Louisiana Department of Insurance

Software Development Standards

Table of Contents

Document last updated on January 30, 2017

Executive Summary

This document represents a basis for the overall design, implementation, development, deployment, and documentation for which all work performed on current applications and systems as well as future systems deployed at LDI must adhere. Included within this basis are the general and specific requirements as defined by the department. These requirements cover the internal and external systems which the department depends on for day-to-day operations. The increasing complex external systems are becoming more important for proper department operation.

In addition, the method of integration for all applications which bonds the applications and systems together and creates a seamless department-wide applications are included within the document. Finally, the structure and storage of all data within the department databases are described.

These standards have been developed with the cost of implementation in mind, and it is believed that these standards will have a minimal cost impact to the department when implemented. There is no risk to the department in employing these suggested standards. The standards have been designed to ensure maximum future flexibility, greatest growth potential and lowest cost of maintenance. A greater risk to the department's operations exists if developers do not follow or correctly employ these minimum standards as more applications are developed.

The complexity and critical nature of the department's automated systems in relation to the operations and function of the department necessitate that these standards be implemented and followed closely. The department's overall IT plan for a completely integrated system is currently in progress and nearing fruition. The plan depends on the standards being implemented and followed, and adherence to the standards will be being monitored.

LDI Software Development Standards Goal

The goal of this document is to establish common standards including system integration for which all present and future automated systems will obey.

These standards will provide the department maximum flexibility, increase the flow of information between different systems within the department and create a foundation for all systems to start from.

These standards cannot be circumvented in any state of an application's or system's lifecycle. Only by all systems embracing these standards will the overall goals of the LDI be achieved.

Implementation Objectives

The overall objective during implementation of these standards is to ensure and verify proper integration and consistency across the automated system or systems being developed or maintained. Correct integration within the department databases and current automated systems is crucial for the operation of the department. The process and scheme of implementation must be consistent with other databases and systems as to minimize the future development and maintenance costs to the department. Finally, only by consistent implementation can the department be secure in relying on the systems for dependable operation, and be secure in the accuracy of the information being stored and processed.

LDI Systems Integration

A primary goal of the departmental software standards is the creation of a set of common interfaces within the department's databases which will allow the flow of information between databases, systems, and applications without error, maximize speed and decrease the need for special interfaces.

The introduction of all new systems **shall** integrate into Department's database paradigm.

New systems working in concert will ensure that special data transformation programs and routines do not have to be developed, deployed, and relied upon for daily department operations unless external sources or regulation requires otherwise.

LDI Acceptable Tools and Technologies by Category

The following tools and technologies are acceptable for the development of NEW LDI Software Systems, unless otherwise approved on a project-by-project basis:

Operating Systems

- Microsoft Windows Server 2012 R2/ 2016
- Microsoft Windows 10

User Interface

- Internal users – IE 10 or greater
- External users – Modern browser supporting HTML5 and CSS3

Database Engines

- SQL Server 201/2016

Programming Languages, Tools and Technologies

- .NET Framework 4.6+
- Visual Studio 2017
- C#.NET 6
- Java Script
- JQuery
- MVC5
- ASP.Net Razor Syntax
- Entity Framework 6
- Axure RP Pro
- XML
- Active Reports 10
- Telerik DevCraft
- SQL Reporting Services

- ReSharper
- Closed XML
- Aspose.Net
- ItextSharp
- LinqPad
- JSON
- HTML 5
- CSS 3

- No compiled code will be used within the system. Example: CLR for SQL queries

Database Design Tools

- SQL Server 2014/
- Microsoft Visio 2016

Administrative Tools and Technologies

- Team Foundation Server 2017
- Microsoft Project Server 2016
- Red Gate SQL Toolbelt
- Red Gate SQL Source Control
- OneNote 2016
- Microsoft Office Professional 2016

Content Management Systems

- Telerik Sitefinity

Documentation Technologies

- Adobe Photoshop CS6
- Adobe Acrobat Pro XI
- Microsoft Office Professional Plus 2016

- **All development and maintenance work will be completed on Department systems using only Department-approved software.**

- **For all maintenance performed, the original development environment and/or application initially used can be utilized.**

LDI Project Management

The department has chosen Microsoft Project as its primary automation tool to assist in managing all IT projects. In addition to the reports which can be generated by Microsoft Project, the department has mandated that the following additional documents be produced.

- Requirements Document
- Design / Definition / Specifications Document
- Project Plan and Work Breakdown Structure
- Screen and system functional mockups
- Execution Schedule
- Scope Document
- Weekly Status Reports
- Issue Descriptions
- Change Requests / Issue Description
- Sign Off Sheets
- Test Plan
- Test Plan Results

Requirements Document, Scope Document, Design / Definition Specifications Document, overall Project Plan, Work Breakdown Structure, and Execution Schedule are all due before programming on a project begins. These documents should at a minimum determine the functionality, operational capability, and features of the system, define the whole organizational structure of the system, explain any critical dates in the timeline of the project, illustrate any possible problems, and define the critical path for project completion.

A Gantt chart is an appropriate method for displaying timelines and the critical path for a project.

Completed weekly status reports including timesheets, change requests / issue description sheets, sign-off sheets, and test plan results are to be given to both IT and the respective division's IT coordinator for project tracking and stored in the central LDI OneNote repository.

At minimum, the Scope Document must include, but is not limited to the following sections:

- Objectives of the project
- Scope of the project
    - In Scope items that will be developed
    - Out of Scope items discovered in interviews of staff that is beyond the original statement of work and should not be considered as items in the project – these items shall be signed off prior to development
- Deliverables that will be produced
- Assumptions of the contractor in development of the Scope Document
- Risks associated with the project

For the Software Development Lifecycle, the LDI uses a hybrid approach for tracking of the overall project and the execution of the actual work. The LDI uses a phased waterfall approach combined with Project Management Institutes (PMI) process groups for the overall management of the project. Agile methodologies are utilized within the phases for project execution and actual work performed, especially during software development. Since Agile is a collection of numerous practices and methodologies combined with an assortment of tools that are dependent on the goals, circumstances, scope and complexity of the project, we use the Disciplined Agile Delivery (DAD) framework. DAD builds on the many practices espoused by advocates of agile software development, including Scrum, Agile Modeling, Lean software development, Extreme Programing, Kanban and others. We have used this combined approach with great success in numerous past projects.

LDI Project and Software Deliverables

All software development projects and maintenance projects at the department have a 1) software and 2) documentation component. These two components should be delivered in a form consistent with Department standards in accordance with the LDI Acceptable Tools and Technologies by Category section of this standard.

All objects written on the SQL server should be written in Transact SQL, no CLRs.

Software Deliverables

LDI requires the following software deliverables:

- Interface and system proof of concept
- Final code and system settings published within the LDI Team Foundation Server
- Documented procedures for publishing the developed application and/or system to the development, staging and production servers. Additionally, any settings or client side configuration required for the application/system to function properly
- Documentation on the software developed including: high level function and functionality

The functional interface mockups and system proof of concept are due prior to primary development. Interfaces and business rules can be constructed using any of the LDI approved tools and technologies. Demonstration, review, acceptance and signoff by key stakeholders and LDI IT is required prior to primary development proceeding.

The final code base, databases, all system settings and documentation, stored within the LDI central OneNote notebook are due at the final system sign-off. Further explanation of how these deliverables will be transferred is detailed in the documentation section of this document.

Documentation Deliverables

LDI requires that all documentation be a consistent organized collection of documents that describe the global structure, purpose, operation, maintenance and data requirements for a program. All documentation for LDI systems will include:

- User Documentation / User Manual
- Program Source Code and Technical Documentation

Program and Technical documentation should have at least these sections:

- System Design Overview
- Operational Environment
- Object Reference
- Database Models
- Entity Relationship Diagrams
- Database Normalization
- Stored Procedure Reference
- Table Reference / Data Dictionary
- Integration with LDI's Integrated Database
- Security Reference
- ASP/ C# Source Code – Locations within the Department servers
- Stored Procedure Source Code
- SQL Script
- Report Descriptions
- Deployment Instructions

All documentation should be submitted in the LDI approved electronic format, Adobe PDF.

Documentation deliverables are due when the software deliverable is completed or according to the contract between LDI and the contractor.

User Documentation/ User Manual

User Documentation – User manual is to be written from a user perspective. The purpose of the document is to empower the user to be self-sufficient.

Application

- How To – Includes any of the following that apply to the project. All documentation is to be written in "User Manual" format. Include screen capture images for easy understanding.

  - Security Maintenance
  - Application Maintenance
  - Application Usage
  - Expirations – Any expirations which will affect the system (i.e. rollover of data, temporary permissions, and site and/or application certificates)

Technical Documentation

All technical documentation should to be written from a design and support perspective. The reader of the documents should be assumed to fully understand the technology and grasp the problem for which the system provides a solution. The only instance when technology should be explained is when the technology is being utilized in a non-standard method, or in a technique that has not been exploited previously by the department. Industry standards are to be followed and referenced at all times.

Standard technical documentation to be produced for all projects:

- System Design Overview
- Operational Environment
- Object Reference
- Database Models
- Entity Relationship Diagrams
- Stored Procedure Reference
- Table Reference / Data Dictionary
- Integration with LDI's Integrated Database
- Security Reference
- All System Source Code
- Stored Procedure Source Code
- Report Descriptions
- Deployment Instructions

Below is a synopsis for each document:

- **System Design Overview**
  - Overview: Provide an overview of the system developed.
  - Program Specifications: Describe the specifications developed in the planning phase of the project.
  - Functions: Specify the system / subsystem functions.
  - System/Subsystem Logic: Describe the logic flow of the entire system/subsystem in the form of a flowchart / diagram.

- **Operational Environment**
  - Operations: Describe the operating characteristics of the user and computer centers or sites where the software will be operational.
  - Equipment: Identify the equipment and software required for the operation of the software to be developed.
  - Support Software: Describe any if needed.
  - API (Application Programming Interface): Describe all APIs used by the system
  - Custom DLLs – Describe DLLs used by the system, which are not industry standard and required for normal operation. Custom developed DLLs shall also include the source code.
  - Interfaces: Describe and define all interfaces to the system. These include interfaces with other Department databases and application systems, external databases and systems to the department, and specific user interface requirements.
- **Object Reference**
  - Overview: Identify and describe all program and data objects developed.
  - Sub Programs: Describe any separate subprograms required for system functionality.
- **Database Models**
  - Overview: Describe the database design and goals and database models developed for system.
- **Entity Relationship Diagrams**
  - Overview: Include all relationship diagrams for system. Due upon request and prior to system deployment for new or substantially updated systems which are being maintained.
- **Database Normalization**
  - Overview: Describe normalization implementation and how it relates to the existing LDI databases.
- **Stored Procedure Reference**
  - Stored Procedures: Define and describe all stored procedures developed for system which are located on the departmental database.
  - Special Dependencies: Define and describe any special or extraordinary stored procedure dependencies.
  - Views: Describe and define all database views.

- **Table Reference / Data Dictionary**
  - Table Schemas
  - Schemas and Table Definitions: Describe all table layouts with types, length and size where appropriate.
  - Database Diagrams: Include all diagrams with identifying primary and foreign keys with all indexes.
  - Data Dictionary: Include all fields by field name with description of information to be stored in data field.
- **Security Reference**
  - Model: Define the security model utilized for the system. Describe all algorithms for security verification and encryption.
  - Database: Include all table, stored procedure, and view permissions by active directory users and groups.
  - User: Include all specific update, view, and create permissions by active directory users and groups.
- **All System Source Code**
  - Source Code: Include all source code of the system divided by module. The source code will be stored within the LDI TFS server.
- **Stored Procedure Source Code**
  - Stored Procedure Source: Include all transact SQL source code for all procedures, divided by procedure.
- **Report Descriptions**
  - Overview: Describe each report. Include reporting requirements, all input parameters and sample output.
- **Deployment Instructions**
  - Overview: Detail technician level instructions for installing the developed system.
  - Dependencies: Describe all required support software by operating system, operating system version, support software, support software version, and required software location on server and client computers.

LDI Application Look and Feel Guidelines

All applications and systems developed for the department should conform as closely as possible to the Windows design metaphor. In addition to conforming to the Windows design metaphor, applications need to conform to the look, feel and function of the Department applications and color. The department has chosen Microsoft products as the Department's standard for base installations on all Department workstations. Designing programs and systems to closely conform to the Microsoft standard will minimize the training costs and reduce the time required for users to become familiar with a new application or system.

All deviations or extensions from this standard must be approved by the IT division. Additional approval may be required from the IT coordinator for the division or divisions for whom the application or system is being developed.

LDI External Browser Compatibility Requirements

In order to for the Department to meet its mission of regulating the state insurance industry, all work produced must be available to the widest possible audience. Therefore, any application which will be used by the general public or industry must be compatible with the greatest range of potential Internet browsers. In order to accomplish this goal, when external-facing systems or applications are being built, updated or maintained, they must be 100% compatible with the following browsers:

- Internet Explorer 10 or higher
- Firefox 31 or higher
- Chrome 28 or higher
- Safari 8 or higher

Additionally, all system and application screens will be built to be web responsive, and mobile compliant, in their overall design to ensure maximum usability with differing screen sizes. The screen resolution guidelines will be will be set by the department on a project-by-project basis.

All code will be verified on these browsers as part of the testing process to ensure 100% compatibility, web responsive and mobile compliant design. Testing results will be submitted to the Department for verification.

Network, Servers and Miscellaneous Standards

- All new servers set up for development which may require input from the Internet must be secured via a web security certificate. It is the responsibility of the developer requiring the new server to consult with LDI Network and Maintenance contractors as well as LDI technicians to determine whether the new server requires a security certificate. LDI network contractors or LDI IT staff will work in concert to load and configure any servers required for software development or maintenance.

- Applications shall not be developed and tested on a production server.

- All new servers set up to house web applications accessible externally must be on the DMZ. There will be no exceptions. It is the responsibility of the developers who are setting up the server along with the Network contractors and IT technicians to see that this standard is enforced.

- Backing up new servers.

  a. All new servers, regardless of whether or not they are used for development, testing or production must be added to the LDI backup solution. There will be no exceptions.

  b. All code on development / testing and production servers shall be placed on a drive other than the C drive. In other words, SOURCE CODE OR SITE DIRECTORIES SHALL NOT be placed on the C: drive.

- Delivery of documentation and source code from a developer to the department will be completed using the Department of Insurance Sign Off sheet. The time and

location for a smooth turnover shall be agreed on by all parties and signed off.

- The LDI standard session is 20 minutes for both internal and external users. Session timeout exceptions need to be evaluated by LDI staff and all the developers onsite in order to set the optimal timeout. Not all session timeouts are the same.

- All programmers and technicians working at the LDI must share and discuss their problems and solutions as well as their progress regarding the applications they are developing. In order to affect this exchange of ideas, weekly meetings will be held which must include the developers and project managers from every developing entity at the LDI.

- Logins, badges, and security: All contractor accounts will be disabled by IT immediately upon completion of a contract when a contractor's services are no longer needed. The badges will be collected by the IT technical staff in charge of Active Directory.

    a. IT Technical staff in charge of Active Directory will disable the Active Directory accounts based on a list submitted by the IT Project Leader.
    b. The IT staff or Contract / DBA will disable all database and/or server accounts from a list submitted by the IT Project Leader.

LDI Development, Test and Production Environments

In order to create the most robust environment for systems development and production, the areas of development and production have been logically and physically separated into the following regions:

- Development
- Testing/Staging
- Production

Only in the **system development environment** are databases and program code objects created and modified, and actual system or program coding occurs. The development environment is characterized by the unique ability of the developer to make dynamic changes to their system development area without prior authorization or coordination, and does not affect testing or the production environment.

During and after development, all system code developed or updated is controlled and cataloged through Microsoft Team Foundation Server. The location and management of the Microsoft Team Foundation Server will be controlled by the LDI IT department. All code developed will be checked in daily to the TFS server once the code is deemed stable by the developer.

The **application testing environment** shall be an exact replica of the production system on which the system is designed to function. This environment allows the developed code to be tested against final production schemas before being moved into production. The primary difference between the development environment and the staging/testing environment is the ability for the developer to make on-the-fly changes to the underlying database. The staging databases are not to be changed without prior coordination. The staging/testing environment is to be used for all system and program testing by the developers, LDI IT staff and LDI users. Testing by developers, LDI IT staff and LDI users will be coordinated to ensure testing is consistent.

A SQL job is scheduled and routinely executed to keep the data in the test environment consistent with the production environment. Certain

database objects require the LDI DBA to manually sync the different environments. All database object syncing will be coordinated with all required parties and executed solely by the LDI DBA.

The **production environment** is logically and physically separated on different servers. LDI internal application databases are also physically separated from external, Internet, and WEB application databases which are located on the network DMZ. This allows for maximum flexibility and security with departmental data. The production environment is designed for maximum uptime and the fastest possible response time. As a result, no on-the-fly changes are allowed within the production environment. All changes to either the databases or system programs have to be scheduled with LDI IT staff, support personnel, and the LDI division responsible for the automated system.

Unhandled exceptions that cannot be corrected immediately should be trapped and the captured information sent to a log file for evaluation. During the testing and debugging phase of development, errors do not have to be trapped on the development and test servers, but these errors are unacceptable in the production environment so the developer should use their own procedures to correct these errors in the development environment. A procedure of adding a “continue” or a “cancel” routine and/or button on the interface is not acceptable.

Methods and Procedures to Move New Systems and Updates into the Production Environment

Once a newly developed system or existing system update has been thoroughly tested by the developer and Department staff, the department staff has approved the system update, and the system signed off, only then can the system be published to the production environment for operations.

LDI staff, maintenance personnel, working in conjunction with the entity responsible for completing the project, will determine the best method for transitioning a new system or updated system into the LDI production environment.

Permissions Structure, Passwords, and Supported Computer Software

All system, application, and database permissions are handled by the Windows Server active directory, and roles defined within the database server.

All users within the department are divided into functional areas or groups. Windows Server has the groups defined within its active directory. Users are defined by which group they belong to. The group itself determines which permissions or abilities the group has. All database roles are a subset of the group as defined within the active directory. Together the group's permissions and database roles describe all actions a user can perform on the departmental network. The department's Active Directory will control the internally developed application; web-based systems must have passwords that meet the requirements of LDI IT SEC-POL-003 as adopted by LDI. To meet the current requirements for password complexity, the password is required to be at least eight (8) characters in length, case sensitive, and contains at least three (3) of the following categories:

- English upper case
- English lower case
- Base 10 digits
- Non-alphanumeric characters (%, &,!,etc.)

No blank passwords are permitted.

No other method or technique of managing users or user's permissions is allowed.

New accounts will be created and appropriately named for the purpose its being used.

Individual user accounts SHALL NEVER be used as a service account for a system or server.

Service accounts will be named for an intended system or function, and will not be used in any other unrelated system or function.
As per IT SEC-POL-010, no non-IT supported software is to be installed on computers. If you wish to test new software, you are responsible to report the software to IT for determination if the licensing criterion for the software is compliance with State mandates and LDI's policy.

Naming Conventions and Data Standards

All tables, stored procedures, database views, code modules, programs, and reports should comply with the Department's naming conventions.

The Department's current naming convention for tables, stored procedures, database views, and reports has the application name followed by an underscore preceding the name of the object. This standard is consistent across all systems.

All code or reference tables have an underscore and the characters code following the table name.

Periodically, the department updates and extends naming convention guidelines. Only the most current naming convention guidelines are to be utilized.

Database fields cannot be an empty string or spaces. Nulls must be used in the fields. Date fields must contain either a date or a null.

Management and Finance System Description

The Employee Portal was developed in 2013 as a new interface and workflow engine to replace our legacy SharePoint Workflow System. This portal will one day replace our departmental Intranet system by combining the features of the current Intranet, the Employee Portal, and the Management and Finance System. This portal will be very employee centric, and will include many features designed to help the employee work as efficiently and effectively as possible. This will be the place that employees visit throughout the day to complete their tasks.

The current iteration of the Employee Portal houses several workflows such as Travel Requests, Contract Requests, Contract Amendment Requests, and Refund Requests. The Employee Portal was developed using Visual Studio Premium 2012 (C#), .NET Framework 4.5, Team Foundation 2013, and SQL 2014 for the backend database. Within Visual Studio, we use the MVC template which separates the input, business and UI logic. Entity Framework 5 is an object-relational mapper that enables us to quickly work with relational data and Razor is the engine that lets us combine HTML markup and server-based code into one file (.cshtml). In the Employee Portal UI, for the most part, we use a

mixture of Kendo UI (part of Telerik DevCraft toolbox) and Microsoft controls to build our forms. We also use Unsemantic CSS framework to place all our controls and make the forms responsive. We also use Aspose.Total and Active Reports 10 to create various reports for the application. The project has been upgraded and is being developed and maintained using Visual Studio 2015 Enterprise and SQL 2016. We have plans to upgrade to Team Foundation Server 2017 in the near future.

The biggest upgrade, however, has been moving from a totally custom solution for the workflow engine to Windows Workflow Foundation (WF). Creating custom workflows can be very complex and time consuming, so we wanted to adopt a framework that gave us all the tools that would allow our developers to be more productive by concentrating on the business logic instead of the lower level processes, and produce programs that are easier to manage and change. A workflow-based application does the same things as other applications. It maintains state, gets inputs, and sends outputs, provides control flow, and executes code. In WF, all of these things are done by activities. Activities can be things such as sending emails, checking AD groups, assigning, updating SQL, etc. We prefer the Flowchart workflows because it better represents how users view processes, and most users are familiar with the flowchart way of modeling. The workflow is created graphically using a designer, and the main logic is defined in one coherent stream.

Key Controls

The LDI does not outsource any key controls.

The contractor is required to provide a quality assurance program that monitors and lists each Interface, Project Solution file and/or Databases in which each staff member has access to Personally Identifiable Information (PII).

The contractor shall use the Principle of Least Privilege so LDI staff members and users will be restricted to the systems that they are to use. Periodic LDI internal audits, will be used to verify the proper use for the Principle of Least Privilege. LDI will require the Contractor to provide assistance in determining records access, information system access and LDI staff access required to perform the audit.

As part of the LDI Quality Control program the IT Director or designee will verify contractor's QA reports and LDI internal audits. These reviews will verify the effectiveness of these reports and internal audits to include the information gathered, how the information is gathered and if the requested changes to the contractor have been successfully implemented. Also, ad hoc Quality Control reviews may take place at any time.

Appendices

Identifying Department Leaders

To provide needed support for IT initiatives, key leaders in the LDI need to be sought out and a working relationship built. The first step should be in identifying all key system at the department and creating owners of these systems in the client community. This ownership inherently builds a relationship between IT and the clients using the systems. These key Department leaders know firsthand the business and the issues the systems need to resolve.

Review Ongoing Status of Projects

To accurately measure the success of a project, there must be a set of standards to measure against. While all projects undertaken are unique and have varying measurements, there are some which are common to all projects.

Common measurements:

- Adherence to timeline
- Adherence to budget
- Percent project complete (based on task from the function specification but not budget or timeline)
- Level of client satisfaction
    - Client response to demonstrations of product during each iteration
- Other measurements defined by the project leader and the primary client
    - Adherence to statement of work
    - Report from test results

A standard report should be developed which takes these measurements into account and reported on a regular interval.

Setting Development Guidelines

The IT Division should set guidelines for not only defining the platforms the applications either execute on or the technology within which they are developed, but should also set the

standards by which the applications are developed. This includes but is not necessarily limited to programming frameworks, languages, naming conventions and documentation.

Resolve Issues which Affect Multiple Projects

By being involved with all the projects at the LDI, the IT Division will from time to time come upon issues which affect multiple projects or the entire organization. These issues should be resolved so that the organization receives a unified result. As an example, over time there have been issues which have come up at the Department (i.e. a consistent numbering schema for entities) where multiple solutions have been implemented by various projects. In the end, a unified solution will need to be implemented and the existing systems converted.