

EVALUATING THE SYSTEMS DEVELOPMENT LIFE CYCLE

Dr. Roy A. Boggs, Florida Gulf Coast University, Rboggs@fgcu.edu
David DeMoya, Florida Gulf Coast University, ddemoya@swfla.rr.com

ABSTRACT

The Systems Development Life Cycle is represented by an increasing number of models, from the Historical-, Incremental-, V-, W-, Spiral-, and Unified models to the Ethics model. Each model has its particular strengths and weaknesses, a set of characteristics that distinguishes it from other models. The following looks at these characteristics and proposes a process for using them to better understand the various SDLC models.

Keywords: Systems Development Life Cycle, SDLC, Systems Analysis and Design

INTRODUCTION

The discovery, design, development and delivery of information systems are often linked together in a process labeled the Systems Development Life Cycle (SDLC). While many other terms and taxonomies are used, the SDLC process is usually iterative and involves distinct steps leading to a deliverable product. Almost every (M)IS student has studied an SDLC model, especially the well known Waterfall model. Other models exist, with names such as the Historical Model, the Incremental Model, The V-Model, the W-Model, the Spiral Model, the Rational Unified Model and the Ethics Model. New ones can be expected.

Which SDLC one prefers is often a matter of personal choice. Sometimes in both private and public agencies one or the other is proscribed, while some are found more in academic literature than in actual practice. Nevertheless, each SDLC has its strengths and weaknesses, and each may provide better functionalities in one situation than in another. SDLCs vary greatly, for example in their scope, end-user involvement, risk assessment, and quality control. There arises then the question of which provides what functionalities and under what expectations. How does one select one SDCL over another, assuming a choice is available?

Without providing a direct answer at this point to such questions, the following submits a set of characteristics against which to evaluate and select SDLCs. The goal is to establish criteria, which may in turn be used by students and practitioners to examine and evaluate SDLCs as they learn about them or examine them for possible use.

TWO SCENERIOS

Scenario One: A large governmental agency is considering a project that would enable client agencies to submit documents via the Internet. The project will involve very large numbers of sensitive documents and data protection is a primary consideration. The new system will be highly structured, risk sensitive, static in scope, and exhibit very limited end-user interaction. The project can be built incrementally and is time management sensitive.

Scenario Two: A large governmental agency is considering a project that would enable it to track the welfare of children assigned to its client agencies. The client agencies will need to interact on a continual basis from portable computers, creating/retrieving, updating and submitting sensitive data. The project must provide for heavy end-user involvement, must be adaptive to constant change while being developed, should provide for possible outsourcing for development, and is not sensitive to time-to-market pressures.

WHICH SDLC?

Given the two actual scenarios above (both are from public agencies, one federal and one state), it becomes clear that one or the other SDLC might not suffice for both scenarios. One has only limited end-user interaction and the other is heavily end-user oriented. One is highly structured; the other must be adaptive. One is to be completed in-house; the other must provide for possible outsourcing. And one is time management sensitive; the other is not sensitive to time-to-market considerations. Both are data sensitive.

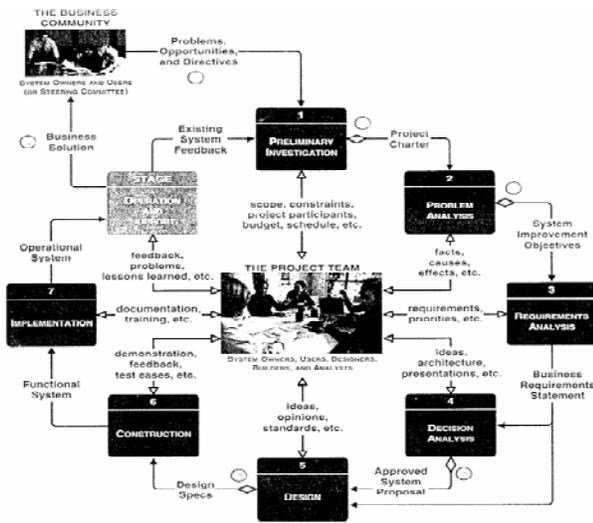
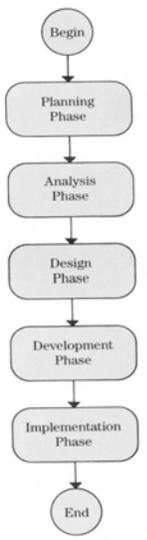


Table I

Without implying that one or the other of the two SDLC models in Table I is the best or that one or the other might be selected for one of the scenarios above, it can be seen that they are quite different. The model on the left represents the traditional Waterfall model, proposed as early in the 1980s by Silver and Silver, and still in popular use. The model on the right is representative of the FAST System Development Phases (pg 85) of Whitten, Bently and Dittman (2001). For a highly structured project, with clear incremental segmentation, one might be better; while for significant end-user interaction the other might be better.

What would be helpful is a set of characteristics against which the SDLCs can be evaluated. As a starting point, the following SDLC characteristics are suggested. These are followed by a table for comparing various SDLCs. The ultimate goal is to arrive at a set of characteristics for both pedagogical and practitioner use. The application is then a better understanding of the various SDLCs that have been, or might be proposed.

SDLC CHARACTERISTICS

The proposed characteristics below are presented in alphabetical order. To do otherwise might suggest that one characteristic is more important than another. The characteristics are themselves neutral. They have been assembled from discussions, both pro and con, in the literature cited in the bibliography. The goal is to discover which combinations of characteristics are represented in which SDLC and to what degree.

First the characteristics.

Adaptive

The model has the ability to react to operational changes as the project is developed. Change orders are easily assimilated without undue project delay and cost increases.

Documented

The model's structure and functions are well document. They have been practitioner tested and results are readily available.

Incremental

Each step of the project is to be completed before another is begun. Project units are distinct and easily identifiable.

Iterative

Certain aspects of the project, for example risk management, are increasingly developed as the project moves toward completion. The project expands from a prototype, in which all aspects of the project are encapsulated.

Maintenance

Systems are dynamic and the model offers to ability to produce a final project that is inherently designed for maintenance. This includes such items as cumulative documentation.

Management Control

Management will have the ability to redirect and if necessary redefine the project once it is begun. A key phrase is 'incremental' management control, with each step under tight management control.

Outsourcing

The project can be designed so that pieces can be outsourced. This might include in-house systems analysis and design with development outsourced – under loose or tight specifications.

Quality Control

Each module of the project can be thoroughly tested before another module is begun. Project requirements are measured against actual results. Milestones and deliverables can be included for each step of the project.

Risk Management

Levels of risk are identifiable and assessment strategies available. Strategies are proved for over-all and unit risks.

Simple

The model is easy to understand and to implement. Even if the model can be expanded to various levels, a logical overview exists.

Time Management

Time-to-market considerations are central. Time-constraints are evident and a strongly structured project formulation is available.

End-User Involvement

The model lends itself to strong and constant end-user involvement. This includes project design as well as interaction during all phases of project development.

The twelve characteristics described above are certainly not all inclusive, but they do offer a solid structure for examining SDLCs for their functionality, strengths and weaknesses. They become ever more useful when combined in a table. One possible table is suggested below.

SDLC ASSESSMENTS

An assessment table, once completed and verified for relative merit, provides one tool for selecting one SDLC over the other, given the parameters of a proposed project. The statement being made here is that one SDLC does not fit all projects. An understanding of a SDLCs strengths and weaknesses is not only valuable, but it becomes useful for a wide range of pedagogical and practitioner related activities.

SDLC Model's Basic Characteristics Assesment Matrix

Assessment Scale	1	2	3	4	5			
	Very poor	Poor	Average	Good	Very Good			
<i>MODELS</i>								
<i>CATEGORY</i>	<i>Waterfall</i>	<i>Spiral</i>	<i>Incremental</i>	<i>V-Model</i>	<i>W-Model</i>	<i>Hist</i>	<i>RUP</i>	<i>Ethics</i>
Adaptive								
Documented								
Incremental								
Maintenance								
Mgmt. Control								
Outsourcing								
Quality Control								
Risk Mgmt.								
Simple								
Time Mgmt.								
Risk Assessment								
End-User								

SUMMARY: THE TWO SCENERIOS

The completed matrix will make an evaluation and selection of a suitable SDLC quickly and easily available. The first scenario given above would look for relevant evaluations in areas related to project structure, in-house development, time management and limited end-user involvement. The second scenario would look for high end-user involvement, easily adaptable, outsourcing possibilities, and with limited time-to-market considerations. There might be several models that would suffice. However, models with unacceptable limitations could be avoided.

There is also a strong pedagogical implication for this activity. There are currently many SDLC models available and they have differing capabilities. New ones are certain to be offered. The SDLC is one of the most important tools for information systems development and management. An overview is not only worthwhile but also important.

BIBLIOGRAPHY

1. Alter, S.A. (1978). Development patterns for decision support systems, MIS Quarterly (3), 33-42.
2. Haug, M., Olsen, E. W., Vallet, G., O. Becart, Editors, (2001). Software Management Approaches: Project Management, Estimation, and Life Cycle Support. Springer-Verlag Berlin Heidelberg.
3. Hoffer, Jeffrey A., George, Joey F., Joseph S Valacich (2002). Modern Systems Analysis & Design 3rd Edition, Prentice Hall.

4. Ikuta, Keiji. (2002). A List of Important Points of Software Development. [To Be Verified]
5. Janson, M.A. & L.D. Smith, (1985). Prototyping for Systems development: A Critical Appraisal. MIS Quarterly (4), 305-315.
6. Kruchten, Philippe (1999). The Rational Unified Process: an Introduction, Addison Wesley Longman, Inc.
7. Marakas, G.M. (1999). Systems Analysis and Design: An Active Approach. Upper Saddle River, NJ: Prentice Hall.
8. Martin, Robert C. Iterative and Incremental Development (IID). [To Be Verified]
9. Mumford, Enid & Mary Weir (1979). Computer Systems in Work Design – the ETHICS Method. Halstead Press, New York.
10. Satzinger, John W., Jackson, Robert B. and Stephen D. Burd (2002). Systems Analysis and Design in a Changing World, Second Edition, Course Technology.
11. Silver, Gerald, and Silver, Joan (1976). Introduction to Systems Analysis, Prentice-Hall.
12. Silver, Gerald, and Silver, Myrna (1989). Systems Analysis and Design, Addison-Wesley Publishing Company, Inc.
13. Weinberg, R.S. (1991). Prototyping and the Systems Development Life Cycle. Journal of Information Systems Management, (2), 47-53.
14. Whitten, Jeffrey L., Bentley, Lonnie D., and Kevin C. Dittmann. (2001). Systems Analysis and Design Methods 5th Edition. McGraw Hill.