System Analysis & Design

CSCI 2783

Design Phase

The design phase of the SDLC uses the requirements that were gathered during analysis to create a blueprint for the future system. A successful design builds on what was learned in earlier phases and leads to a smooth implementation by creating a clear, accurate plan of what needs to be done.

The design phase decides how the new system will operate. Many activities will be involved as the development team develops the system requirements.

Design Phase

The purpose of the analysis phase is to figure out what the business needs. The purpose of the design phase is to decide how to build it. System design is the determination of the overall system architecture—consisting of a set of physical processing components, hardware, software, people, and the communication among them—that will satisfy the system's essential requirements.

Design Phase (Activities)

Activities in the Design Phase	Deliverables
✓ Determine preferred system acquisition strategy (make, buy, or outsource).	– Alternative matrix
✓ Design the architecture for the system.	– Architecture design
✓ Make hardware and software selections.	– Hardware and software specification
✓ Design system navigation, inputs, and outputs.	– Interface design
✓ Convert logical process model to physical process model.	– Physical process model
✓ Update CASE repository with additional system details.	– Updated CASE repository
✓ Design the programs that will perform the system processes.	– Program design specifications
✓ Convert logical data model to physical data model.	– Physical data model
✓ Update CASE repository with additional system details.	– Updated CASE repository
✓ Revise CRUD matrix.	– CRUD matrix
✓ Design the way in which data will be stored.	– Data storage design
✓ Compile final system specification.	– System specification: all of the above deliverables combined and presented to approval committee

Design Phase

During the initial part of design, the project team converts the business requirements for the system into system requirements that describe the technical details for building the system. Unlike business requirements, which are listed in the requirements definition and communicated through use cases and logical process and data models, system requirements are communicated through a collection of design documents and physical process and data models. Together, the design documents and physical models make up the blueprint for the new system.

Design Phase

We should note here that our focus is on the design of the technical system blueprint that will satisfy the system's requirements. An important element of the final, complete information system, however, will be redesigned work flows and procedures that users will follow when using the new system.

Design Phase

The design phase has a number of activities that lead to the system blueprint. An important initial part of the design phase is the examination of several system acquisition strategies to decide which will be used to meet the requirements of the system. Systems can be built from scratch, purchased and customized, or outsourced to others, and the project team needs to investigate the viability of each alternative. The decision to make, to buy, or to outsource influences the design tasks that are performed throughout the rest of the phase.

Design Phase

The project team carefully considers the nonfunctional business requirements that were identified during analysis. The nonfunctional business requirements influence the system requirements that drive the design of the system's architecture. Major considerations of the "how" of a system are operational, performance, security, cultural, and political in nature. For example, the project team needs to plan for the new system's performance: how fast the system will operate, what its capacity should be, and its availability and reliability.

Design Phase

The team needs to create a secure system by specifying access restrictions and by identifying the need for encryption, authentication, and virus control. The nonfunctional requirements are converted into system requirements that are described in the **architecture design document.**

Design Phase

At the same time, architecture decisions are made regarding the hardware and software that will be purchased to support the new system. These decisions are documented in the hardware and software specification, which is a document that describes what hardware and software are needed to support the new application. The actual acquisition of hardware and software is sometimes the responsibility of the purchasing department or the area in the organization that handles capital procurement.

Design Phase

The user’s interactions with the system also must be designed. The system inputs and outputs will be designed along with a plan or roadmap of the way the system’s features will be navigated. Design decisions made regarding the interface are communicated through the design document called the interface design.

Design Phase (System Specification Outline)

- Recommended System Acquisition Strategy
- System Acquisition Weighted Alternative Matrix
- Architecture Design
- Hardware and Software Specification
- Interface Design
- Physical Process Model
- Program Design Specifications
- Physical Data Model
- Data Storage Design
- Updated CRUD Matrix
- Updated CASE Repository Entries

Design Phase (System Specification Outline)

At the end of the design phase, the project team creates the final deliverable for the phase called the system specification. This document contains all of the design documents just described: physical process models, physical data model, architecture design, hardware and software specification, interface design, data storage design, and program design. Collectively, the system specification conveys exactly what system the project team will implement during the implementation phase of the SDLC.

Design Phase (System Acquisition Strategies)

In this class, we should assume that the system will be designed, developed, and implemented by the project team. This is not an entirely realistic assumption. In many projects, the team may recognize that some parts or even all of the new system's software will be acquired from some outside provider. Some organizations have established acquisition policies strongly favoring purchased software.

Design Phase (System Acquisition Strategies)

There are, however, actually three primary ways to approach the creation of a new system:

1. develop a custom application in-house;
2. buy a packaged system and (possibly) customize it; and
3. rely on an external vendor, developer, or service provider to build or provide the system.

Each of these choices has its strengths and weaknesses, and each is more appropriate in different situations. There may be obvious characteristics of the project that suggest the preferred acquisition strategy.

Design Phase (Custom Application)

Many project teams assume that custom development, or building a new system from scratch, is the best way to create a system. For one, teams have complete control over the way the system looks and functions. If the company wants a Web-based feature that links tightly with its existing sales system, the project may involve a complex, highly specialized program.

Design Phase (Custom Application)

Alternatively, a company might have a technical environment in which all information systems are built from standard technology and interface designs so that they are consistent and easier to update and support. In both cases, it could be very effective to create a new system from scratch that meets these highly specialized requirements.

Design Phase (Custom Application)

In some situations, the challenges being addressed with the new system are so significant and demanding that serious systems engineering is required to solve them. In these cases, the developers really cannot find a packaged solution that is capable of meeting the project requirements, and a custom development project is the only real viable choice.

Design Phase (Custom Application)

Custom application development, however, requires a dedicated effort that includes long hours and hard work. Many companies have a development staff that is already overcommitted. Facing huge backlogs of systems requests, the staff just does not have time for another project. Also, a variety of skills—technical, interpersonal, functional, project management, modeling—all have to be in place for the project to move ahead smoothly. IS professionals, especially highly skilled individuals, are quite difficult to hire and retain.

Design Phase (Purchased Software)

Many business needs are not unique, and because it makes little sense to reinvent the wheel, many organizations buy packaged software that has already been written, rather than developing their own custom solution. In fact, there are thousands of commercially available software programs that have already been written to serve a multitude of purposes. Think about your own need for a word processor—did you ever consider writing your own word processing software? That would be very silly, considering the number of good software packages available for a relatively inexpensive cost.

Design Phase (Purchased Software)

Most companies have needs, such as payroll or accounts receivable, that can be met quite well by packaged software. It can be much more efficient to buy programs that have already been created, tested, and proven, and a packaged system can be bought and installed quickly compared with a custom system. Plus, packaged systems incorporate the expertise and experience of the vendor who created the software.

Design Phase (Purchased Software)

Packaged software can range from small single-function tools, such as the server-side download manager, to huge all-encompassing systems, such as enterprise resource planning (ERP) applications that are installed to automate an entire business. Implementing ERP systems is a popular practice in which large organizations spend millions of dollars installing packages by such companies as SAP, Oracle, and Infor and then change their businesses accordingly. Installing ERP software is much more difficult than installing small application packages, because benefits can be harder to realize and problems are much more serious.

Design Phase (Purchased Software)

One problem is that companies utilizing packaged systems must accept the functionality that is provided by the system, and rarely is there a perfect fit. If the packaged system is large in scope, its implementation could mean a substantial change in the way the company does business. Letting technology drive the business can be a dangerous way to go.

Design Phase (Purchased Software)

Systems integration refers to the process of building new systems by combining packaged software, existing legacy systems, and new software written to integrate these. Many consulting firms specialize in systems integration, so it is not uncommon for companies to select the packaged software option and then outsource the integration of a variety of packages to a consulting firm.

Design Phase (Purchased Software)

The key challenge in systems integration is finding ways to integrate the data produced by the different packages and legacy systems. Integration often hinges on taking data produced by one package or system and reformatting it for use in another package or system. The project team starts by examining the data produced by and needed by the different packages and systems and identifying the transformations that must occur to move the data from one to the other.

Design Phase (Outsourcing)

The acquisition choice that requires the least in-house resources is outsourcing, which means hiring an external vendor, developer, or service provider to create or supply the system. Outsourcing has become quite popular in recent years, with both U.S. and non-U.S. service providers available.

Design Phase (Outsourcing)

There is an array of application service providers. Some deliver high-end business applications that can serve the entire enterprise. Some are focused more on serving a small- to medium-sized business clientele. Some ASPs specialize in specific business needs (such as CRM, for example), while some specialize in specific industries (e.g., healthcare).

Design Phase (Influences on Acquisition Strategies)

1. Business Needs
2. In-house Experience
3. Project Skills
4. Project Management
5. Time Frame

Design Phase (Business Needs)

If the business need for the system is common and technical solutions already exist in the marketplace that can fulfill the system requirements, it is usually appropriate to select a packaged software solution. Packaged systems are good alternatives for common business needs.

Packaged software is not suitable for every situation, however. A custom solution should be explored when the business need is unique.

Design Phase (In-House Experience)

If in-house experience exists for all the functional and technical needs of the system, it will be easier to build a custom application than if these skills do not exist. A packaged system may be a better alternative for companies that do not have the technical skills to build the desired system.

Design Phase (Project Skills)

The skills that are applied during projects are either technical (e.g., Java, Structured Query Language [SQL]) or functional (e.g., electronic commerce), and different design alternatives are more viable, depending on how important the skills are to the company's strategy.

Design Phase (Project Management)

Custom applications require excellent project management and a proven methodology. There are so many things that can push a project off track, such as funding obstacles, staffing holdups, and overly demanding business users. Therefore, the project team should choose to develop a custom application only if it is certain that the underlying coordination and control mechanisms will be in place.

Design Phase (Time Frame)

When time is a factor, the project team should probably start looking for a system that is already built and tested. In this way, the company will have a good idea of how long the package will take to put in place and what the final result will contain. Of course, this assumes that the package can be installed as-is and does not need many workarounds to integrate it into the existing business processes and technical environment. The time frame for custom applications is hard to pin down, especially when you consider how many projects end up missing important deadlines.

Design Phase (Alternative Matrix)

An alternative matrix can be used to organize the pros and cons of the design alternatives so that the best solution will be chosen in the end. The matrix is created using the same steps as the feasibility analysis The only difference is that the alternative matrix combines several feasibility analyses into one matrix so that the alternatives can be easily compared. The alternative matrix is a grid that contains the technical, economical, and organizational feasibilities for each system candidate, pros and cons associated with adopting each solution, and other information that is helpful when making comparisons.

Design Phase (Alternative Matrix)

Evaluation Criteria	Relative Importance (Weight)	Alternative 1: Custom Application Using VB.NET	Score (1–5)*	Weighted Score	Alternative 2: Custom Application Using Java	Score (1–5)*	Weighted Score	Alternative 3: Packaged Software Product ABC	Score (1–5)*	Weighted Score
Technical Issues:										
Criterion 1	20		5	100		3	60		3	60
Criterion 2	10		3	30		3	30		5	50
Criterion 3	10		2	20		1	10		3	30
Economic Issues:										
Criterion 4	25	Supporting	3	75	Supporting	3	75	Supporting	5	125
Criterion 5	10	Information	3	30	Information	1	10	Information	5	50
Organizational Issues										
Criterion 6	10		5	50		5	50		3	30
Criterion 7	10		3	30		3	30		1	10
Criterion 8	5		3	15		1	5		1	5
TOTAL	100			350			270			360

Design Phase (Architecture Design)

The objective of architecture design is to determine how the software components of the information system will be assigned to the hardware devices of the system. We must recognize the major functions of the software to understand how the software can be divided into different parts. The most common architecture is the client–server architecture.

Design Phase (Architecture Design)

All software systems can be divided into four basic functions.

1. Data storage
2. Data access logic
3. Application logic
4. Presentation logic

Design Phase (Data Storage)

Most information systems require data to be stored and retrieved, whether a small file, such as a list of lawn chemicals that are no longer authorized for residential applications, or a large database that stores an organization's human resources records. These are the data entities documented in entity relationship diagrams (ERDs).

Design Phase (Data Access Logic)

This is the process required to access data, often meaning database queries in Structured Query Language (SQL).

Design Phase (Application/Presentation Logic)

Application Logic: Involves the logic documented in the data flow diagrams (DFDs), use cases, and functional requirements.

Presentation Logic: Involves the display of information to the user and the acceptance of the user’s commands (the user interface).