

## 2

# System Development Life Cycle (SDLC)

## System Development Life Cycle—Steps in Analysis and Design

The purpose of this chapter is to build on the Tiers of Software Development and to provide a framework for the life cycle of most software development projects. This is important prior to explaining the details of the user interface and analysis tools that are needed to bring software to fruition. Another way of viewing this chapter then is to get a sense of how the tiers of development actually interface with each other and what specific events and tools are used to successfully complete each step. This chapter consists of two sections: the first explains the notion that software goes through three basic phases or cycles, that is, Development, Testing, and Production. The second section provides an example using a seven-stage method called “The Barker Method,” which represents one approach to defining the details of each of the three cycles.

No matter which methodology might be used when designing a system including its related database, the key elements involved in the methodology usually include, at a minimum, business process reengineering and the life cycle for design and implementation. Business process reengineering (BPR), simply defined, is the process used for either reworking an existing application or database to make improvements, or to account for new business requirements. BPR will be discussed in more detail in Chapter 13. The life cycle of the database includes all steps (and environments) necessary to assist in the database’s design and final implementation and its integration with application programs. Irrespective of which design methodology is used, analysts/designers will find that system development projects will usually include the following generic steps:

1. Determine the need for a system to assist a business process
2. Define that system’s goals
3. Gather business requirements
4. Convert business requirements to system requirements
5. Design the database and accompanying applications
6. Build, test, and implement the database and applications

This *traditional method* is the most commonly used design approach and includes at least three primary phases:

1. Requirements analysis
2. Data modeling
3. Normalization

During the first phase, requirements analysis, the development and design team conduct interviews in order to capture all the business needs as related to the proposed system. The data modeling phase consists of the creation of the logical data model that will later be used to define the physical data model, or database structures. After the database has been modeled and designed, the normalization phase is implemented to help eliminate or reduce as much as possible any redundant data. All of the specifics of how this is accomplished will be detailed in the tools of analysis chapters. Below is a more specific description of what activities are included in the Development, Testing, and Production cycles of the SDLC.

## Development

The Development life cycle includes four overall components. Using this perspective, “development” would consist of all the necessary steps to accomplish the creation of the application. This includes feasibility, analysis, design, and the actual coding. Feasibility represents the tasks necessary to determine whether the software project makes business sense. Most organizations would integrate the process of Return-On-Investment (ROI) during this step. ROI consists of the financial steps that determine mathematically whether the project will provide the necessary monetary returns to the business. Focusing solely on monetary returns can be a serious pitfall, since there are many benefits that can be realized via non-monetary returns (Langer, 2005). Feasibility often contains what is known as a high-level forecast or budget. The “high” would represent the “worst case” scenario on cost and the “low,” the best case or lowest cost. The hope of course is that the actual cost and timetable would fall somewhere in between the high and the low. But feasibility goes beyond just the budget; it also represents whether the business feels that the project is attainable within a specific timetable as well. So, feasibility is a statement of both financial and business objectives, and an overall belief that the cost is worth the payback.

Analysis is the ultimate step of creating the detailed logical requirements, or as I will define in Chapter 4, the architecture of the applications and database. As we will see in this book, there are numerous analysis tools that are used along each phase of analysis. Ultimately, the analyst creates a requirements document that outlines all of the needs for the coders to work from, without going back to the users directly for clarification. Analysis, as an architectural responsibility is very much based on a mathematical progression of predictable steps. These steps are quite iterative in nature, which requires practitioners to understand the gradual nature of completion of this vital step in Development. Another aspect