



Government of Ontario IT Standard (GO-ITS)

**Application Development Standard: Standards for SDLC  
(Software Development Life Cycle)**

Document Number: **54**

Version Number: **1.0**

Status: **Approved**

Prepared for the Information Technology Standards Council (ITSC) under the delegated authority of the Management Board of Cabinet

## Foreword

Government of Ontario Information Technology Standards (GO-ITS) are the official publications on the guidelines, preferred practices, standards and technical reports adopted by the Information Technology Standards Council (ITSC) under delegated authority of the Management Board of Cabinet (MBC). These publications support the responsibilities of the Management Board Secretariat (MBS) for coordinating standardization of Information & Information Technology (I&IT) in the Government of Ontario. Publications that set new or revised standards provide enterprise architecture guidance, policy guidance and administrative information for their implementation. In particular, GO-ITS describe where the application of a standard is mandatory and specify any qualifications governing the implementation of standards.

# Table of Contents

<b>1 INTRODUCTION.....</b>	<b>3</b>
1.1 Background .....	3
1.2 Purpose .....	4
1.3 Scope .....	4
1.4 Applicability statements .....	5
1.5 Requirement Levels.....	5
1.6 Recommended Versioning and/or Change Management .....	6
1.7 Publication Details .....	6
1.8 Contact Information .....	6
1.9 Acknowledgements .....	7
1.9.2 Reviewers.....	8
<b>2 APPLICATION DEVELOPMENT IN THE OPS.....</b>	<b>9</b>
2.1 WATERFALL SDLC .....	9
2.2 ITERATIVE SDLC .....	10
<b>3 APPLICATION DEVELOPMENT STANDARD .....</b>	<b>11</b>
3.1 Application Development Standards .....	11
<i>GENERAL STANDARDS</i> .....	12
<i>REQUIREMENT STANDARDS</i> .....	13
<i>ANALYSIS STANDARD</i> .....	14
<i>DESIGN STANDARD</i> .....	16
<i>CODING STANDARD</i> .....	18
<i>TESTING STANDARD</i> .....	20
<i>IMPLEMENTATION STANDARD</i> .....	22
<i>POST-IMPLEMENTATION STANDARD</i> .....	23
<b>4 DOCUMENT HISTORY .....</b>	<b>25</b>
<b>5 REFERENCES.....</b>	<b>25</b>

# 1 Introduction

## 1.1 Background

Application development refers to a software development process used by an application developer to build application systems. This process is commonly known as the Software Development Lifecycle (SDLC) methodology and encompasses all activities to develop an application system and put it into production, including requirements gathering, analysis, design, construction, implementation, and maintenance stages. Examples of the SDLC methodology include e.g., waterfall, iterative, rapid, spiral, RAD, Xtreme and many more.

A SDLC is a well-defined, disciplined, and standard approach used in developing applications which provides:

- a methodical approach to solving business and information technology problems
- a means of managing, directing, monitoring and controlling the process of application/software building, including:
  - a description of the process - steps to be followed
  - deliverables - reports/programs/documentation/etc

Benefits of using a SDLC methodology include:

- Has a proven framework
  - Consistency and uniformity - methods and functions
  - Results/Deliverables
- Facilitates information exchange
- Defines and focuses on roles and responsibilities
- Has a predefined level of precision to facilitate a complete, correct and predictable solution.
- Enforces planning and control

The intent of this GO-ITS 54 document is to describe the standards, which apply when developing applications in the Ontario Public Services (OPS). This document is not intended to be an 'all-inclusive' methodology to application development or software development lifecycle but rather will focus on and outline specific standards that must be followed when building applications. As with any standards document, the Application Development Standards (ADS) document will evolve over time, largely based on contributions from development teams. The ADS document is directed at application developers who will be designing, developing, and maintaining applications for their Clusters/Ministries. This includes external contractors, consultants, and business partners, as well as Ontario Government employees.

This GO-ITS 54 standard was developed in consultation with various stakeholder groups and was sponsored by the Information Technology Standards Council (ITSC) and the Corporate Architecture Branch (CAB) within the Office of the Corporate Chief Technology Officer (OCCTO).

## 1.2 Purpose

This document interprets current industry standards and recommends an application development standard for adoption in the Ontario Public Sector (OPS) for the software/application development lifecycle, consistent with OPS enterprise architecture standards (in particular, compliance with the enterprise architecture checklist), principles, and best practices.

The application development standard will provide:

- Adequate Application Development Standards for all stages of the application development process
- Minimum requirements for application development activities, deliverables and acceptance sign-off
- A general measure for ensuring the application development methodology is in compliance with the application development standard.

## 1.3 Scope

### 1.3.1 In Scope

The application development standard will highlight key characteristics of a software development lifecycle methodology and provide guidance for a generic:

- Waterfall development; and
- Iterative development.

Where applicable, adoption of industry standards methodologies will be recommended and referenced.

### 1.3.2 Out of Scope

The development or selection of a full SDLC methodology or other processes is out of scope:

- Rational Unified Process, Enterprise Unified Process
- Agile Methodology
- Xtreme Programming
- OPS Gating Process
- Enterprise Architecture Process and Methods
- Project Management Processes

## Standards not covered

*Though a large number of detailed standards exist, there are still areas not covered by this standard. One reason for this is to remain sufficiently generic so that each cluster can use new techniques and tools, while still maintaining conformance.*

### Programming Standards

Clusters are expected to maintain standards for the development of the application/software source code. Their purpose is to increase application/software quality, by proper commenting,

limiting module complexity, systematic naming conventions, and other techniques. Such standards are often dependent on the choice of programming language.

## Design Standards

Clusters will also benefit from design standards. These can help ensure that consistent techniques are used, e.g. in conjunction with object-oriented design methods. Guiding principles, such as encapsulation and information hiding, may be defined, and checklists may be developed for use in the design reviews.

## 1.4 Applicability statements

Government of Ontario IT Standards and Enterprise Products apply (are mandatory) for use by all ministries/clusters and to all former Schedule I and IV provincial government agencies under their present classification (Advisory, Regulatory, Adjudicative, Operational Service, Operational Enterprise, Trust or Crown Foundation) according to the current agency classification system.

Kindly refer to [http://intra.pmed.mbs.gov.on.ca/mbc/pdf/Agency\\_Establishment&Accountability-Dir.pdf](http://intra.pmed.mbs.gov.on.ca/mbc/pdf/Agency_Establishment&Accountability-Dir.pdf) for a list of provincial government agencies with their classification under the current classification system, as well as their previous Schedule under the former Schedule system.

Additionally, this applies to any other new or existing agencies designated by Management Board of Cabinet as being subject to such publications, i.e. the GO-ITS publications and enterprise products - and particularly applies to Advisory, Regulatory, and Adjudicative Agencies (see also procurement link, OPS paragraph). Further included is any agency which, under the terms of its Memorandum of Understanding with its responsible Minister, is required to satisfy the mandatory requirements set out in any of the Management Board of Cabinet Directives (*cf.* Operational Service, Operational Enterprise, Trust, or Crown Foundation Agencies).

As new GO-IT standards are approved, they are deemed mandatory on a go-forward basis (Go-forward basis means at the next available project development or procurement opportunity).

When implementing or adopting any Government of Ontario IT standards or IT standards updates, ministries and I&IT Cluster must follow their organization's pre-approved policies and practices for ensuring that adequate change control, change management and risk mitigation mechanisms are in place and employed.

*For the purposes of this document, any reference to ministries or the Government includes applicable agencies.*

## 1.5 Requirement Levels

Within this document, certain wording conventions are followed. There are precise requirements and obligations associated with the following terms:

<b>Must</b>	This requirement is not optional
<b>May</b>	The implementer <i>may</i> choose to take one or more of a selection of options, but <i>must</i> make a choice of one or more, as dictated within the context of the item
<b>Should</b>	The implementer <i>must</i> choose this action, <i>unless</i> business functionality dictates otherwise. Exceptions <i>must</i> be approved by management, as modifications to the standard practice

## 1.6 Recommended Versioning and/or Change Management

GO-ITS 54 Application Development is an OPS mandatory standard. Modifications during the life of the standard must be approved by the organizational owner of the document.

The organizational owner of GO-ITS 54 is the Head of Corporate Architecture Branch, Office of the Corporate Chief Technology Officer, Ministry of Government Services

Ministry of Government Services will submit revised documentation to the Information Technology Standards Council (ITSC) for endorsement, Architecture Review Board (ARB) approval and publication.

## 1.7 Publication Details

Check One	Publish on the Internal or External web site?
<input checked="" type="checkbox"/>	ITSC Web Site at <a href="http://intra.occto.mbs.gov.on.ca/occtoservices/goits_standards">http://intra.occto.mbs.gov.on.ca/occtoservices/goits_standards</a> (Available to the OPS)
<input checked="" type="checkbox"/>	GO-ITS Web Site at <a href="http://www.gov.on.ca/MGS/en/IAandIT/STEL02_047295.html">http://www.gov.on.ca/MGS/en/IAandIT/STEL02_047295.html</a> (Available to the public)

## 1.8 Contact Information

	ADMINISTRATIVE CONTACT	TECHNICAL CONTACT
<b>Full Name</b>	Doretta Ojeda	Joe Fernandes
<b>Job Title</b>	Standards Program Coordinator	Enterprise Application Architect
<b>Organization</b>	Ministry of Government Services	Ministry of Government Services
<b>Division</b>	Office of the Corporate Chief Technology Officer (OCCTO)	Office of the Corporate Chief Technology Officer (OCCTO)
<b>Branch</b>	Technology Adoption Branch (TAB)	Corporate Architecture Branch (CAB)
<b>Section</b>	Technical Standards	Application, Technology and Security
<b>Office Phone:</b>	416-327-2094	416-326-8939
<b>E-mail Address:</b>	Doretta.Ojeda@ontario.ca	Joe.Fernandes@ontario.ca

## 1.9 Acknowledgements

### 1.9.1 Editors – Development Team

Full Name	Cluster/Area
Joe Fernandes	OCCTO – CAB
Doug Croker	OCCTO-CAB
Peter Churchard	OCCTO – CAB
Susan Hess	OCCTO-TAB
Fred Pitt	HSC
Stavros Platis	ETC
Theresa Smith	HSC



**1.9.2 Reviewers**

<b>Check</b>	<b>Area</b>	<b>Date (month/year)</b>
<b>Architecture</b>		
<input checked="" type="checkbox"/>	OCCTO, Corp. Architecture Branch	Dec 4, 2006
<input type="checkbox"/>	Cluster ACT	Not required
<input checked="" type="checkbox"/>	Corporate ACT	Dec 14, 2006
<b>Domains</b>		
<input checked="" type="checkbox"/>	BADWG	Nov 21, 2006
<input checked="" type="checkbox"/>	IADWG	Nov 21, 2006
<input checked="" type="checkbox"/>	AADWG	Nov 14, 2006
<input checked="" type="checkbox"/>	SADWG	Nov 24, 2006
<input checked="" type="checkbox"/>	TADWG	Dec 11, 2006
<b>Infrastructure</b>		
<input type="checkbox"/>	OCCSD, iSERV	Not required
<b>Security</b>		
<input type="checkbox"/>	OCCIO, CSB	Not required
<b>Standards</b>		
<input checked="" type="checkbox"/>	Technical Standards Unit	Dec 01, 2006
<input checked="" type="checkbox"/>	ITSC	Feb 21, 2007
<b>Strategy</b>		
<input type="checkbox"/>	OCCS, SPPM	Not required
<b>Others</b>		
<input checked="" type="checkbox"/>	Solutions Management Forum	Nov 30, 2006
<input checked="" type="checkbox"/>	Project Management – COE	Jan 12, 2007
<input checked="" type="checkbox"/>	Audit	Jan 22, 2007
<input checked="" type="checkbox"/>	Controllership Office	Feb 9, 2007

## 2 Application Development in the OPS

While various application development methodologies have been developed to guide the application development processes, the key application development methodologies used within the OPS are **Waterfall** and **Iterative**. Generally, the critical objectives, activities and deliverables of each of these methodologies remain the same. OCCTO has undertaken the task to identify these various SDLC, and develop a GO-ITS which specifies the application development standard that is applicable across various methodologies. I&IT clusters will use this standard to help guide project teams to develop applications in a consistent, standard and predictable manner.

Effective application development processes are critical to the success of IT projects. Clusters must select and follow one of the many applications development processes that can be categorized as Waterfall or Iterative; however, this Application Development Standard (GO-ITS 54) must be used within the OPS to achieve compliance. This standard clearly defines expected application development activities, measures and deliverables for each phase to help in ensuring that the necessary standards are maintained through the entire life of the project.

**PROJECTS MUST SELECT ONE APPLICATION DEVELOPMENT METHODOLOGY AND USE IT FOR THE DURATION OF THE ENTIRE PROJECT.**

### 2.1 WATERFALL SDLC

The waterfall model is a popular version of the software development life cycle model for software engineering. Often considered the classic approach to the application/software development life cycle, the waterfall model describes a linear and sequential development method with distinct goals for each phase of development.

The seven waterfall phases are:

1. Requirement Gathering – Collecting the business requirements/needs
2. Analysis – Business and Requirement Analysis
3. Design – Architecture and application design
4. Coding - Development/Construction/Programming
5. Testing – Bug fixes, error corrections, quality assurance
6. Implementation – Deploying the application into the production environment
7. Post Implementation – maintenance and review

## 2.2 ITERATIVE SDLC

ITERATIVE AND INCREMENTAL DEVELOPMENT is an application/software development process developed in response to the weaknesses of the more traditional waterfall model.

The iterative process starts with architecturally significant subset of the application/software requirements (often the high risk requirements) and iteratively enhances the evolving sequence of versions until the full application/software is implemented. At each iteration, design modifications are made and new functional capabilities are added. This allows the project team to take advantage of what was being learned during the development of earlier, incremental, deliverable versions of the application/software. The product is defined as completed when it satisfies all of its requirements.

This iterative process uses the elements of the waterfall model in the four iterative phases, which are:

1. Inception – Gathering of business requirements/needs
2. Elaboration – Business/Requirement Analysis and Architecture and application design
3. Construction - Development/Construction/Programming/testing
4. Transition – Implementation of the application

Within the four iterative phase inferences can be made to map to the seven Waterfall phases. Regardless of what SDLC selected, any development initiative will need to go through the activities related to the seven Waterfall phases.

### 3 APPLICATION DEVELOPMENT STANDARD

#### 3.1 Application Development Standards

The application development standards listed are derived and aligned to those listed in the IEEE/EIA 12207.0, Industry Implementation of International Standard ISO/IEC12207:1995 — (ISO/IEC 12207) Standard for Information Technology —Software life cycle processes.

#### GLOSSARY OF TERMS

Term	Description
<b>Application</b>	Computer programs, procedures, rules, and associated documentation and data pertaining to the operation of a computer system.
<b>Application or Software Development Lifecycle</b>	A systematic approach to the creation of software or application. This cycle typically includes a requirements, analysis, design, coding, test, implementation and post-implementation phases.
<b>Audit or Review (Peer Reviews)</b>	An independent review for the purpose of assessing compliance with software requirements, specifications, baselines, standards, procedures, instructions, codes, and other requirements.
<b>Baseline</b>	A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures.
<b>Contract Application Programmer</b>	A person or firm who contracts with the OPS to work on an application development product.
<b>Evaluation</b>	A technique in which requirements, design, code and test results are examined in detail by a person or group to detect problems. The results are documented.
<b>Maintenance</b>	To repair, change, or add to a software product.
<b>Product</b>	A product is the tangible result of any process or work group. This includes, but is not limited to, shrink-wrapped or other software products, components, code, services, and deliverables.
<b>Project Team</b>	A group of people representing different disciplines who collaborate and work together to deliver an application development product.
<b>Sign-off</b>	The term used to describe a point in the application development process where an individual/governance body officially approves and accepts the product.
<b>Walk-through</b>	A review process in which an individual(s) lead their peers through their work product.

## GENERAL STANDARDS

---

- A.1. The I&IT Cluster must issue a written statement establishing and/or selecting a waterfall or iterative software development life cycle methodology (SDLC) as a means for structuring and controlling the process of developing an application software.

**Projects must select one application development methodology and use it for the duration of the entire project.**

If a project needs to change the selected application development methodology, a change request must be issued in accordance to a Change Management Process.

- A.2. The application development project must operate in compliance with Enterprise Architecture Standards, Principles, Processes, and Methods.
- A.3. This standard applies to all application/software development projects including maintenance projects.
- A.4. The project team must:

Document the outputs in accordance with the established Documentation Process (EAPM (Enterprise Architecture Processes and Methods handbook), Artifacts, SDLC).

Place the outputs under the Configuration Management Process and perform change control in accordance with established IT processes. (This statement applies when a Corporate or Cluster Configuration Management Process is available).

Document and resolve problems and non-conformances found in the application/software products and tasks in accordance with established Problem Resolution Process.

- A.5. This standard applies to 'commercial off the shelf' (COTS) software package acquisition and integration. If COTS is being used, the project team must determine that the product satisfies the needs of a particular application development or modification project. The commercial software packages must be compatible with existing OPS IT standards, policies, and guidelines. Software product acquisition procedures must follow the OPS procurement policies, and these products must be reviewed, assessed and tested and reviewed prior to being used. The end-to-end solution must be thoroughly tested as well.
- A.6. Request for Services (RFS) for Contract Application Programming - The project must provide that the procurement of contract programming services be justified with a written request for service(s). The project team must ensure that the contract programmer must adhere to the GO-ITS 54 Standard. The end products of completed contract programming services must be reviewed, tested and approved.
- A.7. The project team must support audit(s) and reviews. The results of the audits/reviews must be documented. Upon successful completion of the audits, if conducted, The project team should:
- a) Update and prepare the appropriate deliverable or changes requested or recommended by the audit
  - b) Obtain sign-off of the responses to the audit.

Throughout the entire project, special consideration must be given, on an ongoing basis to capturing and implementing security and privacy requirements. The post-implementation review must reflect this.

## REQUIREMENT STANDARDS

---

### 1 Requirement Standard

#### 1.1 The project team must gather business and system requirements

Activities include: Review existing systems/process, Describe data/system/process, Identify problem, areas/opportunities, Identify user needs/wants, Conduct interviews, Develop Solution, Identify manual and automated processes, Draw conceptual flow, Identify follow on projects/phases, Identify inputs (functional description), Data entry screens, Inputs from outside sources...

#### 1.2 The project team must establish and document business/application requirements

In order to achieve the above the following items should be followed.

- Document requirements
- Document assumptions
- Document outstanding issues
- Estimate data storage requirements
- Identify legislative/contractual/security/privacy/access requirements
- Document reporting requirements
- Review training requirements
- Conduct initial walkthrough
- Obtain sign-off and approval

#### 1.3 The project team must produce the following minimum set of documentation as part of this phase:

**Enterprise Architecture Documentation:** refer to the Enterprise Architecture Checklist for Change Initiative as the authoritative source for preparing the following Enterprise Architecture artifacts.

- |                                       |   |
|---------------------------------------|---|
| ▪ Resource Type                       | ▪ Conceptual Data Model                             |
| ▪ Program                             | ▪ Service Integration and Accountability Model      |
| ▪ Service                             | ▪ Business Rule Statement / Process Cross Reference |
| ▪ Location Type                       | ▪ Business Process Model                            |
| ▪ Party Type                          | ▪ Business Network Model                            |
| ▪ Role Type                           | ▪ Workflow Model                                    |
| ▪ Target Group Type                   | ▪ Business Scenarios                                |
| ▪ Event Type                          | ▪ Service Objectives                                |
| ▪ Cycle Type                          | ▪ Business Rules Statement / Source Cross Reference |
| ▪ Goals                               | ▪ Business Rule Statement                           |
| ▪ Need Type                           | ▪ Business Rule Source                              |
| ▪ Mandate (Program)                   |   |
| ▪ Strategy                            |   |
| ▪ Target Group / Need Cross Reference |   |

**For Waterfall:** Supplementary document “Business Requirement Document” is required.

- TRA/PIA (As required)

**For Iterative:** Use Enterprise Architecture artifacts to further refine Use-cases, models and scenarios

- TRA/PIA (As required)

**I&IT Cluster specific documentation, as required.**

## **ANALYSIS STANDARD**

---

### **2 Analysis Standard**

#### 2.1 The project team must analyse business and system requirements

Activities include: Analyzing system flow, data model, Name and define fields in data dictionary, data normalization, physical data model, screens, screen navigation, data entry screens, inquiry screens, help screens, online documentation, Analyse reports, Forms, Report distribution system, User generated reports, Identify files, file formats, edit criteria, record volume, record/file purge criteria, Analyse existing system modifications, controls, program to program controls, backup/recovery procedure, system patterns, security, privacy, application security, equipment needed, impacts to other organizations, Access Services/LAN Management/Help Desk, Data Centre (Production System Support/Operations), analyse resource implications, Storage requirements (Tapes, Online transactions (I/O),PADS), User connections, CPU...

#### 2.2 The specific intended use of the system to be developed must be analyzed to specify system requirements. The system requirements specification should describe: functions and capabilities of the system; business, organizational and user requirements; safety, security, information, privacy, interface, operations, and maintenance requirements; design constraints and qualification requirements. The system requirements specification must be documented.

#### 2.3 The system requirements must be evaluated considering the criteria listed below. The results of evaluations must be documented.

- a) Traceability to business needs;
- b) Consistency with business needs;
- c) Testability;
- d) Feasibility of system architectural design;
- e) Feasibility of operation and maintenance.

#### **Application requirements analysis**

#### 2.4 The project team must establish and document application requirements, including the quality characteristics specifications, described below.

- a) Functional and capability specifications, including performance, physical characteristics, and environmental conditions under which the application is to perform;
- b) Interfaces external to the application module/component/service;
- c) Testing requirements;
- e) Privacy and Security specifications, including those related to compromise of sensitive information;
- f) Data definition and database requirements;
- g) Installation and acceptance requirements of the delivered application product at the operation and maintenance site(s);
- h) User documentation;
- i) User operation and execution requirements;

j) User maintenance requirements.

2.5 The project team must evaluate the application requirements considering the criteria listed below.

The results of the evaluations must be documented.

- a) Traceability to system requirements and system design;
- b) External consistency with system requirements;
- c) Internal consistency;
- d) Testability;
- e) Feasibility of application design;
- f) Feasibility of operation and maintenance.

2.6 The project team must conduct review(s).

- Appropriate Enterprise Architecture Checkpoint review
- Internal peer reviews and walkthroughs

2.7 The project team must produce the following minimum set of documentation as part of this phase:

**Enterprise Architecture Documentation:** refer to the Enterprise Architecture Checklist for Change Initiative as the authoritative source for preparing the following Enterprise Architecture artifacts.

**Enterprise Architecture Documentation:** Refer to the Enterprise Architecture Checklist for Change Initiative as the authoritative source for preparing the following Enterprise Architecture artifacts.

- Logical Data Model
- System Functional Requirements
- Logical Application Deployment Model
- Logical Operating Schedule
- Supplementary Specifications

**For Waterfall:** Supplementary documentation is required (if not expressed or captured in the Enterprise Architecture documentation).

- Systems Analysis Document
- Application Requirements and Specification
- Interface Requirements/Specification
- Operational/Support Requirements
- System/Subsystem Specification
- TRA/PIA ((As required)

**For Iterative:** Supplementary documentation required (if not expressed or captured in the Enterprise Architecture documentation).

- Software Requirements Specifications
- Analysis Class
- Use-Case Model
- Use-Case Package
- User-Interface Prototype



- TRA/PIA ((As required)

**I&IT Cluster specific documentation, as required.**

- 2.8 Upon successful completion of the review(s), a baseline for the requirements of the application must be established and formal sign-off must be obtained.

## **DESIGN STANDARD**

---

### **3 Design Standard**

- 3.1 The project team must perform activities/tasks related to design.

Activities include: Design system flow, Develop data model, Create physical data model, Design screens, screen navigation, data entry screens, inquiry screens, help screens, online documentation, Design reports, Forms, Report distribution system, User generated reports, Design Patterns, Existing system modifications, Conduct design walkthrough, Conceptual flow/procedures, Screen design, & Process Implementation...

- 3.2 A top-level architecture of the system must be established. The architecture should identify items of hardware, application/software, and manual-operations. It should be ensured that all the system requirements are allocated among the items. Hardware configuration items, application/software configuration items, and manual operations should be subsequently identified from these items. The system architecture and the system requirements allocated to the items must be documented.

- 3.3 The system architecture and the requirements for the application must be evaluated considering the criteria listed below. The results of the evaluations must be documented.

- a) Traceability to the system requirements;
- b) Consistency with the system requirements;
- c) Appropriateness of design standards and methods used;
- d) Feasibility of the application module/component/services fulfilling requirements;
- e) Feasibility of operation and maintenance.

### **Design - Application**

- 3.4 The project team must transform the requirements for the application into an architecture that describes its top-level structure and identifies the application components. It must be ensured that all the requirements for the application are allocated to its application components and further refined to facilitate detailed design. The architecture of the application must be documented.
- 3.5 The project team must develop and document a top-level design for the interfaces external to the application and between the application components of the application.
- 3.6 The project team must develop and document a top-level design for the database.
- 3.7 The project team should develop and document preliminary versions of user documentation.
- 3.8 The project team must define and document preliminary test requirements and the schedule for Application Integration.
- 3.9 The project team must evaluate the architecture of the application and the interface and database designs considering the criteria listed below. The results of the evaluations must be documented.

- a) Traceability to the requirements of the application;
  - b) External consistency with the requirements of the application;
  - c) Internal consistency between the application components;
  - d) Appropriateness of design methods and standards used;
  - e) Feasibility of detailed design;
  - f) Feasibility of operation and maintenance.
- 3.10 The project team must conduct review(s).
- Appropriate Enterprise Architecture Checkpoint review
  - Internal peer reviews and walkthroughs
- 3.11 The project team must develop a detailed design for each application module/component/service of the application. These should be refined into lower levels containing application units that can be coded, compiled, and tested. It should be ensured that all the application requirements are allocated from the application components to application units. The detailed design must be documented.
- 3.12 The project team must develop and document a detailed design for the interfaces external to the application module/component/service, between the application components, and between the application units. The detailed design of the interfaces should permit coding without the need for further information.
- 3.13 The project team must develop and document a detailed design for the database.
- 3.14 The project team must update user documentation as necessary.
- 3.15 The project team must define and document test requirements and schedule for testing application units. The test requirements should include stressing the application unit at the limits of its requirements.
- 3.16 The project team must update the test requirements and the schedule for Application Integration.
- 3.17 The project team must evaluate the application detailed design and test requirements considering the criteria listed below. The results of the evaluations must be documented.
- a) Traceability to the requirements of the application;
  - b) External consistency with architectural design;
  - c) Internal consistency between application components and application modules/units;
  - d) Appropriateness of design methods and standards used;
  - e) Feasibility of testing;
  - f) Feasibility of operation and maintenance.
- 3.18 The project team must conduct review(s).
- Appropriate Enterprise Architecture Checkpoint review

- Internal peer reviews and walkthroughs

3.19 The project team **MUST** produce the following minimum set of documentation as part of this phase:

**Enterprise Architecture Documentation:** refer to the Enterprise Architecture Checklist for Change Initiative as the authoritative source for preparing the following Enterprise Architecture artifacts.

- Physical Data Model
- Physical Design Document
- System Architecture Document
- Logical Design Document
- Infrastructure Component Placement Diagram
- Infrastructure Pattern Match
- Logical Application Deployment Model
- Functional Group – Application Component Cross-Reference
- Logical Operating Schedule
- Supplementary Specification
- TRA/PIA (As required)

**For Waterfall:** Supplementary documentation is required (if not expressed or captured in the Enterprise Architecture documentation).

- Architecture Design
- System/Subsystem Design
- Application Architecture and Design
- Interface Design
- Database Design
- Screen/Report Design
- TRA/PIA (As required)

**For Iterative:** Supplementary documentation is required (if not expressed or captured in the Enterprise Architecture documentation).

- Design Class
- Design Model
- Design Package
- Software Architecture Document
- Use-Case Realization
- TRA/PIA (As required)

**I&IT Cluster specific documentation, as required.**

3.20 Upon successful completion of the review(s), a baseline for the design of the application must be established and formal sign-off must be obtained.

## **CODING STANDARD**

---

### **4 Coding Standard:**

- 4.1 The project team must review and analyse architecture/design documentation and construct/code to design specifications.
- 4.2 The project team must perform the activities/tasks related to construction or coding.

Activities/Tasks include: Construct the application, components, services including Data entry screens, Inquiry screens, Menu screens, Online help screens, Batch programs, Changes to existing programs, Conversion programs, Build and load files/tables, Build job streams, Develop test cases, Unit test programs, Develop secure code, Develop application/software documentation, Users guide, Turnover documentation, Training materials, Conduct initial turnover walkthrough, Schedule turnover dates.

- 4.3 The project team must develop and document the following:
- a) Each application unit and database;
  - b) Test procedures and data for testing each application unit and database.
- 4.4 The project team must test each application unit and database ensuring that it satisfies its requirements. The test results must be documented.
- 4.5 The project team must evaluate application code and test results considering the criteria listed below.
- The results of the evaluations must be documented.
- a) Traceability to the requirements and design of the application;
  - b) External consistency with the requirements and design of the application;
  - c) Internal consistency between unit requirements;
  - d) Test coverage of units;
  - e) Appropriateness of coding methods and standards used;
  - f) Feasibility of application integration and testing;
  - g) Feasibility of operation and maintenance.
- 4.6 The project team must develop an integration plan to integrate the application units and application components into the application. The plan should include test requirements, procedures, data, responsibilities, and schedule. The plan must be documented.
- 4.7 The project team must integrate the application units and application components and test as the aggregates are developed in accordance with the integration plan. It should be ensured that each aggregate satisfies the requirements of the application and that the application is integrated at the conclusion of the integration activity. The integration and test results must be documented.
- 4.8 The project team must update the required documentation as necessary.
- 4.9 The project team must develop and document, for each requirement of the application, a set of tests, test cases (inputs, outputs, test criteria), and test procedures for conducting Application Testing. The project team must ensure that the integrated application is ready for Application Testing.
- 4.10 The project team must evaluate the integration plan, design, code, tests, test results, and user documentation considering the criteria listed below. The results of the evaluations must be documented.
- a) Traceability to the system requirements;

- b) External consistency with the system requirements;
- c) Internal consistency;
- d) Test coverage of the requirements of the application;
- e) Appropriateness of test standards and methods used;
- f) Conformance to expected results;
- g) Feasibility of application testing;
- h) Feasibility of operation and maintenance.

4.11 The project team must conduct review(s).

4.12 The project team **MUST** produce the following minimum set of documentation as part of this phase:

**Enterprise Architecture Documentation:** refer to the Enterprise Architecture Checklist for Change Initiative as the authoritative source for preparing the following Enterprise Architecture artifacts.

- Refinement of Enterprise Architecture artefact as required

**For Waterfall:** Supplementary documentation is required (if not expressed or captured in the Enterprise Architecture documentation).

- Application Code
- Application Documentation
- User/Operation Manual

**For Iterative:** Supplementary documentation is required (if not expressed or captured in the Enterprise Architecture documentation).

- Software Development Plan
- Build Document
- Implementation Model (Model)

**I&IT Cluster specific documentation, as required.**

4.13 Upon successful completion of the review(s), a baseline for the construction of the application must be established and formal sign-off must be obtained.

## TESTING STANDARD

---

### 5 Testing Standard

5.1 The project team must conduct testing in accordance with the requirements for the application. It must be ensured that the implementation of each application requirement is tested for compliance. The testing results must be documented.

Activities/Tasks include: Application test, Software test, Run parallel test, Document results, User acceptance test, Security test, Develop Test procedures, Document results, Issue production readiness recommendation.

- 5.2 The project team must update the required documentation as necessary.
- 5.3 The project team must evaluate the design, code, tests, test results, and user documentation considering the criteria listed below. The results of the evaluations must be documented.
  - a) Test coverage of the requirements of the application;
  - b) Conformance to expected results;
  - c) Feasibility of system integration and testing, if conducted;
  - d) Feasibility of operation and maintenance.
- 5.4 The project team must support the user testing. The results of the testing must be documented.
- 5.5 Upon successful completion of the test, the project team should:
  - a) Update and prepare the deliverable application product for System Integration, System Testing, Application Installation, or Application Acceptance Support as applicable.
  - b) Establish a baseline for the design and code of the application.

### **System Integration Testing**

- 5.6 The application must be integrated, with hardware configuration items, manual operations, and other systems as necessary, into the system. The aggregates must be tested, as they are developed, against their requirements. The integration and the test results must be documented.
- 5.7 For each requirement of the system, a set of tests, test cases (inputs, outputs, test criteria), and test procedures for conducting System Testing must be developed and documented. The project team must ensure that the integrated system is ready for System Testing.
- 5.8 The integrated system must be evaluated considering the criteria listed below. The results of the evaluations must be documented.
  - a) Test coverage of system requirements;
  - b) Appropriateness of test methods and standards used;
  - c) Conformance to expected results;
  - d) Feasibility of system testing;
  - e) Feasibility of operation and maintenance.
- 5.9 System testing must be conducted in accordance with the requirements specified for the system. It should be ensured that the implementation of each system requirement is tested for compliance and that the system is ready for delivery. The testing results must be documented.
- 5.10 The system must be evaluated considering the criteria listed below. The results of the evaluations must be documented.
  - a) Test coverage of system requirements;

- b) Conformance to expected results;
  - c) Feasibility of operation and maintenance.
- 5.11 The project team must support the system testing. The results of the system testing must be documented.
- 5.12 Upon successful completion of the system testing, if conducted, The project team should:
- a) Update and prepare the deliverable application product for Application Installation and Application Acceptance Support.
  - b) Establish a baseline for the design and code of the application.
- 5.13 The project team **MUST** produce the following minimum set of documentation as part of this phase:

**Enterprise Architecture Documentation:** refer to the Enterprise Architecture Checklist for Change Initiative as the authoritative source for preparing the following Enterprise Architecture artifacts.

- Refinement of Enterprise Architecture artefact as required

**For Waterfall:** Supplementary documentation is required (if not expressed or captured in the Enterprise Architecture documentation).

- Application Test Plan
- Testing Scripts/Scenarios
- Application Testing Procedures
- Software Test Reports
- Defect/Error Reports
- Change requests

**For Iterative:** Supplementary documentation is required (if not expressed or captured in the Enterprise Architecture documentation).

- Defect
- Test Case
- Test Evaluation Report
- Test Plan
- Test Results
- Change Request

**I&IT Cluster specific documentation, as required.**

- 5.14 Upon successful completion of the review(s), a baseline for the testing of the application must be established and formal sign-off must be obtained.

## **IMPLEMENTATION STANDARD**

---

### **6 Implementation Standard**

- 6.1 The project team must develop a plan to install the application product in the target environment as designated. The resources and information necessary to install the application product should be determined and be available. The project team should assist the acquirer with the set-up activities. Where the installed application product is replacing an existing system, the project team should support any parallel running activities that are required. The installation plan must be documented.
- 6.2 The project team should install the application product in accordance with the installation plan. It should be ensured that the application code and databases initialize, execute, and terminate. The installation events and results must be documented.
- 6.3 The project team should support acceptance review and testing of the application product. Acceptance review and testing should consider the results of the Reviews, Audits, Application Testing, Security Testing, and System Testing. The results of the acceptance review and testing must be documented
- 6.4 The project team must conduct review(s).
- Appropriate Enterprise Architecture Checkpoint review
  - Internal peer reviews and walkthroughs
- 6.5 The project team must complete and deliver the application product.
- 6.6 The project team must provide initial and continuing training and support as outlined in the implementation plan.
- 6.7 The project team MUST produce the following minimum set of documentation as part of this phase:
- Enterprise Architecture Documentation:** refer to the Enterprise Architecture Checklist for Change Initiative as the authoritative source for preparing the following Enterprise Architecture artifacts.
- Application Inventory
  - Implementation Document
  - Physical Deployment
  - Calendarized Schedule and State
- For Waterfall:** Supplementary documentation is required (if not expressed or captured in the Enterprise Architecture documentation).
- Application User Manual
  - Application Operator Manual
  - TRA/PIA (As required)
- For Iterative:** Supplementary documentation is required (if not expressed or captured in the Enterprise Architecture documentation).
- Implementation Model (Document)
  - TRA/PIA (As required)
- I&IT Cluster specific documentation, as required.**
- 6.8 Upon successful completion of the review(s), a baseline for the implementation of the application must be established and formal sign-off must be obtained.

## POST-IMPLEMENTATION STANDARD

---

### 7 Post Implementation Standard



- 7.1 The project team must provide, as an integral part of the project team's activities, a plan for a post-implementation review of the application development project.
- 7.2 The project team must perform a post implementation review to support continuous improvement of the implemented application system and to assess whether:
- that application objectives are being achieved.
  - that the application is achieving user's needs.
  - the project team adhered to the selected methodology.
  - Lessons learned.
- 7.3 The project must require that the results or report of a post-implementation review of the application development project be conducted, documented and retained for periodic reviews.

## 4 Document History

Approved by the Information Technology Standards Council (ITSC) – March 21, 2007

Approved by the Architecture Review Board (ARB) – April 24, 2007

## 5 References

IEEE/EIA 12207.0, *Industry Implementation of International Standard ISO/IEC12207:1995 — (ISO/IEC 12207) Standard for Information Technology —Software life cycle processes*

IEEE/EIA 12207.1, *Industry Implementation of International Standard ISO/IEC12207:1995 — (ISO/IEC 12207) Standard for Information Technology —Software life cycle processes – Life Cycle Data*

IEEE/EIA 12207.2, *Industry Implementation of International Standard ISO/IEC12207:1995 — (ISO/IEC 12207) Standard for Information Technology —Software life cycle processes – Implementation considerations*

Kroll, Per; Kruchten, Philippe (2003). *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP*. [ISBN 0-321-16609-4](#).

Kruchten, Philippe (2004). *The Rational Unified Process: An Introduction* (3rd Ed.). [ISBN 0-321-19770-4](#).

Larman, Craig (2004). *Agile and Iterative Development: A Manager's Guide*. [ISBN 0-13-111155-8](#).

Scott, Kendall (2002). *The Unified Process Explained*. [ISBN 0-201-74204-7](#).