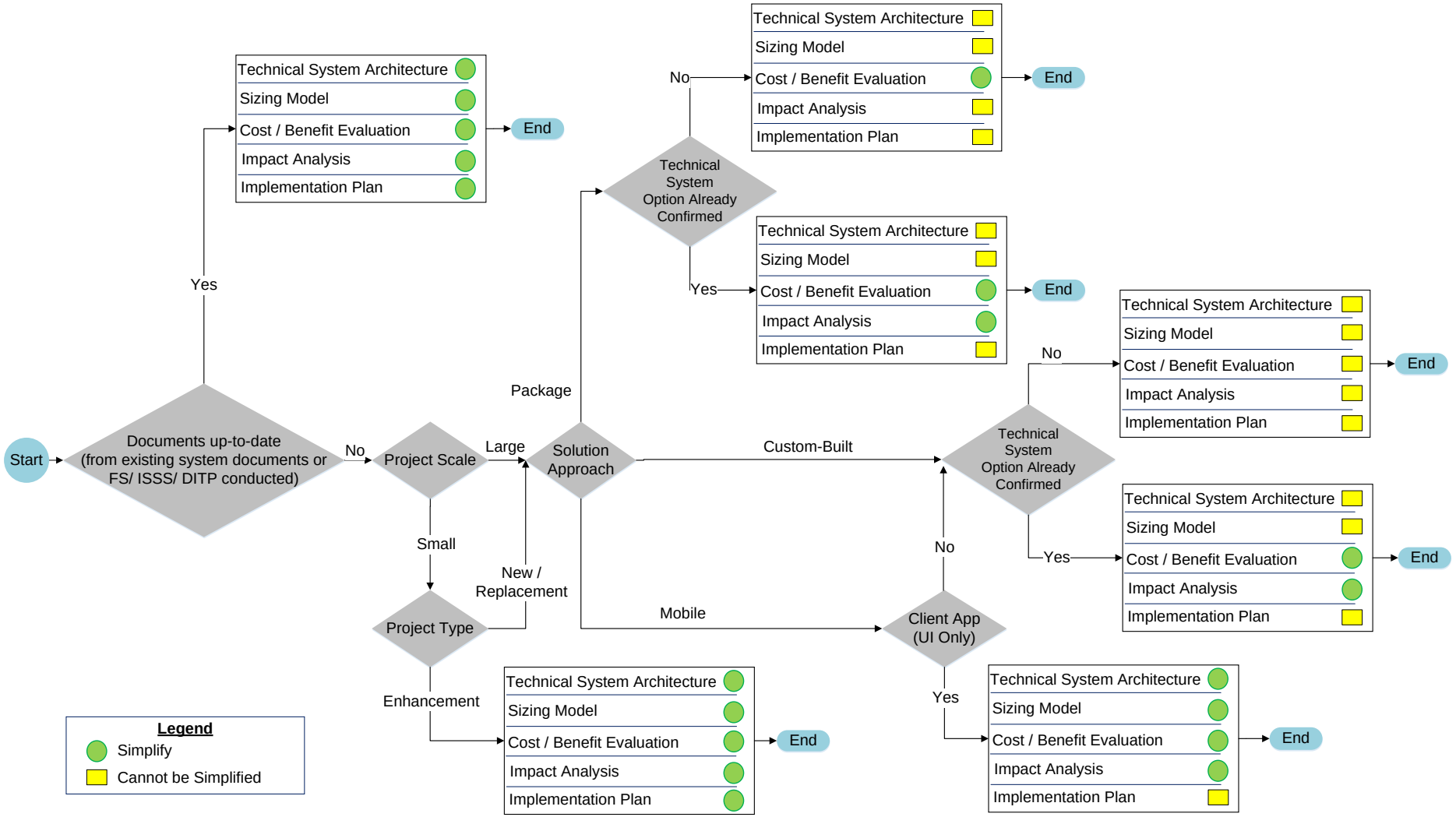

Figure 8 - Decision Tree for TSO

6.2 TECHNICAL SYSTEM ARCHITECTURE

(a) Technical System Architecture defines the standard hardware and hosting platforms, network architecture, as well as storage architecture in the Government business domains in the implementation of business solution. The goal of the technical architecture is to enable the agility to address business needs, and outline a road map for adapting technologies to changes in the IT market place.

(b) Three domains will be covered under the Technical System Architecture of different environments (e.g., development, systems integration testing, user acceptance testing, production and DR):

i) Network Architecture

ii) Storage Architecture

iii) Platform Architecture

6.2.1 Overview & Approach

6.2.1.1 Network Architecture

Network Architecture is to address the linkages and connections between networks, segmentations, different physical nodes (e.g., systems, servers, sites) and their concerns, which should consider one or more of the following topics, such as:

i) Government Backbone Network

- Considerations should be put on interconnection between B/Ds or Government common services if applicable, such as use of GNET.

ii) Inter Data Centre Links

- Bandwidth capability for Wide Area Network (WAN) on new system, replication, storage, backup, restoring and archiving should be considered. Connectivity between different offices and environment (development site, production site, DR site, etc.) should also be taken into account.

iii) Access Network

- Considerations should be given to the reliability and availability of Local Area Network (LAN), provisions of Wi-Fi networks, replacement of standalone network switches to modular network equipment, as well as the supportability of 24/7 services, etc.

iv) Network Access Control

- Network security should be taken into consideration, such as Identity and Access Management (IAM), Defense-in-Depth (DiD) or tiered architecture, Virtual Private Network (VPN) & Multiprotocol Label Switching (MPLS)

connection, user-based network access control, application firewall, network monitoring, control of external and internal user devices, Government shared-service Demilitarised (DM) Zone, etc.

6.2.1.2 Storage Architecture

(a) In the design of storage architecture that will support the applications and systems, consideration should be given to one or more of the following topics:

i) Network Storage

- Shared storage solutions to provide high reliability, performance, and capacity requirements should be considered. Solutions include Direct Attached Storage (DAS), Network Attached Storage (NAS) and Storage Area Network (SAN). Technologies design with the storage solution should also be considered, including redundancy hardware design, multi-site real time data replication, automated multi-tier storage architecture and unified storage architecture.

ii) Completeness of Records

- Considerations should be given to the completeness of record keeping during creation, maintenance and replication by using data management software tool such as Content Management System (CMS) and Database Management System (DBMS) for structured and unstructured data, backup and archiving solution.

(b) Storage Architecture also involves information regarding data store design, e.g., Data resilience, data partitioning, data replication, data backup, etc.

6.2.1.3 Platform Architecture

Platform Architecture design covers client / server platform (e.g., application servers or web servers) and data centre design. Server platform depicts the server architecture of the backend, whereas client platform addresses the systems with which end-users will be physically interacting, and hosting design addresses the design of the facilities that will host the centralised technical components. One or more of the following topics should be covered:

i) Shared Infrastructure Services

- Options on leveraging shared infrastructure services to enhance service levels and reduce operation costs should be considered, such as the examples listed below:
  - e-Government Infrastructure Service (EGIS);
  - Government Cloud Infrastructure (GovCloud);
  - Central Computer Centre (CCC);
  - Central Cyber Government Office (CCGO);

- Common Service Portal (CSP);
- Central Internet Gateway (CIG) System; and
- Government Communication Network (GCN).

ii) Server and OS Consolidation

- It is important to consider the efficiency to use computer server resources such as server virtualisation, better usage of physical space and supporting resources such as electricity and cooling.

iii) Hosting Services

- Consideration should be given to hosting services by physical server versus utilising server virtualisation. Server virtualisation for IT applications is done by pooling computing resources across multiple physical servers. Virtualisation technology is suitable for CPU-intensive systems or systems with high I/O to improve manageability, scalability, and high availability. The adoption of cloud computing and the use of infrastructure and platforms resources as a service should also be considered.

iv) Modularised Hardware

- Considerations should be given to selecting hardware with modularised design as far as possible to enhance maintainability, which will in turn enhance serviceability.

v) Clustering

- Clustering systems, which is essentially the linking together of servers to support a single function, provides high-availability (i.e., redundant nodes to minimise impact of one server failing) and load-balancing (i.e., computational workload shared by multiple computers) to enhance system service levels.

vi) Client (Hardware & OS)

- Considerations should be given to the standardisation and unification of client environment (e.g., OS, web browser, smartphones) for internal usage, and the support of wide variety of hardware and OS being used in the public.

6.2.2 Key Roles & Responsibilities

Table 21 - Key Roles & Responsibilities for Technical System Architecture

Who will be involved?	
Roles Involved	**Key Responsibilities**
1. Systems Analyst	(a) Identify and design the three domains architecture (network, storage, and platform) based on the Functional Specification, Architecture Design, as well as the System Design created in the previous activities. (b) Take into consideration the topics mentioned in the above ***Overview & Approach***, in different environments, including pre-production, development, production, conversion, testing, and DR. (c) Cross check the traceability and relevancy between Technical System Architecture and the Functional Specification, Architecture Design, and System Design. (d) Document and update the RTM for Technical System Architecture.
2. Systems Architect	(a) Provide expertise domain knowledge on network, storage, and platform, such as the industry trend on using SAN, NAS, etc. (b) Liaise with vendors and research on market trends to provide up-to-date technologies that are feasible to the department.

6.2.3 Key Tasks & Deliverables

Table 22 - Key Tasks & Deliverables for Technical System Architecture

What will be delivered?	
Key Tasks	**Tools & Templates**
1. Identify Technical System Architecture Components (a) Review the following documents and identify technical system architecture components in three domains (i.e., network, storage, platform), which will help to conduct the architecture design: i) Functional Specification ii) Architecture Design iii) System Design (b) Consult Systems Architect and capture the insights and knowledge on the three domains of technical system architecture. i) Network Architecture ii) Storage Architecture iii) Platform Architecture (c) Document the following preliminary solutions and prepare material to present to PAT for review, if applicable: i) Network Diagram • Illustrates the network architecture. ii) Data Storage Diagram • Illustrates how the data are stored and what types of storage solution are used. iii) Server Diagram • Illustrates how the servers in different locations are connected with one another.	• Network Diagram *and/or* Data Storage Diagram *and/or* Server Diagram
2. Assessment of Technical System Architecture Components (a) Present the preliminary solutions to PAT to align the expectation on the Technical System Architecture in terms of objective, budget, market trend, and the market availability. (b) Conduct the assessment of whether the solutions are available in the market / through the vendor, as well as the services provided by shared infrastructure would be feasible in the future environment (e.g., the application is used in a remote location with limited network coverage).	

What will be delivered?	
Key Tasks	**Tools & Templates**
3. Design Technical System Architecture (a) Based on the Functional Specification, Architecture Design, and System Design created in the previous process, as well as the feasible preliminary solution to design the technical system architecture for the following domains: i) Network Architecture ii) Storage Architecture iii) Platform Architecture (b) Taken into account the considerations as mentioned in above Overview & Approach, for each of the three domains. (c) Consider the architecture solution for the three domains in terms of backup, restore and archiving (d) Also consider the architecture solution for the following environments: i) Pre-production ii) Development iii) Production iv) Conversion v) Testing vi) Disaster recovery (e) Capture the technical system architecture design by diagrams as mentioned in Tools & Templates in section 6.2.4 below (f) Sub-deliverable's section: Network Architecture; Storage Architecture; and Platform Architecture (g) Sub-deliverable: Technical System Architecture	• Block Diagram *and/or* Network Diagram *and/or* Data Storage Diagram *and/or* Server Diagram

6.2.4 Tools & Templates 

(a) The list below includes the recommended tools, techniques, and checklist applicable to Technical System Architecture:

i) Action Checklist: Please refer to ***Appendix A - Tool 10*** for details

ii) Requirements Traceability Matrix: Please refer to ***Appendix A - Tool 11*** for details

iii) Network Diagram: Please refer to ***Appendix A - Tool 21*** for details

iv) Data Storage Diagram: Please refer to ***Appendix A - Tool 22*** for details

v) Server Diagram: Please refer to ***Appendix A - Tool 23*** for details

(b) The template can be found in ***Appendix C – Section 5.1: Technical System Architecture***.

6.2.5 Hints & Tips

Table 23 - General Tips for Technical System Architecture

(a) Consider security and requirements for high availability and DR in each architecture.
(b) Beware of limitations imposed by existing network, storage or platform architecture.

6.3 SIZING MODEL

(a) The Sizing Model documents the sizing information of the required system, to form a basis for deriving the configuration of the system for costing purposes at later stages.

(b) The sizing information identified would be used as a basis (and supporting document) for determining the hardware and software configuration for the production environment.

6.3.1 Overview & Approach

Depending on the platform, sizing should typically involve the following items:

i) Data Storage

ii) Transaction Rate

iii) Data Access

iv) Server Sizing

v) Network Sizing

6.3.2 Key Roles & Responsibilities

Table 24 - Key Roles & Responsibilities for Sizing model

Who will be involved?	
Roles Involved	**Key Responsibilities**
1. Business Analyst	Assist to collect business operation statistics from Business Users for IT project team to determine sizing requirements.
2. Business User	Provide the business operation statistics.
3. Systems Analyst	(a) Collect and gather existing capacity related documents and information (if applicable). (b) Some of the information such as data volumes and frequencies should have been captured in the CED document. (c) Collect the business operation statistics (e.g., the number of transactions per day, the number of concurrent users). (d) Calculate the sizing model for the system based on the information and statistics captured. (e) Verify and confirm the sizing model results with IT project team.

6.3.3 Key Tasks & Deliverables

Table 25 - Key Tasks & Deliverables for Sizing Model

What will be delivered?	
Key Tasks	**Tools & Templates**
1. Work out the Sizing Model (a) Determine all the data storage requirements within the system, such as: i) Database ii) Flat files (b) For calculating the database storage, project team should gather the following information to assist in the calculation (these information should have been captured in the System Design - Logical Data Model): i) Record Length of each data entity. ii) Number of Records of each data entity. iii) Annual Growth Rate of each data entity. iv) Key Length of each data entity. v) Number of Keys of each data entity. (c) Database storage calculations should include Raw Data Storage and Index Storage. (d) Determine the transaction rate for assisting in the calculations for data access. (e) The following information needs to be captured to calculate the transaction rate: i) Transaction Volume of each function ii) Annual Growth Rate of each function (f) Determine the data access for calculating the CPU sizing for the database server, i.e., the tpm-C (transaction per minute - Benchmark C) (g) For each function identified, determine the data entities in which it will access as well as the access type, i.e., Retrieval [R], Update [U], Insertion [C] or Deletion [D] (h) For each server within the system, the following sizing should be determined: i) CPU sizing ii) Memory sizing iii) Internal disk storage sizing (i) Memory sizing usually include the followings: i) System Software	N/A

What will be delivered?	
Key Tasks	**Tools & Templates**
ii) Application Software iii) Data Buffer iv) System Buffer Cache (usually a percentage of the total of System Software, Application Software and Data Buffer) (j) Internal disk storage sizing usually include the followings: i) Operating System ii) Swap Space iii) Crash Dump Area iv) Application Software v) Working Space vi) Server Log (k) Identify all the different access points and sites of the system and determine the overall network sizing requirement i) For each access point, collect the following to determine the network sizing requirement: • The number of concurrent users during the peak hour • The average bandwidth requirement for each user ii) For system with different sites such as production site to DR site, determine the strategy to be adopted to determine how frequent and what data needs to be transferred across and hence, determine the network sizing requirements. (l) Sub-Deliverable: Sizing Model	

6.3.4 Tools & Templates

(a) Project team may also refer to the guidelines on sizing estimation for specific platform (e.g., Mainframe/Midrange/PC Server) from the vendor.

(b) The template can be found in ***Appendix C – Section 5.2: Sizing Model***.

6.3.5 Hints & Tips

Table 26 - General Tips for Sizing Model

Different solutions may have different parameters to determine the sizing, e.g., database parameters are different to document management systems, it is suggested that Project team could leverage the "Sizing Model Guide" provided by vendors to calculate the sizing model.

6.4 COST / BENEFIT EVALUATION

The Cost / Benefit Evaluation process covers the evaluation in terms of cost effectiveness between the recommended system option and the proposed system option(s). It serves to evaluate the selected technical system options against other options only so that it facilitates project team select the best option in TSO.

6.4.1 Overview & Approach

It covers estimation of the following items:

i) Costs
- Non-recurrent costs (e.g., one-off system implementation costs like hardware cost, software cost, data conversion cost and system development cost and staff effort (in-house or outsourced), etc.)
- Recurrent costs (e.g., system maintenance costs, system operating costs, staff effort (in-house or outsourced), etc.)

ii) Benefits
- Tangible benefits
- Intangible benefits

6.4.2 Key Roles & Responsibilities

Table 27 - Key Roles & Responsibilities for Cost / Benefit Evaluation

Who will be involved?	
Roles Involved	**Key Responsibilities**
1. Business Analyst	(a) Help to identify and estimate the business-related non-recurrent costs and recurrent costs as well as tangible and intangible business benefits of the proposed technical system option(s). (b) Help to verify and walkthrough the estimated business-related costs and benefits with business users. (c) Provide the above estimated business-related costs and benefits to Systems Analyst for them to conduct the cost / benefit evaluation of the selected technical system option against other proposed options.
2. Systems Analyst	(a) Estimate all IT-related non-recurrent and recurrent costs and non-business related tangible and intangible benefits for the proposed technical system options. (b) Confirm the calculations on technical assumptions (e.g., license discount) and costs.

Who will be involved?	
Roles Involved	**Key Responsibilities**
	(c) Verify the estimated costs and benefits with related project stakeholders. (d) Conduct cost / benefit evaluation of the technical system options. (e) Recommend the technical system option that delivers the greatest business value.

6.4.3 Key Tasks & Deliverables

Table 28 - Key Tasks & Deliverables for Cost / Benefit Evaluation

What will be delivered?	
Key Tasks	**Tools & Templates**
1. Perform Cost / Benefit Evaluation (a) Gather existing cost information (both non-recurrent and recurrent costs and benefits) as baseline reference from peers, business users, IT project team or sources such as ISSS or DITP. (b) Based on information gathered, identify gaps between current and future business processes, and estimate the tangible and intangible benefits that entail. (c) Deduce current and future costs, whilst identifying the difference between them. (d) Estimate costs for the following areas for different environments (e.g., Production, UAT and Development Environment): i) Hardware ii) Software iii) Network equipment iv) Site preparation v) System implementation • Data conversion • Implementation fees for vendors • Training vi) System support vii) Business users involved in the operation of the related systems (e) Estimate benefits for the following areas: i) Reduced staff costs	• Cost / Benefit Evaluation Technique

What will be delivered?	
Key Tasks	**Tools & Templates**
ii) Reduced operating expenses iii) Cost avoidance iv) Increased revenue (f) List out the relevant assumptions for the calculation. (g) Sub-Deliverable: Cost / Benefit Evaluation	

6.4.4 Tools & Templates

(a) The list below includes the recommended technique applicable to Cost / Benefit Evaluation:

i) Cost / Benefit Evaluation Technique: Please refer to ***Appendix A - Tool 24*** for details

(b) The template can be found in ***Appendix C – Section 5.3: Cost / Benefit Evaluation***.

6.4.5 Hints & Tips

Table 29 - General Tips for Cost / Benefit Evaluation

(a) When calculating the future costs, leave sufficient buffer to provide greater flexibility in purchasing future costs and services.
(b) The Cost / Benefit Analysis in Project Management Plan (PMP) supports the business case and funding applications of the project, Cost / Benefit Evaluation in this process only aims to justify why a specified technical optional is selected over the others.
(c) Refer to ISSS, DITP or PMP for similar inputs to their Cost/Benefit Analysis, e.g., cost information and potential benefits.

6.5 IMPACT ANALYSIS

(a) Impact Analysis involves studying the effects of the introduction of the proposed system on user environment from both business and technical perspectives. This analysis allows the Project team to clearly identify what and how the proposed system impacts the users and B/Ds.

(b) The following perspectives may be considered when conducting the analysis:

i) The Impact during the implementation of the System

ii) The Impact after the implementation of the System

iii) The savings achieved after the implementation of the System

6.5.1 Overview & Approach

It is recommended to include the following:

i) Summary on system change/enhancement

- Provide a summary of system changes/ enhancements

ii) Effect on organisation and staffing levels

- For each recommended initiative, highlight the impact they may cause on the respective aspects

iii) Significant changes in user operating procedures

- From users perspective, highlight operational changes and potential challenges they may face

iv) Implementation considerations

- Elaborate on the impact the system may have on the organisation, and measures to resolve the mentioned issues. Implementation considerations range from training to effects on inexperienced staff at service level.

v) Savings on replaced equipment and associated costs

- A set of estimated calculations on savings that may be achieved through replacing existing system/ equipment and avoidance of the associated costs

vi) Risk Analysis

- List out a preliminary set of foreseeable risks in terms of Project Management, Staffing and Resources and Change Management arising from implementing and introducing the proposed system

6.5.2 Key Roles & Responsibilities

Table 30 - Key Roles & Responsibilities for Impact Analysis

Who will be involved?	
Roles Involved	**Key Responsibilities**
1. Business Analyst	(a) Review the organisation readiness for the implementation of the proposed system: i) Understand the changes that will probably occur to business and users. ii) Identify the potential impacts of the proposed system to the organisation such as the organisation culture, its business units and stakeholders by referring to the Current Business Process and URD for business impact and the Cost / Benefit Evaluation of the TSO for cost savings. (b) Propose any needs for training and change management to be conducted in association with the implementation.
2. Systems Analyst	(a) Compare the CED against the proposed technical design in the System Specification and TSO processes to determine implementation impact/ considerations. (b) Identify the differences between the current and future environment, and deduce the quantitative and qualitative impact changes may have on the IT environment.

6.5.3 Key Tasks & Deliverables

Table 31 - Key Tasks & Deliverables for Impact Analysis

What will be delivered?	
Key Tasks	**Tools & Templates**
1. Perform Impact Analysis (a) Review and analyse the CED, URD, System Specification and TSO processes and PMP to derive the impact on the organisation such as training, changes in operating procedures, changes in staffing, etc. (b) Identify issues related to organisational, policy or management, and determine what the impacts on the user environment are. i) Refer to CED, TSO documents to identify and derive technical impact.	N/A

What will be delivered?	
Key Tasks	**Tools & Templates**
ii) Refer to the URD to identify and derive business impact. iii) Both Current and Future Business Processes, as well as URD to identify and derive business impact. iv) Refer to Cost / Benefit Evaluation for cost savings. (c) Highlight the identified impacts (both business and technical aspects) to Stakeholders. (d) Propose potential solutions to stakeholders prior to confirming the overall System Design, discuss long term feasibility and consider an incremental delivery approach. (e) Sub-Deliverable: Impact Analysis	

6.5.4 Tools & Templates

(a) There is no recommended tool and technique provided for Impact Analysis.

(b) The template can be found in ***Appendix C – Section 5.4: Impact Analysis***.

6.5.5 Hints & Tips

Table 32 - General Tips for Impact Analysis

(a) Focus on the impacts the new system may have on users, costs and major technical changes e.g., significant changes to users' existing mode of business operation, changes in operation cost, major technical changes that could potentially impact service levels. (b) Refer to PMP for inputs to Impact Analysis, such as risk register and stakeholder register.

6.6 IMPLEMENTATION PLAN

6.6.1 Overview & Approach

The Implementation Plan should include the following:

i) Implementation Strategy

- Describe the implementation approach that will be adopted, based on a desire to minimise dependencies or interfacing requirements, cost and organisational change.
- Highlight the key considerations and assumptions on which the strategy is based.

ii) Implementation Schedule

- Implementation Timeline of the proposed system should take the following activities into account and add appropriate buffer:
  - Requirements
  - Analysis and Design
  - Development
  - Testing
  - Deployment
  - Documentation
  - Meetings and Review

iii) Activities

- Describe each of the activities identified in detail. The document should also highlight contingency plan for critical applications.

6.6.2 Key Roles & Responsibilities

Table 33 - Key Roles & Responsibilities for Implementation Plan

Who will be involved?	
Roles Involved	**Key Responsibilities**
1. Business Analyst	(a) Help the IT project team to collaborate with the key project stakeholders to collect their input on implementation strategy and milestones in implementation. (b) Discuss and confirm the Implementation Plan proposed by the IT project team with key stakeholders of each stakeholder group.

Who will be involved?	
Roles Involved	**Key Responsibilities**
2. Business User	Discuss and confirm milestone dates with Business Analyst and Systems Analyst.
3. Systems Analyst	(a) Define the implementation strategy and propose an Implementation Plan based on input from key project stakeholders. (b) Discuss and confirm milestone dates with Business Analyst and business users. (c) Confirm whether the proposed timeframe is technically feasible (counter-propose if required).
4. Internal Project Manager	Check and review if the proposed Implementation Plan is aligned with the original PMP, and review the need for a change request to amend the PMP as appropriate.
5. PAT / PSC	Review the Technical System Option: i) Technical System Architecture • Network Architecture • Storage Architecture • Platform Architecture ii) Sizing Model iii) Cost / Benefit Evaluation iv) Impact Analysis v) Implementation Plan

6.6.3 Key Tasks & Deliverables

Table 34 - Key Tasks & Deliverables for Implementation Plan

What will be delivered?	
Key Tasks	**Tools & Templates**
1. Work out the Implementation Plan (a) Consult business users for their input on the Implementation Strategy, such as big bang or phased rollout, and devise preliminary implementation milestones/ plan (b) Based on the preliminary Implementation Strategy and Implementation Milestones, fine-tune the milestones with IT project team's input (c) Perform the timeframe estimation by various estimation methods such as:	• Gantt Chart

What will be delivered?	
Key Tasks	**Tools & Templates**
• forecast the schedule according to the available resource once the required manpower resources have been estimated with the resource estimation techniques of Function Point Analysis, Work Breakdown Structure or Product Breakdown Structure which can be found in "*Resources Estimation Guide* [7]"; • forecast timeline for entire project based on past projects experience; • estimate time required by making reference to similar tasks of other project(s); • use the amount of time spent in planning phase to forecast the timeline required for the entire project; or • a mix of these/other techniques. (d) Verify the Implementation Strategy and fine-tune the Incremental Delivery Plan of the Implementation Plan with appropriate stakeholders (e) Based on the Implementation Plan, review the need for amendment to PMP or creation of change request as required (f) Confirm with Stakeholders on whether the proposed timeframe is technically feasible (counter-propose if required) (g) Sub-Deliverable: Implementation Plan	
2. Review and Signoff (a) Cross check the traceability and relevancy between Technical System Architecture, and the System Specification. (b) The selected TSO (i.e., Technical System Architecture, and optional: Sizing Model, Cost / Benefit Evaluation; Impact Analysis; and Implementation Plan) needs to be reviewed by PAT to confirm the correctness and completeness. (c) Once the contents are confirmed to be accurate and complete, the PSC can either signoff the selected TSO or wait and signoff the whole SA&D report after the SA&D Report is completed, subject to the decision of the project owner. (d) Deliverable: TSO	• Action Checklist (TSO)

6.6.4 Tools & Templates 

(a) The list below includes the recommended tool applicable to Implementation Plan:

i) Gantt Chart: Please refer to ***Appendix A - Tool 25*** for details

(b) The template can be found in ***Appendix C – Section 5.5: Implementation Plan***.

6.6.5 Hints & Tips 

Table 35 - General Tips for Implementation Plan

(a) Identify the complexity of each task, whilst keeping stakeholders informed through hosting regular status update meetings.

(b) Allow buffer in between project milestones when unforeseen circumstances.

(c) Consider the delivery lead time for hardware in the implementation plan, such as shipment, import and export clearance.

(d) Apart from tasks relating to putting the proposed system into production, other activities that should be considered include:

i) Prepare for cultural changes;

ii) Design conversion strategies (data to be converted, volume to handle and approach);

iii) Design production procedures;

iv) Create user and technical documentation; and

v) Conduct user training.

REFERENCE

1 “Practice Guide for Agile Software Development” can be found at http://www.ogcio.gov.hk/en/infrastructure/methodology/system_development/agile_software_development.htm

2 “Best Practices for Business Analyst” can be found at http://www.ogcio.gov.hk/en/infrastructure/methodology/system_development/best_practices_for_ba.htm

3 “Practice Guide for Scoping and Planning of Large-scale IT System Development Projects” can be found at http://www.ogcio.gov.hk/en/infrastructure/methodology/proj_mgmt/lsp.htm

4 “Security Risk Assessment & Audit Guidelines” can be found at http://www.ogcio.gov.hk/en/information_security/policy_and_guidelines/index.htm

5 “Practice Guide to Project Management for IT Projects under an Outsourced Environment” can be found at http://www.ogcio.gov.hk/en/infrastructure/methodology/proj_mgmt/pm_practice_guide_outsourced.htm

6 “Practice Guide on Developing Mobile App” (an internal document of HKSAR Government) can be found in ITG InfoStation at http://mobileappdevfac.host.ccgo.hksarg/en/practice_guide_on_development_mobile_native_application.htm

7 “Resources Estimation Guide” (an internal document of HKSAR Government) can be found in ITG InfoStation at https://itginfo.ccgo.hksarg/content/best_practice/resources_estimation.htm

GLOSSARY

Table 36 - Glossary to facilitate the consistency of terms

Term	Definition
Acceptance criteria	Those criteria, including performance requirements and essential conditions, which must be met before project deliverables are endorsed.
Agile	Agile is an iterative and incremental software development method, where requirements and solutions evolve through close collaboration between users and project teams. It promotes evolutionary development and delivery using iterative approach, and also encourages rapid and flexible responses to changing requirements.
Application Design	Detailed design of the various components of the system and the working relationships amongst them, to the extent that the design is sufficiently complete to begin software development.
Architecture Design	Tasks related to designing the architecture of the different software components and their interfaces to establish the framework for the detailed design of the system.
Business Analyst (BA)	BA is responsible for performing the business analysis functions for IT system development projects.
Business Assurance Coordinator (BAC)	BAC is a PAT member who is responsible for ensuring that business issues arising during the project are properly managed and coordinating the quality control activities from business perspectives.
Business case	A document used to justify the commitment of resources and investments to a project. It presents the financial logic behind the project.
Change	One of the dimensions of the project management methodology referring to any additions, deletions and/or modifications to the scope and project plan.
Context diagram	An analysis diagram can be used to provide users with a high-level overview about the IT system.
Change management	A practice to control any additions, deletions, and/or modifications to the scope and project plan.

Term	Definition
Change request	A formal request to change the scope, design, method, or other planned aspects of a project, usually including estimates of the effect to the project cost and schedule. It may arise through changes in the business or issues in the project. The document should be logged, assessed, and approved before a change to the project is made.
Contractor Project Manager	Lead person (outsourced individual), who is responsible for coordinating contractor resources to fill specified roles such as installation, customisation, training, or support, etc. The contractor project manager works closely with the internal project manager and also manages the project management aspects such as project resources, timeline, budget, risks, and issues, specifically within the activities being carried out by outsourced staff.
Control	One of the project phases of the project management methodology, referring to comparing actual performance with planned performance, analysing variances, assessing trends to effect process improvements, evaluating possible alternatives, and recommending appropriate corrective action as needed.
Cost / benefit evaluation	An evaluation on the costs and benefits of the selected technical system options against other feasible options that supports the decision of the selected technical system option.
Data conversion	Describes how data are extracted, transformed and loaded from the source data stores to the target data stores. If the goal is to evolve or replace a legacy system, data conversion will identify data from the old system to be converted into the new system's format.
Data model	Documents and organises data, how it is stored and accessed, and the relationships among different types of data. A data model helps with data planning and identifies potential problems that future data users might encounter.
Deliverable	Any tangible, verifiable outcome or item produced by the project.
Dependency	Relationships between activities, For example, one activity may be a result of several other "dependent" activities wherefore that activity may not begin until all dependent activities are complete.
Design approach	Describes how the available technology and architecture will be utilised to meet the identified requirements.
Developers	People who design and implement the system. They can be project team members (for in-house projects) or contractor team members (for outsourced projects).

Term	Definition
Departmental Information Technology Plan (DITP)	DITP is a medium-term IT Plan, covering a period of normally one to three years depending on the business planning horizon of the B/D. It should take stock of the progress of IT development of the B/D; identify areas where IT can be exploited to improve the overall efficiency, effectiveness and resources utilisation of the B/D; and set out IT requirements of the B/D in stages in the planned period.
Enterprise Architecture	Enterprise Architecture provides an enterprise view on IT capabilities and resources and helps to enable business-IT alignment. It establishes the framework, tools and techniques to help organisation to improve the alignment between its business goals and IT initiatives.
Feasibility Study (FS)	A research on the economic viability of proposed projects and provides a thorough analysis of the business opportunity, including a look at all the possible roadblocks that may stand in the way of the cooperative's success
Financial	One of the dimensions of the project management methodology describing the processes involved in estimating, budgeting, and controlling costs so that the project can be completed within the approved budget.
Functional Requirements	Define the capabilities and functionality of a proposed system from a business perspective.
Functional Specification	High-level description of a proposed system's behaviour which includes components that provides a functional overview of the system with respect to its features, its relationships and interactions with other systems and components.
Impact Analysis	An analysis that explains the effects of the introduction of the proposed system on the user environment from both business and technical perspectives.
Information Systems Strategy Study (ISSS)	A departmental ISSS is conducted to (i) assess the current ICT requirements of the department, and identify its long-term IT potential with regard to its business and operational strategies; and (ii) recommend a departmental IT strategic plan covering development and implementation over the next five years.
Integration	One of the dimensions of the project management methodology, referring to the processes and activities needed to identify, define, combine, unify and coordinate the various processes and project management activities within the project dimensions.

Term	Definition
Internal Project Manager	Lead person (Government resource) accountable for project planning and delivery. As the person responsible for meeting the project objective and ensuring project success, the project manager is expected to provide oversight and input to project management aspects such as project work plan, budget, quality, risks, and issues, as well as management of contractor resource performance through working closely with the contractor project manager in an outsourced project.
Issue	One of the dimensions of the project management methodology describing any problem that is presented to the project team for resolution. It can be a question (that requires clarification) about any aspect of the project (or project-related topic), statement of concern, suggestion, potential change request, problem, barrier to progress, or potential risk.
Iterative approach	A system development approach which starts the system development with a high-level SA&D and then followed by repeated cycles of implementation activities including detailed requirements analysis, design, coding and testing to deliver a target group of functions/components that may be rapidly rolled out. Each cycle will normally be short, e.g., within one to three months.
Milestones	A point in time of the project usually used to mark a significant point of the project and to highlight reporting dates and/or other key dates throughout a project.
MoSCoW	MoSCoW (Must, Should, Could and Won't) Analysis is a technique that helps users to prioritise each requirement based on its importance.
Non-functional Requirements	Specify criteria of how the system can perform and maintain these functions and features (i.e., how the system should work) from a business perspective. The non-functional requirements can be grouped into different categories such as audit, control and security, data requirements, service level targets, usability, etc.
Organisation	One of the dimensions of the project management methodology describing the processes that organise, manage and lead the project team.
Phase	A group of related project activities that come together with the completion of a specific deliverables.
Prioritised Requirements List (PRL)	A list of requirements for the project, which have been prioritised using the MoSCoW technique. Agile's concept on prioritisation of requirements suggests users and project team to use one single prioritised requirement list to improve transparency, clarity and control.

Term	Definition
Process flow	Provides a visual representation of the steps in a process.
Project Assurance Team (PAT)	The PAT looks after the quality assurance work on behalf of the PSC from the business, user and technical perspectives. They are individuals who have knowledge or expertise in the specific subject matter area that is part of the project scope. The PAT consists of Business Assurance Coordinator (BAC), User Assurance Coordinator (UAC) and Technical Assurance Coordinator (TAC), that is a balanced representation of the Business User and technical interests.
Project Management Plan (PMP)	A formal, approved document that defines how the project is executed, monitored and controlled. It is used as a live document during the course of the project and is composed of subsidiary management plans from other project dimensions and other planning documents.
Project Owner	The Project Owner is the person who often stands to win or lose the most in terms of the outcome of the project; accepts full authority for the project, accepts accountability for the performance of the project (and who wants to do the project) and provides resources.
Project Steering Committee (PSC)	The PSC is accountable to the project owner for the progress and performance of the project. They decide on all actions needed in order to complete the project.
Project team	Includes people who design and implement the system. They can be project team members (for in-house projects) or contractor team members (for outsourced projects).
Quality	One of the dimensions of the project management methodology referring to the degree to which a set of inherent characteristics fulfils requirements.
RACI	A tool to identify roles and responsibilities in an organisation, including people who are responsible, accountable, be consulted and be informed about an activity in a project.
Requirement	A singular documented need of what a particular system, product or service should be or performs. A requirement is a statement that identifies a necessary attribute, capability, characteristic, or quality of a system in order for it to have value and utility to a user. Requirements must be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for System Design. Requirements are categorised as functional, non-functional, or project requirements.

Term	Definition
Risk	One of the dimensions of the project management methodology which is the possibility of an event occurring that will have a negative impact on the achievement of objectives. Risk is measured in terms of impact and likelihood.
Risk register	A record of identified risk relating to an initiative, including their status, history, owner, impact and action taken to mitigate.
Schedule	The planned dates for performing schedule activities and for meeting schedule milestones.
Scope	One of the dimensions of the project management methodology stating the requirements specified for the end result. The overall definition of what the project is supposed to accomplish, and a specific description of what the end result should be or accomplish.
Security requirements	Any requirements that deal with security (e.g., identity and access management, user registration, roles based access, privacy requirements).
Stakeholders	A stakeholder is a person or an organisation with a (legitimate) interest in a given situation, project, action or enterprise. Within an IT project, a stakeholder is often defined as any person (or organisation) that can impact/can be impacted by the success of the project positively or negatively.
Systems Analyst	Specialises in analysing, designing and implementation of systems. Systems Analysts assess the suitability of information systems in terms of their intended outcomes and liaise with end-users, vendors and programmers in order to achieve these outcomes.
Systems Architect	Systems Architect defines the structure, behaviour, and more views of a system, and is also responsible for the overall design of the systems. For small projects, this role is usually taken over by the Systems Analyst.
System Development Life Cycle (SDLC)	Describes the phases and major processes in the course of system development of administrative computer systems.
System Specification	Documents the functional design for each functional development component identified and a consolidated listing of the specifications created during the preceding design tasks.

Term	Definition
Technical Requirements	Supplement the Non-Functional Requirements and are not raised by the Business Analyst, but from the IT project team. It allows the IT project team to define, document and validate the detailed operational, performance and technical architecture requirements and all relevant technical factors which may impact the technical architecture.
User Experience	Describes the overall experience and satisfaction a user has when using a product or system. It most commonly refers to a combination of application and business topics, such as selling over the web, but it applies to any result of interaction design.
User Requirements Document (URD)	Describes what the new IT system looks like from a business perspective. It is a critical deliverable (also sometimes named as "User Requirements Specification") for SDLC as it documents all the analysed requirements and related information elicited from stakeholders for system analysis and design purposes
Wireframe	A skeleton representation of the future system user interface, it does not contain detailed information such as colour, logo and graphics. It is intended for understanding the spacing and layout of the user interface only.