# Table of Contents

# Table of Contents

# Table of Contents

# Table of Contents

# *Module 1* Introduction

# Introduction

**In this module, you will:**

- *Consider a variety of project scenarios*
- *Get an overview of the agenda, topics covered, and objectives*
- *Agree on workshop norms and logistics*
- *Survey the workshop materials*

Welcome to ***Business Analysis Simulation***. This course has been designed as a capstone to the advanced analysis program. It is intended to tap into the knowledge, skills, and experience that you have developed throughout the previous workshops of ***Advanced Business Analysis*** and ***Bringing Business Solutions to Life*** as well as your analysis expertise.

This workshop is designed for practical application of various industry recognized tools and techniques to successfully complete a project. Throughout the workshop, real world situations will be introduced and project teams will have to use problem solving, decision making, leadership and technical skills to determine the best way to handle and overcome the challenges. Oral and written communication will also be essential to elicit and validate information as well as to seek and gain approval at key points throughout the process. This workshop will prove your competencies in various key business analysis knowledge areas.

This module will provide information on the workshop objectives and topics of discussion, the logistics for this workshop, and some guidelines for obtaining the maximum benefit from the workshop.

# Introduction

## Has this happened to you?

1. The design team understands the requirements, but does not understand the business processes or people that will be using the system.
2. The requirements correctly document system functionality, but the database developer has multiple questions about data elements that the business analyst can not answer.
3. There is confusion between the business users, project sponsor and analysis team about the scope and related components of a project.
4. The design team has told you that many of the documented business requirements are not technically feasible and can not be implemented.
5. During design, a user suggests that a new piece of data needs to be captured, displayed on the user interface and stored in the database. None of this was captured in the requirements.
6. During testing, there is disagreement between the analyst, designer and test teams about the definition of a term in a requirement.
7. After the requirements are signed off, the programmers developing the related application say that they do not understand what is written in the document.
8. The test team identifies too many bugs to fix before the planned implementation date.
9. A new application has so many requirements to test in a limited time period, that it is hard to determine which test cases to write.
10. The users identify more enhancement requests during user acceptance testing, than the original requirements written during analysis.
11. The help desk calls to ask you if anything has changed in a desktop application. They report that a large number of users are calling with problems. You are aware that the application was updated over the weekend, but apparently the users and helpdesk were not properly informed.
12. The users complain there is new functionality in an existing application and they do not know how to use it.
13. After implementation, users complain the system does not do what they expect it to do.

# Introduction

## Workshop Objectives

***Business Analysis Simulation*** has been designed to help you accomplish a great deal in a short period of time. The objectives of this workshop are:

- To review those activities a BA may be involved with to analyze, design, test and implement a solution.
- To review tools and techniques a BA can use on the project
- To gain practical experience – through hands on exercises

### Workshop Agenda

To meet these objectives, ***Business Analysis Simulation*** includes the following modules:

1. **Introduction –** Recognize workshop objectives, agenda and logistics, meet the analyst team and get up-to-speed on a case study project.
2. **The Building Blocks –** Review the analysis tools and tasks relevant to successful completion of the workshop. Discuss success criteria. Begin the thought process of dealing with real world scenarios which might impact the analysis work or the project outcome.
3. **Analysis –** Get right into the project work performing analysis, reviewing requirements, modeling use cases, creating activity diagrams and a traceability matrix.
4. **Design –** Define the technical components of the solution. Build data models and select a solution based on the analysis work. Create storyboards to represent the desired user interface. Gain approval to continue into implementation.
5. **Testing** – Create test cases tracing back to the requirements and use cases. Perform simulated user acceptance testing and document the results. Discuss and decide whether to move into the implementation phase.
6. **Implementation** – Analyze project impact, and determine how to handle various real world project scenarios. Determine whether to implement in a go / no go meeting.
7. **Follow Through** – Wrap up all the work and complete the project. Evaluate and respond to issues, close out the project. Brainstorm BA skills that have been mastered throughout the program and what it takes to be a superstar BA.

## Workshop Approach

1. **Knowledge, tools and techniques** from ***Advanced Business Analysis*** and ***Bringing Business Solutions to Life*** will be applied to all exercises on a new case study project.
2. **Real world scenarios** will be used throughout the workshop for participants to problem solve, make decisions and take actions to overcome challenges.
3. **Written and oral communications** will be critical to gain acceptance through several approval review points.
4. This workshop will employ the "**time box**" approach to complete the exercises – exercises will be limited to pre-defined periods of time.
5. The end result will be a review of the **project implementation** based on the criteria initially defined by each team.

### Getting the most from this workshop

Generally, what people get from any workshop is directly related to what they put in. To help maximize the value you derive from ***Business Analysis Simulation***, consider doing the following:

- Set cell phones and pagers to vibrate or off
- Respect agreed-upon class start, end, and break times
- Keep an open mind
- Participate at your comfort level
- Think about and apply workshop information
- Ask questions any time
- Share your relevant stories
- Respect others' views
- Put aside other work and problems, and avoid outside interruptions
- Most importantly - Have fun!

# Introduction

## Workshop Logistics

Use the space below to record the logistics for this workshop.

| | |
|---|---|
| **Facilitator's name** | |
| **Facilitator's e-mail** | |
| **Start/end times** | |
| **Lunch (approximate)** | |
| **Breaks** | |

## Workshop Materials

The materials used in this workshop include:

1. **Participant Guide**, including:
   - Workshop discussion material, worksheets and templates
   - Case study background material
   - BA tools, techniques & memory joggers
2. **Appendices**, including:
   - Recommended reading
   - Glossary of terms
   - Index
   - Case study exercises
   - Blank paper
3. **Handouts for case study**
   - Suggested solutions for exercises

# *Module 2* The Building Blocks

# The Building Blocks

**In this module, you will:**

- *Review the role of the business analyst and the system development life cycle*
- *Review and define criteria for ensuring successful project outcome*
- *Review the new project initiative for a case study company in preparation for analysis, design, testing and implementation activities*

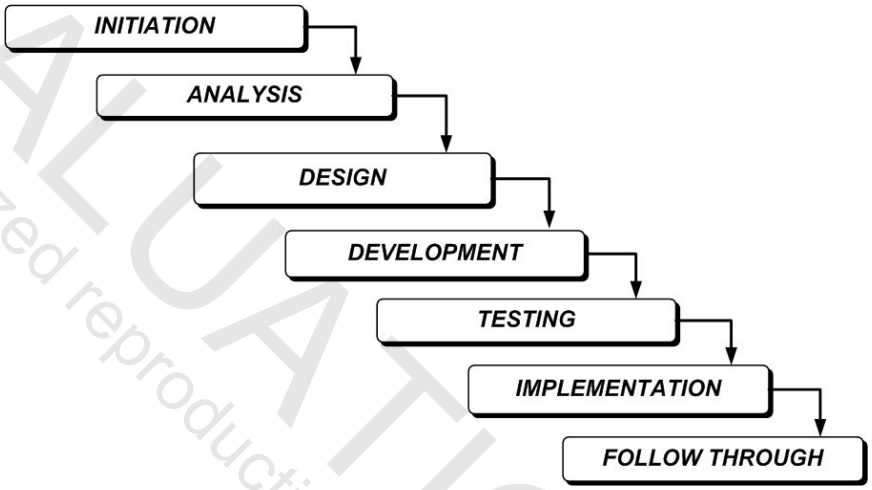


Figure 2.1 The SDLC

## The Building Blocks

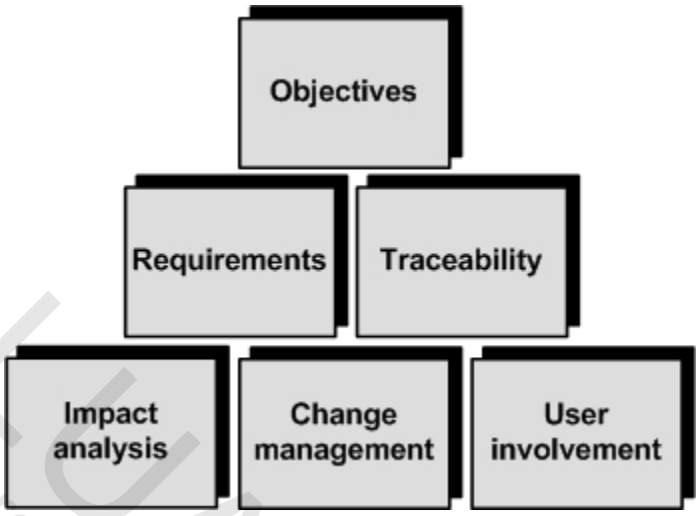


Figure 2.2 The Building Blocks

- **Project objectives** – are provided in the project scope document
- **Clear requirements** – will be created in the form of use cases, activity diagrams, class/object diagram, entity relationship diagram and story boards
- **Traceability** – will be established between requirements and two other project components
- **Impact analysis** – is necessary to evaluate the assumptions, constraints and potential risks and consider what must be done to plan for, manage or eliminate the impact
- **Change management** – will be performed throughout the project to handle a variety of real life scenarios
- **User involvement** – will be provided as inputs to establish business needs, validate information, gain approval and perform acceptance testing

# The Building Blocks

## The Scope of Business Analysis

Business analysis is a complex set of skills and capabilities, tools and techniques, challenges and rewards that defines job role vital to organizations today. The International Institute of Business Analysis (IIBA) defines it as follows:

> ***Business analysis is the set of tasks and techniques used to work as a liaison among stakeholders in order to understand the structure, policies, and operations of an organization, to recommend solutions and enable the organization to achieve its goals.***[1]

The key word to note in the definition of business analysis is "business" – the organization is looking to identify and implement solutions to solve ***business problems***. This is a critical objective that we will be coming back to time and again in this workshop.

Solutions to business problems often include the following three components to ensure that a holistic and complete perspective of the solution is taken. The ultimate project implementation plan should reflect the integrated nature of the solution:


Figure 2.3 The Solution Components

Many times, it is the process component – new processes, or re-designed processes, that define the systems and organization components. In this workshop, the impact and change to all three components will be discussed. Although they are depicted as equal in size in the illustration above, we may discover that one component has more change associated with it than another.

[1] International Institute of Business Analysis (IIBA), Business Analysis Body of Knowledge (BABOK) version 2.0, 2009

## The System Development Life Cycle and this Workshop

The focus of this workshop will be the activities a business analyst may be involved with in order to analyze, design, develop, test and implement a solution, once the project has been approved and scope and requirements agreed upon. As in real life we will continuously verify that the work being performed is still aligned to the original business case and the requirements.

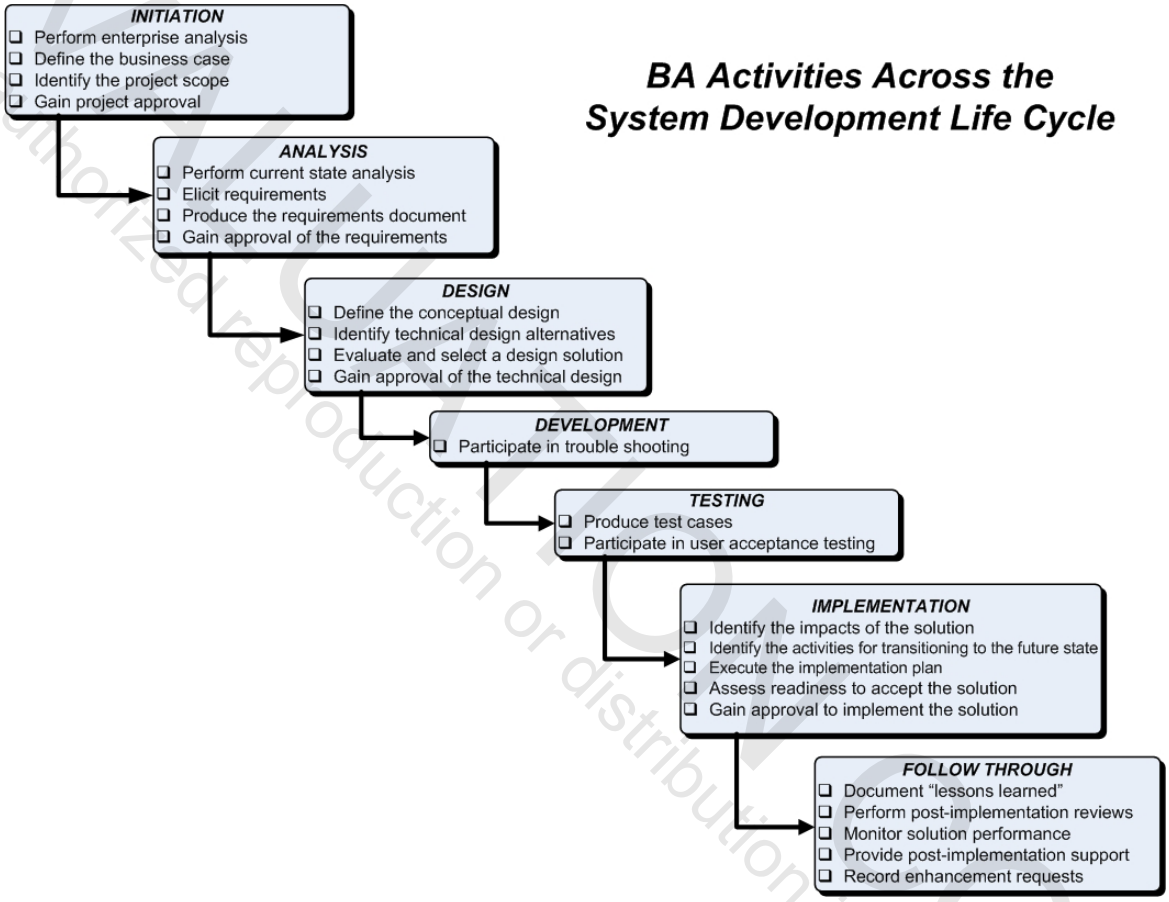


Figure 2.4 The SDLC

# The Building Blocks

## Critical Success Factors

One of the most difficult tasks to undertake is to predict whether a project will be successful. But, what is the definition of success? We can think about the criteria for a successful implementation from two perspectives:

1. **The project manager perspective** – on-time, within budget, managed scope, completed by resources assigned.
2. **The business analyst perspective** – fixing issues, achieving goals and objectives, managing scope, fulfilling requirements, meeting stakeholder's expectations.

Success from the business analyst perspective can be defined as:

> ***Ensuring that the solution, as expressed by the requirements, process changes, system design and organizational changes, meets the stakeholders' needs and expectations, and delivers the benefits identified in the business case.***

### Success criteria

What are your criteria to ensure the successful implementation of a solution? These should be components that the analysts will have some control over and will be used once all activities are complete to evaluate the project outcome.

## Assumptions

**Assumptions**

- ***Things that are believed to be true, but have not been confirmed.***
- ***Business assumptions are provided to the project team to inform them of key stakeholder expectations.***
- ***Requirements assumptions are added by the Business Analyst to transfer business domain knowledge to the project team.***
- ***Assumptions may be a source of potential project risk.***

When little or no information is available, but the outcome is necessary to consider because of the impact to the project, the analyst must make an assumption. These assumptions should be written down, so that they can be communicated to management, considered during design and assessed as new information becomes available. Some examples of assumptions are as follows:

*All workstations in the company will support the updated version of the software.*

> It will be necessary to validate the hardware, operating system and software compatibility to remove this assumption.

*The existing agents will be able to handle the new call volume for the sales promotion.*

> Until the promotion gets underway and the actual call volume is determined, it will be hard to assess if this assumption is correct or needs to be revised.

*The documented process accurately reflects the normal sequence of tasks actually being performed.*

> Documentation easily becomes outdated and otherwise inaccurate. People can perform the activities in different order, perform "work arounds" when they encounter issues, and change work patterns to prevent boredom.

*Information provided during stakeholder interviews is correct.*

> We should always validate with more than one stakeholder to better understand and confirm complaints, effects, needs and requirements.

*The newly designed processes will create efficiencies and save time performing the work.*

> Until people are properly trained and have overcome the learning curve there will initially be some performance concerns.

# The Building Blocks

## Constraints

Unlike assumptions, constraints are usually fact based and set limitations on the design. They should also be identified and documented prior to design work so that the solution will fit within the boundaries of the constraints. **Constraints may be documented as quality of service requirements.**

**Constraints**

- ***Business constraints describe limitations on the projects flexibility to adopt a desired solution.***
- ***Technical constraints define architectural decisions, specify restrictions, and identify standards that must be adhered to.***

Business constraints may be based on budget, time limits or resources, other project dependencies, management approval, regulations, legislation or other legal restrictions, auditing or accounting compliance, security or personnel access, user skill sets, etc. Some examples of business constraints are:

*The new accounting system must be in compliance with the Sarbanes-Oxley Act 2002.*

*Security clearance is revoked for level 1 and level 2 employees from 6:00pm to 6:00am from Monday through Friday.*

*The server project must be implemented before the new human resources project can be started.*

Technical constraints deal with system specific components like network and server utilization, software features, programming languages, compatibility, interoperability, access methods, technical investments, security restrictions, transmission rates, storage capacity, etc. Some examples of technical constraints are:

*All desktop workstations will use the Windows 7 image with standard applications.*

*The server's storage capacity for the database is limited to 200 Gb.*

*The IP voice traffic shall not fall below the standards for toll quality voice transmission.*

**Assumptions**

# The Building Blocks

## Constraints

# *Module 4* Design

# Design

**In this module, you will:**

- *Identify data model elements (class, objects, entities)*
- *Create a data model (class diagram, entity relationship diagram)*
- *Produce a storyboard to represent the user interface*
- *Evaluate technical alternatives and recommend a solution*

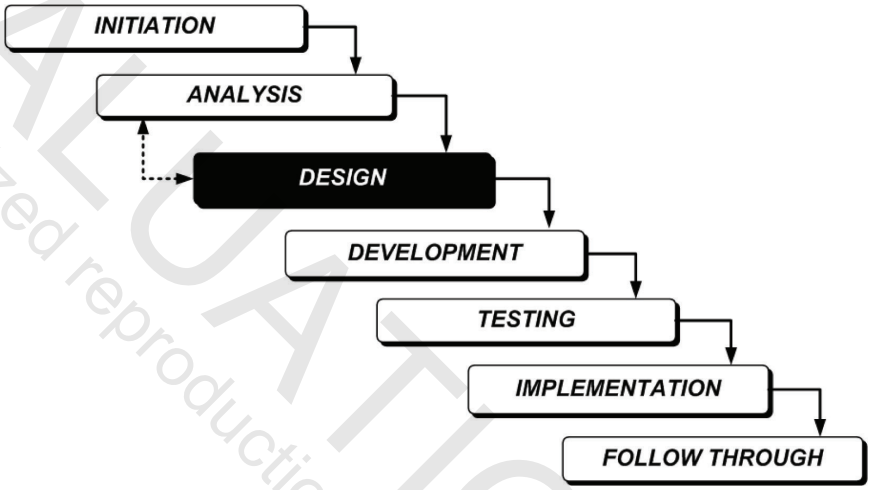


Figure 4.1 SDLC

## The Design Phase

Welcome to technical design. Remember that the analysis phase, focused on requirements, is all about the "what" while the design phase is focused on the "how."

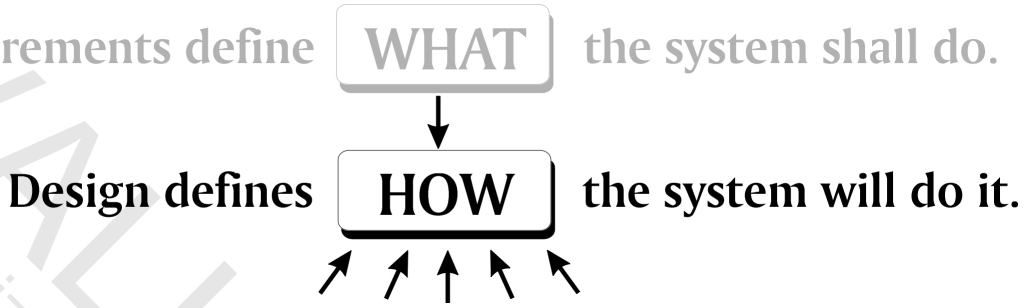


Figure 4.2 What not How

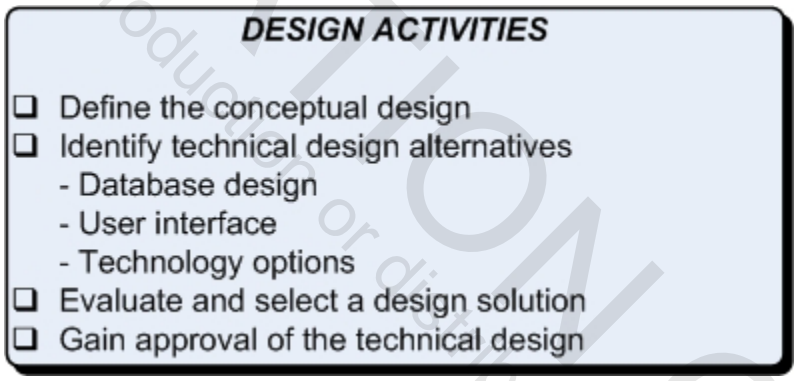


Figure 4.3 Design Phase Activities

# Design

## Object Oriented Approach

As mentioned in the introduction, there are two approaches to analysis and design. Object orientation is an approach that views a system as a collection of things or objects that work together to accomplish tasks. The **Unified Modeling Language** (UML) is the tool to depict requirements and design elements based on object orientation.

Object orientation includes the use of specific concepts and UML elements which are described on the following pages:

- Class
- Object
- Message
- Class Diagram

Entities defined in Information Engineering (IE) have many of the same characteristics as the UML class/object concept with some subtle differences. For the purpose of this workshop, the class and entity terms are used interchangeably in identifying the data elements of a process or system.

### Criteria for Identifying "Things"

1. Is a real or tangible thing (noun)
2. Is important in the context of the business
3. Is something for which we store or use data
4. Is something without which we can not do the process
5. Has two or more attributes

## UML Model Elements

### Class

***A class is an abstraction of a set of objects that share common attributes, behaviors, and meanings.***

For example, **`Product`** is a class which abstracts the things in a warehouse. Class membership is based on one or more *attributes* of the objects. In the case of the `Product` class, every member object knows its `PartNumber` and its `PartClass`, which are attributes of each and every object in the class. The behavior of each member of the class is specified by the services `purchase()` and `inquire()`.

An example class is given below.

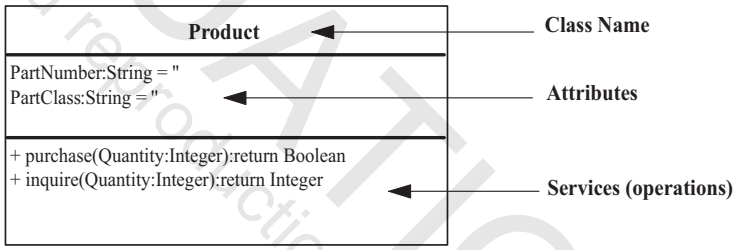


Figure 4.4 A Class

A class encapsulates behavior (services) and state (attributes). Classes are used, initially, to model the vocabulary of a system and to organize thinking about that system. The prototypes of a class are given in the figure below.

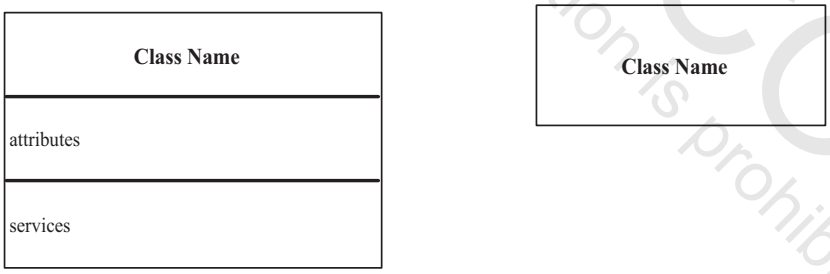


Figure 4.5 Class Examples

Depending on the phase of the SDLC and the amount of information available, a class may be defined with all three parts filled in to some degree, or it may just appear as a name. You may choose the level of detail that is appropriate for the audience understanding.

# Design

## Object

***An object is an instance of a class. An object has a well-defined boundary which hides the details of its behavior and the data on which it operates.***

For example, `DataCompressionKit` and `NetworkRouter` are objects in the class `Product`. When brought to life in a computer system, an object knows its attributes and the services it can perform as defined in its parent class. In the following figure, the state of `DataCompressionKit` is specified by the values assigned to the attributes `PartNumber` and `PartClass`, and the behavior is defined by the services `purchase()` and `inquire()`.

An example object is given below.

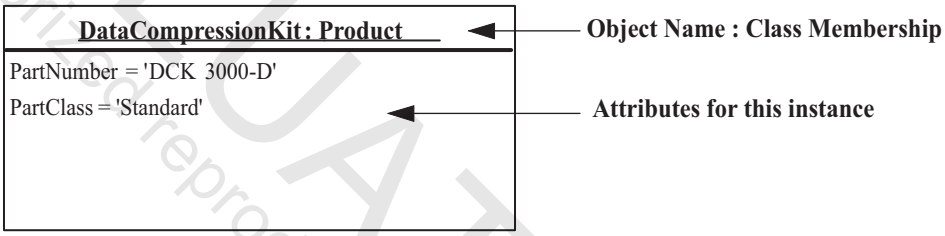


Figure 4.6 An Object

## Message

***A message is the way that objects interact. The message is the function and therefore depicts the functional behavior of the system.***

Using messages between objects is described as the *sender-receiver paradigm*. When an object receives a message, it "comes alive" and reacts to the message through a corresponding service.

An example of a product `inquire` message sent from the `HugeAircraft` object instance of the class `Customer` to the `DataCompressionKit` object instance of the class `Product` is given below. `Inquire()` is a service defined in the `Product` class.

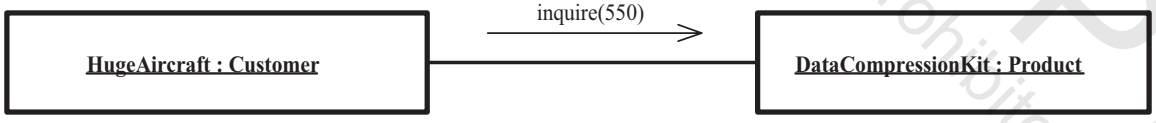


Figure 4.7 A Message

Eventually, you will notice that class : objects cluster into groups. Ultimately, you will define how they relate to one another by constructing a *class diagram*, which presents a structural view of the system.

## Class Diagram

The following figure illustrates a class diagram containing classes, associations with *multiplicity* indicators[3], and a hybrid called an *association class*. The model is interpreted as follows:

- A `Customer` class may or may not be associated with multiple instances of the `Contact` classes.
- A `Contact` class must be associated with at least one instance of the `Customer` class.
- The association class, `CUSTOMER_CONTACT`, may be viewed either as an association with class properties, or as a class with association properties. In this case, `Description` is a property belonging to a specific `Customer` to `Contact` instance pair.

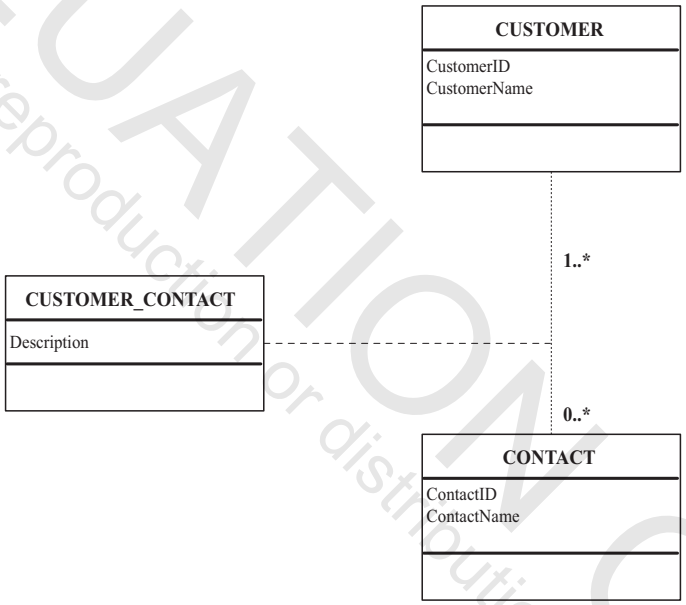


Figure 4.8 A Class Diagram

**Class Diagram Symbols**

| Multiplicity | Explanation |
|---|---|
| 0 | None |
| 1 | One |
| N | Specific number |
| * | Infinite number (many) |
| .. | Separator |

[3] The term *multiplicity* is used in the UML in place of IE *cardinality*.

# Design

## Information Engineering Approach

IE uses a different model to represent data, although the key concepts of "things" and relationships are the same. This is called an Entity Relationship Diagram (ERD).

*One customer may submit zero to many purchase orders – this is shown with 0 indicated with an "o" and "crows feet" to indicate many possible purchase orders.*

*One purchase order is associated with only one customer – This is shown with the "||" connector indicating one and only one.*


Figure 4.9 Entity Relationship Diagram

Here's how IE works:

1. Each **entity** is represented by a rectangle
2. A line represents the **relationship** between two entities
3. There are many types of relationships between entities. This is called **cardinality** and is represented by a line called a **connector**
4. Each relationship is read two ways: from left to right; and from right to left

| IE Cardinality Symbol | Explanation |
|---|---|
| | Simple relationship (used at conceptual level, only) |
| | One to many (used at conceptual level, only) |
| | Many (used at conceptual level, only) |
| | One or more; |
| | None or more; |
| | One and only one; |
| | None or one; |

Figure 4.10 Entity Diagram Symbols

## Designing Usability into the Solution

What does the user interface look like? How does a user navigate around the system? What is the layout of each screen or web page? Is everything just a few "clicks" away? The design of the user interface determines how users will interact with the system, and how the system will respond to user actions. Fonts, colors, windows, dialog boxes, prompts, and menu bars all go into the design considerations. The interface needs to capture inputs and produce outputs in the form of information, graphics or pictures.

**Storyboard**

***A storyboard is a graphical representation of the user interface and the sequence of steps a user goes through to complete a process.***

### An example of a storyboard

The illustration that follows is of a storyboard for a process to check a customer's order. After entering the customer number, a screen displays showing the available processes that can be performed. If the user selects the last option, "Check order status" the Order History screen displays. The user selects an order and the details of that order display on the Order Details screen. The arrows indicate navigation. Here, the user can go back and forth between Order History and Order Details, and the Main Menu.

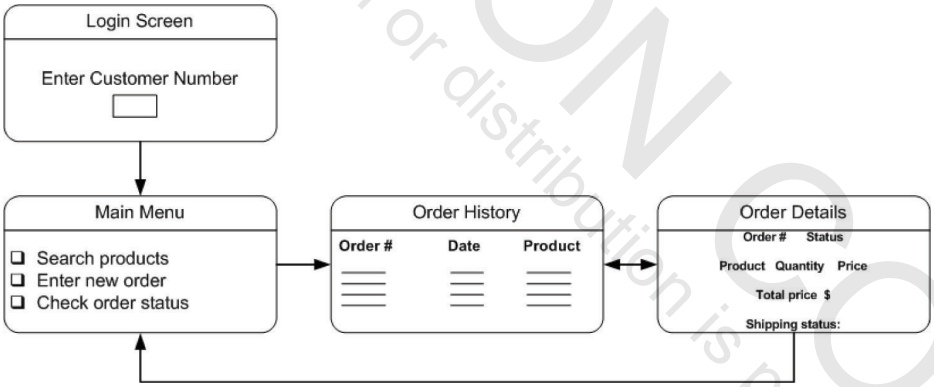


Figure 4.11 Storyboard Example

Notice that the items on the Main Menu are likely to be the use cases identified during analysis. The business analyst will need to help identify the design-specific business rules for developing this storyboard – how many orders should the page display? Should the Order History screen display all orders past and pending? Or representing just 12 months of history? The answer might depend on how easy it is to retrieve history. Perhaps orders over 5 years have been archived?

# Design

## Creating a storyboard

An effective way to create a storyboard is to construct the conversation a user would have with someone, such as a customer order representative, in order to complete a task. Then change perspectives: if the manual processes were to be replaced by an automated system, what tasks would the system perform in response to the user? If written correctly, this dialog should be well represented in the use case. In some situations, business analysts may actually use dialogs and storyboards in the analysis phase in order to capture requirements and produce use cases that can be more easily validated with visuals, in addition to words.

Have a dialog with a user or customer or observe them performing a process and:

- Listen for / look for the steps or tasks in the process → inputs and outputs in the solution
- Listen for / look for different ways to complete the process → navigation and alternative scenarios
- Make sure you understand when the user has completed the process → the last step in the "story" and the post-condition
- Imagine what can go wrong in the process → exception scenarios

There are many guidelines, books and websites on designing user interfaces. Some companies have "human factors engineers" or graphics designers to help with this critical design activity.

The audience for the storyboard at this point in the project includes the entire technical team and the users. Remember – it does not show everything, but the high-level concept of what the user interface looks like.

## Technical Solution Selection

The team may be faced with the following decisions in the selection process:

- Should a manual process be automated, or simply redesigned? In other words, what level of automation should be included in the solution?
- Should a new system be built or should a commercial off-the-shelf (COTS) software package be purchased and customized?
- Is this a system that can be hosted and managed by a third party service provider?
- Should existing systems be modified as an alternative to buying or building a new system?
- Is an investment in the technology infrastructure needed? Or can the proposed solution run on the existing hardware and software?
- Does the existing infrastructure have the capacity to add new functionality without negatively effecting existing processing capabilities?
- What does the user interface look like?
- What are the impacts of the potential solution alternatives to other systems, processes or people within the organization?
- What are the technical capabilities of the users and their level of acceptance for adapting to the new system?
- Will the technology potentially reduce or replace the need for people performing the work?
- How stable is the technologies and architectures being considered? (stable and well known, new or leading edge, ahead of the curve or bleeding edge).

Note, some of these decisions may have been addressed in a feasibility study in the initiation phase. In some organizations, the BA is also involved with that activity.