Maine State Government
Dept. of Administrative & Financial Services
Office of Information Technology (OIT)

Software Development Lifecycle Procedure

1.0 Purpose

Application Owners will use this Procedure to create the Software Development Lifecycle (SDLC) artifacts for major application projects, as required by the SDLC Policy[1].

This SDLC Procedure is instituted to support the SDLC Policy. The SDLC Policy mentions artifacts to be created through the life of a major application project. This Procedure specifies the contents of such artifacts. However, beyond the contents, this Procedure does not take a position on the appearance and formatting of such artifacts.

2.0 Definitions

2.1 **Acceptance Plan**: This artifact validates the successful deployment and formal acceptance by the Agency of the deployed solution.

2.2 **Architectural Diagram**: This artifact is a visual representation of the solutions architectural design.

2.3 **Deployment Plan**: This artifact is a detailed checklist used by the project team to ensure deployment readiness.

2.4 **Infrastructure Planning Document**: This artifact captures the provisioning blueprint for the infrastructure intended to stand up the application project. This includes complete details of all relevant client devices, terminal emulators, browsers, etc., as well as the full complement of web, application, and database servers.

2.5 **Logical Database Design Document**: This artifact is structured in terms of Entities and Relationships. While it is not expected to finalize all the Attributes of all the Entities, it is indeed expected to contain primary, alternative, and secondary keys, as well as their individual Domains. The Domain of an Attribute is defined by its data type, its size, and its permissible range of values.

2.6 **Object-Relational Model Document**: This artifact applies only if the Business Logic is encoded in an object-oriented programming language. It captures the mapping between objects in the memory with those on the disk. Most contemporary applications implement the Business Logic layer in an object-oriented programming language and house the Data layer in a database. This dichotomy necessitates a mapping between the two kinds of objects. At a minimum, this document should capture the correspondence between classes and tables. It is not mandatory to map inheritances and relationships.

[1] https://www.maine.gov/oit/sites/maine.gov.oit/files/inline-files/sdlc-policy.pdf

Reference 6.1 provides further guidance on this topic.

2.7 **Operations Manual**: This artifact captures all instructions necessary for application operation and administration, including executing batch jobs, restarting aborted/failed jobs, reviewing logs, and all weekly/quarterly/yearly periodic procedures.

2.8 **Physical Database Design Document**: This artifact is structured strictly in terms of the SQL Data Definition Language (DDL). In other words, this is the core of the 'Create & Alter Script' to be executed downstream by the DBA.

2.9 **Problem Statement**: This artifact captures the high-level summary of what is expected and what is not expected from the application project. It may or may not define every deliverable. However, all downstream deliverables must align with the Problem Statement. It should include the background and context of the application project, including why it is being undertaken, and what business values it is meant to produce. This is the only document that absolutely needs to be signed off by all Application Owners, without exception. Finally, if the contents of the Problem Statement are generated under the context of a Project Charter, then that adequately suffices in terms of SDLC Policy compliance and no duplication of effort is warranted. However, it remains the burden of the Application Owners to prove they have generated the entire contents of the Problem Statement, without exception, albeit under a different context.

2.10 **Process Flow Diagram** (also known as a Flow Chart): This is the most widely used process modeling tool and has been in continuous use for more than fifty years.

2.11 **Roles & Privileges Document**: This artifact captures the various user/actor roles that are required in the application, as well as their access privileges to all the entities and transactions. The primary focus of this artifact should be at the business function level, rather than at the underlying database level.

2.12 **Use Case**: A Use Case is a well-defined sequence of actions undertaken jointly by the user and the application, which produces a predictable result of value to the user. Use Cases are utilized to document business requirements and evaluate the users' ability to accomplish their work in a productive manner. Use Cases encompass input/output screens and forms, reports, tasks/transactions, interfaces, and batch jobs. At a minimum, a Use Case specification should include its name, version, summary, detailed primary scenario, detailed secondary scenarios, detailed exception scenarios, triggers, pre-conditions, post-conditions, assumptions, related business rules, code modules, authors, and last update date.

2.13 **User Interface Prototype Document**: This artifact captures the salient details of the input/output screens and forms, edit rules, report layouts and parameters, etc. While actual implementations will include minutiae not covered by their prototypes, the expectation is that actual implementations will not violate the salient prototype details.

2.14 **User Manual**: This artifact captures all the instructions necessary for an average end-user to successfully interact with the application. More specifically, it covers data-entry, query, reporting, workflow, tasks/transactions, batch job schedules and their outcomes, etc. It does not cover operational and administrative aspects of the application.

3.0 Applicability

The applicability of this Procedure is identical to that of the SDLC Policy.

4.0 Responsibilities

4.1 Application Owners: The Application Owners are responsible for creating the SDLC artifacts according to this Procedure and submitting them to the Enterprise Project Management Office (PMO).

4.2 Associate CIO, Applications: The Associate CIO, Applications, is responsible for enforcing this Procedure.

5.0 Directives

5.1 Compliance with the SDLC Policy is measured by the successful creation of the SDLC artifacts according to this SDLC Procedure.

5.2 It is not the intent of the SDLC Policy and Procedure to create duplicate work. Should the Application Owners generate the contents of the SDLC artifacts under different names or contexts, then that adequately suffices in terms of compliance with the SDLC Policy.

5.3 The artifacts, corresponding to each SDLC Discipline, are enumerated below. The artifacts are elaborated in the Definitions section, in alphabetical order.

Discipline	Artifacts
Business Modeling	• Current *Process Flow Diagrams* • Legacy *User Manual (Only if available, not mandatory.)* • Legacy *Operations Manual (Only if available, not mandatory.)*
Requirements	• Problem Statement • Desired *Process Flow Diagrams* • Desired *Use Cases* • Architectural Diagram
Analysis & Design	• *Roles & Privileges Document* • *User Interface Prototype Document* • *Logical Database Design Document* • *Physical Database Design Document* • *Object-Relational Model Document* (*Only if the Business Logic is encoded in an object-oriented programming language.)*
Test	• See the Application Deployment Certification Policy[2].
Implementation / Deployment	• *Acceptance Plan* • *Infrastructure Planning Document* • New Operations & User Manuals • *Deployment Plan*

[2] https://www.maine.gov/oit/sites/maine.gov.oit/files/inline-files/application-deployment-certification_0.pdf

5.4 The artifacts enumerated above constitute the minimum requirements for SDLC Policy compliance. At their discretion, Application Owners may choose to create additional SDLC documentation in support of their specific application projects.

5.5 Submission of the SDLC artifacts enumerated above does not preclude the necessity of submitting other project management documentation to the Enterprise PMO, concerning the overall management of the application project. The Enterprise PMO determines the scope and nature of the required project management documentation.

5.6 What matters in terms of SDLC Policy compliance is the content, not the title. Alternative SDLC methodologies may use broad-based umbrella titles, such as Design Specifications, Requirements Document, etc. Contents of such titles may overlap with multiple Requirements and Analysis & Design artifacts enumerated above. Should it be more convenient to the Application Owners to generate the contents of the artifacts under Requirements and Analysis & Design under broad-based umbrella titles, then that adequately suffices in terms of SDLC Policy compliance and no duplication of effort is warranted. However, it remains the burden of the Application Owners to prove that they have indeed generated all the required contents in their entirety, without exception, albeit under different titles.

6.0 References

6.1 Agile Data, Mapping Objects to Relational Databases: O/R Mapping in Detail, Last Updated on October 6, 2006[3]

7.0 Document Information

Initial Date: August 31, 2009
Last Revision Date: March 30, 2018 – To update Document Information.

Point of Contact: Director, Project Management Office, OIT
Approved By: Chief Information Officer, OIT

Legal Citation: Title 5, Chapter 163: Office of Information Technology[4].
Waiver Process: See the Waiver Policy[5].

[3] http://www.agiledata.org/essays/mappingObjects.html

[4] http://legislature.maine.gov/statutes/5/title5ch163sec0.html

[5] https://www.maine.gov/oit/sites/maine.gov.oit/files/inline-files/waiver.pdf