Systems Development Life Cycle and IT Audits

Tommie W. Singleton, Ph.D., CISA, CMA, CPA, CITP

In systems development, the temptation to skip certain prescribed tasks associated with documentation, combined with the fastpaced life of IT professionals, can create an environment that is not able to properly employ the best practices of systems development. However, the employment of best practices has proven over the years to provide returns in both efficiencies and effectiveness.

In all types of audit, the employment of any set of "best practices" is generally seen by auditors as a positive impact on the quality of the information, systems or operations being audited. In the case of the systems development life cycle (SDLC), some practices provide additional benefits in terms of IT audits. Specifically, throughout the steps in the SDLC, documentation is being created that provides valuable potential sources of evidence for IT auditors. In other words, employing SDLC as it is prescribed in the industry is a control.

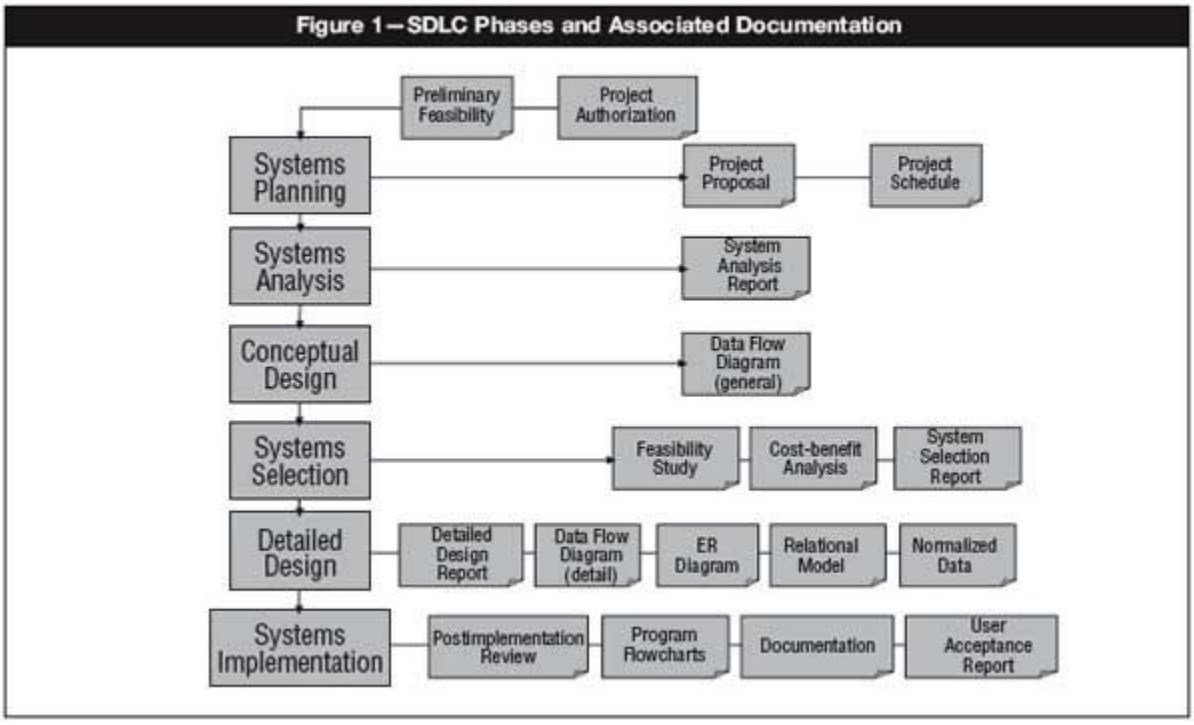
In this article, the conventional phases of the SDLC—and how each one can provide this potential evidence—will be discussed. Different groups use different lists of steps in the SDLC, but almost all agree on the same elements. Herein, a list of eight phases is used to demonstrate this process of analyzing an entity's SDLC. A summary of six of the eight phases and examples of related documentation are depicted in **figure 1**. Other documentation should exist; those contained in the figure are for illustrative purposes.

Phase One: Systems Planning

In phase one, systems are planned using a strategic approach. Executives and others evaluate the effectiveness of systems in terms of meeting the entity's mission and objectives. This process includes general guidelines for system selection and systems budgeting. Management develops a written long-term plan for systems that is strategic in nature. The plan will change in a few months, but much evidence exists that such planning pays dividends in terms of effective IT solutions over the long term.

This phase is similar to IT governance, and the two are quite compatible. Thus, the first thing an IT auditor would like to see is the implementation of IT governance activities.

During this phase, several documents will be generated. They include the long-term plan, policies for selection of IT projects, and a long-term and short-term IT budget, as well as preliminary feasibility studies and project authorizations. Project proposals should have been documented when submitted to management, and a project schedule should exist that contains the approved projects (see **figure 1**).



The presence of these documents illustrates a structured, formal approach to systems development and, as such, illustrates an effective planning system for IT projects and systems. It also demonstrates a formal manner of approving IT projects.

IT auditors will want to verify the presence of a systems planning phase (or IT governance activities) and take a sample of the documents to verify the effectiveness of that system. The same audit procedure will be true for all of the other seven phases and, therefore, will not be repeated in the narratives of phases two through eight.

Phase Two: Systems Analysis

In the systems analysis phase, IT professionals gather information requirements for the IT project. Facts and samples to be used in the IT project are gathered primarily from end users. A systems analyst or developer then processes the requirements, producing a document that summarizes the analysis of the IT project.

The result is some kind of documentation, such as a systems analysis report (see **figure 1**). Other documentation should exist. In effect, systems analysis illustrates the entity's ability to be thorough with its systems development.

Phase Three: Conceptual Design

Next comes the conceptual design phase. In phase two, systems analysis, the requirements have been gathered and analyzed. Up to this point, the project is on paper and each user group has a slightly different view of what it should be. At this point, a conceptual design view is developed that encompasses all of the individual views.

A variety of possible documents could be the output of this phase. **Figure 1** uses a data flow diagram (DFD), developed to a general level at this point, as an example. The point is that one or more of these documents should exist if the entity is following the SDLC thoroughly.

Phase Four: Systems Evaluation and Selection

During the systems evaluation and selection, managers and IT professionals choose among alternatives that satisfy the requirements developed in phases two and three, and meet the general guidelines and strategic policies of phase one.

Part of the analysis of alternatives is to do a more exhaustive and detailed feasibility study—actually, several types of feasibility studies. A technical feasibility study examines whether the current IT infrastructure makes it feasible to implement a specific alternative. A legal feasibility study examines any legal ramifications of each alternative. An operational feasibility study determines if the current business processes, procedures and skills of employees are adequate to successfully implement the specific alternative. Last, a scheduled feasibility study relates to the firm's ability to meet the proposed schedule for each alternative. Each of these should lead to a written feasibility report.

Another aspect of this phase is a cost-benefit analysis. Quantifying tangible and intangible costs and benefits, an accountant should be able to determine the value of each alternative. This phase is associated with how to assess the value of IT.

Finally, since a definitive choice among alternatives is being made, a selection report should be written to explain the reasons behind the choice and, possibly, include the costbenefit and feasibility studies.

Phase Five: Detailed Design

At this point, IT professionals have chosen the IT solution. The DFD design created in phase three is "fleshed out"; that is, details are developed and (hopefully) documented. Examples of the types of documentation created include use cases, Unified Modeling Language (UML) diagrams, entity relationship diagrams (ERDs), relational models and normalized data diagrams. Other systems design documents could also exist. IT professionals often do a walk-through of the software or system to see if any defects in the system can be detected during development. That walk-through should also be documented.

To summarize this phase, a detailed design report should be written to explain the steps and procedures taken. It would also include the design documents referred to previously.

Phase Six: Programming and Testing Systems

For in-house development of applications, current best practices include the use of object-oriented (OO) programs and procedures. IT auditors should be interested in the IT programming shop's choice of tools and procedures. Some businesses are locked into legacy systems and applications and, thus, would not be expected to use OO (e.g., banks). IT auditors would also be interested in programming flow charts as documentation.

No element of SDLC is more important than systems testing. Perhaps none of the phases has been more criticized than testing for being absent or performed at a substandard level. Sometimes management will try to reduce the costs of an IT project by cutting out or reducing the testing.

Sound testing includes several key factors. The testing should be done offline before being implemented online. Individual modules should be tested, but even if a module passes the test, it should be tested in the enterprise system offline before being employed. That is, the modules should be tested as stand-alone and then, in conjunction with other applications, tested systemwide. Test data and results should be kept, and end users should be involved in the testing.

Figure 1 does not include this phase, but clearly the test results should be documented. The IT auditor will want to gain some assurance that proper testing of applications and systems has occurred before they are being put into operations.

Phase Seven: Systems Implementation

At this point, the system should be ready to deploy. The last step before deployment is a user acceptance sign-off. No system should be deployed without this acceptance. The user acceptance report should be included in the documentation of this phase.

After deployment, however, the SDLC processes are not finished. One key step after implementation is to conduct a postimplementation review. This reviews the cost-benefit report, traces actual costs and benefits, and sees how accurate the projections were and if the project is able to produce an adequate return. The systems design is also reviewed and compared to the performance of the system to see if the information requirements processes (phases two and three) were performed adequately. In general, the time, costs and requirements are the three main elements of any IT project, and those elements should be benchmarked somehow.

This step also reviews all of the system documentation to determine if it is adequate for the next phase: maintenance. If it is developed properly and according to SDLC best practices, it will be adequate.

At a minimum, a user acceptance report and a postimplementation report should be documented during this phase.

Phase Eight: Systems Maintenance

IT professionals and IT auditors know that 80 percent of the costs and time spent on a software system, over its life cycle, occur after implementation. It is precisely for this reason that all of the previously mentioned SDLC documentation should be required. Obviously, the entity can leverage the 80 percent cost by providing excellent documentation. That is the place for the largest cost savings over the life of the system. It is also the argument against cutting corners during development by not documenting steps and the system.

As changes occur, there should be change authorizations, change implementation and testing documents created during those changes. Testing during the maintenance phase should be able to use most of the original test data and test results, significantly reducing the time and effort necessary to adequately test the changes.

Conclusion

Employing the best practices of SDLC is not just a good idea in the IT industry; it serves as a control over systems development for IT auditors and provides documentation that the IT auditor can use to gain assurance over the adequacy and effectiveness of the entity's SDLC procedures.

IT auditors are able to verify that SDLC best practices are operating effectively by examining documentation that should have been created during the various phases. Of course, IT auditors would use other means of verification, such as inquiry and checklists, but the presence of proper SDLC documentation illustrates the level of application of the best practices in SDLC. A review of a sample of the documents will provide evidence that the entity is using SDLC best practices, which provides some assurance that systems are being developed efficiently and effectively.

Endnotes

[1] The majority of the content from this article is taken from: Hall, James; Tommie Singleton; *IT Audit and Assurance, 2nd Edition*, Thomson-Southwestern Publishing, 2005.

Tommie W. Singleton, Ph.D., CISA, CMA, CPA, CITP
is an assistant professor of information systems at the University of Alabama at Birmingham (USA), Marshall IS Scholar, and director of the Forensic Accounting Program. Prior to obtaining his doctorate in accountancy from the University of Mississippi (USA) in 1995, Singleton was president of a small, value-added dealer of accounting information systems using microcomputers. In 1999, the Alabama Society of CPAs awarded Singleton the 1998-1999 Innovative User of Technology Award. Singleton is the ISACA academic advocate at the University of Alabama at Birmingham. His publications on fraud, IT/IS, IT auditing and IT governance have appeared in numerous journals, including the *Information Systems Control Journal*.

Information Systems Control Journal, formerly the IS Audit & Control Journal, is published by the ISACA. Membership in the association, a voluntary organization of persons interested in information systems (IS) auditing, control and security, entitles one to receive an annual subscription to the Information Systems Control Journal.