Government of Ontario IT Standard (GO-ITS)

Number 56

OPS Enterprise Architecture Artefacts

Appendix B – Corporate Enterprise Architecture Review Requirements Guidebook

Version 1.7

Status: Approved

Note:

- This appendix contains hyperlinks that are intended for use on the OPS intranet only. In particular, hyperlinks that point to artefact templates and examples will not work outside the OPS intranet.

- However, for external accessibility, the artefact templates have been included as separate files in Appendix D – *Artefact Template Files*.

- See Appendix C – *Corporate Enterprise Architecture Artefact Template Information* for more information and instructions on how to access the included template files.

Table of Contents

The creation of artefacts recorded in the Corporate Architecture Review Requirements document delivers specific project benefits. A decision not to create these artefacts exposes a project to specific risks as outlined in this guide. The following is intended to convey the context in which these artefacts were assigned their optional or mandatory status.

Note: For more information regarding requirements for Acquired Solutions, consult the separately published document "Acquisition and Integration of Acquired Solutions". Also, refer to this guide for specific requirements when selecting products listed on the Government of Ontario Information Technology Standard (GO-ITS) or Vendor of Record (VoR) Agreements.

Row 1: Contextual

Row: 1	**WHAT**	**Column 1**
Artefact Type:	**Resource Type**	
What/Purpose:	This artefact identifies and classifies the types of resources that are required by, or produced by, the government enterprise for a variety of purposes including planning, budgeting, resource management, and performance measurement.	
Description:	This artefact lists the types of resources that are important to the business that are either "used" by some business processes or produced by them. The list also provides a business definition for each Resource Type. Resource types typically include assets or information. Human resources are not included (see Party Type and Role Type). Each resource type that may contain sensitive or personal information should be classified as such. Resources can also be associated with a process. See Resource Type artefact template. See the Resource Type artefact example.	
Rank:	Mandatory	
Benefit:	This artefact distinguishes the types of resources that are required by, or produced by the government enterprise for a variety of purposes including planning, budgeting, resource management (for tangible resources it can assist with stock management and order fulfillment) and performance measurement.	
Risk:	Financial and Public Exposure: resources may not be available when required.	

Artefact Dependencies:

Pre-requisite artifacts are:

- None

Artefacts dependent on this artefact:

- Information Model
- Conceptual Data Model
- Business Network Model
- Business Process Model
- Business Scenarios
- State Transition Model

Recommended Practice:

Resources named in the resource type artefact should be traceable to the Conceptual Data Model (CDM) and/or Information Model (IM), where entities/subject areas for the resource exist. If the CDM/IM has multiple entities/subject areas for a given resource then those entities/subject areas should be identified in the resource type description. If there are no entities/subject areas for a resource, that should be noted as well.

Row: 1	HOW	Column 2

Artefact Type: **Line of Business Profile**

What/Purpose: For the OPS, a Line of Business equates to an area of government mandate. The Line of Business description identifies and defines a required business focus (mandate and/or purpose) of government, at a strategic level.

The Line of Business Profile is a composite artefact that defines the key elements of a Line of Business. The key elements of a Line of Business are:

- Line of Business name and description
- Mandate
- Owner
- Program Portfolio

Description: When implemented, a Line of Business operates at a strategic level, developing and managing a set of strategic goals and priorities. It implements those goals/purposes by creating and managing a series of *Programs*. The Line of Business Profile documents the key elements that are relevant to its definition.

See the Line of Business Profile artefact template.
See the Line of Business Profile artefact example.

Rank: Optional

Benefit: This artefact ensures identification and documentation of individual areas of government mandate and the identification of the programs that are used to implement the mandate.

Risk: Public Exposure: Operating without a mandate.

Financial Exposure: Lack of an effective means of classifying the areas of government mandate may result in ambiguity and/or duplication of Programs and Services.

Artefact Dependencies

Prerequisite:
- Mandate

Artefacts dependent on this artefact:
- Program
- Governance Model
- Party Type

Row: 1	HOW	Column 2

Artefact Type: **Program**

What/Purpose: A program is a mandate conferred from the governors of the enterprise to achieve goals, expressed as outcomes, and impacts that address the identified needs of a target group within a jurisdiction. Programs are delivered through a collection of services that contribute to the program goals and comply with the program strategy. Programs receive allocated funding, or a mechanism is established for program funding and resourcing by the governors.

Description: This artefact describes each program by expressing the name, description, type and owner of the program.

See Program template.
See Program example.

Rank: Optional

Benefit: A program is a logical unit for a "services framework" in that a given instance of a public service or internal service is logically and coherently defined by a specific program. Programs set the scope and focus of a given business model.

Risk: Public Exposure: Operating without a mandate.
Financial: Inability to document the appropriate allocation of funds to programs.

Artefact Dependencies

Prerequisite:

- Line of Business Profile
- Mandate
- Party Type

Artefacts dependent on this artefact:

- Program Profile
- Information Model
- Program Logic Model
- Business Function Model
- Strategy

Recommended Practice: Please refer to **GO-ITS 56.1, *Defining Programs and Services in the OPS***

Row: 1	**HOW**	**Column 2**

Artefact Type: **Service**

What/Purpose: A Service provides specific results (service outputs) that satisfy the needs of a target group (e.g. the client) and contribute to the achievement of the program goals.

Description: This artefact describes a service by identifying the target group (e.g. client) whose need is being satisfied along with a measurable output (service delivery unit) that has value from the client's perspective.

See Service artefact template.
See Service artefact example.

Rank: Optional

Benefit: The services will satisfy the needs of the client and contribute to the achievement of Program goals.

Risk: Financial and Public Exposure: The initiative may not be aligned with or may be counter to program goals.

Artefact Dependencies

Prerequisite:

- Program
- Target Group
- Need Type

Artefacts dependent on this artefact:

- Service Profile
- Information Model
- Program Logic Model
- Business Function Model
- Service Life Cycle
- SIAM
- Business Process Model
- Service Objectives
- Performance Matrix
- Strategy

Recommended Practice: Please refer to **GO-ITS 56.1, *Defining Programs and Services in the OPS***

Row: 1	**HOW**	**Column 2**

Artefact Type: **Program Profile**

What/Purpose: A **Program Profile** is a composite artefact that defines the key elements of a Program. The key elements of a program are:

- Type
- Program name and description
- Mandate
- Target Group
- Target Group Need
- Program Goal(s) described as Outcome(s) and Impact(s)
- Program Owner (Accountable Party)
- Program Management/Delivery Strategy
- Program Classification by Need
- Service Portfolio
- Strategic Goals

Description:

See Program Profile template.
See Program Profile artefact example.

Rank: Mandatory

Benefit: Clear and consistent definition of program.

Complete view of all the elements that comprise a program in order to provide a single view for the business owner.

Provides ability to test for potential inconsistencies in the primitive artefacts developed for the program.

Risk: Loss of project time due to the ongoing need to synthesize a complete program view from its constituent elements.

Artefact Dependencies: Pre-requisites artefacts are:

- Program
- Program Mandate
- Target Group
- Party Type
- Role Type
- Need Type
- Strategy
- Goal
- Service

Artefacts dependent on this artefact:

- None

Recommended Practice: See GO-ITS 56.1, *Defining Programs and Services in the OPS*.

Row: 1	WHERE	Column 3

Artefact Type: **Location Type**

What/Purpose: This artefact identifies and classifies the types of business locations that are important and required by a government enterprise. The consistent use of a standard set of location types across all OPS programs assist in identifying and comparing service delivery, logistics, access to supply sources, technology infrastructure opportunities and requirements at the enterprise level (e.g. across ministries, programs and projects).

Description: This artefact lists the types of locations of interest to the business. Locations are categorized as being physical in nature. They identify where services are produced and/or consumed, processes are performed, and where resources and parties are located. This artefact does not include *channels*; e.g., over-the-counter, mail/fax or telephone. Channels are identified in the Service Profile.

See Location Type artefact template.
See Location Type artefact example.

Rank: Mandatory

Benefit: From the business perspective, location type helps to identify where services are produced and/or consumed, processes are performed, and where resources and parties are located. From the technology perspective, this artefact supports the logical model task of identifying the technology infrastructure and network connectivity required to support a given location type.

Risk: Financial & Public exposure: Business may not understand impact of business changes on geographical coverage of service delivery operations. IT may lack business requirements for technology infrastructure and network connectivity (lack of coverage in access).

Security: Lack of identification of secured points of access.
Public Exposure: A service may not be offered at a required location type, or may be offered at an inappropriate location type.

Artefact Dependencies: Pre-requisites artefacts are:

- None

Artefacts dependent on this artefact:

- Business Network Model

Row: 1	WHERE	Column 3

Artefact Type:	**Geographic Area Type**
What/Purpose:	This artefact identifies and classifies the types of geographic areas (e.g., regions and districts) that are required by an OPS enterprise to carry out its mandate, deliver its program(s), and distribute its services. The Geographic Area Type is usually based on the type of mandate (e.g., social, economic, or stewardship) that is providing the authority for the program being delivered, including the various business drivers, logistics, transportation and communications infrastructure, and population centers.
Description:	This artefact lists the types of geographic areas within which an OPS enterprise administers programs. Geographic area type is a spatial concept that defines types of natural or administrative areas. See Geographic Area Type artefact template. See Geographic Area Type artefact example.
Rank:	Optional
Benefit:	Awareness of the types of geographic areas affected by a change initiative. For example, the City of Toronto amalgamation in the year 2000 required the mapping of service delivery from 6 municipalities to 4 administrative areas.
Risk:	Public Exposure: Lack of knowledge of geographic area types could result in jurisdictional disputes.
Artefact Dependencies:	Pre-requisite artefacts are: • Location Artefacts dependent on this artefact: • None

Row: 1	WHO	Column 4

Artefact Type:	**Party Type**
What/Purpose:	This artefact identifies and classifies parties of interest to the service, to help ensure that all party types are accounted for when conducting needs analysis for a given ('As Is' or 'To be') service.
Description:	This artefact lists the types of parties of interest to the enterprise. Types of parties include individuals and organizations. Organizations are further classified into Government of Ontario, Broader Public Sector and Non-Government categories. See Party Type artefact template. See Party Type artefact example.
Rank:	Mandatory
Benefit:	Classifying parties is essential to understanding roles, needs and accountabilities.
Risk:	Privacy: Inability to identify FIPPA requirements. Financial & Public Exposure: Lack of an effective means of classifying parties may result in ambiguity in roles, responsibilities, authorities and accountabilities.
Artefact Dependencies:	Pre-requisites artefacts are: • None Artefacts dependent on this artefact: • Role Type • Target Group Type
Recommended Practice:	See the *Common Data Elements Model, Party Subject Area*.

Row: 1	WHO	Column 4

Artefact Type:	**Role Type**
What/Purpose:	This artefact supports analysis and design of service delivery mechanisms. It assists in analysis of roles, responsibilities, authorities, and accountabilities. It also supports analysis of gaps and overlaps in responsibilities.
Description:	This artefact lists the types of roles played by parties (individuals and organization) of interest to the business. A role is defined by a set of functions or relationships played by a party. See Role Type artefact template. See Role Type artefact example.

Rank: Mandatory

Benefit: This artefact supports the analysis of roles, responsibilities, authorities, and accountabilities. It supports the analysis of gaps and overlaps in responsibilities. It also addresses pluralistic nature of government business relationships, if required.

Risk: Public Exposure and Security: Lack of an effective means of classifying roles may result in erroneous or incomplete assignment of functions to individuals and organizations.

Privacy: Improper or inadequate identification for permission management.

Artefact Dependencies: Pre-requisites artefacts are:

- None

Artefacts dependent on this artefact:

- Program Profile
- Service Profile
- Business Process Model
- Business Scenario

Row: 1	**WHO**	**Column 4**

Artefact Type: **Target Group Type**

What/Purpose: A target group type is a classification of that part of the population whose needs the program has a mandate to satisfy. By classifying target groups, program owners can make explicit decisions about how the needs will be met.

Description: This artefact lists the types of groups targeted by a program. A target group comprises two (2) sub-groups: client groups and interested parties. The needs of client groups are intended to be met directly from the program outcomes and indirectly through anticipated program impacts. Interested parties are generally intended to benefit indirectly, that is, from program impacts.

See Target Group Type artefact template.
See Target Group Type artefact example.

Rank: Mandatory

Benefit: This artefact supports program design. Through the name of the target group, it clearly identifies the characteristics of the targeted group. Not all target groups will be clients of the program's services. By defining target group to include client groups and interested parties, program managers can make deliberate decisions about the services that will be delivered within the resource envelope that is available and the level of maturity of the program.

Risk: Public Exposure: Lack of identification of a specific target group. Unanticipated impacts could result if interested (affected) parties are not identified.

Artefact Dependencies: Pre-requisite artifacts are:

- None

Artefacts dependent on this artefact:

- Party
- Role
- Mandate
- Target Group/Needs Cross Reference
- Conceptual Data Model
- Program Profile
- Service Profile

Recommended Practice: Identify all of the groups that the program is mandated to serve. Classify as interested party or client group. The distinction must be made within the context of the discussion about the services the program will offer. Identifying services and client groups is an iterative process when defining a program that is influenced by resource availability and other factors.

Row: 1	**WHEN**	**Column 5**

Artefact Type: **Event Type**

What/Purpose: Event types identify and classify events important to the OPS enterprise. An event is a point in time occurrence that may trigger a process. Events may cause changes in state, in the life cycle of a business component (e.g., program, service, organization, role, resource, etc.). Trigger events (e.g., requisitioning a commodity) are used to define business scenarios that explore the structure and behaviour of a given business model.

Description: This artefact lists types of events that trigger business processes. An event type is a classification of types of point in time occurrences that result from processes or trigger processes.

See Event Type artefact template.
See Event Type artefact example.

Rank: Mandatory

Benefit: This artefact ensures that the business model can respond to each event e.g. Routine response, like requisitioning a commodity or planned event like an emergency response.

Risk: Financial & Public exposure: Lack of an effective means of classifying events may result in the incomplete capture of event instances to which the business model must respond.

Artefact Dependencies: Pre-requisite artefacts are:

- None

Artefacts dependent on this artefact:

- Business Process Model
- Business Scenario

Row: 1	**WHEN**	**Column 5**

Artefact Type: **Cycle Type**

What/Purpose: A cycle is a recurring sequence of activities that occur within a preset interval of time; e.g., the seasons or a driver licence renewal, whereas an event is a point in time occurrence e.g., Cancel Appointment. A cycle type is a classification of recurring internal or external cycles that trigger one or more events.

Description: This artefact lists the types of cycles that affect the business.

See Cycle Type artefact template.
See Cycle Type artefact example.

Rank: Optional

Benefit: This artefact ensures that business model can address each cycle.

Risk: Lack of an effective means of classifying cycles may result in the incomplete capture of cycle instances to which the business model must address.

Artefact Dependencies: Pre-requisites artefacts are:

- Event Type

Artefacts dependent on this artefact:

- None

Row: 1	WHY	Column 6

Artefact Type:	**Need**
What/Purpose:	This artefact identifies the needs of a target group that the program intends to satisfy.
Description:	This artefact lists the needs to be satisfied by a program. A need is a condition or situation in which something is required, desirable, or useful for a given target group. It is expressed as a statement of the problem or condition of the target group that the program is intended to address. See Need artefact template. See Need artefact example.
Rank:	Mandatory
Benefit:	Identifies the needs to be satisfied by a program.
Risk:	Financial & Public exposure: Incorrect identification of needs may result in programs and services attempting to address inappropriate issues.
Artefact Dependencies:	Pre-requisites artefacts are: • Mandate Artefacts dependent on this artefact: • Program Profile

Row: 1	WHY	Column 6

Artefact Type:	**Goal**
What/ Purpose:	This artefact formalizes programs by defining specific goals, enabling program and service design and performance measurement design (measures of goal-directed change in level of need). It formalizes motivation for change in change initiatives.
Description:	This artefact expresses a desired change to a target group. Program goals state the desired change to a target group and are expressed as outcomes (measurable results directly attributed to the program) and impacts (results influenced by the program). See Goal artefact template. See Goal artefact example.

Rank: Mandatory

Benefit: This artefact explicitly identifies what a program will do. It will assist in the identification of appropriate performance metrics.

Risk: Public exposure: Mismanagement of public programs.

Artefact Dependencies: Pre-requisites artefacts are:

- Mandate
- Need

Artefacts dependent on this artefact:

- Program Profile
- Service Objectives

Row: 1	**WHY**	**Column 6**

Artefact Type: **Mandate**

What/Purpose: This artefact is used to articulate:

- A program's mandate (i.e. authoritative command(s));
- The target group(s);
- The target group's need(s); and
- The jurisdiction(s) within which it has the right to exercise authority.

It Identifies the program's right to exist.

Description: This artefact articulates the authoritative command provided by the governing bodies. It lists the source or instrument that provided the program with its right to exist.

See Mandate artefact template.
See Mandate artefact example.

Artefact Dependencies: Prerequisites:

- Target Group Type
- Need

Rank: Mandatory

Benefit: This artefact articulates the authoritative commands a program receives from its governors. The mandate instruments provide a source for deriving business rules.

Risk: Financial & Public exposure: Incorrect identification of mandate types may result in an incomplete or incorrect understanding of the mandate instances that articulate what a program or service will and will not do, resulting in an inappropriate response.

Recommended Practice:	• The target group identified in the mandate statement must correspond to the target group identified in Target Group artefact. • The needs identified in the mandate statement must correspond to those identified in the Needs artefact. • Abide by the plain language guidelines. • A program's "mission statement" is often a good source for a mandate statement.

Row: 1	**WHY**	**Column 6**
Artefact Type:	**Target Group / Needs Cross Reference**	
What/ Purpose:	A target group type is a classification of that part of the population whose needs the program is intended to satisfy. This artefact matches the target group (client groups and interested parties) to the need, since not all members (sub-types) of the target group have all needs. This allows program managers to define services to meet specific needs of target groups.	
Description:	This artefact cross references the types of groups targeted by a program to the needs that the program is mandated to meet. See Target Group / Needs Cross Reference artefact template. See Target Group / Needs Cross Reference artefact example.	
Recommended Practice:	If the cross reference matrix is too sparse, look to redefining the needs and target groups; if too full, the target groups may not have been defined specifically enough.	
Rank:	Mandatory	
Benefit:	This artefact allows program managers to define services to meet specific needs of target groups. Managers can make explicit decisions about the direction of their program. It provides the basis for defining program outcomes and impacts.	
Risk:	Without this artefact, needs of specific target groups may be overlooked and unintended impact(s) may result.	
Artefact Dependencies:	Pre-requisites artefacts are: • Need • Target Group Type Artefacts dependent on this artefact: • Program Profile	

Row 2: Conceptual

Row: 2	WHAT	Column 1
Artefact Type:	**Information Model**	
What/Purpose:	An Information Model (IM) describes all the things (terms, facts and concepts) that are important to the enterprise (e.g. ministry, cluster) or a specific domain (e.g. an OPS program), and shows how these things are directly inherited from or related to the OPS enterprise business concepts as articulated in the OPS Business Architecture Concept Model (OPS BACM).	
Description:	This model not only can describe domain specific concepts, terms and information from a pure business perspective, it also describes domain specific high-level requirements with an intention of scoping out and defining data requirements for I&IT solutions. Some of the uses of the IM are: • To specify, analyze and represent business concepts, and to facilitate common understanding and stakeholder agreement on the meaning of terms and relationships; • To assist the development of new enterprise strategy and planning initiatives either for an entire enterprise or a specific domain within the enterprise; • To identify in-scope entities and their relationships required or involved in support the business activities of an enterprise or a business domain; • For a large project or initiative that has enterprise scope or impact, to show the traceability and semantic alignments of domain specific business concepts to the OPS enterprise business concepts as articulated in the Information Model; • To show the link between project or domain specific business architecture and high-level data architecture; and • To serve as a reference model in the creation of a domain specific conceptual data model. See Information Model Example 1. See Information Model Example 2.	
Rank:	Optional	
Benefit:	This model has a number of benefits: • It clearly specifies, and represents business concepts and their relationships for an entire enterprise or a specific domain. • It identifies and contains the project scope. • It shows traceability and business concept alignments from	

a project to the ministry (or cluster), or from a ministry (or cluster) to the OPS enterprise business concepts.

Risk: The risks of not developing this model include:

- Incompleteness in enterprise strategy and planning due to lack of a holistic view of the enterprise business visions and goals, and the important inter-relationships among business entities within the enterprise.
- Failure to communicate clearly to achieve the common understanding among people within the enterprise about the business goals, objectives, and requirements.
- Failure to show alignment and traceability from a project to the ministry (or cluster), or from a ministry (or cluster) to the OPS enterprise business concepts.
- Failure from a project perspective due to a lack of: scope definition, information sharing requirements, functional requirements, and communication in the event of turnover in the project team.

Artefact Dependencies: Pre-requisites artefacts are:

- Resource Type
- Location Type
- Program Profile
- Service Profile
- Business Process Model
- Party Type
- Role Type
- Event Type
- Mandate

Artefacts dependent on this artefact:

- Conceptual Data Model
- Business Rule Profile

Recommended Practice: The IM must be diagrammed using one of the following notations, and must be accompanied by a level of metadata as specified in GO-ITS 56.3, Information Modelling Standard:

- Entity Relationship Diagram, or
- Unified Modeling Language (UML) Class Diagram representing only entity classes, without showing any methods on these classes.

Row: 2	**WHAT**	**Column 1**

Artefact Type: **Conceptual Data Model**

What/Purpose: A Conceptual Data Model (CDM) represents the structure of the information about in-scope, high-level business entities and their relationships. It gives a formal representation of the data needed to run an enterprise or a business activity. It is used primarily to enhance communication with business staff and to clarify

business rules involving the business information.

Description: The Conceptual Data Model is the precursor to the logical data model. It suppresses technical details by including only the business entities that have a business meaning, the important relationships among these entities and the representative attributes of the entities.

The Conceptual Data Model is required for any project that goes forward to Row 3 and needs to do further business requirement analysis at a more detailed level.

See Conceptual Data Model Example 1.
See Conceptual Data Model Example 2.

Rank: Mandatory

Benefit: It is used during the planning phase of a project to identify and contain the project scope.

Risk: The risks of not developing this model:

- Failure in achieving the understanding of the common definitions, semantics, information, and knowledge across all business domains within an organization.
- Failure in partitioning the organization's information and scoping subsequent projects.
- Failure in identifying missing important information needs and their implications.
- Failure in identifying some key functional requirements related to the missing subject area groupings.
- Failure in assessing information sharing requirements across business units or functional areas.
- An incomplete picture of the needs of an organization, a project or an application may result in erroneous recommendations regarding the development of some solution areas.
- Lack of foundation to develop a coherent database strategy.
- Lack of assurance that business goals and objectives will be properly supported.
- Lack of clear project scope and vision may jeopardize requirements definition, data analysis and design effort, project estimation activities. The accuracy of the estimation of subsequent projects may be affected.

Artefact Dependencies: Pre-requisites artefacts are:

- Resource Type
- Location Type

- Program Profile
- Service Profile
- Party Type
- Role Type
- Event Type
- Mandate
- Information Model (if created)

Artefacts dependent on this artefact:

- Logical Data Model
- Fact and Dimension Matrix

Recommended Practice: A Conceptual Data Model must be diagrammed using one of the following notations, and must be accompanied by detailed metadata as specified in GO-ITS 56.3, Information Modelling Standard:

- Entity Relationship Diagram (ERD), or
- Unified Modeling Language (UML) Class Diagram representing only entity classes, without showing any methods on these classes.

Artefact Type:	**Conceptual Data Model (Acquired Solution)**
Guidance:	The Conceptual Data Model (CDM) (Acquired Solution) is a fully attributed CDM which has been extended to include additional detail about the business data requirements. It includes specification about the data content and structural requirements at a sufficient level of detail to inform the Request for Proposal (RFP). It includes all the data elements used to support the business processes and functional requirements.
Required:	• All data entities must be identified, defined and fully attributed. • All business relevant unique identifiers must be identified and defined. • Relationships among data entities must be clearly defined. • Many-to-many relationships that represent additional business data requirements must be resolved to show these additional data requirements. • Domains of data attributes with significant business value must be defined or at least described in the description of the attribute. • Information classification must be specified for key data entities. • Volume and volatility must be specified for key data entities. • Data retention requirements must be stated for key data

	entities. See CDM (Acquired Solution) example.
Not Required:	• Data types and sizes for all data attributes. • Attributive entities. • Resolution of many-to-many relationships for technical implementation purposes, i.e. resulting in associative entities without additional data attribute(s) • Data model normalization to 3rd normal form. • Definition of constraints related to implementation such as domain, referential integrity, etc.
Recommended Practice:	Refer to the most current version of *Guidance for the Acquisition and Integration of Acquired Solutions*.

Artefact Type: **Interface Data Requirements Document**

What/Purpose: The purpose of the Interface Data Requirements document is to capture the application interface data requirements between the Acquired Solution and other business applications with which the Acquired Solution will interface.

Specifying the interface data requirements early in the project life cycle will provide product vendors with sufficient detail about the data requirements between the Acquired Solution and applications with which the Acquired Solution is intended to interface.

The Interface Data Requirements Document should be stated in the form of interface file or message layout, including key data and other business data used for information exchange and database updates, for all automated data interfaces.

Required:

- Names of all applications that require (input, output, or both) data interface with the target application
- Names and descriptions of all data interfaces with the target application including:
    - Purpose of interface;
    - Interface type;
    - Method used; and
    - Frequency of all data interfaces.

- Detailed interface data requirements include
    - Interface file record type, record layouts for each record type, and data fields in each file record layout

including the data fields served as key identifiers for data integration; or

- Interface information exchange message types, message layouts for each message type, and data fields for each applicable message layout including the data fields served as key identifiers for data integration.

See Interface Data Requirements Document example.

Not Required: Logical Data Model for the entire solution

Rank: Mandatory[1]

Recommended Practice:	Refer to the most current version of *Guidance for the Acquisition and Integration of Acquired Solutions*.

Row: 2	**WHAT**	**Column 1**

Artefact Type: **Semantic Model**

What/Purpose: The term "semantic" refers to the model's use in establishing the vocabulary that will be used by the enterprise to talk about its business.

Description: A **semantic model** is a diagram depicting major things of interest (expressed as Terms) to the business, and how they relate to each other (expressed as Facts). The model represents the basic vocabulary for expressing rules. The purpose of the model is to structure basic knowledge of the business.

The OPS Semantic Model defined here is not the same as the "semantic model" described by E.F. Codd, or other definitions in the information modeling literature.

Example Semantic Model.doc
See Semantic Model artefact example.

Rank: Optional

Benefit: The semantic model has a number of benefits:

- It is a powerful management tool for clarifying business concepts

[1] Mandatory for acquired solutions where there are application interface data requirements between the acquired solution and other business applications.

- It establishes a common business vocabulary
- It clarifies the relationships between business components
- It clarifies the understanding between Business and IT participants
- It provides clearer handoffs between architects and between architects and developers
- It provides a foundation for subsequent models, including the conceptual data model, high-level business object model and class model.

Risk:

- Misunderstanding of business
- Miscommunication between business and IT
- Data and class models developed at the logical level may not support business requirements

Artefact Dependencies:

Pre-requisites artefacts are:

- None

Artefacts dependent on this artefact:

- None

Recommended Practice:

- The **major things of interest** are the essential ingredients needed to run the business.

  A major thing of interest can be a person (referred to in our practice as a Party or the Role the Party plays), a place (referred to in our practice as a business Location where Services are delivered or consumed), an event (referred to in our practice as an Event that triggers a business Process), or a thing (referred to in our practice as a Resource used by or produced by business Processes).

- A semantic diagram has two basic components: terms and facts.

  Term: A basic word or word phrase in plain language that business (program) owners recognize and share in the business. Terms are always nouns or qualified nouns. Nouns may be singular or plural, to make the relationships read naturally. Terms represent things that are indivisible – that is, not composite. Terms that have a collective sense such as inventory, personnel etc., should be decomposed. Terms must always represent things we can know something about.

  Each term must have a written definition. This catalogue of terms and definitions is the basis for dialog with the business.

  Fact: A simple declarative sentence that relates terms. Facts represent common or shared verbs or verb phrases of the business. Every fact must always be expressed using a complete sentence. Facts follow a strict subject-verb-object structure.

- For all Change Initiatives, the semantic models being developed should be set in the context of the OPS business architecture methodology, which is based on the "Public Sector Reference Model" and defined in GO-ITS 56.1, Defining Programs and Services in the Ontario Public Service. For example, a semantic model could be developed to support the understanding of the delivery of each service.

- A semantic model may be made up of one or many semantic diagrams.

- Each semantic diagram shows any or all three known structural relationships – Generalization, Aggregation and Association.

    - **Generalization:** Generalization relationships show how different but similar pieces can be generalized into a common type. Generalization shows "is-a" relationships, also known as inheritance or subtyping. A line with a large, hollow arrowhead pointing from a sub-type to a super-type is used to represent this type of relationship.

    - **Aggregation:** Aggregation relationships show components that are either included as part of other components, or contained in other components. An aggregation relationship is sometimes called a "Whole-Part" relationship. A line with a hollow diamond head pointing from the part to the whole is used to represent this type of relationship.

    - **Association:** Association relationships show components that relate to each other in some way other than generalization or aggregation. Associations may describe actions, processes, structures or rules. A line with a small filled-in arrowhead is used to represent this type of relationship. The form of the relationship is "Noun – Verb - Noun" where the verb characterizes a relationship between the two components and tends to allow the model to read like a series of natural language sentences. The direction of the arrow identifies how the relation must be read.

Row: 2	**WHAT**	**Column 1**
Artefact Type:	**Fact and Dimension Matrix**	
What/Purpose:	The Fact and Dimension Matrix represents a high-level, enterprise-wide view of business information requirements that are within the scope for a decision support solution. The matrix illustrates different data analysis perspectives (i.e. dimensions) and may lead to the development of a data warehouse and/or data mart to support business intelligence initiatives. This artefact is used to: • Present different views of business decision support	

information requirements.

- Identify areas in the Conceptual Data Model where additional details related to decision support type of information requirements are needed.
- Identify the scope of information requirements for a decision support project
- Assist in prioritizing the dimensions in which the solution should be designed and data be analyzed first.

Description: The Fact and Dimension Matrix provides information about the in-scope, high-level facts (known as fact groups), the perspectives for performance measures (known as dimensions), and the relationships between the facts and perspectives that will be implemented in the decision support system.

This artefact will be developed by leveraging data definitions from existing conceptual or/and logical data models of the source data stores.

See Fact and Dimension Matrix artefact example

Rank: Optional

This artefact is considered as **mandatory** for a project that develops or acquires data warehouse and/or data mart based solutions for decision support initiatives.

Benefit: This artefact illustrates an understanding of the scope and content of the information requirements as set out by the business, at a high level, for the decision support solution.

Risk: If this artefact is not developed, it will result in:

- An inability to properly support the business goals and objectives
- Missing key information requirements as set out by the business
- The construction of separate data marts that lack a framework to tie the data together.
- Missing a foundation for partitioning the organization's data and scoping the breadth of the decision support solution.

Artefact Dependencies: Pre-requisites artefacts are:

- Conceptual Data Model

Artefacts dependent on this artefact:

- Logical Dimensional Model

Recommended Practice: A matrix that represents the common or potential dimensions (i.e., performance measurements) used across the enterprise

against the fact groups (i.e. primary business processes of the organization). Intersections are marked where a dimension may exist for a fact group.

Row: 2	How	Column 2

Artefact Type: **Service Life Cycle**

What/Purpose: A service life cycle identifies all the processes required to manage and deliver a service. It provides a standard mechanism to record processes. The result of analyzing the processes may uncover common processes.

Description: Processes are organized into the following six cycles:

Plan:
- Plan service goals, establish delivery strategy, forecast demand, plan capacity etc.

Design:
- Design delivery processes, ensure compliance with applicable policy, standards and rules
- Design products and instruments (e.g. licence cards)
- Design service agreements

Develop:
- Promote service to client
- Acquire new client
- Execute client service level agreements
- Establish supply arrangements
- Provision delivery infrastructure

Operate:
- Includes all processes required to accept and fulfill a service request
- Manage client inquiries

Monitor:
- Monitor effectiveness, efficiency and quality of service
- Ensure compliance with service level agreements
- Account for service delivery against program mandate

Decommission:
- Merge service instance with another service
- Decommission service delivery infrastructure
- Phase out service

See Service Life Cycle artefact template.
See Service Life Cycle artefact example.

Rank: Optional

Benefit: Facilitates the identification of candidate processes for redesign or automation.

Risk:	Without having a detailed list of processes associated with each service you may not be able to justify redesign.
Artefact Dependencies:	Pre-requisites artefacts are: • Service Profile Artefacts dependent on this artefact: • Business Process Model

Row: 2	**How**	**Column 2**
Artefact Type:	**Business Function Model**	
What/Purpose:	The business function model identifies and displays both graphically and textually, in a structured format, the processes that the business performs. Functions provide context and a high-level or strategic view that allows key stakeholders to focus on the critical business without getting caught up on details such as organizational units or data flows. The business function model therefore highlights what the business ought to be doing in order to deliver services.	
Description:	A hierarchical diagram and a table describing business processes which are typically grouped by business functions. Business functions are at the top level of the diagram with the descriptions in the table, and do not imply any sequential order, precedence or dependence. Functions are defined as a collection/grouping of ongoing processes, which, together, completely support the business. Note that functions represent major organizational actions that may cross-organizational or departmental boundaries. Business functions, which describe what work an organization does, can therefore be decomposed or detailed into processes that describe how the work is accomplished. Business Processes can be either delivery or management processes. Management processes are internal and administer the resources and operations of the organization. Delivery processes are client facing and supply the outputs of services. The function model indicates whether each process is a management or a delivery process. The model is therefore a useful means for further detailed analysis for mapping requirements and resources. See Business Function Model artefact template. See Business Function Model artefact example.	
Rank:	Mandatory	

Benefit:

- Provides a clear understanding of the types of actions that the organization needs to perform, without the complexities of *who*, *when*, *where* and *how*.
- Helps to identify and analyze functions that are not supported by core business capabilities (or resources) which may be candidates for out-sourcing.
- Serves as a solid starting point to identify and decompose the detailed processes that comprise the business.
- Helps uncover common functions and the key processes they include.
- Clarifies which processes deliver output and which, provide internal support and management
- Provides a means for mapping/tracing functional requirements to business processes.
- Useful for project scoping and roadmap definition
- Assists in grouping or categorizing complex business actions into manageable and logical pieces

Risk: Not doing a Business Function Model runs the risk of orphaning or overlooking functions or processes.

Artefact Dependencies:

Pre-requisite artefacts are:

- Program Profile: To identify program goals and strategies that will define needed business functions and processes.
- Service Profile: To identify the services for which business functions and processes are needed. To utilize the processes identified as 'key processes'.

Artefacts dependent on this artefact:

- Business Process Model - details how the business carries out work
- Business Scenario - describes the sequence of events for which the processes are carried out

Recommended Practice:

A Business Function Model is a structured hierarchy with detailed description of the function in a table, similar in format to an organizational chart, which involves identifying 3 components.
1) Functions: A logical set of ongoing cross-organizational actions that a business must perform in order to meet its business objectives and continue in existence.
2) Processes: What must be done to perform the functions. Processes are separated into management processes and delivery processes.
3) Connectors: Represent hierarchical relationships between functions and processes.

Guidelines and Syntax Checklist:

- All functions and processes are connected to at least one other function or process (i.e., no orphans)
- Label names are descriptive
- Name a function with a noun or wording ending in "ing." For example, "Communications" or "Licensing"
- Begin the name of a process with an active verb to construct a simple imperative statement that will describe the intent of the activity. For example, "Approve Order" or "Create Notice."

Connectors are not named (assumed to be read as "consists of").

Row: 2	**How**	**Column 2**

Artefact Type: **Service Integration Accountability Model**

What/Purpose:

A **Service Integration Accountability Model** (SIAM) is a diagram that shows how services are integrated in order to deliver the final deliverable to the end client. The power it provides is that it illustrates the required horizontal accountability relationships between service providers in the delivery of services. It illustrates the inter-relationships of services, their providers and the party role consuming the service. It will show how the output of one service is the input of another service.

Horizontal accountability describes the relationship that a service provider has to target group members for the provision of the service's output. The definition of "service" is provided in GO-ITS 56.1, Defining Programs and Services in the Ontario Public Service.

Description:

A SIAM is an analytical tool that:

- **Ensures** that the final valued service output meets the

target group needs and identifies all intermediate outputs and the services producing them so as to ensure the highest possible degree of enterprise performance in producing the final outputs.

- **Enables** the business designer to confirm that all the necessary services (along with appropriate service outputs for accomplishing the service objectives) have been identified.
- **Aids** in setting the context for forming service level agreements (SLA) between service providers. The model supports the determination of who the provider of the services will be. It can also be used to make explicit and identify the nature of the accountability (i.e., formal agreement such as a SLA or Memorandum of Understanding (MOU) or informal agreement such as a handshake).
- **Provides** the basis for workflow modeling. It can be translated using a straightforward procedure into models showing work and material flow used in other methodologies, e.g. RUP use cases and activity diagrams. It ensures consistent and levelled process integration across all organizational boundaries.

There are two forms of the model: Service Integration Model (**SIM**) and Service Integration and Accountability Model (**SIAM**).

- A **SIM** portrays the linkage between services and output types, without reference to organization structure or physical locations. The services can range from very abstract (e.g. Channel Service) to very specific (e.g. Ontario Driver License Counter Channel Service). This is a representation of the "service value chain" and aligns with the industry-accepted concept of "supply chain management". The output from one service represents an input that supports an essential process or processes in another service. Moving "upstream" in the value chain results in more granular inputs and the model provides the basis for an SOA strategy.
- A **SIAM** portrays all the information of a SIM plus the organizational assignments for services. The services and output types must be defined with sufficient specificity to assign each service in the model to an organization.

See Service Integration and Accountability Model artefact example.

Rank: Mandatory

Benefit: Provides a representation of services and the accountability for

providing required inputs.

Risk: Without a clear understanding of points of accountability, the instruments for formalizing that accountability (e.g., SLA, Contract, etc.) may be overlooked.

Artefact Dependencies:

Pre-requisites artefacts are:

- Service Profile for services that are in-scope.
- Service for services that are out-of-scope.
- Party Type

Artefacts dependent on this artefact:

- None

Recommended Practice:

- Using the SIAM effectively requires modeling performance of various services in order to determine horizontal accountabilities.
- Services need to be profiled.
- Business Functions and/or processes are not represented on the SIAM model. The SIAM portrays only accountability relationships with external service providers and clients. See the Business Function and Business Process models.
- A service that provides its service output to another service in general should provide it to two or more services. While this may generally be the case, it does not always hold true.
- If a service depicted on a SIAM is in scope for the project, then a Service Profile must be created. If a service depicted on a SIAM is not in the project's scope and has been included in the SIAM to set context, then a Service artefact must be created.
- Each organization must be included in the Party Type artefact located in row 1 column 4.
- Each role must be included in the Role Type artefact located in row 1 column 4.
- All services are peers.
- Each service contains all the processes required to produce its output.

Row: 2	**HOW**	**Column 2**

Artefact Name: **Service Profile**

What/Purpose: A **Service Profile** is a composite artifact that defines the key attributes of a service. The key elements of a service are:

- Service Name and Description
- Type of Service (Public or Support)
- Service Output
- Service Output Type Classification
- Contribution to Program Goal

- Service Owner
- Service Provider
- Strategy (Service Delivery Model)
- Client Group
- Needs Addressed
- Performance Measures
- Business Level Performance Requirements
- Key Service Feature
- Key Service Processes

Description: This artefact describes a Service by documenting its key attributes.

See Service Profile template
See Service Profile example

Rank: Mandatory

Benefit:
- Clear and consistent definition of service
- Complete view of all the elements that comprise a service in order to provide a single view for the business owner
- Provides ability to test for potential inconsistencies in the primitive artefacts developed for the service
- Provides an early business interpretation of the system metrics
- Establishes traceability to logical quality level metrics for availability, recoverability, and scalability.

Risk: Loss of project time due to the ongoing need to synthesize a complete service view from its constituent elements.

Artefact Dependencies: Pre-requisites artefacts are:
- Program Profile

Artefacts dependent on this artefact:
- Business Process Model
- Service Integration Accountability Model
- Quality Level Metrics

Recommended Practice: See GO-ITS 56.1, *Defining Programs and Services in the OPS*.

Row: 2	HOW	Column 2

Artefact Name: **Business Process Model**

What/Purpose: A Business Process Model is a diagram or a series of diagrams, which describe the operational aspect of a business. It describes how tasks are structured, what their relative order is, how they are synchronized, what resources are being passed, and who performs them.

This model helps to understand accountabilities of different roles involved in the business process and brings clarity to work that crosses and/or is within organizational boundaries.

Description: A **business process** a linked sequence of activities or tasks that delivers a service, manages a resource, or operates an organization. Business processes are a set of repeatable, coordinated activities with inputs, outputs and resources, which could interact with people or organizations, contribute to achieving business goals.

There are two types of **business processes:**

1. A **Service Delivery Process** directly supports or enables the delivery of one or more service outputs.
2. A **Management Process** supports the management of an organization, including the planning, design, provision and monitoring/evaluation of programs and services.

Rank: Mandatory

Benefit:

- Supports business process redesign or re-engineering exercises by:
    a. Providing a crosscheck of responsibilities for each identified role and can be used to identify redundancies, bottlenecks and inefficiencies;
    b. Providing a mechanism to discover opportunities for automation; and
    c. Identifying common processes or tasks across multiple services through the analysis of this model.
- Facilitates the analysis and transformation to system functional requirements and non-functional requirements including mapping of data to processes for CRUD (Create, Read, Update, Delete) analysis.
- Supports the placement/cross-reference of business rules in relation to business processes.
- Identifies security issues by clarifying resources shared across role and/or organizational boundaries.
- Aids in the creation of the Conceptual PIA and TRA.

Risk: Missing or incorrectly defined business processes lead to:

- Incomplete designs and missed opportunities;
- Ambiguity and confusion about role responsibilities, sequencing of processes and key hand-offs; and
- Lack of understanding with respect to the impact of information on security.

Artefact Dependencies: Pre-requisites artefacts are:

- Service Profile
- Business Function Model
- Role Type (where role is architecturally significant)
- Party Type (where party is architecturally significant)
- Resource Type
- Event Type

Artefacts dependent on this artefact:

- Business Scenarios
- System Functional Requirements

Recommended Practice:

1. A business process model must be created using one of the two following notations (in either case the roles are shown as swimlanes):
   - UML Activity Diagram, see http://www.omg.org
   - Business Process Modeling Notation (BPMN) see http://www.bpmi.org/

   See UML Activity Diagram or BPMN Business Process Model artefact examples.

2. Begin the name of a business process with an active verb to construct a simple imperative statement that will describe the intent of the process. For example, "Approve Order" or "Create Notice".
3. When process mapping, the process steps should be decomposed to their level of usefulness; that is, to understand the problem or issue at hand. If a process is not decomposed to a sufficient level of granularity, it may not be useful. If it is decomposed too much, the detail may make it incomprehensible and it may cost too much in time and money.
4. Place significant resources on the model and indicate where they originate and who will receive/utilize them. Resources need to be associated with the processes which access or update the information. Co-ordinate information resources with the resource type to clarify whether the information is sensitive or personal in nature.
5. For "As-Is" process modeling, focus on the processes actually practiced, rather than formally documented procedures that may or may not be followed.

6. Avoid introducing system activities / constraints and focus on what the user does.
7. Recommend re-use:
   - Business process patterns based on best practices can be (re) used to design new processes.

Row: 2	HOW	Column 2

Artefact Name:	**SOA Service Specification**
What/Purpose:	Where a discrete, common and repeatable business process has been identified, this specification is used to detail the characteristics and capabilities of the business process as a candidate application service. The specification requires further elaboration during the logical design phase (see Logical Design Document for further details).
Description:	This specification provides information about a discrete, common and repeatable business process that may be a candidate to become an application service. The specification is made up of: 1. **Business Process Model:** Diagrammed depiction of the candidate business process 2. **Business Scenario:** Textual description supplementing the Business Process Model 3. **Parameters:** Specific input and output parameters passed and returned from the process. Create this specification if your project is using a service-based (SOA) approach or an opportunity to automate one or more common, repeatable business processes has been identified. See SOA Service Specification template. See SOA Service Specification example 1. See SOA Service Specification example 2.
Rank:	Mandatory for projects following a service-based approach.
Benefit:	Provides an explicit and detailed description of the nature and use of a proposed module. It describes what the service needs to do, not (necessarily) how it should do it. The specification is to be used by an Application Architect to evaluate feasibility to construct a candidate application service.
Risk:	Missed opportunity for business to provide input into to design of an automated service.
Document	Pre-requisites:

Dependencies:

- Service Profile
- Business Process Model
- Business Scenario
- Business Rules Profile
- Conceptual Data Model

Document dependent on the SOA Service Specifications is:

- Logical Application Design Document (SOA Application *Service Model Template*)

Row: 2	**WHERE**	**Column 3**

Artefact Name: **Business Network Model**

What/Purpose: To understand the major flows of information, and provide a framework for analysis of volumes, frequencies, service levels, and other scaling factors for the enterprise.

Description: Business network models show the flow of information, resources and material between business location types. See Business Network Model artefact example.

Rank: Mandatory

Benefit: Provides an understanding of the major flows of information, resources and materials. It may assist in the analysis of volumes, frequencies and service levels.

Risk: Inadequate availability and flow of resources.
The business model may not reflect all of the necessary business requirements. This could result in network designs and implementations that would not support existing or future business requirements.

Artefact Dependencies:

Pre-requisites artefacts are:

- Location Type
- Resource Type

Artefacts dependent on this artefact:

- None

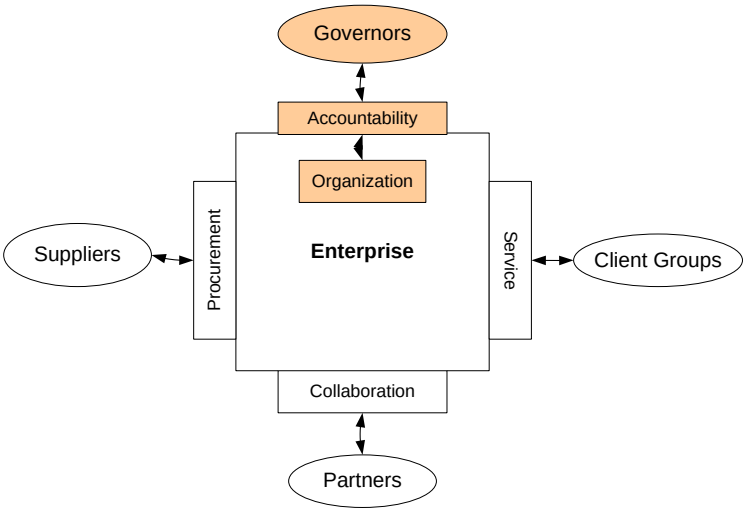
Row: 2	**WHO**	**Column 4**

Artefact Name: **Governance Model**

What/Purpose: A Governance Model represents interactions between enterprise governors (external to the enterprise) and the organizations that make up the enterprise (see the enterprise model below). These

include management and governance organizations and service providers

Description: Identifies the external entities, to which the enterprise (organization/program/service) is accountable for results, resource usage and compliance with rules. The Governance Model leads to the identification of internal management processes required to produce the above.



See Governance Model example 1.
See Governance Model example 2.

Rank: Optional

Benefit:
- Identifies accountability /reporting relationships, management functions and related processes that business initiatives have to implement
- Directives from the governing body provide mandates and constraints needed for program management.

Risk:
- Failure to understand the governance relationships.
- Failure to identify reporting structures will lead to gaps in information management, reporting or the identification of processes needed for governance purposes.
- Underestimation of the resources, effort and time needed to develop and implement programs and services.

Artefact Dependencies: Pre-requisite artefacts are:
- Organization Chart

Artefacts dependent on this artefact:
- Party Type

Recommended Practice:

Governance model is an important artefact when undertaking a horizontal business transformation initiative since it ensures that accountability (in the target state) is explicit.

The scope of the enterprise should be clearly defined so that the context for the governance model is clear.

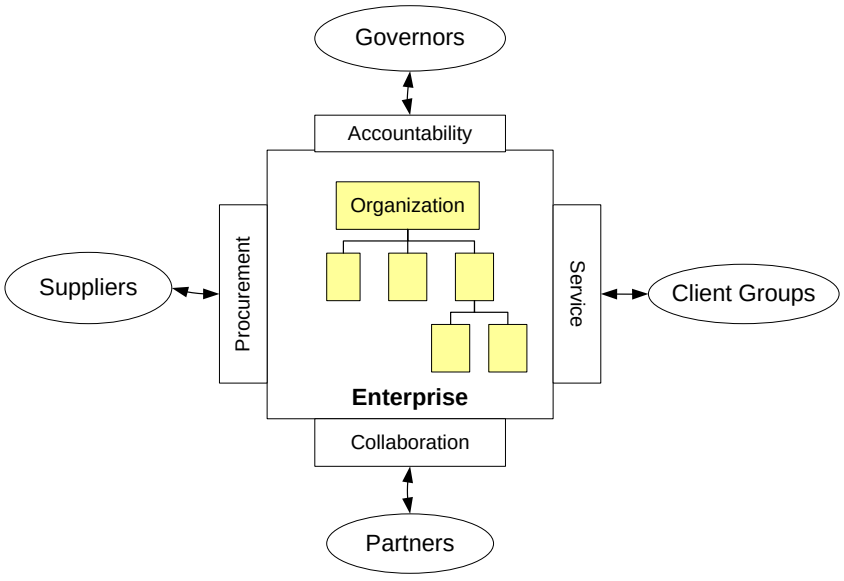
Business processes that are classified as "Governance Management" should be identified. External governing/regulatory entities must correspond to a "party type".

Row: 2	WHO	Column 4

Artefact Name: **Organization Chart**

What/Purpose: An Organization Chart represents the internal accountabilities of an enterprise.

Description: Organization represents a view of the enterprise (see the following figure of the enterprise model). The organization is defined within the larger enterprise context.



An **organizational chart** is a chart which represents the structure of an organization reporting relationships. The chart usually shows the management and staff positions that make up an organization. The chart also shows relationships between staff in the organization which can be:

- **Line** - direct relationship between superior and

subordinate.

- **Lateral** - relationship between different departments on the same hierarchical level.
- **Staff** - relationship between a managerial assistant and other areas. The assistant will be able to offer advice to a line manager. However, they have no authority over the line manager actions.
- **Functional** - relationships between specialist positions and other areas. The specialist will normally have authority to insist that a line manager implements any of their instructions.

There are three different types of organization charts:

- **Hierarchical**: An **hierarchical organization** is structured in a way such that every entity in the organization, except one, is subordinate to a single other entity. This is the dominant mode of organization among large organizations; most corporations, governments, and organized religions are hierarchical organizations.
- **Matrix:** Large organizations often use **matrix management**. Large projects are organized with teams that work on a functional, rather than a project, basis. Under matrix management, all people who do one type of work are in a pool. For example, all architects may be in one architect department and report to an architect manager. These same architects may be assigned to different projects and report to a project manager while working on that project. Therefore, each architect may have to work under several managers to get his or her job done.
- **Flat organization** (also known as *horizontal organization*) refers to an organization structure with few or no levels of intervening management between staff and managers. The idea is that well-trained workers will be more productive when they are more directly involved in the decision making process, rather than closely supervised by many layers of management.

See Organization Chart example 1.
See Organization Chart example 2.

Rank: Optional

Benefit: A quick way to identify what organization units and managers need to be involved (for input, advice or approval purposes) in business initiatives that will affect their program, services, business processes or resources.

Risk: A missing or incorrect Organization Chart requires project staff to find out who needs to be involved through informal methods. This method may lead to gaps in information and approvals through the exclusion of key parties.

Artefact Dependencies: Pre-requisite artefacts are:

- Role

Artefacts dependent on this artefact:

- None

Recommended Practice: An Organization Chart may be prepared and used without reference to any other artefacts. However, it is desirable that the terms used in organizational unit names are the same words used to identify programs, services, business processes and resource types

In order to accommodate the fluid nature of organizations do not include the names of individuals in positions.

Generally speaking, when undertaking a major business transformation, organization structure would be developed after determining the strategies and services and processes. In this way, organization structure would best reflect "how to carry out the business of the enterprise".

Row: 2	**When**	**Column 5**

Artefact Name: **Business Scenario**

What/Purpose: A **business scenario** is a description of an event or a series of events. It is also an account of a projected course of action, events or situations that is used to check for completeness of the business model. Scenarios describe ways that work processes are carried out in a business.

Description: A **business scenario is** a textual description of expected and alternate sequences of events. It is used to test and elaborate designs and is the basis for defining business requirements.

A Business Scenario must be created using one of the two following notations:

- OPS Business Scenario
- UML Business Use Case, see http://www.omg.org

See OPS Business Scenario template.
See OPS Business Scenario example.

See UML Business Scenario example.

Rank: Mandatory

Benefit: Helps understand different ways that events could unfold, by providing a narrative description. Using plain language, it describes what the business does in response to an event.

Risk: Missing or incorrect Business Scenarios increases the risk that the business is not able to respond effectively to plausible situations (expected and alternate).

Artefact Dependencies: Pre-requisites artefacts are:
- Business Process Model
- Event Type
- Business Rule Profile

Artefacts dependent on this artefact:
- System Functional Requirements

Row: 2	**WHEN**	**Column 5**

Artefact Type: **State Transition Diagram**

Why/Purpose: A State Transition Diagram shows change in components of business models over time in response to business events.

State transition modeling is a useful technique to:
- Discover business processes and rules;
- Confirm list of business events, resources;
- Communicate with business audience.

State Transition Diagrams complement business scenarios describing possible states of business components and transitions among them without explaining how the transitions happen.

Description: A State Transition Diagram describes all of the states that a business component can have, the events under which the component changes state (transitions), the conditions that must be fulfilled before the transition will occur, and the processes required for the transition.

See State Transition Diagram artefact example.

Recommended Practice:

Modeling guidelines

Use State Transition Diagrams to:

- Show the lifecycle of a service or process output;
- Identify information required to define a state; and
- Validate functional scope of business modeling

Modeling standards

- UML (www.omg.org)

Artefact Dependencies:

Elements used in a state transition diagram are either derived from or created in other architecture artefacts. In other words, they relate to the traceability between various artefacts.

Prerequisites:

- Resources;
- Events; and
- Business Rules.

Artefacts dependent on this artefact:

- Business Rules; and
- Business Process.

Rank: Optional

Benefit: Can facilitate the discovery of state-based business rules. May help to discover business processes, scenarios, and events. May help business analysts in development of business requirements.

Risk: Business rules may be missed, and the effects of change may not be accounted for when designing the 'to be' business.

Row: 2	**WHY**	**Column 6**

Artefact Type: **Service Objectives**

What/ Purpose: Check for alignment of business objectives with business/program/policy goals.

Description: Service objectives measure the extent to which a service output contributes to program goals (effectiveness), quantify compliance with service standards (quality) and quantify relationships between units of service delivery and consumption of resources required to deliver service (efficiency).

See Service Objectives artefact template.

See Service Objectives artefact example.

Rank: Mandatory

Benefit: Provides the metadata required for supporting performance measurement.

Risk: Acceptance of the 'to be' business model may be problematic without the means of measuring performance.

Artefact Dependencies: Pre-requisites artefacts are:

- Goal Type
- Service Profile

Artefacts dependent on this artefact:

- None

Row: 2	**WHY**	**Column 6**

Artefact Type: **Business Rule Source**

What/ Purpose: This artefact is used to document the authoritative sources of Business Rules and provide information about those sources. It is important to document the source of a business rule because it provides the rationale for its existence and makes it easier to manage change.

An enterprise articulates business rules to meet specific objectives including:

- Meeting program management accountabilities (e.g. constraints such as privacy).
- Achieving program strategies (to achieve program outcomes).
- Enabling the delivery of effective, efficient and quality services.

Description: Business Rule Source types include**:**

Authoritative instruments that articulate the authority of the program. One or more of these sources is listed in the Mandate artefact.

Authoritative instruments that articulate constraints of the program. Program constraints can be OPS operational policy, legislation such as FIPPA etc.

See Business Rule Source artefact template.

See Business Rule Source artefact example.

Rank: Mandatory

Benefit: The business rule sources govern the existence of a business rule. It is important to document the source of a business rule because it provides the rationale for its existence.
It is important to document the source of a business rule because it links a business rule to an authoritative instrument. When the instrument changes (e.g. legislative change) it is easier to manage change.

Risk: The risks of not producing this artefact include:
- Assuming a rule is authoritative even though it does not have an authoritative source.
- Lack of rationale for business rules.

Artefact Dependencies: Pre-requisites artefacts are:
- Mandate

Artefacts dependent on this artefact:
- None

Recommended Practice:
- This artefact is developed incrementally as the business architecture is built.
- The source should be the actual source used. That is, a program may have a specific policy that was put in place to implement a legislative requirement. The assumption would be made that the policy accurately reflects the requirement of the legislation and it is not necessary to go to the original source document.
- Business rules for the "as is" description of the program are codified in sources such as existing policy, legislation etc.
- The business architecture for the "to be" description of the program may identify the need for new business rules. However, the business architecture does not represent an "authoritative" source. This artefact lists only authoritative sources (see Business Rule Profile artefact for explanation of how to handle "notional business rules").
- The source type would list the type of instrument:
  - Legislation

- Policy
- Treaty
- Contract etc.

- The Name would list the name of the specific source instrument, for example, a specific Act.

- The reference would include:
  - Location of the source; and
  - Specific sections of the source that are relevant.

Row: 2	**WHY**	**Column 6**
Artefact Type:	**Business Rule Profile**	
What/ Purpose:	Business Rules govern, constrain or influence behaviour in order to achieve program strategies and hence program outcomes, meet program management accountabilities (e.g., constraints such as privacy) and enable the delivery of effective, efficient and quality services.	
Description:	A business rule statement is a plain language statement which describes/documents a business rule. It is atomic, that is, cannot be subdivided. Business rule statements are associated with their source (an authority identified in the business rule source artefact), the process to which they apply, and an identifier for each rule. Business rules either already exist, are modified, or are new (proposed). If a proposed business rule does not have a source, it will be identified as "Unassigned". A Business Rule Source must be identified as part of the initiative. Business rules should be made explicit so they can be managed independently of process and application. Business rules represent a starting point for identifying business requirements and system rules for an automation project. See Business Rule Profile artefact template. See Business Rule Profile artefact example.	
Rank:	Mandatory	
Benefit:	Making business rules explicit allows them to be managed independently of process and provides a business with agility when changes occur. Documenting business rules as part of a change initiative allows managers to explicitly approve the set of rules. These rule statements can then be encoded in applications and represent	

explicit instructions to developers. When business rules are made explicit they can be reused.

Risk: The program may not be operating in accordance with government direction.
When business rules are not explicit, application developers must infer the rules. The rules are then encoded in applications and are not explicitly known to the business.

The implications of changing legislation or government direction are not easily understood.

Artefact Dependencies:

Pre-requisites:

- Business Rule Source
- Mandate

Artefacts dependent on this artefact:

- State Transition
- Business Scenario
- Business Process

Recommended Practice:

Business rule statements should follow plain language guidelines:

- Uses terms and wording that is clear and acceptable to both the Business and IT
- Is complete, readily usable, readily understandable, simple

Business rule statements conform to "Subject, Verb, Object, Constraint /Permission".

- Both **Subject** and **Object** must be valid Terms (see the Semantic Model artefact type for a definition)
- **Action/Verb Phrase** must include one of CAN/MAY/MUST or one of CAN NOT/MAY NOT/MUST NOT.
- **Constraint/Permission** must use Terms for any measures or targets (e.g. "by Fiscal Year End" as opposed to "by March 31st".

Row: 2	**WHY**	**Column 6**

Artefact Name: **Program Logic Model**

What/Purpose: The program logic model is a diagrammatic representation that helps design Programs. It is used to test that each service contributes to the achievement of at least one of the Program's goals and that the program goals contribute to the government's strategic goals or directions. Therefore ensuring that the correct services are in place to achieve the stated program goals and that the correct program goals have been stated to achieve the strategic direction set by government.

Description: A model that provides a visual representation of the alignment of services to programs as it shows how outputs produced by services contribute to program outcomes, program outcomes support program impacts and impacts support strategic outcomes of Government. Alignment occurs when the output of a service contributes to the outcome of a program. The output must demonstrate a measurable impact on reducing the target group needs.

Note that other (peer) programs may be required to achieve the strategic goals.

See Program Logic Model example.

Rank: Optional

Benefit: Supports results-based planning and the development of performance measures.

Helps define programs and services that align with government priorities. Allows program managers to map services and their outputs to the strategic direction set by government.

Targeting Outcomes of Programs focuses on outcomes in planning, implementing, and evaluating programs.

Facilitates program planning and delivery.

Enhances buy-in and team building among program stakeholders and participants by promoting participation and ownership.

Aids stakeholders in understanding the goals, expectations and outcomes associated with the program.

Demonstrates how different components of a program, such as program goals, program impact and service outputs are linked.

Assists in identifying unintended consequences of the program.

Risk: If these linkages are not analyzed, there is a risk that new Program designs will not further the strategic priorities of the government or that continuing Programs will fall out of alignment with them.

Artefact Pre-requisite artefacts are:

Dependencies:

- Program Goals (outcomes and impacts)
- Service (Names and Service Outputs)
- Government Strategic Goals (sourced from Results Based Planning)

Artefacts dependent on this artefact:

- Business Scenarios

Artefact Dependencies:

Pre-requisite artefacts are:

- Goals (outcomes and impacts), including government strategic goals (sourced from Results Based Planning)
- Service (Names and Service Outputs)

Artefacts dependent on this artefact are:

- Service Profile
- Service Life Cycle

Recommended Practice:

Ensure that government strategic goals and priorities are reflected in this artefact.

Row 3: Logical

Row: 3	WHAT	Column 1

Artefact Type: **Logical Data Model**

What/Purpose: A Logical Data Model (LDM) represents in full detail the in- scope business entities, their relationships, and their attributes.
It is used to describe the data requirements and needs in support of the in-scope business activities in as much detail as possible without any regard to the physical implementation environment or performance considerations.

Description: The Logical Data Model is a fully attributed and normalized data model with detailed definition of the data entities, relationships, and attributes. This model is independent of physical constraints and considerations, such as organizational ownership, geographic location, or technology of implementation.

It can be used:

- To enhance communication between IT and business;
- To discover, uncover, and clarify business rules involving the business information;
- To understand all the required business information and data;
- As a common reference to describe how business activities (functions) in the scope to produce their respective outcome by manipulating data (CRUD) and exchanging messages (flows); and
- To provide the underlying structure of a physical data model (PDM).

A Logical Data Model should be directly traceable to the corresponding Conceptual Data Model.

A Logical Data Model must be diagrammed using one of the following notations, and must be accompanied by detailed metadata as specified in GO-ITS 56.3, Information Modelling Standard:

- Entity Relationship diagram, or a
- Unified Modeling Language (UML) Class Diagram representing only entity classes, without showing any methods on these classes.

See Logical Data Model artefact example 1.
See Logical Data Model artefact example 2.

Rank: Mandatory

Benefit: This model has the following benefits:

- It facilitates a complete understanding of in scope business entities and associated information requirements.
- It provides assistance and enablement for further technical analysis, design, and physical implementation.
- It is authoritative and provides a formal data definition of business concepts.
- It can be used as a reference when integrating data from multiple data sources.
- To further elaborate business information requirements and establish a baseline for the creation of a physical data model with verifiable traceability.
- It can be used to assess the alignment of existing physical data models with business requirements.

Risk: The risks of not developing this model include:

- Lack of full data analysis leads to data definition ambiguities and inaccuracies.
- Lack of data normalization leads to data redundancies, and inefficient solution design and implementation later.

Artefact Dependencies: Pre-requisites artefacts are:

- Conceptual Data Model

Artefacts dependent on this artefact:

- Physical Data Model
- Logical Dimensional Model

Artefact Type:	**Logical Data Model (Acquired Solution)**
Guidance:	A Logical Data Model for Acquired Solution must be produced only if gaps exist between data requirements in the Conceptual Data Model (Acquired Solution) and the data solution acquired, and when the resolution for the data gaps requires database solution customization. The Logical Data Model for Acquired Solution must be based on both the Conceptual Data Model (Acquired Solution) and the data solution acquired. It may be extended with new data entities, attributes, and relationships to cover data requirements of the customized solution components for both the persistent and interface data solution.
Required	Logical Data Model (Acquired Solution) that includes: • Model Diagram showing new data elements, i.e. data entities, attributes, and relationships connecting the existing and

	extended data entities; • Data dictionary including definitions for new data entities, relationships, attributes, and additional requirement specifications of data types, sizes, attribute data value domains, and key identifiers; and • Data requirements of the customized solution components for both the persistent and interface data solution. See LDM (Acquired Solution) example.
Not required	Logical Data Model for the entire solution
Recommended Practice:	Refer to the most current version of *Guidance for the Acquisition and Integration of Acquired Solutions*.

Row: 3	**WHAT**	**Column 1**
Artefact Type:	**Logical Dimensional Model**	
What/Purpose:	This artefact provides information about the logical design of the data mart being built. It includes the details of the fact entities, facts, dimensions, dimension attributes, and the relationships connecting the facts with the dimensions. The Logical Dimensional Model is used to illustrate the capability of the data mart design to meet the decision support requirements and to address the specific information requirements. The Logical Dimensional Model also illustrates the traceability from the high-level business information requirements, as presented in the Fact and Dimension Matrix, to the data mart design. It is also used to visually communicate the appropriate level of granularity and business hierarchy of the information required for decision support reporting.	
Description:	The Logical Dimensional Model constitutes the logical design of a solution for the decision support needs. It includes sufficient details of the data mart design without addressing the particular technology implementation. The Logical Dimensional Model should be built from the Fact and Dimension Matrix and re-use as much as possible the existing data definitions and inherent business hierarchical relationships from existing Logical Data Model(s) (i.e. data sources) to achieve the consistency of data across the business domain. See Logical Dimensional Model artefact example	

Rank: Optional

This artefact is considered as **mandatory** for a project that develops or acquires data warehouse and/or data mart based solutions for decision support initiatives.

Benefit: This artefact illustrates a clear understanding of the business information requirements. In particular, it assists in:

- Creating sufficient detail for a solution design that can be used for future planning and implementation.
- Explaining the contents of the decision support requirements and solution to the business sponsor.

Risk: If this artefact is not developed, it will result in:

- An increased risk that a decision support solution will not meet the business needs nor can be integrated into a longer-term solution.
- The subsequent solution model (i.e. physical dimensional model) being defined directly from the source system data and metadata leading to a lack of conformed dimensions hence impacting opportunities for data integration.
- The lack of a unified view of conformed dimensions and the increased risk of creating non-additive facts, mixing fact granularity or missing an important dimension.

Artefact Dependencies: Pre-requisite artefacts are:

- Fact and Dimension Matrix
- Logical Data Model

Artefacts dependent on this artefact are:

- Physical Dimensional Model

Recommended Practice: Each decision support solution is represented as a star schema(s) with the fact entity consisting of quantitative and/or qualitative measurements and being joined to a set of dimension entities which include descriptive attributes. The entire decision support solution set is represented as a series of interconnected star schemas. The interconnection is based on conformed dimensions.

Artefact Type: **System Functional Requirements**
Note: This artefact is defined with two versions, a generic version and an object-oriented version. The generic version consists of a single component. The object-oriented version consists of two components: the System Use Case Model and the Use Case Specification.

Version: **Generic**

What/Purpose: This artefact captures and presents information that defines the required functional capabilities of the system.

Description: Functional requirements capture the intended behaviours of the system. These behaviours may be expressed as services, tasks or functions the system is required to perform.

The System Functional Requirements described by using non-UML notation should be documented using clear, specific, well-articulated statements and/or detailed mathematical functional descriptions. Non-UML techniques may be used for small systems with minimum enterprise architecture impact and for non-OO systems.

The System Functional Requirements should include:

- Introduction and scope
- Specific requirements
- Functional requirement 1…n
- Security requirements
- Supporting information

See System Functional Requirements artefact template.

Rank: Mandatory (either generic or UML version)

Benefit: This artefact captures the functional requirements used to architect and design the software system. It specifies the system's intended functions and its environment, and serves as a contract between the customer and the developers. This information is an essential input to activities in system architecture, design, and test.

Risk: Failure to capture this information can result in:

- Requirements mismanagement
- Missing functional requirements
- Gaps in security requirements
- Inability to trace business functions to functional requirements to the source code that realize them
- Difficult to understand system requirements
- Project delays or failure

Dependencies:	Pre-requisite documents are: • Business Scenario • Business Rule Profile Documents dependent on this artefact are: • Logical Application Design Document • Physical Application Design Document • System Architecture Document

Artefact Type:	**System Functional Requirements (generic version) Guidance for Acquired Solution**
Guidance:	Provides the minimum high-level functional capabilities required *before procurement* of the intended Acquired Solution. Functional requirements capture the intended behaviours of the Acquired Solution. These behaviours may be expressed as services, tasks or functions the system is required to perform. This specification informs the requirements section of the Request For Proposal (RFP).
Required:	Document sections: ▪ Introduction and scope ▪ Specific requirements ▪ Functional requirement 1…n ▪ Security requirements; and ▪ Supporting information (e.g., sample screens, reports).
Not Required:	System Use Case Model Use-Case Specification
Specific Guidance for GO-ITS/VoR:	For Checkpoint 1 – The following elements are required when software products listed on the Government of Ontario Information Technology Standard (GO-ITS) or Vendor of Record (VoR) Agreements is selected as the Acquired Solution (specifically Curam, Siebel and OpenText). The System Functional Requirement artefact is a key artefact, which highlights the project's unique system functional needs. Articulating these requirements forms the basis for ensuring appropriate due diligence is undertaken in selecting the right Acquired Solution.

	Projects should focus their documentation on the following key elements: ▪ Specific requirements ▪ Functional Requirement 1…n ▪ Security requirements
Recommended Practice:	Refer to the most current version of *Guidance for the Acquisition and Integration of Acquired Solutions*.

Row: 3	**HOW**	**Column 2**
Artefact Type:	**System Function Requirements (continued)**	
Version:	**Object-Oriented**	
	System Use Case Model	
What/Purpose:	The System Use Case Model captures the functional requirements of the system to be developed using UML diagrams. It is the foundation for the system to be built.	
Description:	The System Use-Case Model is the foundation for most of the analysis and design artefacts and it describes the system interactions with the clients and / or with other external systems. It captures the functional requirements of the system to be developed and includes a collection of packages containing mainly use-case diagrams traceable to the corresponding Business Scenarios, business use-cases and use-case realizations. The System Use-Case Model is correct only when it describes the system's functionality. Each system use-case must have an associated Use-Case Specification document that should include a brief description, flow of events, special requirements, pre-conditions and post-conditions. The System Use-Case Model is generally used in all phases of the development cycle and is an essential input to designing and testing the system. See *System Use Case Model* – diagram example. See Use Case Specification artefact template.	
Rank:	Mandatory (either generic or UML version)	

Benefit: The System Use-Case Model is a model of the system's intended functions and its environment, and serves as a contract between the customer and the developers. The use-case model is used as an essential input to activities in analysis, design, and test.

Risk: Failure to create this model can result in:

- Missing functional requirements.
- Incapacity of tracing requirements from Business Use-Cases to Use-Case Realizations.
- Difficult to understand system requirements.
- Project delays or failure.

Artefact Dependencies: Pre-requisite artefacts are:

- Business Scenario
- Business Use Case Model
- Business Rule Profile

Artefacts dependent on this artefact are:

- Logical Application Design Document
- System Architecture Document
- Physical Application Design Document

Recommended Practice: The System Use Case Model includes (diagrams should be supported by descriptions):

- Use-Case Package diagram
- Use-Case diagram
- Actors diagram
- Class diagrams to show traceability

Diagrams Model	UC	O	C	S /C	S	A	Com	D
Use-Case	M		m		m	M		

UC=Use-Case; O=Object; C=Class; S/C = Sequence/Collaboration
S=Statechart; A= Activity; Com=Component; D=Deployment; M=Major; m=minor

Row: 3 **HOW** **Column 2**

Artefact Type: **System Function Requirements (continued)**

Version: **Object-Oriented (continued)**

Use-Case Specification

What/Purpose: To provide detailed information for each use-case included in the System Use-Case Model. Each system use-case must have an associated Use-Case Specification file.

Description: Each System Use-Case should include descriptions, which are stored in separate files called Use-Case Specification.

Use-Case Specifications contain a description of the flow of events describing the interaction between actors and the system. The specification typically contains information such as preconditions, post-conditions, special requirements and key scenarios.

The detailed descriptions are based on the detailed business descriptions associated with business use-cases. If the Business Use-Case Model was not created then the detailed descriptions of the system use-cases are based on the knowledge of the business domain experts.

See Use-Case Specification artefact template.

Rank: Mandatory (when the System Use-Case Model was created)

Benefit: Use-Case Specification captures the required system behaviour from the perspective of the end-user in achieving one or more desired goals

Risk: Failure to create this specification can result in:

- Incomplete System Use-Case Model
- Incomplete description of the system's functionality.
- Failure to optimize the Application Design Model and to identify all reusable design elements.

Artefact Dependencies: Pre-requisite artefacts are:

- Business Scenario
- Business Use Case Model
- Business Rule Profile

Artefacts dependent on this artefact are:

- Logical Application Design Document
- System Architecture Document
- Physical Application Design Document

Recommend Practice: The Use-Case Specifications includes:

- The name of the use case
- Basic Flow of Events
- Alternative Flows
- Sub flows
- Key Scenarios
- Pre-conditions
- Post-conditions
- Extension Points

- Special Requirements
- Business Rules Specification

Row: 3	HOW	Column 2
Artefact Type:	**System Architecture Document**	
What/Purpose:	The System Architecture Document provides a comprehensive overview of the solution, using a number of different architectural views to depict various aspects of the system.	
Description:	The System Architecture Document describes the architecturally significant decisions, which have been made on the project. It defines a complete, high-level overview of the system, by including all UML and non-UML architectural views to describe different aspects of the system, and by emphasizing the logical partitioning and the functional components. The document also contains criteria used to partition the application's functionality and architectural patterns. The application building blocks and components description are the most important sections of this document. See the System Architecture Document template:	
Rank:	Optional	
Benefit:	The System Architecture Document serves as a communication medium between the software architect and other project team members or clients, regarding architecturally significant decisions, which have been made on the project.	
Risk:	The failure of creating this artefact can result in: • Missing the big picture • Incapacity of aligning the software solution to enterprise best practices • Incapacity of identifying and using common components • Additional costs	
Artefact Dependencies:	Pre-requisite artefacts are: • System Functional Requirements or System Use-Case Model Artefacts dependent on this artefact are: • None	
Recommended Practice:	The System Architecture Document includes: • Introduction	

- Architectural Representation
- Architectural Goals and Constraints
- Use-Case or Requirements View
- Logical View
- Implementation View
- Process View
- Deployment View
- Data View

Row: 3	HOW	Column 2

Name: **Logical Application Design Document**
Note: This document is defined with two versions, a generic version and an object-oriented version, the Logical Application Design Model.

Version: Generic

What/Purpose: The Logical Application Design Document provides a view for representing the transformation of the functional requirements. It specifies and illustrates how the functional requirements are transformed and mapped onto classes and their interrelationships. Logical design is intentionally infrastructure independent and provides an accurate, detailed and complete description of the logical design for the entire application.

Description: The design must document how each of the requirements specified in the System Functional Requirements and Supplementary Specifications will be logically accomplished or realized as a well-defined set of interactions between various objects. The design also identifies interfaces reflects details such as scalability, availability, and security, and leverages existing system designs whenever possible.

The design should reflect the application architecture principles, practices, and standards; ensure the requirements traceability by cross-referencing the system requirements with logical design elements e.g. subsystems, modules; and align all aspects of the information, application, technology, security, and integration architecture, to solve specific business requirements. The design should also ensure that the application security design is reflected in the architecture.

Logical design is infrastructure independent; it does not have any details of any implementation technology.

See Logical Application Design Document template.
See Logical Application Design Document example.

UML version of this document is the high-level Logical Application Design Model.

Note: A usage pattern based approach is being encouraged and projects must clearly articulate and demonstrate the use of usage patterns in the application architecture and design. The design must clearly articulate how patterns are being used (including usage patterns) and used in the project solution.

Where an Application Usage Pattern(s) are being leveraged, clearly demonstrate the use of the Application Usage Pattern(s) in the application architecture design. The Logical Application Design Document must:

- *State the Application Usage Pattern(s) including the Application Usage Pattern name, version # and date. There is no need to copy the entire usage pattern document;*
- *Incorporate the Application Usage Pattern diagram(s) into the Logical Application Design Document. Label the reused Application Usage Pattern(s) and shade or colour code them to differentiate the usage pattern diagrams from the project-specific design elements. The logical design diagrams include the component models, base classes and interactive diagrams.*

The entire project solution design will be a collection of patterns extended with project-specific design elements (i.e., application packages, components).

See Application Usage Pattern template.

When a service-based approach (i.e., comply with SOA design principles) is being used to assemble an application, the SOA Application Service Model template must be used to articulate and illustrate the architectural design for *each* discrete "Service" being proposed or developed.

See SOA Application Service Model template.
See SOA Application Service Model example.

Note: At the logical level, there are some items that may not be detailed enough. For example:

- Static model (classes will not be fully attributed)
- Data usage will be linked to the logical data model, hence providing only a logical view.
- Detailed WSDL interface spec. may not necessarily be

available

Rank: Mandatory (either generic or UML version)

Benefit:
- Captures the most essential design elements. It reflects some of the more common choices as well as important design considerations that should be taken into account during the design phase.
- Enables reusability of usage patterns from a solution development and deployment perspective
- Enables improved consistency and alignment to existing architecture designs and solutions

Risk: Failure to create the Logical Application Design Document can result in:
- Incorrect and inconsistent abstractions of design.
- Failure to ensure design consistency in the reuse of usage patterns.
- Failure to identify reusable objects.

Document Dependencies: Pre-requisite documents are:
- System Functional Requirements or System Use-Case Model
- SOA Service Specification
- Supplementary Specifications

Documents dependent on the Logical Application Design Document are:
- Physical Application Design Document or Physical Application Design Model

Artefact Type:	**Logical Application Design Document – Guidance for Acquired Solution**
Guidance:	Provide a concise description of the logical design for the entire application. The design must document how each of the requirements specified in the System Functional Requirements and Supplementary Specifications will be logically accomplished. Some sections of the LADD may not be relevant to an Acquired Solution implementation. Those components can be viewed as "black-boxes" or self-contained solutions and their internal logical designs do not need to be

	documented. The logical design should be at a level of granularity suitable for Acquired Solutions, more coarse grain than the artefacts needed for custom designed/developed solutions. The integration architecture needs to be well-documented with close attention to all connectivity and interface points. Document, at a high-level, the structures that can be "seen", configured, or changed (integration and connectivity points, API's, protocols, and standards). Connectivity points and interface requirements to external sources need to be well-understood and documented to provide the vendor with sufficient information to respond to the integration requirements. Any custom interface design requirements must be reflected in this document.
Required:	The following items will be high-level at the pre-Request For Proposal (RFP) stage and more refined post-RFP: • Logical Application Design Goals and Considerations; • Logical Application Design Overview; • Logical Application Design – Diagrams and Descriptions • Component Model; • Custom Application Design; • Integration and Interface Design; • Security Designs; • Logical Design Mechanisms; • Error handling or recovery mechanism; and • Requirement Traceability (to functional requirements.
Not Required:	Components and sub-systems design, and Detailed Sub-system Design and Module Design.
Specific Guidance for GO-ITS/VoR	For Checkpoint 1 – The following elements are required when solution software products listed on the Government of Ontario Information Technology Standard (GO-ITS) or Vendor of Record (VoR) Agreements is selected as the Acquired Solution (specifically Curam, Siebel and OpenText): • Component Model; • High-Level Integration and interface design; and • Interface Specifications (or requirements).
Recommended Practice:	Refer to the most current version of Guidance for the Acquisition and Integration of Acquired Solutions.

Row: 3	WHERE	Column 3

Name: **Solution Pattern Match**

What/Purpose: Solution Pattern Match is used to optimize solution reuse by providing evidence of the selection, use and/or creation of usage patterns (including Application Usage Patterns and Infrastructure Usage Patterns).

This encourages the use of common architecture designs based on OPS best practices as well as the opportunity for rationalizing both application and infrastructure optimization. It provides an opportunity to identify, capture and reuse patterns or improve and update those that currently exist.

Description: A Solution Pattern Match is completed to ascertain the applicability of patterns to project specific needs.

The Solution Pattern Match consists of:

- a selected pattern;
- how the pattern fits with the project's intended use of it;
- how the pattern Quality Level Metrics (QLM) fit with the solution specific requirements;
- a reference to available hosting environments
- any other architectural patterns being used together with a rationale

Project teams will undertake a Solution Pattern Match to determine whether approved patterns (including Application Usage Patterns and/or Infrastructure Usage Pattern) are available for reuse or creation/update of an existing pattern is appropriate. The project team will analyse the business and system functional requirements for its project and match the requirements against the functionality available in the approved patterns.

Improvements to existing approved usage patterns or entirely new usage patterns may also be proposed. If the project team is planning to contribute **new Application Usage Patterns or Infrastructure Usage Pattern(s),** use the following template to describe and illustrate the design.

See Solution Pattern Match template.

Rank: Mandatory

Benefit:
- Speed up development and implementation efforts by leveraging approved architecture usage patterns;
- Reduce cost in design and provisioning of existing infrastructure.

Risk:
- Increased costs to design and provision same application functionality multiple times
- Opportunities for application and infrastructure rationalization may be missed resulting in increased costs in IT solution delivery.
- The continued proliferation of 'unique' solution designs that are provisioned in 'unique' hosting environments

Document Dependencies: Pre-requisite:
- System Functional Requirements
- Supplementary Specification

Documents dependent on the Solution Pattern Match are:
- Logical Application Deployment Model
- Logical Application Design Document or Logical Application Design Model
- Physical Deployment Model

Row: 3	**HOW**	**Column 3**

Name: **Logical Application Deployment Model**

What/Purpose: To depict the logical design at a sufficient level of detail so that:
- Stakeholders can verify that their requirements are being taken into account and can input into the design before proceeding to physical design and implementation.
- Transformation issues can be identified and addressed.

Description: Annotated, logical technology diagrams (including depiction of reused application usage patterns), system software, hardware, and network components that address system functional requirements, as well as non-functional requirements such as:

- Security and privacy requirements.
- Disaster Recovery View. The Disaster Recovery View is used to capture information related to IT recovery solution/mechanism in the event of a disaster (See Disaster Recovery View template).
- Quality Level Metrics: includes Conventional, Extended and Adaptiveness/Modifiability (see Quality Level Metrics template).

See Logical Application Deployment Model template.

Note: A usage pattern based approach is being encouraged and projects must clearly articulate and demonstrate the use of usage patterns in the application architecture and design. The design must clearly articulate how patterns are being used (including usage patterns) and used in the project solution.

Using the Application Usage Pattern(s) in the Logical Application Deployment Model:

- Clearly state the Application Usage Pattern(s) including the name, version # and date. There is no need to copy the entire pattern document;
- Incorporate the appropriate Application Usage Pattern diagram(s) into the Logical Application Deployment Model – Master View diagram. Label the reused Application Usage Pattern(s) and shade or colour code them to differentiate the pattern diagram from the project-specific design elements;
- illustrate the project-specific design elements (i.e., application packages) which collectively will constitute the solution design;

See Application Usage Pattern template.

Rank: Mandatory

Benefit: Provides a starting point to engage other groups, especially operations, to provide input early into the design.

Provides an authoritative source of requirements and design constraints for input into the solutioning process.

Risk: Designs may not follow approved patterns and therefore may not represent best practices. The result may be a suboptimal, under-performing design. Designs may proceed too far without taking essential security/privacy/recovery and quality requirements into account.

Dependencies: Pre-requisite documents are:

- Logical Application Design Document or Logical Application Design Model
- Supplementary Specifications
- System Functional Requirements

- Application Usage Pattern

Documents dependent on the Logical Application Deployment Model are:
- Physical Deployment Model
- IT Solution Concept

Artefact Type:	**Logical Application Deployment Model – Guidance for Acquired Solution**
Guidance:	Provide high-level annotated logical technology diagrams that depict system software, hardware and network components that address system functional requirements as well as non-functional requirements before procurement or selection of the Acquired Solution. This artefact informs the requirements section of the Request for Proposal (RFP) and provides prospective vendors sufficient information to understand the IT environment in which the Acquired Solution will reside.
Required:	Document sections: ▪ Master Diagram (including integration nodes) ▪ Security View ▪ Quality Level Metrics; and
Not Required if Issuing an RFP:	▪ Disaster Recovery View ▪ Operationally significant aspects (e.g. indication of scale out/scale up strategy, system management/monitoring servers/agents, etc.).
Additional Requirements Specifically for Solutions Listed on GO-ITS and VoR	For Checkpoint 1 – The following is required when solution software products listed on the Government of Ontario Information Technology Standard (GO-ITS) or Vendor of Record (VoR) Agreements are selected as the Acquired Solution (specifically Curam, Siebel and OpenText): • Disaster Recovery View
Recommended Practice:	Refer to the most current version of *Guidance for the Acquisition and Integration of Acquired Solutions*.

Row: 3	**WHY**	**Column 6**
Name:	**Supplementary Specification**	

What/Purpose: This document captures system requirements that are not captured in use cases or the System Functional Requirements. It focuses on the non-functional requirements of the proposed solution.

Description: The Supplementary Specification captures special requirements, which are not captured in the System Functional Requirements document and other non-functional requirements documents. These requirements may include:

- Legal and regulatory requirements and application standards.
- System and development environment requirements, compatibility requirements, and design constraints.
- Integration with existing systems
- Security Specification

See the Supplementary Specification template.

Rank: Mandatory

Benefit: To identify and describe the non-functional requirements regarding system performance and reliability, standards, integration, development and design constraints, volume and sizing, databases, special hardware, network architecture, network connections, availability, disaster recovery, security and assumptions and issues.

Provides an understanding of the limitations of the environment in which a system will be implemented.

Risk: The failure of creating this document can result in:

- Missing special requirements.
- Application failure to achieve expected performance.
- Project delays and additional costs.

Dependencies: Pre-requisite documents are:

- None

Documents dependent on the supplementary specification are:

- Logical Application Design Document or Logical Application Design Model
- Logical Deployment Model
- Physical Application Design Document or Physical Application Design Model

Artefact Type:	**Supplementary Specification - Guidance for Acquired Solution**
Guidance:	Initially the Supplementary Specification informs the Request for Proposal. Later it is refined to contribute to system integration by providing the non-functional requirements for the automated solution.
Required:	Document sections: • Reused Component Requirements (if applicable) • Purchased Components Requirements • Licensing Requirements, Legal and Regulatory, Copyright, Other Notices, and Standards
Not Required:	• Physical Data Specification • Logical Data Specification Logical Infrastructure and Deployment Specification • Physical Infrastructure and Deployment Specification • Interfaces Specification • Design Specification • Development Specification
Specific Guidance for GO-ITS/VoR:	For Checkpoint 1 – The following elements are required when solution software products listed on the Government of Ontario Information Technology Standard (GO-ITS) or Vendor of Record (VoR) Agreements is selected as the Acquired Solution (specifically Curam, Siebel and OpenText): • Reused Component Requirements; and • Purchased Components Requirements.
Recommended Practice:	Refer to the most current version of *Guidance for the Acquisition and Integration of Acquired Solutions*.

Row 4: Physical

Row: 4	WHAT	Column 1

Artefact Type: **Physical Data Model**

What/Purpose: A Physical Data Model (PDM) defines the physical implementation of the logical data requirements using a particular technology within an intended implementation platform and environment. It shows how each data element will be implemented and stored on the data store.

Description: The Physical Data Model is primarily concerned with physical limitations, performance and space requirements. For implementation purposes, objects that appear in the conceptual or logical data models may be combined or subdivided, and new objects may be introduced in order to reduce response time, to accommodate the physical limitations of the computing environment, and to improve maintainability.

For example, it may introduce new implementation objects, such as database triggers, primary key and foreign key constraints, and check constraints in the Relational Database Management System environment to ensure that business rules in the logical data model are fulfilled during the physical implementation.

It may also introduce new implementation objects such as indexes that do not contribute to the business information requirements of the system application. These new objects may be created in order to speed up response time, ensure that the application fits within the physical limitations of the computing environment, improve maintenance turnaround, etc.

A Physical Data Model should be directly traceable to the corresponding Logical Data Model.

A Physical Data Model must be diagrammed using one of the following notations, and must be accompanied by detail level of metadata as specified in GO-ITS 56.3, Information Modelling Standard, and a mapping or design document which provides traceability back to the LDM:

- Relational diagram, or
- Other formal graphical representations (e.g. tree diagram for XML model)

See Physical Data Model artefact example 1.
See Physical Data Model artefact example 2.
See Physical Data Model artefact example 3.

Rank: Mandatory

Benefit: Provides an opportunity to address physical implementation issues independent of the business meaning of the data.

Risk: The risks of not developing this model include:

- Poor system performance.
- Difficulty maintaining the database.
- Lack of utilization of some functions and features offered by a specific technology.
- Loss of productivity.

Artefact Dependencies: Pre-requisite artefacts are:

- Logical Data Model

Artefacts dependent on this artefact are:

- None

Artefact Type:	**Physical Data Model (Acquired Solution)**
Guidance:	The Physical Data Model for Acquired Solution must illustrate the database solution design including all the solution configuration changes and/or customization changes. The data dictionary for the PDM must include data name mappings between the solution table column names and business attribute names. When a vendor's Physical Data Model is not provided, the project will need to fully understand both the current capabilities and limitations of the data solution in order to determine the appropriate configuration and/or customization required. The actual content and format of the Physical Data Model depends on the availability of detailed data solution specifications provided by the vendor. The detail of the vendor's data solution specification may be in any combination of the following forms: • Physical Data Model; • DDL of the database; • Database Interface file layouts and specification; • Report samples and screen shots; • Functionality description, process specifications, user manual and guide; • XML schema definitions of database, interface file layouts, and/or message layouts; and/or • Solution Module Scripts / Codes representing business

	application logic, data access and database updates.
Required	A Physical Data Model that includes: • all the configured and customized solution components; and • data dictionary with data name mappings between the table column names and business attribute names A set of physical interface data files and message data layouts specifications for the automated application interfaces including those customized and configured interface solutions. Obtain the vendor solution Physical Data Model if the service support agreement is such that the OPS is responsible for the future maintenance and support of the product database. See Customized Solution PDM example See Complete Physical Data Model example
Recommended Practice:	Refer to the most current version of *Guidance for the Acquisition and Integration of Acquired Solutions*.

Row: 4	**WHAT**	**Column 1**
Artefact Type:	**Database Inventory**	
What/Purpose:	This is a list that provides an account of exiting data states (i.e., files, databases, datamarts, etc.) and their pertinent characteristics. This list facilitates the further analysis of data conversion and data conversion strategies from current to target platforms.	
Description:	A listing of all of the files and physical databases within the area of the project, problem domain, or area of investigation. The area of interest could be defined along organizational, functional, or systems lines. See Database Inventory artefact example.	
Rank:	Optional	
Benefit:	This listing provides information about the physical data environment.	
Risk:	The ability to support data and the ability to ensure security and privacy may be compromised by insufficient knowledge about its implementation.	
Artefact Dependencies:	Pre-requisite artefacts are: • None	

Artefacts dependent on this artefact are:

- None

Row: 4	WHAT	Column 1

Artefact Type: **Physical Dimensional Model**

What/Purpose: This artefact describes the internal data structures used by the data warehouse or a data mart.

The Physical Dimensional Model provides information about the physical implementation of the data mart using a specified database management system (relational or multi-dimensional). It is primarily concerned with physical limitations, performance and space requirements.

Description: The Physical Dimensional Model is used to show how and where each data element will be implemented and stored on the database. The relationships in the Logical Dimensional Model are transformed and presented by the primary keys in the dimension tables and the corresponding foreign keys as a part of the multi-part key in the fact table.

The Physical Dimensional Model should be built from the Logical Dimensional Model.

See the Physical Dimensional Model example.

Rank: Optional

This artefact is considered as **mandatory** for a project that develops or acquires data warehouse and/or data mart based solutions for decision support initiatives.

Benefit: This artefact focuses on meeting a specific set of information retrieval and analysis requirements by designing the database in such a way as to be easy and efficient to query.

Risk: Not creating this artefact may result in inefficient query processing and handling, slower response times, or under-utilization of the decision support solution.

Artefact Dependencies: Pre-requisite artefacts are:

- Logical Dimensional Model

Artefacts dependent on this artefact are:

- None

Recommended Practice: **Physical Dimensional Model Diagram:** A model diagram which presents / describes the internal data structures. It consists of one or more fact tables(s) each with a multi-part key, and a set of smaller dimension tables, each with a single-part primary key that corresponds exactly to a component of the multi-part key in the fact table.

Row: 4	**HOW**	**Column 2**

Name: **Physical Application Design Document**
Note: This Document is defined with two versions, a generic version and an object-oriented version, the Physical Application Design Model.

Version: Generic

What/Purpose: The detailed Physical Application Design Document provides a comprehensive view for representing the transformation of the Logical Application Design constructs into detailed application design specifications. It takes into account the technology infrastructure and provides an accurate, detailed, and complete description of the detailed physical design for the entire application.

Description: The design must present how the high-level logical application design specified in the Logical Application Design Document will be physically accomplished based on selected implementation technology and infrastructure.

This design document must illustrate, in greater detail, the physical application architecture/design describing how to physically accomplish the system requirements described in the logical application design. It should provide a clear understanding of the structure of the application including custom applications and integration interfaces. This should be sufficiently detailed to guide developers in understanding the architectural foundation upon which to construct/assemble the solution.

Almost all details should be known at this point in the design process. Detailed physical design describes how to utilize the identified implementation environment on a selected infrastructure to physically implement the subsystem's logical design, module design, and system interface design. Design objectives in the Quality Level Metrics such as reliability,

availability, scalability, interoperability, and the use of common components/services and Application Usage Pattern(s) and Infrastructure Usage Pattern(s) should be adequately reflected in the physical design. Leverage existing system designs whenever possible.

The detailed design must align with application architecture principles, practices and standards and ensure traceability with system functional requirements. It must align all aspects of the information, application, technology, security, and integration architecture to solve specific business requirements.

The physical application design must also illustrate the application security design in detail. This includes design countermeasures for application security concerns such as input validation, authentication, authorization, cryptography, session management, parameter manipulation, audit and logging and exception management.

The Physical Application Design Document is implementation technology and infrastructure specific.

See Physical Application Design Document template.
See Physical Application Design Document example.

The **UML version** of this Document is the Physical Application Design Model.

Note: When a usage pattern based approach is being used in the logical application design, projects must continue to illustrate the use of usage patterns in the physical application architecture and design.

At the physical level, clearly demonstrate the reuse and refinement of Application Usage Pattern elements (e.g. base classes). The design must clearly illustrate how the usage patterns are physically implemented and leverage in the solution design.

See Infrastructure Usage Pattern template.

When a service-based approach (i.e., comply with SOA design principles) is being used to assemble an application, the SOA Application Service Model template must be used to articulate and illustrate the architectural design for *each* discrete "Service" being proposed or developed.

See SOA Application Service Model template.
See SOA Application Service Model example.

At the physical level, all items that were not completed in logical architecture must be well-defined with all required design details related to implemented technology. The SOA Application Service Model template must be fully described and illustrate the physical architecture.

Rank: Mandatory (either generic or UML version)

Benefit: The Physical Application Design Document is a platform-specific design describing how to physically accomplish the system requirements realized in the high-level Logical Application Design Document. It provides a clear understanding of the structure of an application and is used as essential input to activities in implementation and test.

Risk: Failure to create the Physical Application Design Document can result in:

- Confusion in implementing the design
- Security risks
- Misuse of the development language packages or third party packages
- Project delays and potentially additional cost due to lack of clear architecture design requirements.

Dependencies: Pre-requisite documents are:

- Solution Pattern Match
- Application Usage Pattern
- Logical Application Design Document or Logical Application Design Model
- Logical Application Deployment Model

Documents dependent on the Physical Application Design Document are:

- Physical Deployment Model

Artefact Type:	**Physical Application Design Document - Guidance for Acquired Solution**
Guidance:	The Physical Application Design Document provides an accurate, detailed, and complete description of the physical design for the entire solution including the Acquired Solution components. It describes how to utilize the identified implementation

	environment on a selected platform to physically implement the Acquired Solution components and sub-systems design, integration design, and interface specifications.
Required:	Physical Application Design Goals and Considerations Physical Application Design Overview Physical Application Design – Diagram and Description Component Model Custom Application Design (if required) Integration and Interface Design Security Design Requirement Traceability (to functional requirements).
Not Required:	Components and sub-systems design Detailed subsystem design and module design

Row: 4	**HOW**	**Column 2**
Artefact Type:	**Physical Application Design Document (continued)**	
Version:	Object-Oriented	
	Physical Application Design Model	
What/Purpose:	The detailed Physical Application Design Model is an abstraction of the implementation model and it can be used to generate source code. It elaborates the Logical Application Design Model. In addition, it incorporates development environment classes that are specific to technologies and languages used to implement the model.	
Description:	The detailed Physical Application Design Model is a platform-specific model. Create the Physical Application Design Model by using a UML Modeling Tool and make the language-specific packages visible. In the event these packages were not added when the model was created, add them by using modeling tool-specific instructions. When a service-based approach (i.e., SOA) is being used to assemble an application, the SOA Application Service Model template must be used to build upon the logical design and articulate the physical architectural design for *each* discrete "Service" being proposed or developed. See SOA Application Service Model artefact template. See SOA Application Service Model example.	

The Physical Application Design Document is a **generic design** artefact that can be used when UML is **not used** as a modeling notation. More details regarding this artefact can be found in the OPS Application Architecture Guidebook.

Rank: Mandatory (either generic or UML version)

Benefit: The detailed Physical Application Design Model provides a clear understanding of the physical structure of an application.

Risk: Failure to create this artefact can result in:

- Confusion in implementing the design;
- Misuse of the development language packages or third party packages; and
- Project delays and additional costs.

Artefact Dependencies: Pre-requisite artefacts are:

- Logical Application Design Document or Logical Application Design Model
- Logical Application Deployment Model

Artefacts dependent on this artefact are:

- Physical Deployment Model

Recommended Practice: The Physical Application Design Document includes (diagrams should be supported by descriptions):

- Static Model
- Package diagram
- Class Diagrams
- State-Transition Diagrams
- Component Diagrams
- Implementation Diagrams
- Dynamic Model (Use-Case Realization)
- Interaction diagrams: Sequence and Collaboration diagrams
- Statechart / Activity diagrams
- Trace (class) diagram to use-cases

Diagrams Model	UC	O	C	S /C	S	A	Com	D
Design		m	M	M	M	m		

UC=Use-Case; O=Object; C=Class; S/C = Sequence/Collaboration
S=Statechart; A= Activity; Com=Component; D=Deployment; M=Major; m=minor

Row: 4	HOW	Column 2

Artefact Type: **Application Inventory**

What/Purpose: This is a systematic capturing of information and its related attributes or information. It can be used for a great variety of applications, with varying levels of detail, and for different types of audiences.

Description: A grouping of automated functions into a package called an application or application portfolio helps people understand the purpose of the application, who is responsible for it, what part of the business it supports, etc. In a more formal analysis, application portfolios are defined in packages aligned with technology requirements. As a result of grouping analysis, different server technologies, locations of applications, application distribution strategies, etc. may be chosen for different packages or groups of automated function.

The following criteria is often used to partition automated functions in an application architecture:

- Common middleware components—industry guidelines used to define packaging of many automated functions such as message handlers
- Characterization of business process behaviours—grouping automation functions that support similar business behaviours such as event driven vs. collaborative vs. information retrieval

Functional characteristics of similar processes—grouping functions that support similar business tasks such as planning, research, dispatch, and enrolment.

See the Application Inventory template.
See the Application Inventory example.

Rank: Mandatory

Benefit: This listing provides information about the application environment.

Risk: The ability to support applications and the ability to outsource them may be severely restricted.

Artefact Dependencies:

Pre-requisite artefacts are:

- Physical Application Design Document or Physical Application Design Model
- Physical Deployment Model

Artefacts dependent on this artefact are:

- None

Recommended Practice:

The Application Inventory includes:

- Introduction
- Purpose
- Application Inventory List

Row: 4	**HOW**	**Column 2**

Artefact Type: **Application Implementation Document**
Note: This artefact is defined with two versions, a generic version and an object-oriented version, the Application Implementation Model.

Version: Generic

What/Purpose: The Application Implementation Document provides detailed information regarding the physical realization (files) of the logical design elements (classes, components).

Description: The Application Implementation Document provides all the information needed to construct a system and to put it into operation. It captures the physical realization (files) of the logical components (packages). These files are both deliverable components, such as executables, and components from which the deliverables are produced, such as source code files. The Application Implementation Document also includes information about the physical realization of the language specific and third party components used in the application.

See the Application Implementation Document template.
The **UML version** of this artefact is the *Implementation Model.*

Rank: Mandatory (either Generic or UML version)

Benefit: Ensure physical implementation elements match to high-level logical application design elements.

Risk: Failure to create this artefact can result in:

- Lack of synchronization during Elaboration and Construction phases, between the logical design and physical implementation.
- Difficulties during testing, debugging and bugs fixing.
- Difficulties in maintaining and enhancing the product
- Project delays and additional costs.
- Failure to optimize the application distribution.
- Performance issues.
- Ineffective communication between various components.

Artefact Dependencies: Pre-requisite artefacts are:

- Physical Application Design Document or Physical Application Design Model
- Physical Deployment Model

Artefacts dependent on this artefact are:

- None

Recommended Practice: The Application Implementation Document includes:

- Introduction
- Implementation Overview
- Implementation Design
- Subsystem Overview
- Detailed Subsystem Design

Row: 4	**HOW**	**Column 2**
Artefact Type:	**Application Implementation Document (continued)**	
Version:	**Object-Oriented**	
	Application Implementation Model	
What/Purpose:	The Application Implementation Model captures the physical realization (files) of the logical components (packages) and elements presented in the Logical Application Design Model.	
Description:	The Application Implementation Model maps the high-level logical application design elements to their physical implementation. It also includes information about the physical realization of the language specific and third party components used in the application. These files are both deliverable components, such as executables, and components from which the deliverables are produced, such as source code files.	

The Implementation Model provides all the information needed to construct a system and to put it into operation. It recommends the order in which these components should be implemented, tested and assembled.

The *Application Implementation Document* is a **generic design** artefact that can be used when UML is **not employed** as a modeling notation.

Rank: Mandatory (either Generic or UML version)

Benefit:
- Provides a clear understanding of the physical structure of an application.
- Ensure physical implementation elements match to high-level logical design elements.
- Ensure proper component integration.

Risk: The failure of creating this model can result in:
- Lack of synchronization between the logical design and physical implementation;
- Difficulties during testing, debugging and bugs fixing;
- Difficulties in maintaining and enhancing the product; and
- Might result in project delays.

Artefact Dependencies: Pre-requisite artefacts are:
- Physical Application Design Document or Physical Application Design Model
- Physical Deployment Model

Artefacts dependent on this artefact are:
- None

Recommended Practice: The Application Implementation Model includes:
- Logical components.
- Implementation subsystems and correspondent physical components. The subsystems are implemented as directories, which include the components (files).
- Both deliverable components, such as executables, and components from which the deliverables are produced, such as source code files.
- Mapping of logical components to physical components.
- Third party physical components (file names).
- Programming language specific libraries (files).

Diagrams	UC	O	C	S /C	S	A	Com	D

Models								
Implement ation				m			M	

UC=Use-Case; O=Object; C=Class; S/C = Sequence/Collaboration
S=Statechart; A= Activity; Com=Component; D=Deployment; M=Major; m=minor

Row: 4 WHEN Column 3

Name: **Physical Deployment Model**

What/Purpose: The Physical Deployment Model depicts the physical technology infrastructure implementation at a sufficient level of detail so that:

- All stakeholders (business, architects, application developers, technology infrastructure developers, IT operations, security/privacy, etc.) can verify that their concerns have been taken into account by the system technology infrastructure implementation.
- Transformation issues from design to implementation can be addressed.

Description: Technology constrained, detailed, annotated physical technology infrastructure diagrams (including applicable Infrastructure Usage Patterns), system software, hardware and network components that address high/medium level annotated diagrams providing a listing and placement of logical infrastructure component or service types necessary for the system technology infrastructure.

The placement of infrastructure components should satisfy both system and non-system functional requirements, and show system technology constraints such as:

- Technology standards e.g., SMTP, JEE, .NET, MAPI, LDAP, MPLS, TCPIP, X.509, SNMP etc.
- Product choices e.g., Servers-"R"-Us Application Server 4.1, Tachyon Systems LANBlinder switch, Big Desktop Co. Word Processor Suite 5.5, HUGENet DSL Service, Acme Access Control, Ace SNMP Server Monitor 1.5, etc.
- Provisioning and sizing information e.g., 100 Mbps VLAN, 3 clustered Application Servers, 10 GB RAM + 500 GB RAID Level 5 disk, 100 UTP ports, etc.

- Quality Level Metric related details e.g., 750 ms server response time, 920 tps per application server, 10000 hours MTBF, 99.99% uptime, etc.
- Infrastructure numbering and naming schemes e.g., server naming scheme, LAN subnet IP address ranges, etc.
- Security information e.g., 128 bit SSL, SHA-256 hashing, access list configuration scheme, authentication token types, 4096 bit PKI key, etc.
- Location details e.g., contingency site: Disaster Co, 13th Floor, 4444 Volcano Road, Antarctica Printer 23456, Floor 8, Ferguson Block, etc.

If GO-IT Standard exemption is required projects must provide answers to the following:

- What GO-IT Standard exemptions have the project received?
- When were the exemptions received?
- What were the reasons for the exemptions?

See Physical Deployment Model template.
See template for Quality Level Metrics section.
See the Disaster Recovery View template.

Note: A usage pattern based approach is being encouraged and projects must clearly articulate and demonstrate the use of usage patterns in the physical deployment model. The design must clearly articulate how patterns are being used (including usage patterns) and used in the project solution.

Specifically, illustrate the Infrastructure Usage Patterns in the Physical Deployment Model by:

- Clearly stating the Infrastructure Usage Pattern(s) being used including the Infrastructure Usage Pattern name, version # and date. There is no need to copy the entire pattern document in the Physical Deployment Model;
- Incorporate the appropriate Infrastructure Usage Pattern physical diagrams into the Physical Deployment Model – Master View diagram. Label the reused Infrastructure Usage Pattern and shade or colour code it to differentiate the pattern diagram from the project-specific deployment elements;
- Illustrate the project-specific deployment elements (i.e., infrastructure components);

- Complete the remaining infrastructure design requirements as required for the Physical Deployment Model.

See Infrastructure Usage Pattern template.

Rank: Mandatory

Benefit:
- Enables the reuse of the Infrastructure Usage Pattern(s) from a deployment perspective
- Enables the appropriate design and selection of hardware and software components prior to detailed solutioning (including ITS blueprint).
- Provides traceability to non-functional requirements.

Risk: Failure to create the Physical Deployment Model can result in:
- Inappropriate system performance
- Lack of integration with other applications
- Security risk
- System instability
- Deployment failure
- Project delays

Dependencies: Pre-requisite documents are:
- Solution Pattern Match
- Application Usage Pattern
- Infrastructure Usage Pattern
- Logical Application Deployment Model
- Physical Application Design Document
- IT Solution Concept

Documents dependent on the Physical Deployment Model are:
- Detailed Design / Solution Blueprint

Row: 3	**WHO**	**Column 4**

Artefact Type: **User Interface Design**

What/Purpose: Enables the development of user interfaces that meet business requirements and are consistent, intuitive, and easy to use User interfaces can take many forms, but always accomplish two fundamental tasks: communicating information from the machine to the user, and communicating information from the user to the machine.

Description: The user interface (UI) includes display screens, data entry

screens, reports, messages -- and how an application invites interaction with the user.

The total "user experience" designed into the interface includes aesthetics, response time, and content.

The best way to ensure quality user interface designs is to use an orderly and well-defined design process that is specifically geared to producing quality results.

See the User Interface Design example.

Rank: Optional

Benefit:
- Helps the business user understand interaction with the system
- Facilitates the evaluation of the extent to which the system meets business needs
- Involves stakeholders in the design process and reduces project risk
- Provides an early means to assure compliance with OPS standards and legislative requirements (e.g., Accessibility for Ontarians with Disability Act).

Risk: Failure to satisfy user requirements.

Artefact Dependencies: Pre-requisite artefacts are:
- System Functional Requirements or Use-Case Specification

Artefacts dependent on this artefact are:
- None

Row: 4	**WHEN**	**Column 5**

Name: **Operating Schedule**

What/Purpose: The Operating Schedule documents appropriate planning for the ongoing operations of the solution. This document can be used as a source for the analysis of the timing and resource requirements of the solution over time.

It is used to outline the underpinning technology architecture of the solution for the purpose of business and IT planning, service level management, procurement, implementation, operations & administration, and auditing & control.

Description: Describes the sequencing, timing, duration and other time-related characteristics of events and processes/activities/tasks. It should also indicate relationships and dependencies between processes.

An operational sequence diagram(s) and table(s) are used to illustrate and describe process behaviour.

See Operating Schedule template.

Rank: Mandatory

Benefit:
- Aids in the clear understanding of the sequencing of events and activities, resource requirements, and technology requirements.
- Enables initial thought process in assessing responses and reactions to operational issues
- Establishes the guidance to minimize application downtime or interruptions

Risk: Insufficient understanding of the order of events, insufficient understanding of resource requirements over time, and lack of understanding the technology requirements of the proposed solution.

Artefact Dependencies: Pre-requisite documents are:
- System Functional Requirements
- Physical Application Design Document or Logical Application Design Model
- Physical Deployment Model

Revision History

Revision	Revision Date YYYY-MM-DD	Effective Date YYYY-MM-DD	Row & Column	Type	Remarks
1.7	2012-07		Various	Edit	Improved instructions in the following templates: • Application-Inventory.dot • Application-Usage-Pattern.dot • Disaster-Recovery-View.dot • Infrastructure-Usage-Pattern.dot • Logical-Application-Deployment-Model.dot • Logical-Application-Design-Document.dot • Logical-Operating-Schedule-and-States.dot • Physical-Application-Design-Document.dot • Physical-Deployment-Model.dot • Quality-Level-Metrics.dot • SOA-Application-Service-Model.dot • Solution-Pattern-Match.dot • Supplementary-Specification.dot • System-Functional-Requirements.dot
			Various	Edit	Enhanced security requirements displayed in the following examples: • logical-deployment-component-diagram-model-example.doc • logical-deployment-location-diagram-model-example.doc • physical-deployment-model-example.doc
1.6	2011-04		(3, 4)	Edit	Removed the Functional Group – Application Component Cross-Reference artefact from the requirements.
			(3, 2)	Edit	Updated the Logical Application Design Document with application security requirements.
			(3, 3)	Edit	Updated the Logical Application Deployment Model to include Infrastructure Component Placement Diagram content, remove Operational Views, and clarify logical security requirements.

Revision	Revision Date YYYY-MM-DD	Effective Date YYYY-MM-DD	Row & Column	Type	Remarks
			(3, 3)	Edit	Removed the Infrastructure Component Placement Diagram as a separate artefact in the requirements. Merged with Logical Application Deployment Model.
			(3, 3)	Edit	Replaced the Infrastructure Pattern Match with the Solution Pattern Match.
			(3, 5)	Edit	Removed the Logical Operating Schedule & State artefact from the requirements.
			(4, 2)	Edit	Updated guidance in the Physical Application Design Document and Model to assist projects to further demonstrate the use of the Application and Infrastructure Usage Patterns in the physical solution design.
			(4, 3)	Edit	Updated guidance in the Physical Deployment Model to assist projects to match and illustrate the use of the Infrastructure Usage Pattern.
			(4, 5)	Edit	Renamed the Calendarized Schedule to Operating Schedule. Incorporated content from the Logical Operating Schedule and Calenderized Schedule.
			(3, 6)	Edit	Updated the Supplementary Specifications with security requirements.
			(3, 3); (4, 3)	Edit	Updated the Quality Level Metrics template with a streamlined set of metrics.
			(4, 2)	Edit	Physical Application Design Document updated with application security requirements.
			(4, 3)	Edit	Updated the Physical Deployment Model to clarify physical security requirements.
			(1,1); (1,2); (2,1)	Edit	Resource Type, Program, and Information Model: updated references to renamed artefacts and reference models.
			(2,2)	Edit	Replaced "SOA Service Description Profile" artefact with updated artefact named "SOA Service Specification"

Revision	Revision Date YYYY-MM-DD	Effective Date YYYY-MM-DD	Row & Column	Type	Remarks
			(3,2); (4,2)	Edit	Renamed "Service Model" to "SOA Application Service Model" in the Logical & Physical Application Design Documents and Models.
			(2,2)	Edit	Renamed "SOA Service Profile" to "SOA Service Specification." Updated the description and guidance. Provided an end-to-end example to demonstrate traceability.
			(3,2) & (4,2)	Edit	Updated guidance in the SOA Application Service Model template and included an end-to-end example to demonstrate traceability.
			(2,6)	Edit	Performance Matrix removed from the requirements.
			(2,2)	Edit	Service Profile, Performance Measure Section: Simplify instructions. Redefine definitions of effectiveness, efficiency and customer satisfaction to align with the *Performance Measurement Reference Guide* published by Ministry of Finance. Process Checklist Section: section removed.
			(1,6)	Edit	Strategy artefact removed from the requirements.
			(2,5)	Edit	State Transition Diagram updated with improved description and examples.
1.5	2010-05	2010-06-28	All	New	Provided artefact dependency statements for all artefacts.
			All	Edit	Improved consistency of formatting.
			n/a	Edit	Reversed chronology of the revision history.
			(2, 2)	Edit	Modified the Service Profile artefact description. Updated both the template and example.
			(3, 4)	Edit	Updated the description, guidance and template for the Functional Group – Application Component Cross-Reference.
			(3, 4)	Edit	Removed the Detailed Workflow Specification artefact from the requirements.
			(3, 4)	Edit	Updated the description of the User Interface Design.

Revision	Revision Date YYYY-MM-DD	Effective Date YYYY-MM-DD	Row & Column	Type	Remarks
			(3, 5)	Edit	Renamed the Logical Operating Schedule to "Logical Operating Schedule and States." Update the description.
			(4, 3)	Edit	Updated the description of the Physical Deployment Model to capture GO-ITS compliance status.
			(4, 5)	Edit	Renamed the Calendarized Schedule and States to "Calendarized Schedule." Updated the description.
			(2, 1)	New	Specified, for Acquired Solutions, a variant of the Conceptual Data Model, and an Interface Data Requirements Document. Provided an example for each artefact.
			(3, 1)	Edit	Enhanced the guidance for the Logical Data Model (Acquired Solution) and provided an example.
			(3, 2)	Edit	Enhanced the guidance for the System Functional Requirements for Acquired Solutions.
			(3, 2)	Edit	Enhanced the guidance for the Logical Application Design Document for Acquired Solutions.
			(3, 3)	New	Provided guidance for the Logical Application Deployment Model for Acquired Solutions.
			(3, 6)	Edit	Enhanced the guidance for the Supplementary Specification for Acquired Solutions.
			(4, 1)	New	Specified, for Acquired Solutions, a variant of the Physical Data Model, and provided two examples.