



# Analyzing Requirements and Defining Microsoft .NET Solution Architectures

Written by Paul Pu  
[www.torontocollege.com](http://www.torontocollege.com)



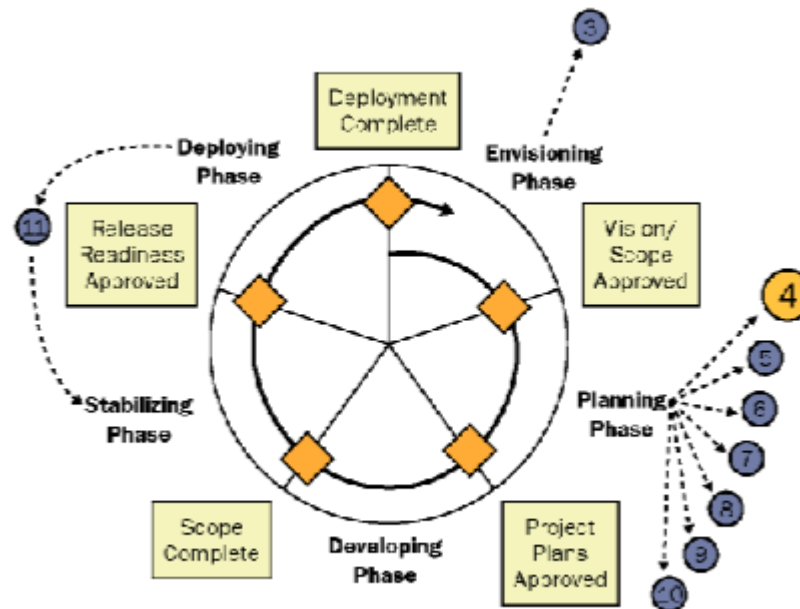
# An Introduction to the Planning Phase

---

- During the planning phase, the team defines the solution: what to build, how to build it, and who will build it. During this phase the team prepares the functional specification, works through the design process, and prepares work plans, cost estimates, and schedules for the various deliverables.



# An Introduction to the Planning Phase

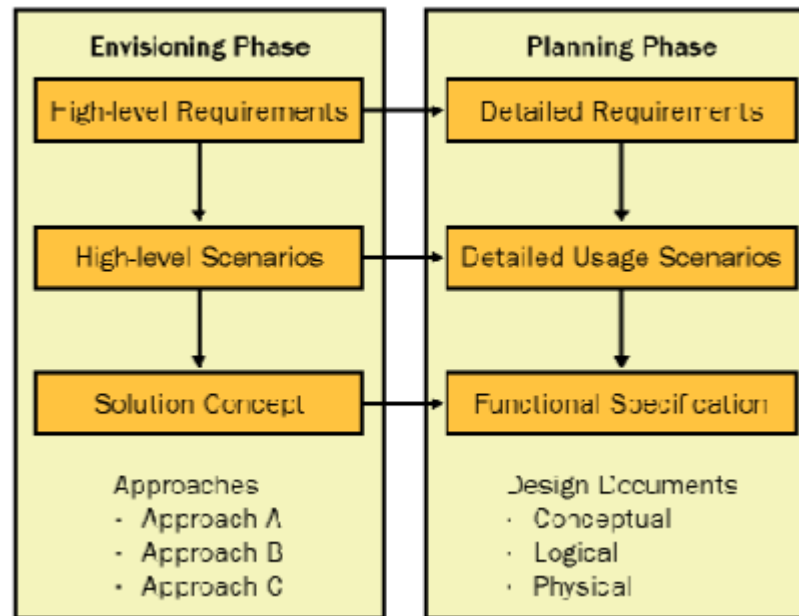


Written by Paul Pu  
[www.torontocollege.com](http://www.torontocollege.com)



# An Introduction to the Planning Phase

---

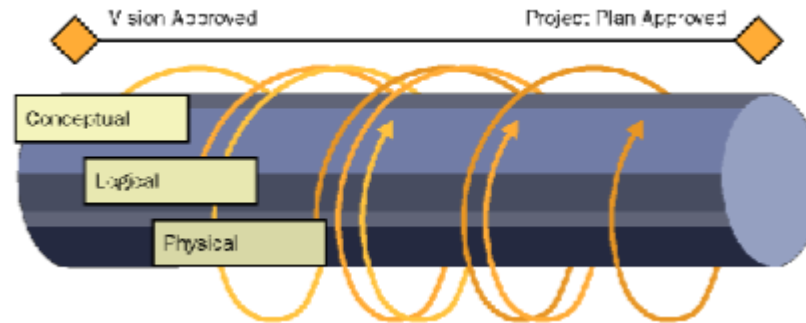


Moving from the envisioning phase to the planning phase



# The Three Design Processes: Conceptual, Logical, and Physical

---



The planning phase of the MSF Process Model



## The Three Design Processes

---

Type of design	Perspective	Purpose
Conceptual	View the problem from the perspective of the user and the business. Defines the problem and solution in terms of usage scenarios.	Defines the problem and solution in terms of usage scenarios.
Logical	View the solution from the perspective of the project team. Defines the solution as a logical set of cooperating services.	Defines the solution as a logical set of cooperating services.
Physical	View the solution from the perspective of the developers. Defines the solution's services and technologies.	Defines the solution's services and technologies.



## Roles and Responsibilities in the Planning Phase

---

- Product management ensures that the plan meets the customer needs.
- Program management ensures that the resources can accomplish the project plan.
- Development ensures that the plan is technically feasible.
- Testing ensures that the plan meets the requirements.
- Release management evaluates the design for ease of deployment, management, and support.
- User experience ensures that users will be able to use the product.



## Milestones and Deliverables of the Planning Phase

---

**The final deliverables at this milestone of the planning phase are:**

- Functional specification (baseline).
- Master project plan (baseline).
- Master project schedule (baseline).
- Updated master risk assessment document.



## An Overview of the Functional Specification

---

During the planning phase, the focus of the project team begins to shift from problem definition to solution design. One of the primary deliverables at this milestone is the **functional specification**. The functional specification defines what will be built, how it will be built, and when it will be built.



# What Are the Goals of a Functional Specification?

---

## **Some of the goals of a functional specification are:**

- Consolidate a common understanding of the business and user requirements.
- Break down the problem and modularize the solution logically.
- Provide a framework to plan, schedule, and build the solution.
- Serve as a contract between the team and the customer for what will be delivered.



## Elements of the Functional Specification

---

**The possible elements of a functional specification are listed below. Each element might be a separate document.**

**Conceptual design summary.** This section provides a summary of the conceptual design of the solution and includes information such as solution overview and solution architecture. The following artifacts from the conceptual design are used in the functional specification:

- Use cases
- Usage scenarios
- Context models such as screen shots of existing systems and photocopies of current user manuals

These artifacts can exist in various forms. For example, context models can be in the form of screen shots of an existing system or photocopies of current user manuals or reports; use cases can be in a use case documentation database; and conceptual user interface (UI) prototypes can be in electronic form.



## Elements of the Functional Specification

---

**Logical design summary.** This section provides a summary of the logical design and includes information such as users, objects, and attributes. The following artifacts from the logical design phase are included in the functional specification:

- Task and task-sequence models
- Logical object and service models
- Conceptual models of the proposed solution
- UI screen flows
- Logical database model
- System architecture



## Elements of the Functional Specification

---

**Physical design summary.** This section provides a summary of the physical design document and includes information from key sections of the document, such as the application and infrastructure sections. The following artifacts from the physical design phase are included in the functional specification:

- Component packaging
- Component distribution topology
- Technology usage guidelines
- Infrastructure architecture and design
- Description of UI screens
- Physical database model



## Elements of the Functional Specification

---

**Standards and processes.** This section includes information about the standards and processes that the team uses as guidelines for performing various tasks for the project. In addition, this section includes details of quality and performance metrics that will be used. These metrics are gathered during tests and help achieve the goals defined by the requirements.



## An Overview of the Conceptual Design Process

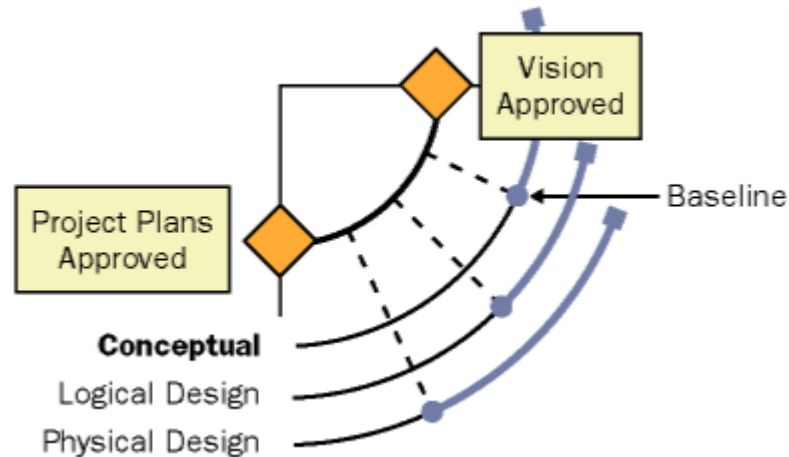
---

Conceptual design is the process of gathering, analyzing, and prioritizing business and user perspectives of the problem and the solution, and then creating a high-level representation of the solution.



## Conceptual design in the MSF Process Model

---



The conceptual design occurs in the MSF Process Model



# What Are the Goals of Conceptual Design?

---

## **What Are the Goals of Conceptual Design?**

- Understanding the business problem to be solved.
- Understanding the requirements of the business, the customer, and the end user.
- Describing the target future state of the business.



## What Are the Goals of Conceptual Design?

---

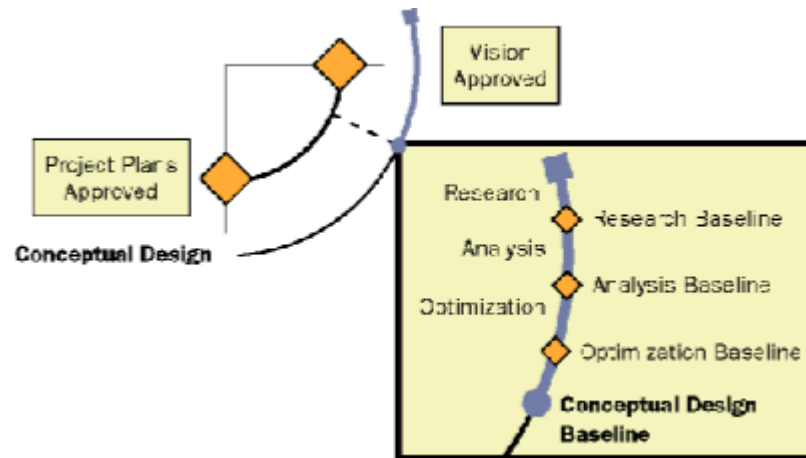
Conceptual design is not	But it helps you to
The complete functional specification	Begin the functional specification
A definition of system components	Identify the parts of the business problem that will be addressed by the eventual components
A technology solution	Records the business activities and portray their boundaries and their relationships



# What Are the Steps in Conceptual Design?

What Are the Steps in Conceptual Design?

- Research. Analysis. Optimization.



Steps in conceptual design

Written by Paul Pu  
[www.torontocollege.com](http://www.torontocollege.com)



# Building the Conceptual Design

---

## How to Restate Requirements?

- When you restate requirements, keep the following criteria in mind:
- Requirements must be well defined. A well-defined requirement is a complete sentence and typically uses "shall," "may," "must," or "should."
- Requirements must be concise. Each requirement must address one item only.
- Requirements must be testable. Each requirement should have specific inputs resulting in known outputs.
- Requirements should be organized in a hierarchy of related requirements. You need to group related requirements together under a single high-level requirement to form feature sets.
- Requirements should be written in the language of the business and should not use jargon.



## Building the Conceptual Design

---

### Draft Requirement Statements in the Envisioning Phase

Requirement ID	Requirement
1	Identify best customers by product and location (profit analysis and geographical analysis); these are the customers on which the sales team should focus.
2	Identify decreases in a customer's sales.
3	Identify best customers.
4	Identify top buyers.



## Building the Conceptual Design

---

Requirement ID	Requirement
1.1	Must be able to analyze customer data
1.1.1	Must be able to analyze profit levels by product
1.1.2	Must be able to analyze profit levels by customer
1.1.3	Must be able to analyze profit levels by region



## Building the Conceptual Design

---

### **How to Categorize Requirements?**

After refining requirements, you categorize them as **user**, **system**, **operations**, and **business requirements**.



# Building the Conceptual Design

---

## User requirements

User requirements define the non-functional aspect of the user's interaction with the solution. They help you determine the user interface and performance expectations of the solution in terms of its **reliability**, **availability**, and **accessibility**. In addition, they help you identify the training that the users will need to effectively use the solution. A successful solution satisfies both the organization's need for technology and the user's expectations for employing that technology.

Example:

- A cashier should be able to complete more than one transaction per minute.
- A customer should be able to complete the purchase of a product on the shopping Web site within five minutes.



# Building the Conceptual Design

---

## **System requirements**

System requirements specify the atomic transactions and their sequence in the system, and help the project team define how the new solution will interact with the existing systems.

## **Examples**

- All enterprise line-of-business applications supporting near-real-time user notifications must either implement the approved notification component or have received a waiver during final design review.



# Building the Conceptual Design

---

## **Operations requirements**

Operations requirements describe what the solution must deliver to maximize operability and improve service delivery with reduced downtime and risks. It addresses the following key elements of operations:

- Security
- Availability and reliability
- Manageability
- Scalability
- Supportability



# Building the Conceptual Design

---

## **Examples of operations requirements**

- Consider the example of a music store that wants to implement an e-commerce Web site. Some of the operations requirements for this site include:
- Availability and reliability. Customers should be able to access the site and use its resources at any time within stated service levels.
- Scalability and flexibility. The solution should be able to handle varying volumes of users and transactions. In addition, the site must be designed so that it can be modified and upgraded without affecting availability or performance. Both the infrastructure and business processes must be scalable and flexible.



# Building the Conceptual Design

---

## **Examples of operations requirements**

- Performance manageability. The site design should include a system for managing the total system throughput and response time within stated service levels.
- Strong security. The data, services, and devices in the system must be protected from unauthorized access. The system should also provide authentication and secure transactions.
- Administrative manageability. The site should allow the administrators to perform their tasks both onsite and remotely.
- Recoverability. The site should be able to recover from critical failure without major impact, or within stated service levels.



# Building the Conceptual Design

---

## **Business requirements**

Business requirements describe the organization's needs and expectations for the solution. Business requirements define what the solution must deliver to capitalize on a business opportunity or to meet business challenges. To identify business requirements, you need to consider the organization as a valid entity with its own set of needs from the solution. These requirements exist at the managerial decision-making level and provide the context in which the solution will operate.



# Building the Conceptual Design

---

## **Examples of operations requirements**

- Call-center managers must be able to view the last, current, and average call times for each telephone operator.
- Cashiers can override an item's price to a specified amount without the supervisor's code.
- The solution must be designed, built, and deployed as quickly as possible.
- The solution must be able to interact and communicate with other business processes, applications, and data sources.



# Building the Conceptual Design

---

## How to Refine Use Cases Diagrams?

During the envisioning phase, the project team creates a use cases diagram that specifies all high-level use cases in the organization. The purpose of this use cases diagram is to list the key use cases in the system, to define of the scope of solution, and to provide a basis for the solution concept. To create a conceptual model of the design, you need to refine the use cases that are within the scope of the solution by using the information gathered during the research step of conceptual design.

To refine the use cases diagram, you perform the following tasks:

- Create subordinate use cases.
- Create usage scenarios for each subordinate use case.
- Validate each use case and usage scenario against the original interviews, against other documentation, and with the user.
- Refine requirements with the validated use cases and usage scenario information.

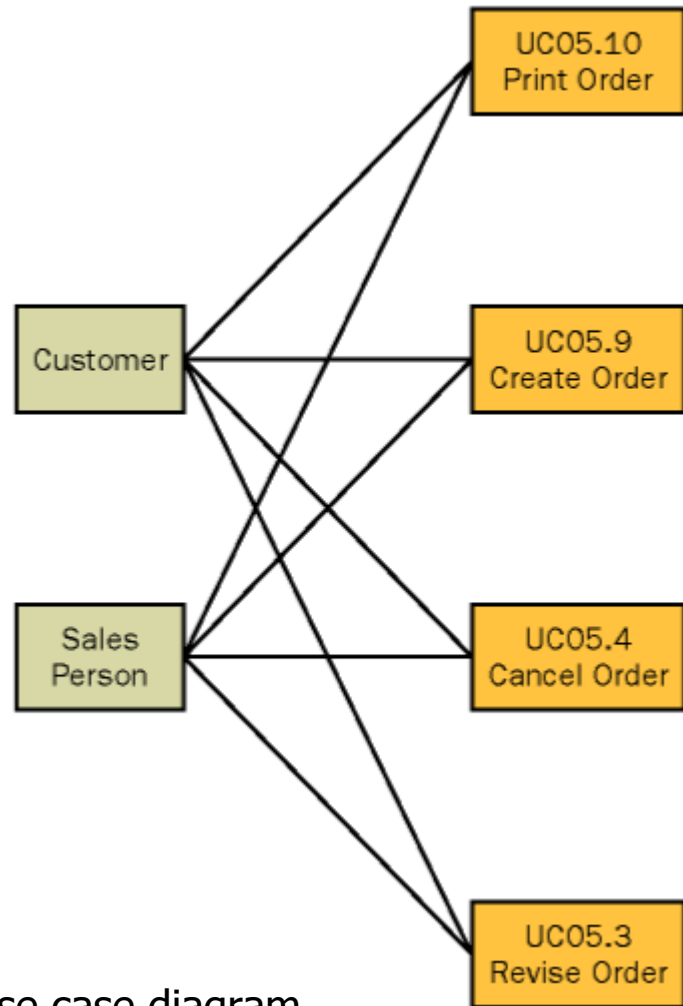


# Building the Conceptual Design

---

## **Creating subordinate use cases**

To create subordinate use cases, you revisit each use case in the use cases diagram that is within the scope of the project. You then identify each task associated with the use case, and model them as subordinate use cases for the higher-level use case. You also identify all actors that perform the tasks and the relationship between the various tasks and actors.



Refined use case diagram



Written by Paul Pu  
[www.torontocollege.com](http://www.torontocollege.com)



Written by Paul Pu  
[www.torontocollege.com](http://www.torontocollege.com)



# Building the Conceptual Design

---

Usage scenarios for subordinate use cases



# Building the Conceptual Design

---

## How to Select an Application Architecture?

- The key deliverable of the conceptual design is the conceptual model of the solution. To be able to create a conceptual model, you need to understand the services that the solution must provide.
- **A service** is defined as a unit of application logic that includes methods for implementing an operation, a function, or a transformation.



## Building the Conceptual Design

---

### **Services that a solution typically provides are:**

- **User services.** User services are units of application logic that provide the user interface in an application. The user services of an application manage the interaction between the application and its users. To design efficient user services, you need a thorough understanding of the users of the application, the tasks that they will perform, and the typical interactions they will have with the application to perform their activities.
- **Business services.** Business services are units of application logic that enforce business rules in the correct sequence. A business service hides the logic of implementing business rules and transforming data from the user services, other business services, and data services.



## Building the Conceptual Design

---

### **Services that a solution typically provides are:**

- Data services. Data services are units of application logic that provide the lowest visible level of detail for manipulating data. You use data services to implement business schema on the data store being used by the application. Data services are used to manage all kinds of data—static, structured, and dynamic. You use data services in all scenarios in which a user or business service needs to access or work with data.
- System services. System services are the units of application logic that provide functionality outside the business logic. Common system services include:
- Backup services: Backup services, Error handling services, Security services, Messaging services



# Building the Conceptual Design

---

[Examples of Services](#)



## Building the Conceptual Design

---

- **Application architecture**

You also need to know how the services are organized in the solution. The services are organized according to the application architecture. Application architecture consists of definitions, rules, and relationships that form the structure of an application. It shows how an application is structured but does not include implementation details. It focuses on the solution and not the technologies that will be used to implement the solution.



## Building the Conceptual Design

---

**Some of the application architectures that are used are:**

- Client/server architecture
- Layered architecture
- Stateless architecture
- Cache architecture
- Layered-client-cache-stateless-cache-server architecture



# Building the Conceptual Design

---

## **Client/server architecture**

- The client/server architecture is a two-tier approach that is based on a request-and-provide strategy. The client initiates a session with the server and controls the session, enlisting the server on demand. The client requests the server for one of its services. Upon receiving the request, the server performs the required operation and returns the result to the client.



# Building the Conceptual Design

---

## **Layered architecture**

- Layered architecture is an evolved version of the client/server architecture and is composed of hierarchical layers. The various services in the application are clearly positioned in specific layers in such a way that a service cannot communicate with other services except the ones in the adjacent layer. Layers encapsulate services and protect one service from another while providing a simplified set of interfaces for shared resources. User, business, and data services are examples of a layered architecture.



# Building the Conceptual Design

---

## **Stateless architecture**

- Stateless architecture is a version of the client/server or layered architecture in which each client request contains all the information that is required by the server to understand and process the request. No information is stored on the client.



# Building the Conceptual Design

---

## **Cache architecture**

- Caching is another approach in which the application provides a means for processing some client requests without forwarding the request to another device. To implement this kind of application architecture, you need to identify what can and cannot be cached. You also need to define ways to manage the lifetime of items in the cache.



## Building the Conceptual Design

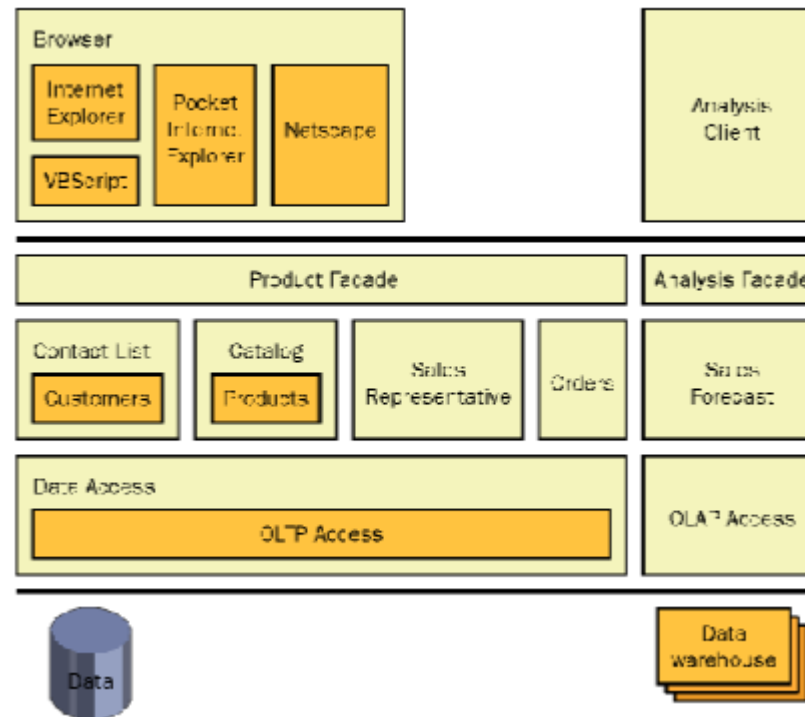
---

### **Layered-client-cache-stateless-cache-server architecture**

- Layered-client-cache-stateless-cache-server architecture is the browser-based version of the Windows Distributed interNet Applications (DNA) architecture. It combines the layered-client-server, client-cache, and cached-stateless-server approaches by adding proxies throughout the system as necessary.



## Building the Conceptual Design



Conceptual model of solution for Adventure Works Cycles



# Building the Conceptual Design

---

## Optimizing the Conceptual Design

- The final step of conceptual design is optimization. During optimization, you begin designing the solution by evolving the solution concept as it will be in the final application.