AU4426_C016.fm Page 209 Tuesday, January 23, 2007 3:58 PM

Chapter 16

Introduction to the Traditional SDLC

As is expected of any profession that is still relatively young, IT has evolved — and is still continuing to evolve — from highly individual seat-of-the-pants techniques for developing and maintaining systems to formal, well-documented methodologies. In the early days, when what is now called information technology (IT) was referred to as data processing, there were no methodologies or formal guidelines for developing systems. Systems were developed under what IT now knows was the mistaken belief that their life span would never exceed five years, and thus long-term maintainability was not considered a major concern.

Data processing was an art rather than a science, with no two systems being developed in the same way. As a result, it was difficult to predict the length of a project, its cost, and the degree to which it would solve the problem that had initiated it.

If the process was an art, the practitioners were definitely artists. Two separate departments frequently performed systems analysis and programming with only minimal communication between them. The systems analysts gathered customer requirements and gave them to the programming staff, who then worked their black magic to translate those requirements into computer systems. Because there were few common processes and because programmers were frequently unwilling to share the secrets of their success, individual programmers were viewed as creative beings who were essential to the continued running of the systems they had

AU4426_C016.fm Page 210 Tuesday, January 23, 2007 3:58 PM

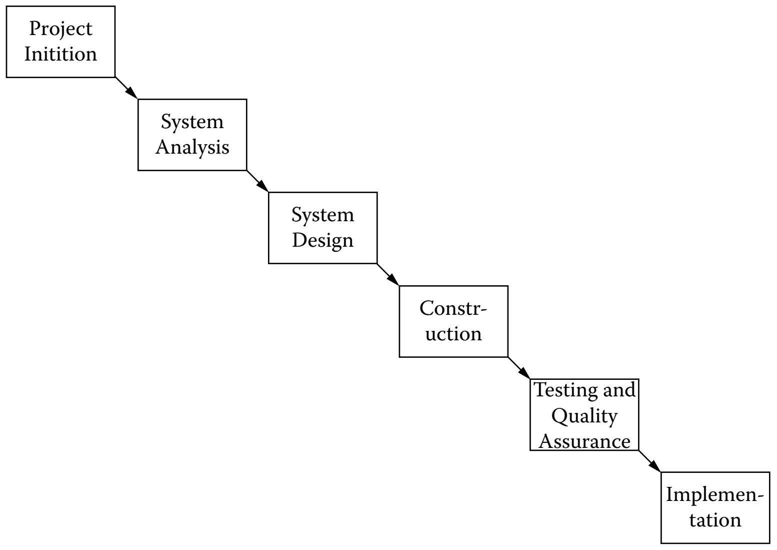
developed. In SEI CMM terms, the industry was at Level 1. (Exhibit 2.1 outlines the five levels of the CMM maturity path.)

As business applications expanded beyond the Finance department, it became obvious that there must a better way to manage data processing and, in particular, a better way to develop systems. The result was the creation of formal methodologies centered around what was commonly referred to as a system development life cycle (SDLC). The objective of these methodologies was to document and institutionalize the best practices of system development. Although Six Sigma had not become part of the business vocabulary, the goal of an SDLC was one of the Six Sigma tenets: the desire to reduce variation.

In its simplest form, an SDLC divides the software development process into a number of clearly defined phases, each of which is further divided into steps. Progress through the steps is measured by the completion of forms and checklists. Because the phases were viewed as sequential steps, with the output from one phase becoming the input to the next, a traditional SDLC was often called “a waterfall.” And, like water flowing over a precipice, the underlying premise of the waterfall approach to system development was that all motion was forward. Once a phase was completed, there was no returning to it. Exhibit 16.1 shows the phases in a typical traditional SDLC.

Although a number of industry experts and major consulting firms developed their own versions of the methodology, each with forms for every step of the process, there were many common elements. The forms

Exhibit 16.1 Traditional SDLC (The Waterfall)



AU4426_C016.fm Page 211 Tuesday, January 23, 2007 3:58 PM

varied; the philosophy did not. Use of an SDLC, the proponents claimed, would ensure that system development followed a common, sequential process with all critical information being properly documented.

Although there were shortcomings, with the development of a life cycle, the data processing industry had made a major step in transforming itself from seat-of-the-pants programming to software engineering. SDLC can be viewed as the foundation of the modern IT department. In SEI terms, it was an attempt to reach Level 2 and have repeatable processes.

Advantages of the Traditional SDLC

Although sometimes criticized for its rigidity, a traditional SDLC provided and continues to provide benefits for many organizations. In addition to the reason it was initiated — namely, adding structure to a previously unstructured process — the waterfall approach to system development has two primary advantages:

1. The explicit guidelines allow the use of less-experienced staff for system development, as all steps are clearly outlined. Even junior staff members who have never managed a project can follow the "recipes" in the SDLC "cookbook" to produce adequate systems. Reliance on individual expertise is reduced. Use of an SDLC can have the added benefit of providing training for junior staff, again because the sequence of steps and the tasks to be performed in each step are clearly defined.
2. The methodology promotes consistency among projects, which can reduce the cost of ongoing support and allow staff to be transferred from one project to another. Although coding techniques are not specified in a typical SDLC, the extensive documentation that is an inherent part of most methodologies simplifies ongoing maintenance by reducing reliance on the original developers for explanations of why the system was constructed as it was and which functions are included in which program modules.

SDLC Disadvantages

While there is no doubt that the waterfall approach to system development is superior to a totally unstructured environment, there are some known disadvantages:

1. If followed slavishly, it can result in the generation of unnecessary documents. Many methodologies have forms for every possible

AU4426_C016.fm Page 212 Tuesday, January 23, 2007 3:58 PM

scenario. Inexperienced staff may believe that all are required and may, for example, insist on three levels of customer sign-off when only one is needed. This can have the effect of complicating the process and extending the project schedule unnecessarily. To prevent this from occurring, most organizations view an SDLC as a set of guidelines and use only the steps and forms that apply to the specific size and type of project under development. Many even provide templates for their staff, outlining which steps are required for specific types and sizes of projects. An 18-month mainframe development project might require different processes than a two-week Web front end.

2. It is difficult for the customer to identify all requirements early in the project; however, the sequential "river of no return" approach dictates this. The philosophy of the SDLC means that there are no easy ways to mitigate this problem and still remain true to the methodology.
3. The customer is involved only periodically, rather than being an active participant throughout the project. This can result in misunderstandings on both sides: IT and the customer.
4. As a corollary to the previous two points, the waterfall approach is usually applied to large projects with long development cycles. The combination of incomplete specifications, infrequent communication, and long elapsed time increases the probability that the system will be "off track" when it is finally delivered. As highway engineers know, an error of only a few degrees, if left uncorrected for thousands of miles, will result in the road going to the wrong destination.
5. Similarly, the long development cycle increases the possibility that by the time the system is delivered, business changes may have invalidated the initial design or that the project champions may have left the company or been reassigned, taking with them the impetus for the project.

How Six Sigma Can Help

Just as Six Sigma helps companies eliminate defects, reduce costs, and improve customer satisfaction in their manufacturing processes, it can increase the effectiveness of the traditional system development process. While not all Six Sigma tools are applicable to all projects, the judicious use of some tools and strict adherence to the concepts of customer focus and fact-based decisions can increase the probability of successful implementation of waterfall projects by:

AU4426_C016.fm Page 213 Tuesday, January 23, 2007 3:58 PM

- Keeping the customer involved throughout the process
- Identifying a manageable scope for the project
- Ensuring continued commitment to the project, even if key players leave the organization

Just as the definition phase of the DMAIC model seeks to identify customers and to understand the current process before making any changes, the first step in applying Six Sigma to traditional system development is to understand what constitutes the waterfall approach and who is typically involved in each phase.

Although phases and their names may vary between different SDLC methodologies, this book describes a six-phase life cycle:

1. Project Initiation
2. System Analysis
3. System Design
4. Construction
5. Testing and Quality Assurance
6. Implementation

Each of these phases is described in detail in subsequent chapters.

The concepts discussed in this section can be applied to all traditional life cycles, regardless of the names given to the phases or the individual steps included in each phase.

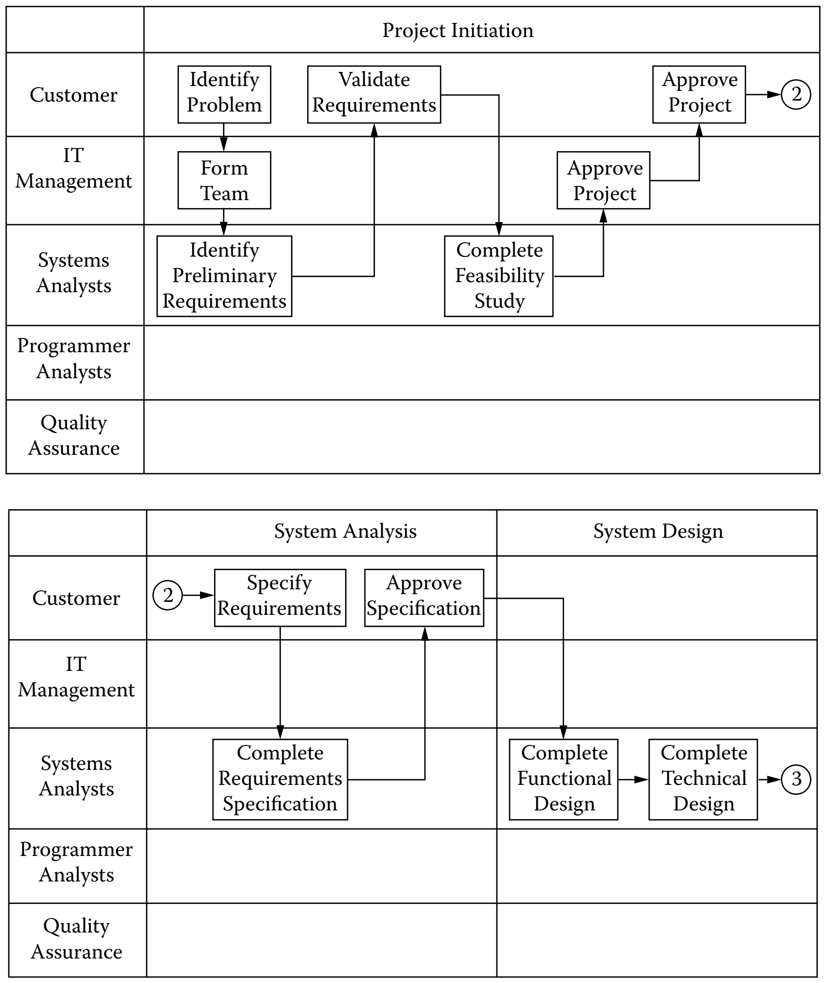
Before a process can be improved, a Six Sigma company knows that it is important to understand which departments and functions are normally involved in that process so that workflow and dependencies are clearly delineated. As the GWC Order Entry team learned, a functional process map provides a pictorial outline of major steps and responsibilities. Exhibit 16.2 presents a functional process map for the traditional life cycle, showing the departments that are normally involved in each phase of the life cycle. Although some individuals may perform multiple roles (e.g., systems analyst and programmer analyst) because the functions differ, they are shown as separate rows on the process map.

A review of the process map confirms the third disadvantage of the waterfall approach, the fact that customers are involved only periodically. This presents the first challenge of adapting Six Sigma concepts to the traditional SDLC — that of maintaining a focus on customers.

Exhibit 16.3 shows how the Six Sigma DMAIC phases map to the traditional SDLC. It should be noted that although the waterfall SDLC is considered a sequential process, its first two phases are a combination of Define, Measure, and Analyze, with functions being repeated at different levels of detail in later phases.

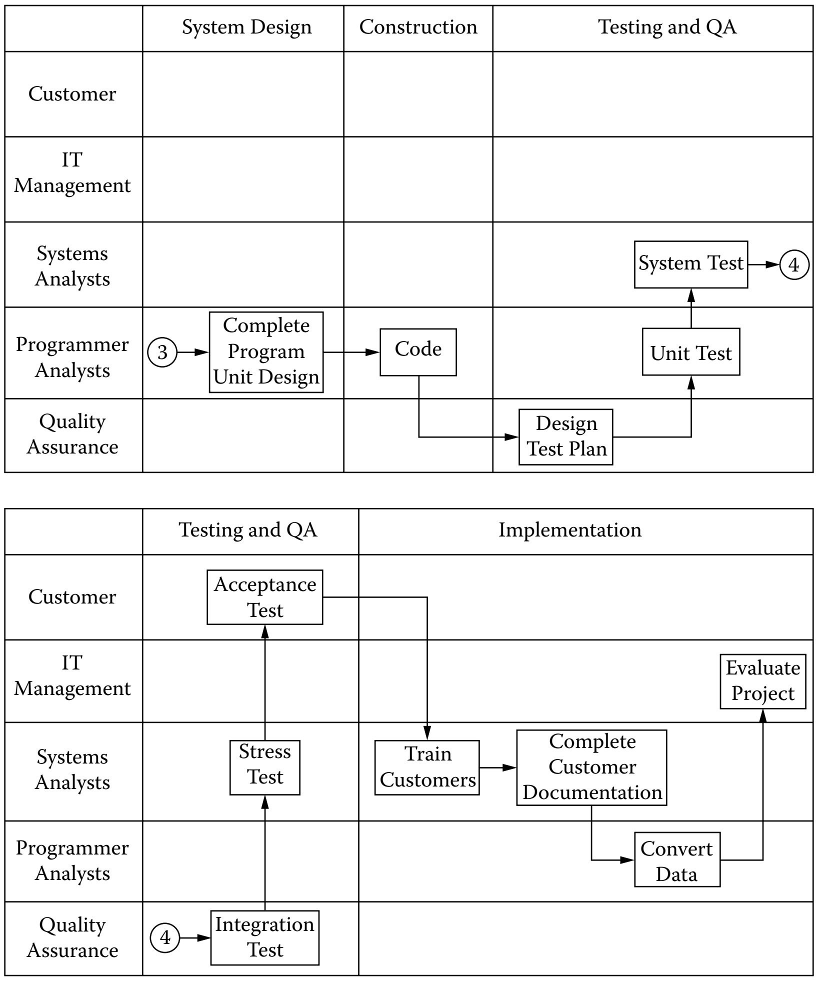
AU4426_C016.fm Page 214 Tuesday, January 23, 2007 3:58 PM

Exhibit 16.2 Traditional SDLC Functional Process Map



AU4426_C016.fm Page 215 Tuesday, January 23, 2007 3:58 PM

Exhibit 16.3 Traditional SDLC Functional Process Map (continued)



AU4426_C016.fm Page 216 Tuesday, January 23, 2007 3:58 PM

Exhibit 16.4 Mapping Traditional SDLC Phases to Six Sigma's DMAIC

SDLC Phase	*Six Sigma (DMAIC)*
Project Initiation	Define, Measure, Analyze
System Analysis	Define, Measure, Analyze
System Design	Analyze
Construction	Improve
Testing and Quality Assurance	Improve
Implementation	Improve and Control

Exhibit 16.5 Mapping Traditional SDLC Phases to DFSS's IDDOV

SDLC Phase	*DFSS (IDDOV)*
Project Initiation	Identification, Definition
System Analysis	Definition, Development
System Design	Development
Construction	Development
Testing and Quality Assurance	Verification
Implementation	Verification

The Role of DFSS

As discussed in Chapter 10, although classic Six Sigma can be employed at any stage of the life cycle, the greatest benefits of DFSS are achieved if the techniques are applied to new products or processes. The traditional SDLC is an excellent candidate for DFSS improvement. Judicious use of several DFSS tools, notably the value chain map and QFD, can help mitigate the disadvantages of the waterfall SDLC by more clearly identifying customers and their requirements and ensuring that key requirements, the customers' CTQs, are satisfied by the resulting system.

It can also be argued that the traditional SDLC phases have a more exact correspondence to the DFSS phases than they do to DMAIC. Exhibit 16.4 provides a map of the waterfall phases to IDDOV. That increased correlation is logical because both methodologies are designed to be employed from the earliest stages of a product or system's creation, rather than being applied after the product or system has been implemented. It is, however, significant that no phase of the traditional SDLC corresponds to DFSS's Optimization. This is one of the shortcomings of the waterfall approach to system development.

AU4426_C016.fm Page 217 Tuesday, January 23, 2007 3:58 PM

The remainder of Section IV outlines each of the six system development life-cycle phases, showing the tasks normally completed in each and how the use of Six Sigma and DFSS can contribute to a more successful project.

AU4426_C016.fm Page 218 Tuesday, January 23, 2007 3:58 PM