Chapter 6 – Foundations for Systems Design

Table of Contents

- Chapter Overview
- Learning Objectives
- Notes on Opening Case and EOC Cases
- Instructor's Notes (for each section)
  - Key Terms
  - Lecture notes
  - Quick quizzes
- Classroom Activities
- Troubleshooting Tips
- Discussion Questions

Chapter Overview

This chapter is the first chapter in Part Three of the textbook. The third part of the book moves from the discussion of analysis concepts to design concepts.

There are three major themes in this chapter. The first major theme explains systems design and how it fits into a development project. The second major theme is a description of the six activities of Systems Design. Finally this chapter discusses issues of system controls and security.

The first section defines design as distinct from analysis. The objective of systems design is to define, organize, and structure the components of the final solution system that will serve as the blueprint for construction. There are various components that need to be designed, including items such as the application software, the database, the user interface, the network, interfaces to external systems, and internal controls.

The next major theme the six activities that support Core Process 4, Design the application. These six activities are: design the environment, design application architecture and software, design user interfaces, design system interfaces, design the database, and design system controls and security.

The final major component in this chapter is a discussion of the principles and issues involved in designing the system controls and addressing security issues. This topic is included as the first design activity because it must be considered during all other design activities.

Learning Objectives

After reading this chapter, the student should be able to:

- Describe systems design and contrast it with systems analysis
- List documents and models used as inputs to or output from systems design
- Explain each major design activity
- Describe security methods and controls

Notes on Opening Case and EOC Cases

Opening Case

Security and Controls at New Mexico Health Systems: This case highlights the importance of having systems that are "hardened" against malicious attacks. In the case, an outside consultant gives a brief overview of the effectiveness of existing security and controls. In this company (NMHS) the existing systems appear to have secure systems. However, a new system is under development, which provides a customer portal that has access to internal database. Obviously, this external portal high risk and considerable danger to the security of the internal data. Although specifics are not provided in the case, it does emphasize the need to include controls and consider security needs during all phases of the development project.

EOC Cases

County Sheriff Mobile System for Communications (CSMSC): Police departments in today's environment have a very high need to be able to communicate in real time with centralized databases and dispatch offices. Plus since all of this information is personal and confidential this communication must be secure. County sheriff departments typically have tough situations. Most counties do not have the amount of budget as state high patrol agencies, but they often have to cover large areas with many different types of highways and terrain. In this case, a typical county sheriff department has a particular need to provide real time communication and data access between the mobile units and the central office. Students are asked to consider what specific controls that might be needed for radio, cellular, and satellite transmissions to ensure that the data transmitted is secure and only received by authorized law enforcement personnel. This helps students think about specific security issues.

Community Board of Realtors (running case)**:** Community Board of Realtors is a professional organization that supports real estate offices and agents. In this chapter's case, the students are asked to list specific design tasks that correspond to the five activities of Core Process 4, Design system components. This exercise helps students think about the elements of design from an overview perspective.

Spring Breaks 'R' Us Travel Services (SBRU) (running case)**:** SBRU is an online travel services that books spring break trips to resorts for college students. The SBRU system has three Web subsystems that use normal Web pages. The fourth subsystem is a social networking subsystem that supports online chatting. The students are asked to consider system integrity in this Internet and social networking

environment, including what encryption should be considered.

On the Spot Courier Services (running case)**:** On the Spot is a small, but growing, courier service that needs to track customers, package pickups, package deliveries, and delivery routes. In this chapter the processing requirements for the customers, the home office, and the delivery drivers is reviewed. Customers basically have standard Web page access. The home office has access and update capability to all customer, driver, schedules, and package information. The drivers have real time access and update information about package pickup and delivery. The students are asked to consider issues of fraud by employees. Also to consider what kinds of access controls should be implemented. And finally to research how to set up a digital certificate.

Sandia Medical Devices (running case)**:** Sandia Medical Devices is a company that specializes in medical monitoring through remote, mobile telecommunication devices. As described in previous chapters, the Real-Time Glucose Monitoring (RTGM) system will include processing components on servers and on mobile devices, such as smartphones, with data exchange via 3G and 4G phone networks. Users will include such patient and health care personnel as physicians, nurses, and physician assistants. In the United States, the Health Insurance Portability and Accountability Act of 1996 (HIPAA) mandates certain responsibilities regarding the privacy and security of electronic protected health information (ePHI). The students are asked to questions about HIPPA applicability to the Sandia Medical Devices system. Questions deal with the privacy of information on mobile devices that are outside of a secure environment as well as security within a third-party host for the servers.

Instructor's Notes

What is Systems Design

Key Terms

- **none**

Lecture Notes

Design is an abstract concept that has different meanings depending on the context. Design also can include many different activities. Depending on the size and scope of the project and the product being designed, the design activities may be simple, one-person activities or the design activities may be complex involving many people, models, and resources.

Analysis, Design, and Implementation

Design is defined in Chapter 1 as “those system development activities that enable a person to describe in detail how the resulting information system will actually be implemented.” Design cannot be done unless accurate requirements are documented during analysis activities. In other words, design depends on the information that is extracted during analysis. On the other hand, implementation cannot occur without the design. Hence design is the first step to move into implementation. In fact, design acts as a

bridge between analysis and implementation. Figure 6-1 illustrates these relationships.

Design Models

During iterative development, analysis, design and implementation sequence is repeated many times as each part of the system is iterative developed. Analysis is a model building activity to capture the requirements. Design is also a model building activity, however, the objective is to represent the solution system. In other words, design models represent the final solution that must be implemented.

Some projects, are large and complex. Others are small and informal. The extent to which formal design models are built depends on the type, complexity, size, and formality of the project and the development method.

Figure 6-2 identifies the primary object-oriented models that are included in the Unified Modeling Language (UML), and which also are the ones taught in this textbook. By the end of the course, the students should understand and be proficient in both understanding and in developing all of these models.

Quick Quiz

Q: What is the primary design objective during systems development?

A: Describe in detail how the new system will be implemented.

Q: What are the major analysis models that you have learned in the first five chapters of the text?

A: Domain class diagram, use case diagram, activity diagram, SSD, State Machine

Q: What are the major design models that will be taught?

A: Design class diagram, Interaction diagram, State Machine, Package diagram, Component diagram, Deployment diagram

Q: Which design models are dependent on which analysis models?

A: Domain class → Design class, Use case, activity, SSD → Interaction Diagram, State Machine → State Machine.

Design Activities

Key terms

Application component: a well-defined unit of software that performs one or more specific tasks

Lecture Notes

As shown in Figure 6-3 there are five identified activities that comprise Core Process 4, Design System Components. The five activities are:

1. Describe the environment.

2. Design the application components.
3. Design user interface.
4. Design the database.
5. Design the software classes and methods.

An alternative way to think about these five activities is with a design question, either a how-to question or a what-is question. The following questions correspond to the activities.

1. How will this system interact with other systems and with the organization's existing technologies?
2. What are the key parts of the information system and how will they interact when the system is deployed?
3. How will users interact with the information system?
4. How will data be captured, structured, and stored for later use by the information system?
5. What internal structure for each application component will ensure efficient construction, rapid deployment, and reliable operation?

Describe the Environment

Most new information systems must fit into an already existing technology environment. Hence this activity is more focused towards defining and understand the environment rather than designing the technology infrastructure from scratch. Essentially there are two key elements that must be defined,

- External systems, meaning other systems that the new system must interface with, and
- Technology architecture, meaning the complete environment of hardware and software that supports the new system.

Design the Application Components

As defined above an application components is a well-defined unit of software. Therefore designing the components refers to defining the set of components that will be needed and how they all interconnect together. Many new systems consist of pre-packaged components and in-house programmed components. These components can range from small special purpose modules, such as one to calculate sales tax or to generate a GUID, to complete subsystems, such as a payment processing subsystem. Decisions must be made to define the components, decide whether to buy or build each component, what programming language is required, etc.

Various models are used to design the application components, including package diagrams, deployment diagrams, and component diagrams. Examples are given in Figure 6-5.

Design the User Interface

This is one of the most critical and yet one of the most difficult design activities. It is critical because, to the users, the interface is the system. That is all they see and interact with. A poorly designed user

interface causes a poorly used and respected system. It is difficult because of the many types of devices and options that must be supported. Not only are there many devices and options, but new ones appear on the horizon frequently. User interface design is also complex because it has many characteristics of analysis. It cannot be done in isolation, but requires heavy user involvement throughout.

Many techniques can be used to do user interface design, including models, mock ups, samples, story boards, and so forth.

Design the Database

Information systems, by definition, always contain a major database component. If the database is not designed correctly, then the information requirements will not be met correctly. Database design is a fairly mature activity, with standard rules and approaches. If the data analysis was done thoroughly, then the actual design of the database can proceed rapidly. During design, however, is a good time to double check the accuracy of the data definitions done during analysis.

Design the Software Classes and Methods

This activity is often thought of as detail design. Depending on the complexity of the system either a minimal set of models can be generated, or a thorough and complex design can be done. The primary objective of this activity is to define and design the classes and methods that are needed to carry out each use case. A complex design will utilize such models as design class diagrams and interaction diagrams.

Quick Quiz

Q: What are the five activities included in Systems Design?

> A: Describe the environment, Design the application components, Design the user interface, Design the database, Design the software

Q: Designing the environment focuses on two major issues, what are they?

> A: External systems and the required interfaces, and the total technology environment.

System Controls and Security

Key terms

integrity controls: controls that reject invalid data inputs, prevent unauthorized data outputs, and protect data and programs against accidental or malicious tampering

input controls: controls that prevent invalid or erroneous data from entering the system

value limit controls: controls that check numeric data input to ensure that the value is reasonable

completeness controls: controls that ensure that all required data values describing an object or transaction are present

data validation controls: controls that ensure that numeric fields that contain codes or identifiers are correct

field combination controls: controls that review combinations of data inputs

output controls: controls that ensure that output arrives at the proper destination and is accurate, current, and complete

fraud triangle: a model of fraud that states that opportunity, motivation, and rationalization must all exist for a fraud to occur

security controls: controls that protect the assets of an organization from all threats, with a primary focus on external threats

access control: a control that limits a user's ability to access resources, such as servers, files, Web ages, application programs, and database tables

authentication: the process of identifying users who request access to sensitive resources

multifactor authentication: the process of using multiple authentication methods for increased reliability

access control list: a list attached or linked to a specific resource that describes users or user groups and the nature of permitted access

unauthorized users: people who aren't allowed access to any part or function of the system

authorization: the process of allowing or restricting a specific authenticated user's access to a specific resource based on an access control list

registered users: people who are authorized to access the system

privileged users: people who have access to the source code, executable program, and database structure of the system

encryption: the process of altering data so unauthorized users can't view them

decryption: the process of converting encrypted data back to their original state

encryption algorithm: a complex mathematical transformation that encrypts or decrypts binary data

encryption key: a binary input to the encryption algorithm—typically a long string of bits

symmetric key encryption: an encryption method that uses the same key to encrypt and decrypt the data

remote wipe: a security measure that automatically deletes sensitive data from a portable device when unauthorized accesses are attempted

asymmetric key encryption: an encryption method that uses different keys to encrypt and decrypt the data

public key encryption: a form of asymmetric key encryption that uses a public key for encryption and a private key for decryption

digital signature: a technique in which a document is encrypted by using a private key to verify who wrote the document

digital certificate: an institution's name and public key (plus other information, such as address, Web site URL, and validity date of the certificate) encrypted and certified by a third party

certifying authorities: widely accepted issuers of digital certificates

Secure Sockets Layer (SSL): a standard set of methods and protocols that address authentication, authorization, privacy, and integrity

Transport Layer Security (TLS): an Internet standard equivalent to SSL

IP Security (IPSec): an Internet standard for secure transmission of low-level network packets

Hypertext Transfer Protocol Secure (HTTPS): an Internet standard for securely transmitting Web pages

Lecture Notes

The terms control and security are blanket terms used for a wide variety of issues to mitigate risks and dangers of all information systems. These risks range from simple risks like incorrect data input to more serious dangers of revealing confidential information and even to malicious destruction of data or systems. Some controls are embedded within specific software applications, while other security measure are system wide and affect the entire environment.

System controls and security are not identified as a separate design activity in Figure 6-3 because they must be integrated in all other design activities. They should be integral to every other design.

Designing Integrity Controls

Integrity controls are mechanisms and procedures that are built into a system to safeguard the system and the information within it. Some of the controls—called integrity controls—must be integrated into the application programs that are being developed and the database that supports them. Other controls—usually called security controls—are part of the operating system and the network. Integrity controls ensure correct system function by rejecting invalid data inputs, preventing unauthorized data outputs, and protecting data and programs against accidental or malicious tampering. The primary objectives of integrity controls are to:

- Ensure that only appropriate and correct business transactions occur.
- Ensure that the transactions are recorded and processed correctly.
- Protect and safeguard the assets of the organization (including hardware, software, and information).

Input Controls

Input controls prevent invalid or erroneous data from entering the system. Input controls can be applied to data entered by people or data transmitted from internal or external systems.

- **Value limit controls**—These check numeric data inputs to ensure that the amount entered is reasonable.
- **Completeness controls**—These ensure that all required data values describing an object or

transaction are present.

- **Data validation controls**—These ensure that numeric fields containing codes or identifiers are correct.
- **Field combination controls**—These review various combinations of data inputs to ensure that the correct data are entered.

Output Controls

Output controls ensure that output arrives at the proper destination and is accurate, current, and complete. It is especially important that outputs with sensitive information arrive at the proper destination and that they not be accessed by unauthorized persons. Common types of output controls include:

- Physical access controls to printers
- Discarded output control—Physical control of discarded printed outputs containing sensitive data is a must because "dumpster diving" is an effective way to access data without authorization.
- Access controls to programs that display or print
- Formatting and labeling of printed outputs—System developers ensure completeness and accuracy by printing control data on the output report.
- Labeling of electronic outputs—Electronic outputs typically include internal labels or tags that identify their source, content, and relevant dates.

Redundancy, Backup and Recovery

Redundancy, backup, and recovery procedures are designed to protect software and data from hardware failure and from such catastrophes as fire and flood. Many organizations that need continuous access to their data and systems employ redundant databases, servers, and sites. If one site or server fails, the other(s) is (are) still accessible and the organization continues to function. Backup procedures make partial or full copies of a database to removable storage media, such as magnetic tape, or to data storage devices or servers at another site.

Fraud Prevention

System developers often pay inadequate attention to an equally serious problem: the use of the system by authorized people to commit fraud. In order for fraud to occur, certain conditions are always present:

- Opportunity—the ability of a person to take actions that perpetrate a fraud.
- Motivation—a desire or need for the results of the fraud. Money is the usual motivation.
- Rationalization—an excuse for committing the fraud or an intention to "undo" the fraud in the future.

System designers have little or no impact on motive and rationalization, but they can minimize or

eliminate opportunity by designing and implementing effective controls. The following list itemizes some important factors can reduce the risk of fraud:

- Separation of duties so that no one person has access to disbursements and record keeping
- Records and audit trails that record all transactions
- Monitoring procedures to review records and audit trails
- Asset control and reconciliation to limit access to physical inventory and reconcile inventory levels
- Security designed into the systems and infrastructure

Designing Security Controls

In addition to the objectives enumerated earlier for integrity controls, security controls have two objectives:

- Maintain a stable, functioning operating environment for users and application systems (usually 24 hours a day, 7 days a week) —focuses on security measures to protect the organization's systems from external attacks from hackers, viruses, and worms as well as denial-of-service attacks.
- Protect information and transactions during transmission across the Internet and other insecure environments—focuses on the information that is sent or received via the Internet.

Access Controls

Access controls limit the ability of specific users to access such specific resources as servers, files, Web pages, application programs, and database tables. Most access control systems rely on these common principles and processes:

- **Authentication**—the process of identifying users who request access to sensitive resources. Users can be authenticated through user names and passwords, smart cards, challenge questions and responses, or such biometric methods as fingerprint and retinal scans or voice recognition.
- **Access control list**—a list attached or linked to a specific resource that describes users or user groups and the nature of permitted access (e.g., read data, update data, and execute program).
- **Access controls** restrict which persons or programs can add, modify, or view information resources. Access controls are generally implemented through the operating system (as security controls) or through the DBMS to restrict access to individual attributes, tables, or entire databases.
- **Authorization**—the process of allowing or restricting a specific authenticated user's access to a specific resource based on an access control list.

There are three user categories or types and the role of the access control system in allowing or restricting their access:

- **Unauthorized users**—people who aren't allowed access to any part or functions of the system.

- **Registered users**—those who are authorized to access the system. Normally, various types of registered users are defined depending on what they are authorized to view and update.
- **Privileged users**—people who have access to the source code, executable program, and database structure of the system, including system programmers, application programmers, operators, and system administrators.

Data Encryption

No access control system is perfect, so designers must anticipate access control breaches and provide other measures to protect the confidentiality of data. The primary method of maintaining the security of data—data on internal systems and transmitted data—is by encrypting the data. **Encryption** is the process of altering data so unauthorized users can't view them. **Decryption** is the process of converting encrypted data back to their original state. An **encryption algorithm** is a complex mathematical transformation that encrypts or decrypts binary data. An **encryption key** is a binary input to the encryption algorithm—typically a long string of bits.

With **symmetric key encryption** the same key encrypts and decrypts the data. A significant problem with symmetric key encryption is that sender and receiver use the same key, which must be created and shared in a secure manner. Security is compromised if the key is transmitted over the same channel as messages encrypted with the key.

An additional security measure for portable devices is a technique commonly called **remote wipe**, which automatically deletes sensitive data from portable devices under certain conditions, such as repeated failure to enter a valid user name and password or an attempt to access a database or file from an unauthorized application.

Asymmetric key encryption uses different but compatible keys to encrypt and decrypt data. **Public key encryption** is a form of asymmetric key encryption that uses a public key for encryption and a private key for decryption. The two keys are like a matched pair. Once information is encrypted with the public key, it can be decrypted only with the private key. It can't be decrypted with the same public key that encrypted it. Some asymmetric encryption methods can encrypt and decrypt messages in both directions. That is, in addition to using the public key to encrypt a message that can be decrypted with the private key, an organization can also encrypt a message with the private key and decrypt it with the public key, which is the basis for Digital Signatures.

Digital Signatures and Certificates

A **digital signature** is a technique in which a document is encrypted by using a private key to verify who wrote the document. If you have the public key of an entity and then that entity sends you a message with its private key, you can decode it with the public key. The encoding of a message with a private key is called digital signing.

A certificate—or **digital certificate**—is an institution's name and public key (plus other information, such as address, Web site URL, and validity date of the certificate), encrypted and certified by a third party. Many third parties, such as VeriSign and Equifax, are very well known and widely accepted certifying authorities. An entity that wants a certificate with its name and public key goes to a certifying authority and buys a certificate. The **certifying authority** encrypts the data with its own private key

(signs the data) and gives the data back to the original entity.

Secure Transactions

Secure electronic transactions require a standard set of methods and protocols that address authentication, authorization, privacy, and integrity. Netscape originally developed the **Secure Sockets Layer (SSL)** to support secure transactions.

SSL was later adopted as an Internet standard and renamed **Transport Layer Security (TLS)**, although the original name—SSL—is still widely used.

TLS is a protocol for a secure channel to send messages over the Internet. Sender and receiver first establish a connection by using ordinary Internet protocols and then ask each other to create a TLS connection. Sender and receiver then verify each other's identity by exchanging and verifying identity certificates as explained previously.

IP Security (IPSec) is a newer Internet standard for secure message transmission. IPSec is implemented at a lower layer of the network protocol stack, which enables it to operate with greater speed.

Hypertext Transport Protocol Secure (HTTPS or HTTP-S) is an Internet standard for securely transmitting Web pages.

Quick Quiz

Q: What are the three objectives of integrity controls?

> A: Have correct business transactions, record and process these transactions correctly, protect all assets of the organization.

Q: Name four types of input controls.

> A: Value limit controls, completeness controls, data validation controls, field combination controls

Q: What are the two objectives of transaction logging?

> A: To discourage fraudulent actions by recording who did what, and to provide a recovery mechanism.

Q: What is the objective of output controls?

> A: To ensure that output arrives at the proper destination (location and person), and is accurate, current, and complete.

Q: For fraud to occur, three things need to be present. What are they?

> A: Opportunity, Motivation, and Rationalization

Q: What are several techniques used to reduce the opportunity for fraud?

> A: Separation of duties, good audit trails, monitor procedures and activities, have good access controls and good security.

Q: What are the two objectives of security controls?

A: Maintain a stable and safe environment, and protect data during transmission.

Q: What is the difference between authentication and authorization?

A: Authentication is to identify a person (or foreign system) to know who it is, and authorization is the process to allow that identified person access to certain data and assets.

Q: Name and define three types of user roles for security access.

A: Unauthorized users - cannot have access to the assets, registered users-known users who can have access, Privileged users-known users who also have access to the security system itself.

Q: How does symmetric encryption work?

A: Data is encrypted and decrypted using an algorithm that has an encryption key. The same key is used to encrypt and decrypt.

Q: How does asymmetric encryption work?

A: There are two keys, one to encrypt and one to decrypt, which are different keys. The encryption key cannot decrypt, only the decryption key can.

Q: What is the difference between public key encryption and asymmetric key encryption?

A: They are the same, except that one of the keys (the encryption key) is made public.

Q: What is digital signing?

A: It is a public key encryption method, where either key may be the encryption key. The opposite key is always used to decrypt. In other words, encryption/decryption can go either direction, i.e. with either key combination. So a document is signed if it is encrypted with the private key, so that public users can decrypt it with the public key.

Q: What is a certifying authority use for?

A: Since asymmetric encryption/decryption is very inefficient, it is only used to send a symmetric key. The original public key is sent in a signed certificate from a well-known and widely accepted certifying authority. Then asymmetric encryption/decryption is used to send a symmetric key. Finally the encrypted message is sent using the symmetric key. The certifying authority provides a secure method, with a digitally signed certificate by a widely accepted authority which has published its public key (built into web browsers), of sending out an organizations public key.

Classroom Activities

There are two major themes in this chapter that the students should understand.

The first major theme concerns the concepts related to design – what it is, what feeds into it, and what are the activities of design. This material is covered in the first two major headings. Figure 6-3 and 6-4 are two key figures that the students should understand.

One interesting activity is to have students think about a server based information system, such as a grocery store checkout system and a smaller smartphone kind of app, such as a book reader. Discuss

what components need to be designed in each instance. Base the discussion on the five activities identified in Figure 6-3.

Another interesting activity is to invite some guest speakers who can give a short interpretation of things like database design, user interface design, security design, network design, and interfaces to foreign systems. Care is required if this is chosen so that the quest speakers stay on track and only take a few minutes. It is also useful to summarize so that the overview of all the areas is understood.

A classroom activity for controls is a little harder to create. Some case histories about fraud and embezzlement due to lack of controls is also interesting for the students. Often times accounting professors will have examples of fraud due to lack of process as well as information system controls.

Security is such a broad and critically important area now, that it is imperative that students understand the importance of security for every new system. One interesting classroom activity is simply to browse several of the fraud and virus protection and history sites. There are several available. Another interesting activity is to have a securities expert come in and relate some case histories. Or with a little Internet research you can find several interesting virus attack cases. (Searches for Most expensive computer virus, Cost of computer malware, etc.)

One other interesting activity is to act out Figure 6-15. Select three students to be (1) the client computer (2) the server computer, and (3) a certifying authority and then have the students pass messages to each other that are encrypted, signed certificates, symmetric keys, etc. to simulate establishing a secure and encrypted connection between a client and a server.

Troubleshooting Tips

None of the concepts in this chapter are particularly difficult. The major problem is that there are quite a few terms and concepts to learn.

The one area that is a little complex and can be difficult to understand is the concept of an asymmetrical key (public key) system and particularly how a secure session is set up (Figure 12-25). Doing the suggested class activity discussed above will help students understand the process. The only way to really understand it is to go through the process step by step. It cannot be glossed over and understood.

Discussion Questions

1. Analysis versus Design

Figure 6-4 shows the relative amount of time spent on analysis (Core process 3) and design (Core process 4) both within individual iterations and across all iterations. First, can you explain why Figure 6-4 is constructed with the relative weights as shown across the project? Second within an iteration, can analysis and design activities occur simultaneously? Why or why not? How does a developer know when to do analysis and when to do design? Does he/she even try to distinguish? Sometimes there is a problem that a developer wants to get to design before analysis is complete. Is this a danger? What is the danger? How can it be avoided?

2. Design Modeling

Although this chapter does not discuss design modeling in great depth, more will come later, a short

discussion of design modeling may be appropriate. How useful is design? What kind of models or tools might be useful? Is it good practice to create design models? How formal should they be? Should they be saved and archived? What are design models good for – during design, during construction, during system maintenance? Can we just skip design and building models? What are the dangers with that approach?

3. Designing Integrity Controls

The rapid increase in e-commerce and the need for online access has increased the need for integrity controls. Integrity controls are integrated into the application and the database. Business managers want integrity but might not be willing to pay for the time and effort needed to develop comprehensive integrity controls. What arguments can a project manager present to persuade business owners that this investment in non-functional software is beneficial? How much effort is needed compared to developing functional business software?

4. Designing Security Controls

What are the differences between integrity controls and security controls? Who, on a typical software development team, should develop security controls? Is it better to implement security controls directly within application software, or should security software be maintained separately? Because the developers of security controls know a great deal about your system, what can you do to protect the system from internal dangers?