Author: Timothy Trimble
Email: timothytrimble@me.com
Author site: www.timothytrimble.com
Developer site: www.timothytrimble.info
Document: SDLCGuidelines-002.doc, version 002

Software Development Life Cycle – Guidelines

Introduction

The process of designing and implementing software can be a challenge at both the individual and enterprise project level. Along with the fine art of communications between developers and customers, it is easy to understand why software projects are usually delayed, don't match customer expectations, or suffer from "feature creep". Which is why the Software Development Life Cycle, (SDLC, also called Systems Development Life Cycle) was originally developed.

There are many books and articles that describe the best methodologies for designing and developing software. This document is just a small sampling of one of those methodologies. It is not intended to be "written in concrete" and the only process. However, it is strongly recommended that for any significant software development project, that some type of "best practices" methodology be utilized. If you want to learn more do a Google search or Amazon.com search on "SDLC".

The following outline is a guide for the standard SDLC process, utilized in various forms by most software development organizations and educational institutions. This document is intended to be a "living" document and is thus to be modified and adjusted to fit the needs of the developer or consultant. Feel free to email me your comments and recommendations.

1.0 – First Contact

Upon first contact with the intended client, a problem is presented to the developer, in which the client is seeking a solution. This is actually where the very first steps of analysis begins, by determining if the developer has the expertise, resources, and willingness to accept the project presented by the client, and if the client is willing to work with the developer for defining the project requirements.

2.0 – Requirements Analysis

If the project is accepted, then the next step is to begin the Requirements Analysis phase of the project. This task is usually assigned to a Project Manager or a Software/Systems Analyst. Within this phase are the following steps:

2.1 – Client Interviews

The client is interviewed by the analyst for the purpose of defining the actual requirements.

2.2 – Business Analysis

The analyst examines the business model of the client for gaining an understanding of how the business functions, and how the requirements fit the need for benefiting the business.

2.3 – Systems Analysis

If there are existing software systems in place, which are currently providing related business processes, then the job of the analyst is to examine the current systems and to determine what can be utilized from the existing systems (data, routines, source, etc.). The analyst will also examine why the current systems are not meeting the needs of the client, how the systems fit (or do not fit) into the clients business processes, and how anticipated changes in the business process could affect the design of future systems.

2.3 – Requirements Definitions Document

In many occasions, the client does not have a Requirements Document prepared. And usually, the client does not have a clear picture as to what is needed. In these situations, it is good to present a Requirements Definitions Document to the client after the Analysis process has been completed. This is a way to show the client that we have a full understanding of their requirements. If the client approves of the contents of the document, then it can be used as a foundation for a Specifications Document.

3.0 – Specifications

Once the Analysis process has been completed then the Specifications for the project can be defined. The following two deliverables are key to presenting the fact that we have a clear understanding of the requirements, and we have the capability to deliver a solution:

3.1 – Rapid Prototype

As a part of the Specifications process, it is usually a good idea to include a Rapid Prototype of the proposed solution. This can be as simple as a few screens to represent the proposed User Interface, or it can even include some basic functionality.

3.2 – Specifications Document

The Specifications Document is key for conveying the features, functionality, and UI of the proposed project. A REQUIRED step in the SDLC, the Specs has to be reviewed and signed off by the client, management for the vendor (us), and in some cases a representative of the development team. This prevents feature creep and confirms that the client and the development team both understand what the

deliverables are to be. The Specifications Document should contain the following:

- **Document Description** – A description of what the document contains.
- **Document Revisions** – If it is expected that there will be some changes to the document by multiple individuals and that the project is fairly complex then it would be a good idea to include a table that shows the Revision #, Date, Type of Revision, Author, and Description of the revision.
- **Table of Contents**
- **Introduction/Background** – Brief description of the project and why it is needed.
- **System Requirements** – This should contain a description of the target platform(s) for the project, and a description of the development environment that will be utilized for the project.
- **Installation** – A description of how the proposed completed application is to be installed.
- **User Interface Definitions** – Define any standards or guidelines that are to be followed for the general UI objects. If specific screen layouts need to be defined, they can be defined here or individually under the Target Functionality area.
- **Target Functionality** – Define each function and feature point of the application, grouped by main areas (or modules) of functionality.
- **Database Definitions** – Define the structure of the application database. Include a reference to an ERD if possible.
- **Rapid Prototype** – Describe the Rapid Prototype application, how to use it, and what areas of the specifications it addresses. IMPORTANT: With the use of a Rapid Prototype, be sure to educate the client that this does not represent a significant amount of work toward the completion of the project. It is only a demonstration and may possibly be throw away code.
- **Specifications Approval** – This section provides a signature area for the approval of the specifications as defined.
- **Addendum** – If there are any Change Requests submitted then they are to be attached as addendum to the specifications.

Each feature of the Specifications Document should be numbered and sub-numbered. This numbering scheme will be utilized as a reference number for the Project Plan, Test Plan, and any other documents that need to reference specific tasks within the project. A small sample of this scheme follows:

1.0 – System Requirements
    1.1 – Client Platform
    1.2 – Development Platform
2.0 – Installation
3.0 – User Interface
    3.1 – Button Objects
        3.1.1 – Navigation Buttons
        3.1.2 – Data Management Buttons

4.0 – Development

This phase of the SDLC is where the actual development processes go. As each area of functionality is implemented, it should be flagged by the project manager as such, and thus be handed off to an Alpha tester. Usually Alpha testing can begin at around 50% completion of the development process, depending on how the application is designed. In some cases, Alpha testing cannot begin until the application is Alpha Complete.

4.1 – Data Conversions

If there is to be any data converted from legacy systems, then it should be done as a part of the development process. True Beta Testing cannot begin until the data conversion process has been completed.

5.0 – Alpha Testing

During the development phase, Alpha testing can begin. It is best to have a issue tracking system in place for reporting issues and tracking their resolution.

5.1 – Test Plan

For a successful test process, it is highly recommended that a test plan be defined. This test plan is based on all the features defined in the specifications, by function number. This test plan should be followed by both internal and client testers during the Alpha and Beta test phases of the project.

6.0 – Alpha Complete

This is a milestone marker. When all of the defined functionality is in place then the application can be flagged as Alpha Complete. Once this stage is reached, then a freeze should be made on the inclusion of additional functionality via Change Requests.

7.0 – Beta Testing

Once Alpha Complete has been reached, then the application is ready for Beta Testing by both the internal staff, and by designated client staff. Again, an issue tracking system should be in place – one that is easily accessible by both the internal staff and the client. During the testing process, issues should be classified into the following categories:

- **Category 1** – These are issues that cause a severe crash (CTD: Crash To Desktop), an application freeze, or corruption/loss of data.
- **Category 2** – These issues result in undesired functionality or application errors.
- **Category 3** – These issues should be suggestions, minor UI changes, or issues that do not affect the data or functionality of the application.

8.0 – Beta Complete

This is another milestone marker. Once the application has been deemed as free of Category 1, and/or Category 2 issues, then it can be flagged as Beta Complete. Some

organizations will release upon reaching Cat 1 Free, with the intent of resolving Cat 2 issues with an impending patch release.

9.0 – Documentation

During the Beta Testing process, it is a good time to begin the documentation process. Initially, the User Guide and associated on-line help should be developed. Depending on the complexity of the application and the use of source commenting, the writing of internal technical documentation might also be desired.

10.0 – Release Candidate Review

This is a milestone marker. Representatives from the vendor and from the client should meet and decide if the application has reached the point of being a Release Candidate. Between this milestone and Production Implementation, the Release Candidate is usually released to a wider audience and a final set of stress tests are gone through.

11.0 – Production Implementation

This very critical phase can make or break a project. Proper planning needs to be done and a lot of coordination is required to ensure that the implementation of the application goes smoothly. Sub-phases for this process include the following:

11.1 – Technical Training

Training of the client's technical support team, for understanding the installation and configuration process, and for being able to act as a first level support and interface to the vendor.

11.2 – User Training

If you don't win the users, then the application will not be successful. Make sure that the users are fully trained and comfortable with the application. Get them excited about the implementation process, and maybe even recruit some of them for assistance with the implementation.

11.3 – System Cut-Over

This is the process of turning the switch off of one system and turning it on for the new system. Data might need to be converted/migrated as a part of this process, using tools and processes that have already been tested and approved during the Beta Test phase.

11.4 – Post Implementation Support Procedures

It is important that the process of supporting the application be clearly defined before the application is accepted by the client. Various support plans can be offered, as well as the initiation of the design and development process for future enhancements to the application.

12.0 – Application Acceptance

This final milestone marks the successful completion of the project. A major meeting with the client's management should be held and a sign-off on the completed project should be formally done. The client might also be recruited for being a spokesperson for a case study or for marketing purposes.

This is the end of the SDLC Guidelines document. Again, feel free to contact me if you have any comments or questions.

Timothy Trimble
Writer and Technologist
Author: http://www.timothytrimble.com
Developer: http://www.timothytrimble.info