



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

## **Software Development Lifecycle Policy**

### **1.0 Purpose**

In order to be successful, all *major application projects* must undergo a well-defined development lifecycle. This Policy establishes the minimum requirements and responsibilities for such a lifecycle in Maine State Government.

Software development is a complex endeavor, susceptible to failure, unless undertaken with a deliberate and systematic methodology. The Maine State Software Development Lifecycle (SDLC) is a methodology for implementing an *application project* by following a sequence of standard steps and techniques. In combination with sound Project Management, the SDLC improves the capability of application projects to deliver as expected, on time, and within budget. Besides increasing the success rate of application projects, a statewide SDLC also facilitates statewide collaboration in application projects. Finally, a statewide SDLC increases the efficiency and effectiveness of professional training for the State's software development personnel.

### **2.0 Definitions**

- 2.1 Application Owners:** The Project/Product Manager, the Executive Sponsor, and the Technical Leader are jointly and collectively identified as the Application Owners.
- 2.2 Application Project:** The [Project Management Institute](http://www.pmi.org)<sup>1</sup> defines a Project as a temporary endeavor undertaken to create a unique product, service, or result. By extension, an Application Project is a temporary endeavor undertaken to create a unique application product, be it a new application or the upgrade of an existing application.
- 2.3 Discipline:** Disciplines are the specialized activities that take place over the life of an application project. The Disciplines of this SDLC are Business Modeling, Requirements, Analysis & Design, Implementation, Test, and Deployment.
- 2.4 Major Application Project:** An Application Project is deemed to be major if it meets the threshold of Enterprise Portfolio submission. At the discretion of the Associate CIO Applications, or the Enterprise PMO, or the Application Owners, an Application Project that does not meet the threshold of Enterprise Portfolio submission may still be deemed to be major due to its complexity, operational impact, security impact, business criticality, media or political exposure, etc. Should it appear that a large application project has been decomposed into smaller projects, some or all of which fall below the threshold of Enterprise Portfolio submission, then the Enterprise PMO or the Associate CIO, Applications, may designate any or all of the smaller projects as major.

---

<sup>1</sup> <http://www.pmi.org>

2.5 **Phase:** Phases represent the sequential evolution of an application project through time. The Phases of this SDLC are Inception, Elaboration, Construction, Transition, and Production.

### 3.0 **Applicability**

3.1 This Policy applies to all major application projects, both new applications and upgrades of existing applications, under the purview of the Chief Information Officer (CIO), irrespective of hosting location.

3.2 At its discretion, the Associate CIO, Applications, may sanction a simplified implementation of this SDLC for certain application projects.

### 4.0 **Responsibilities**

4.1 *Application Owners:* The Application Owners are responsible for executing this SDLC and submitting the resulting artifacts to the Associate CIO, Applications. This submission consists of the names and signatures of the Application Owners, and the actual artifacts. (Artifacts are the documents, diagrams, etc., that are created as a result of following the SDLC.)

4.2 Application Owners and Enterprise Project Management Office (PMO): The Application Owners and the Enterprise PMO must jointly consider this SDLC as an integral part of the overall project plan.

4.3 Associate CIO, Applications: The Associate CIO, Applications, is responsible for enforcing this Policy.

### 5.0 **Directives**

5.1 A software application typically undergoes several development lifecycles, corresponding to its creation and subsequent upgrades. Each such development lifecycle constitutes a project. Such projects continue until the underlying technology ages to the point where it is no longer economical to invest in upgrades and the application is considered for either continued as-is operation or retirement. The Maine State SDLC defines a standard methodology for the creation and upgrades of software applications. It does not address routine maintenance that occurs as part of the operational management of an application. Nor does it address retirement.

5.2 The Maine State SDLC is a subset of the Enterprise Unified Process, which is a derivative of the IBM Rational Unified Process. This SDLC is structured in two dimensions: *Phases* and *Disciplines*.

5.3 This SDLC is sequential with respect to the *Phases*, while iterative, or incremental, with respect to the *Disciplines*. More specifically, each Phase ends with a go/no-go decision on whether to proceed to the next Phase. Each Discipline ends with a set of deliverables encapsulating the results of a specific activity, which is incrementally refined over multiple iterations of the Phases. While it may appear that a one-to-one correspondence exists between the elements of the Phases and the elements of the Disciplines, this is absolutely not the case. In reality, designing, coding, and testing continue incrementally

across multiple Phases. More detailed coverage of this two-dimensional model is provided in References [6.1] and [6.2].

5.4 The Maine Office of Information Technology will propose, adopt, and implement Procedures, Best Practices, etc., in support of this Policy.

5.5 This SDLC is meant to operate under the umbrella of the Enterprise PMO.

## **6.0 References**

6.1 [Ambler, Scott W. A Manager's Introduction to The Rational Unified Process \(RUP\), December 4, 2005<sup>2</sup>.](#)

6.2 [Ambler, Scott W. Introduction to the Enterprise Unified Process \(EUP\), October 15, 2005<sup>3</sup>.](#)

6.3 [Software Development Lifecycle Procedure<sup>4</sup>.](#)

## **7.0 Document Information**

Initial Issue Date: August 31, 2009

Last Revision Date: March 6, 2017 – 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<sup>5</sup>.](#)

Waiver Process: See the [Waiver Policy<sup>6</sup>.](#)

---

<sup>2</sup> <http://www.ambysoft.com/downloads/managersIntroToRUP.pdf>

<sup>3</sup> <http://www.enterpriseunifiedprocess.com/downloads/eupIntroduction.pdf>

<sup>4</sup> <http://maine.gov/oit/policies/SDLCProcedure.pdf>

<sup>5</sup> <http://legislature.maine.gov/statutes/5/title5ch163sec0.html>

<sup>6</sup> <https://www.maine.gov/oit/sites/maine.gov.oit/files/inline-files/waiver.pdf>