Marketing & Sales Requirements Specification

Version 1.0

August 28, 2018

The responsibility to communicate that information rests on the shoulders of the marketing and sales teams. Typically, marketing has a predominant role at the beginning of a potential sale. For example, a marketing team may develop a new radio campaign to help spread awareness about a product launch. A sales team works to finalize a deal by communicating directly with leads and addressing their concerns. Another advantage sales and marketing teams exploit, is collaboration. Rather than operating as independent units, strong information and idea sharing between the teams can help improve results and create a seamless experience for prospective buyers. Sales includes operation and activities involved in promoting and selling goods and services and marketing includes the process or technique of promoting, selling and distributing a product or service.

Table of Contents

1. [1. Executive Summary 3](#_Toc191724230)

[1.1 Project Overview 3](#_Toc191724231)

[1.2 Purpose and Scope of this Specification 3](#_Toc191724232)

1. [2. Product/Service Description 3](#_Toc191724233)

[2.1 Product Context 3](#_Toc191724234)

[2.2 User Characteristics 3](#_Toc191724235)

[2.3 Assumptions 3](#_Toc191724236)

[2.4 Constraints 3](#_Toc191724237)

[2.5 Dependencies 4](#_Toc191724238)

1. [3. Requirements 4](#_Toc191724239)

[3.1 Functional Requirements 5](#_Toc191724240)

[3.2 User Interface Requirements 5](#_Toc191724241)

[3.3 Usability 5](#_Toc191724242)

[3.4 Performance 6](#_Toc191724243)

[3.4.1 Capacity 6](#_Toc191724244)

[3.4.2 Availability 6](#_Toc191724245)

[3.4.3 Latency 6](#_Toc191724246)

[3.5 Manageability/Maintainability 6](#_Toc191724247)

[3.5.1 Monitoring 6](#_Toc191724248)

[3.5.2 Maintenance 6](#_Toc191724249)

[3.5.3 Operations 6](#_Toc191724250)

[3.6 System Interface/Integration 7](#_Toc191724251)

[3.6.1 Network and Hardware Interfaces 7](#_Toc191724252)

[3.6.2 Systems Interfaces 7](#_Toc191724253)

[3.7 Security 8](#_Toc191724254)

[3.7.1 Protection 8](#_Toc191724255)

[3.7.2 Authorization and Authentication 8](#_Toc191724256)

[3.8 Data Management 8](#_Toc191724257)

[3.9 Standards Compliance 8](#_Toc191724258)

[3.10 Portability 8](#_Toc191724259)

1. [4. User Scenarios/Use Cases 9](#_Toc191724260)
2. [5. Deleted or Deferred Requirements 9](#_Toc191724261)
3. [6. Requirements Confirmation/Stakeholder sign-off 10](#_Toc191724262)
4. [APPENDIX 11](#_Toc191724263)

[Appendix A. Definitions, Acronyms, and Abbreviations 11](#_Toc191724264)

[Appendix B. References 11](#_Toc191724265)

[Appendix C. Requirements Traceability Matrix 11](#_Toc191724266)

[Appendix D. Organizing the Requirements 13](#_Toc191724267)

# Executive Summary

## Project Overview

The project entitled Marketing & Sales. Because many small businesses don’t have the expertize or even need to pursue classical marketing strategy, the sales manager handles marketing duties as part of his responsibilities. The sales division takes the lead in settings strategies and decides what marketing communications it needs to support efforts.

## Purpose and Scope of this Specification

In scope

Increase sales by promoting wine products

Build brand awareness

Target new customers

Enhance customer relationships.

Increase profit

Increase the reach of company’s product at more number of customer

To achieve sales target

Monitor the sales of the business

**Out of Scope**

The following items are out of scope:

This module is for marketing and sales only

# Product/Service Description

Wine distillery objectives is to produce a high quality wines. The wines that will be created will be distributed to small store here in Nasugbu Batangas. The winery will provide an enjoyable lifestyle and adequate standard living for the employees and customer.

## Product Context

Over the course of human history, and using a system of trial, error, and careful observation, different cultures began producing fermented beverages. Mead, or honey wine, was produced in Asia during the Vedic period and the Greeks, Celts, Saxons, and Vikings also produced this beverage. In Egypt, Babylon, Rome, and China, people produced wine from grapes and beer from malted barley. In South America, people produced *chicha* from grains or fruits, mainly maize; while in North America, people made *octli*(now known as "pulque") from agave, a type of cactus.

## User Characteristics

Create general customer profiles for each type of user who will be using the product. Profiles should include:

Adult people(18 and above)

May have ability to communicate by processing their product.

Understand the flow of system that they use.

The language that provide must be read and understand.

Have an experience to socialize to the main manager of the product.

## Assumptions

List any assumptions that affect the requirements, for example, equipment availability, user expertise, etc. For example, a specific operating system is assumed to be available; if the operating system is not available, the Requirements Specification would then have to change accordingly.

Customer (Assumptions about the customer needs, perceptions

Social media is always profitable venture

## Constraints

Describe any items that will constrain the design options, including

Time

Schedule (Date that need to be met for the strategy to succeed)

Data (gathering data requirements)

System resource constraints (limits on disk space, other hardware limitations)

Other design constraints (design or other standards, such as programming language or framework)

## Dependencies

List dependencies that affect the requirements.

Accounting and Finance

The sales report will given to the accounting and finance to know if the sales are improving or not. The marketing and sales needs some information also in accounting like the price of the product

Supply Chain Management Systems

The Marketing and sales module needs an inventory of the finish product information to market the product and also need some information like what is the available product

# Requirements

Describe all system requirements in enough detail for designers to design a system satisfying the requirements and testers to verify that the system satisfies requirements.

Organize these requirements in a way that works best for your project. See Appendix DAppendix D, Organizing the Requirements for different ways to organize these requirements.

Describe every input into the system, every output from the system, and every function performed by the system in response to an input or in support of an output. (Specify what functions are to be performed on what data to produce what results at what location for whom.)

Each requirement should be numbered (or uniquely identifiable) and prioritized.

See the sample requirements in Functional Requirements, and System Interface/Integration, as well as these example priority definitions:

Priority Definitions

The following definitions are intended as a guideline to prioritize requirements.

* Priority 1 – The requirement is a “must have” as outlined by policy/law
* Priority 2 – The requirement is needed for improved processing, and the fulfillment of the requirement will create immediate benefits
* Priority 3 – The requirement is a “nice to have” which may include new functionality

It may be helpful to phrase the requirement in terms of its priority, e.g., "The value of the employee status sent to DIS **must be** either A or I" or "It **would be nice** if the application warned the user that the expiration date was 3 business days away". Another approach would be to group requirements by priority category.

A good requirement is:

* Correct
* Unambiguous (all statements have exactly one interpretation)
* Complete (where TBDs are absolutely necessary, document why the information is unknown, who is responsible for resolution, and the deadline)
* Consistent
* Ranked for importance and/or stability
* Verifiable (avoid soft descriptions like “works well”, “is user friendly”; use concrete terms and specify measurable quantities)
* Modifiable (evolve the Requirements Specification only via a formal change process, preserving a complete audit trail of changes)
* Does not specify any particular design
* Traceable (cross-reference with source documents and spawned documents).

## Functional Requirements

In the example below, the requirement numbering has a scheme - BR\_LR\_0## (BR for Business Requirement, LR for Labor Relations). For small projects simply BR-## would suffice. Keep in mind that if no prefix is used, the traceability matrix may be difficult to create (e.g., no differentiation between '02' as a business requirement vs. a test case)

The following table is an example format for requirements. Choose whatever format works best for your project.

For Example:

| Req# | Requirement | Comments | Priority | Date Rvwd | Reviewed / Approved |
| --- | --- | --- | --- | --- | --- |
| * BR\_LR\_05 | * The system should associate a supervisor indicator with each job class. | * Business Process = “Maintenance | * 3 | * 7/13/04 | * Bob Dylan, Mick Jagger |
| * BR\_LR\_08 | * The system should handle any number of fees (existing and new) associated with unions. | * Business Process = “Changing Dues in the System” * An example of a new fee is an initiation fee. | * 2 | * 7/13/04 | * Bob Dylan, Mick Jagger |
| * BR\_LR\_10 | * The system should capture and maintain job class status (i.e., active or inactive) | * Business Process = “Maintenance” * Some job classes are old and are no longer used. However, they still need to be maintained for legal, contract and historical purposes. | * 2 | * 7/13/04 | * Bob Dylan, Mick Jagger |
| * BR\_LR\_16 | * The system should assign the Supervisor Code based on the value in the Job Class table and additional criteria as specified by the clients. | * April 2005 – New requirement. It is one of three new requirements from BR\_LR\_03. | * 2 |  |  |
| * BR\_LR\_18 | * The system should provide the Labor Relations office with the ability to override the system-derived Bargaining Unit code and the Union Code for to-be-determined employee types, including hourly appointments. | * April 2005 – New requirement. It is one of three new requirements from BR\_LR\_04. * 5/11/2005 – Priority changed from 2 to 3. | * ~~2~~ * 3 |  |  |

## User Interface Requirements

The user interface must be secure, convenient and extensible. Security requirements include the need to protect authorization information from unauthorized access, the maintenance of customer confidentiality. The user interface should be efficient in both speed and use.

User Confidence. User must be confident use the interface without making unintended purchases. Users have very exacting expectations of any system which involves money. It is essential that such systems provide the user with confidence in their design and implementation. Otherwise a system is likely to encounter overwhelming consumer resistance and fail to gain acceptance.

Efficiency. The user interface should be efficient in terms of both speed and user interaction. Unnecessary user interactions should be avoided. Unnecessary communications introduce both delay and additional potential failures.

Security of customer record. The user must be aware of all information that they need to give which is to be communicate to the management of the product to accept the records form them.

## Usability

The user can easily use it if they will want to explore how the process of our system will work, every objectives of it must achieve that belongs for every process and the effectiveness and also the satisfaction in a quantified context of use.

## Performance

Specify static and dynamic numerical requirements placed on the system or on human interaction with the system:

Static numerical requirements may include the number of terminals to be supported, the number of simultaneous users to be supported, and the amount and type of information to be handled.

Dynamic numerical requirements may include the number of transactions and tasks and the amount of data to be processed within certain time period for both normal and peak workload conditions.

All of these requirements should be stated in measurable form. For example, "95% of the transactions shall be processed in less than 1 second" rather than “an operator shall not have to wait for the transaction to complete”.

### Capacity

Sales teams apply capacity planning by quantifying individual sales performance, developing sales goals based on the aggregate of the total “production capacity” then finding ways to increase that capacity through sales. It’s the process by which you measure and quantify the current capability of your sales team, compare that capacity to the sales goals you want to hit.

### Availability

Include specific and measurable requirements for:

Hours of operation

Level of availability required

Coverage for geographic areas

Impact of downtime on users and business operations

Impact of scheduled and unscheduled maintenance on uptime and maintenance communications procedures

reliability (e.g., acceptable mean time between failures (MTBF), or the maximum permitted number of failures per hour).

### Latency

Include explicit latency requirements, e.g., the maximum acceptable time (or average time) for a service request.

## Manageability/Maintainability

### Monitoring

Include any requirements for product or service health monitoring, failure conditions, error detection, logging, and correction.

### Maintenance

Specify attributes of the system that relate to ease of maintenance. These requirements may relate to modularity, complexity, or interface design. Requirements should not be placed here simply because they are thought to be good design practices.

### Operations

Specify any normal and special operations required by the user, including:

periods of interactive operations and periods of unattended operations

data processing support functions

backup and recovery operations

safety considerations and requirements

disaster recovery and business resumption

## System Interface/Integration

* **Ease of use.** Look for software with a user-friendly graphical interface.
* **Entry of sales information.** Most systems allow you to enter inventory and also the report. Many systems make it easy to enter sales manually when needed by letting you search for report.
* **Updating product information.** Once a sale is entered, these systems automatically update inventory.
* **Sales tracking options.** Different businesses get paid in different ways. For example, repair or service shops often keep invoices open until the work is completed, so they need a system that allows them to put sales on hold.
* **Security.** In retail, it's important to keep tight control over cash receipts to prevent theft. Most of these systems provide audit trails so you can trace any problems.

### Network and Hardware Interfaces

Specify the logical characteristics of each interface between the product and the hardware or network components of the system. This includes configuration characteristics (e.g., number of ports, instruction sets), what devices are to be supported, and protocols (e.g., signal handshake protocols).

### Systems Interfaces

Marketing and Sales system interfaces is compose of Dashboard including Sales, Products, Customer, Suppliers, and Sales Report. Sales interface will generate all the transactions including customer cash and change and some information of the company. In Product interface, wine products are displayed will all the information and also you can add, delete and update the selected wine product. Customer interface will provide all the information of the customer that bought the wine products including dates, wine name, address etc. In supplier interface will generate the suppliers of the raw materials in producing wines and also you can add new supplier and necessary information and then displayed it. Sales report generate the recommended sales report the wine distillery.

1. System1-to-System2 Interface

The <external party> will create and send a fixed length text file as an email attachment to [System2mail@u.washington.edu](mailto:heppsmai@u.washington.edu) to be imported into the System2 system for payroll calculation. This file must be received on EDIT day by 4:00 PM in order to be processed in the EDIT night run. The requirements below document the file specifications, data transfer process, and specific schedule. This file is referred to as "FileName" in this document.

File Structure and Format

* 1. The FileName file is a fixed length text file.
  2. The FileName file is an unformatted ASCII file (text-only).
  3. The FileName file contains a batch totals record and several detail records.

File Description: Batch Totals Record

* 1. The batch totals record can be placed at the beginning, in the middle, or at the end of the file.
  2. The batch totals record contains the following:

Record Type (value: XA)

Process Type (value: A)

Batch Number (3 digit number assigned by Payroll Dept)

Origin Code (AIG)

Total number of detail records

Total deduction amount

File Description: Detail Records

* 1. The FileName file contains a row for each record meeting xxx criteria.
  2. Each row in the FileName file contains the following fields, comma-delimited and encased in double-quotes where the data includes commas or spaces:
* Employee Id
* Record Type
* Process Date (MMDDYY)
* XYG Number
* Element Code
* Amount
* Amount Sign
* Year Flag
* Total Amount
* Total Amt Sign

## Security

### Protection

Specify the factors that will protect the system from malicious or accidental access, modification, disclosure, destruction, or misuse. For example:

encryption

data integrity checks

### Authorization and Authentication

Specify the Authorization and Authentication factors. Consider using standard tools such as PubCookie.

## Data Management

types of information used by various functions

frequency of use

data access rules

data entities and relationships

integrity constraints

data retention

valid range, accuracy, and/or tolerance

units of measure

data formats

default or initial values

## Standards Compliance

Specify the requirements derived from existing standards, policies, regulations, or laws (e.g., report format, data naming, accounting procedures, audit tracing). For example, this could specify the requirement for software to trace processing activity. Such traces are needed for some applications to meet minimum regulatory or financial standards. An audit trace requirement may, for example, state that all changes to a payroll database must be recorded in a trace file with before and after values.

## Portability

If portability is a requirement, specify attributes of the system that relate to the ease of porting the system to other host machines and/or operating systems. For example,

Percentage of components with host-dependent code;

Percentage of code that is host dependent;

Use of a proven portable language;

Use of a particular compiler or language subset;

Use of a particular operating system;

The need for environment-independence - the product must operate the same regardless of operating systems, networks, development or production environments.

# User Scenarios/Use Cases

Provide a summary of the major functions that the product will perform. Organize the functions to be understandable to the customer or a first time reader. Include use cases and business scenarios, or provide a link to a separate document (or documents). A business scenario:

Describes a significant business need

Identifies, documents, and ranks the problem that is driving the scenario

Describes the business and technical environment that will resolve the problem

States the desired objectives

Shows the “Actors” and where they fit in the business model

Is specific, and measurable, and uses clear metrics for success

# Deleted or Deferred Requirements

Identify any requirements that have been deleted after approval or that may be delayed until future versions of the system. For example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Req# | Business Requirement | Status | Comments | Pri | Date Rvwd | SME Reviewed /Approved |
| * BR\_LR\_01 | * The system should validate the relationship between Bargaining Unit/Location and Job Class. | * April 2005: Deleted. * This requirement has been replaced by BR\_LR\_036 and BR\_CC\_33. | * Business Process = “Assigning a Bargaining Unit to an Appointment” | * 1 | * 7/13/04 | * Bob Dylan, Mick Jagger |
| * BR\_LR\_02 | * The system should validate that the supervisor indicator is correct according to job class. * Deferred to Phase 2B: 3/29/2005 | * April 2005: Deferred to Phase 2B. | * Business Process = “Assigning a Bargaining Unit to an Appointment” | * 3 | * 7/13/04 | * Bob Dylan, Mick Jagger |
| * BR\_LR\_03 | * The system should derive the bargaining unit code, union code, and supervisor indicator from the job class code and location. | * April 2005: Deleted * Replaced by BR\_LR\_16 and BR\_LR\_17. | * Business Process = “Assigning a Bargaining Unit to an Appointment”; This will eliminate the need, typically, for the user to enter the bargaining unit code, union code and supervisor indicator. | * 1 | * 7/13/04 | * Bob Dylan, Mick Jagger |

# Requirements Confirmation/Stakeholder sign-off

Include documentation of the approval or confirmation of the requirements here. For example:

|  |  |  |
| --- | --- | --- |
| Meeting Date | Attendees (name and role) | Comments |
| * 7/13/07 | * Bob Dylan, Labor Relations SME * Mick Jagger, Labor Relations SME * Ringo Starr, Technical Project Manager * Debbie Harry, Technical Analyst * Janis Joplin, Technical Analyst * Fred Meyer, Project Manager | * Confirmed BR\_LR\_01 – BR\_LR\_15 |
| * 04/15/05 | * Bob Dylan, Labor Relations SME * Mick Jagger, Labor Relations SME * Ringo Starr, Technical Project Manager | * Deferred / Deleted: BR\_LR\_01 - BR\_LR\_04, BR\_LR\_07, BR\_LR\_12, BR\_LR\_14, BR\_LR\_15, BR\_LR\_06, BR\_LR\_17 |

APPENDIX

The appendixes are not always considered part of the actual Requirements Specification and are not always necessary. They may include

Sample input/output formats, descriptions of cost analysis studies, or results of user surveys;

Supporting or background information that can help the readers of the Requirements Specification;

A description of the problems to be solved by the system;

Special packaging instructions for the code and the media to meet security, export, initial loading, or other requirements.

When appendixes are included, the Requirements Specification should explicitly state whether or not the appendixes are to be considered part of the requirements.

Definitions, Acronyms, and Abbreviations

Define all terms, acronyms, and abbreviations used in this document.

References

List all the documents and other materials referenced in this document.

Requirements Traceability Matrix

The following trace matrix examples show one possible use of naming standards for deliverables (FunctionalArea-DocType-NN). The number has no other meaning than to keep the documents unique. For example, the Bargaining Unit Assignment Process Flow would be BUA-PF-01.

For example (1):

| **Business Requirement** | **Area** | **Deliverables** | **Status** |
| --- | --- | --- | --- |
| * BR\_LR\_01 * The system should validate the relationship between Bargaining Unit/Location and Job Class.---Comments: Business Process = "Assigning a Bargaining Unit to an Appointment" (Priority 1) | * BUA | * BUA-CD-01 * Assign BU Conceptual Design | * Accepted |
| * BUA-PF-01 * Derive Bargaining Unit-Process Flow Diagram | * Accepted |
| * BUA-PF-01 * Derive Bargaining Unit-Process Flow Diagram | * Accepted |
| * BR\_LR\_09 * The system should provide the capability for the Labor Relations Office to maintain the job class/union relationship.---Comments: Business Process = "Maintenance" (Priority 1) | * BUA | * BUA-CD-01 * Assign BU Conceptual Design | * Accepted |
| * BUA-PF-02 * BU Assignment Rules Maint Process Flow Diagram | * ReadyForReview |

For example (2):

| **BizReqID** | **Pri** | **Major Area** | **DevTstItems DelivID** | **Deliv Name** | **Status** |
| --- | --- | --- | --- | --- | --- |
| * BR\_LR\_01 | * 1 | * BUA | * BUA-CD-01 | * Assign BU Conceptual Design | * Accepted |
| * BR\_LR\_01 | * 1 | * BUA | * BUA-DS-02 | * Bargaining Unit Assignment DB Modification Description | * Accepted |
| * BR\_LR\_01 | * 1 | * BUA | * BUA-PF-01 | * Derive Bargaining Unit-Process Flow Diagram | * Accepted |
| * BR\_LR\_01 | * 1 | * BUA | * BUA-UCD-01 | * BU Assign LR UseCase Diagram | * ReadyForReview |
| * BR\_LR\_01 | * 1 | * BUA | * BUA-UCT-001 | * BU Assignment by PC UseCase - Add Appointment and Derive UBU | * Reviewed |
| * BR\_LR\_01 | * 1 | * BUA | * BUA-UCT-002 | * BU Assignment by PC UseCase - Add Appointment (UBU Not Found) | * Reviewed |
| * BR\_LR\_01 | * 1 | * BUA | * BUA-UCT-006 | * BU Assignment by PC UseCase - Modify Appointment (Removed UBU) | * Reviewed |
| * BR\_LR\_09 | * 1 | * BUA | * BUA-CD-01 | * Assign BU Conceptual Design | * Accepted |
| * BR\_LR\_09 | * 1 | * BUA | * BUA-DS-02 | * Bargaining Unit Assignment DB Modification Description | * Accepted |
| * BR\_LR\_09 | * 1 | * BUA | * BUA-PF-02 | * BU Assignment Rules Maint Process Flow Diagram | * Accepted |
| * BR\_LR\_09 | * 1 | * BUA | * BUA-UCD-03 | * BU Assign Rules Maint UseCase Diagram | * Reviewed |
| * BR\_LR\_09 | * 1 | * BUA | * BUA-UCT-045 | * BU Assignment Rules Maint: Successfully Add New Assignment Rule | * Reviewed |
| * BR\_LR\_09 | * 1 | * BUA | * BUA-UCT-051 | * BU Assignment Rules MaintUseCase: Modify Rule | * Reviewed |
| * BR\_LR\_09 | * 1 | * BUA | * BUA-UCT-053 | * BU Assignment Rules MaintUseCase - Review Assignment Rules | * Reviewed |
| * BR\_LR\_09 | * 1 | * BUA | * BUA-UCT-057 | * BU Assignment Rules MaintUseCase: Inactivate Last Rule for a BU | * Reviewed |
| * BR\_LR\_09 | * 1 | * BUA | * BUA-UI-02 | * BU AssignRules Maint UI Mockups | * ReadyForReview |
| * BR\_LR\_09 | * 1 | * BUA | * BUA-TC-021 | * BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Success | * ReadyForReview |
| * BR\_LR\_09 | * 1 | * BUA | * BUA-TC-027 | * BU Assignment Rules Maint TestCase: Modify Rule - Success | * ReadyForReview |
| * BR\_LR\_09 | * 1 | * BUA | * BUA-TC-035 | * BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Error Condition | * ReadyForReview |
| * BR\_LR\_09 | * 1 | * BUA | * BUA-TC-049 | * BU Assignment Rules Maint TestCase: Modify Rule - Error Condition | * ReadyForReview |

For example (3):

| **BizReqID** | **CD01** | **CD02** | **CD03** | **CD04** | **UI01** | **UI02** | **UCT01** | **UCT02** | **UCT03** | **TC01** | **TC02** | **TC03** | **TC04** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| * BR\_LR\_01 |  |  | * X |  | * X |  | * X |  |  | * X |  | * X |  |
| * BR\_LR\_09 | * X |  |  | * X |  | * X |  |  | * X |  | * X |  | * X |
| * BR\_LR\_10 | * X |  |  | * X |  |  |  |  | * X |  | * X |  |  |
| * BR\_LR\_11 |  | * X |  |  |  |  |  |  |  |  |  |  |  |

Organizing the Requirements

This section is for information only as an aid in preparing the requirements document.

Detailed requirements tend to be extensive. Give careful consideration to your organization scheme. Some examples of organization schemes are described below:

By System Mode

Some systems behave quite differently depending on the mode of operation. For example, a control system may have different sets of functions depending on its mode: training, normal, or emergency.

By User Class

Some systems provide different sets of functions to different classes of users. For example, an elevator control system presents different capabilities to passengers, maintenance workers, and fire fighters.

By Objects

Objects are real-world entities that have a counterpart within the system. For example, in a patient monitoring system, objects include patients, sensors, nurses, rooms, physicians, medicines, etc. Associated with each object is a set of attributes (of that object) and functions (performed by that object). These functions are also called services, methods, or processes. Note that sets of objects may share attributes and services. These are grouped together as classes.

By Feature

A feature is an externally desired service by the system that may require a sequence of inputs to affect the desired result. For example, in a telephone system, features include local call, call forwarding, and conference call. Each feature is generally described in a sequence of stimulus-response pairs, and may include validity checks on inputs, exact sequencing of operations, responses to abnormal situations, including error handling and recovery, effects of parameters, relationships of inputs to outputs, including input/output sequences and formulas for input to output.

By Stimulus

Some systems can be best organized by describing their functions in terms of stimuli. For example, the functions of an automatic aircraft landing system may be organized into sections for loss of power, wind shear, sudden change in roll, vertical velocity excessive, etc.

By Response

Some systems can be best organized by describing all the functions in support of the generation of a response. For example, the functions of a personnel system may be organized into sections corresponding to all functions associated with generating paychecks, all functions associated with generating a current list of employees, etc.

By Functional Hierarchy

When none of the above organizational schemes prove helpful, the overall functionality can be organized into a hierarchy of functions organized by common inputs, common outputs, or common internal data access. Data flow diagrams and data dictionaries can be used to show the relationships between and among the functions and data.

Additional Comments

Whenever a new Requirements Specification is contemplated, more than one of the organizational techniques given above may be appropriate. In such cases, organize the specific requirements for multiple hierarchies tailored to the specific needs of the system under specification.

There are many notations, methods, and automated support tools available to aid in the documentation of requirements. For the most part, their usefulness is a function of organization. For example, when organizing by mode, finite state machines or state charts may prove helpful; when organizing by object, object-oriented analysis may prove helpful; when organizing by feature, stimulus-response sequences may prove helpful; and when organizing by functional hierarchy, data flow diagrams and data dictionaries may prove helpful.