



P U R D U E U N I V E R S I T Y

COMPUTER AND INFORMATION  
SYSTEMS TECHNOLOGY

**Documenting Requirements**  
**Prof. Kevin C. Dittman**  
***“Draft”***

## TABLE OF CONTENTS

Table of Contents.....	2
An Introduction to requirements discovery and specification .....	3
Analyzing requirements .....	8
Specifying requirements .....	8
Validating requirements .....	9
Documenting requirements .....	10
Requirement Number.....	10
Requirement Title.....	10
Requirement Text .....	10
Requirement Parent.....	10
Requirement Type .....	10
Revision Date and Revision Number .....	10
Verification Method .....	10
Precedence/Criticality .....	10

## AN INTRODUCTION TO REQUIREMENTS DISCOVERY AND SPECIFICATION

Requirements discovery for a system is dependent on the analysts' ability to first discover and then analyze problems and opportunities that exist in the current system—thus; requirements address the problems and opportunities!

**Problem analysis** is the activity of identifying the problem, understanding the problem (including causes and effects), and understanding any constraints that may limit the solution.

Lets examine the concepts of system requirements and the process of discovering and documenting those requirements. What are system requirements?

A **system requirement** (also called a business requirement) is a description of the needs and desires for an information system. A requirement may describe functions, features (attributes), and constraints.

System requirements define the services the system is to provide and prescribe constraints for its operation. In documenting the system requirements for a new information system, an analyst will likely identify dozens of unique requirements. To simplify the presentation of requirements and to make them more readable, understandable, and traceable, requirements are often categorized as functional versus non-functional.

A **functional requirement** is a function or feature that must be included in an information system in order to satisfy the business need and be acceptable to the users.

A functional requirement is an action of the system and usually is written using an action verb phrase. For example:

- process a checking account deposit,
- calculate the GPA for a student, and
- capture the account holder identification information,

are all examples of functional requirements. Many practitioners like to preface the above phrases with the words, "*The system shall...*" This is a simple testing mechanism that you can use to verify if the proposed requirement is indeed a functional requirement. If the resulting sentence makes sense, then it can be determined to be a functional requirement. Using this test we can rephrase the previous statements above to verify that they are indeed functional requirements.

- The system shall process a checking account deposit
- The system shall calculate the GPA for a student
- The system shall capture the account holder identification information

Consider the following phrases:

- The system shall be available 24 hours a day, even days a week.

- The system should be accessible via the Internet.

Since these phrases do not include a strong action verb phrase (often identified by the use of the word “be”) they are not functional requirements, they are considered non-functional requirements.

A **Non-Functional Requirement** is a description of the features, characteristics, and attributes of the system as well as any constraints that may limit the boundaries of the proposed solution.

There are many classifications of non-functional requirements. They include:

- **Performance**  
Performance requirements represent performance the system is required to exhibit to meet the needs of users. This includes performance or response times.
- **Data**  
Such requirements typically result from legacy system constraints or established external interfaces. Included shall be requirements, if any, on databases and data files to be included in the system.
- **Information Security and Privacy**  
The requirement types include, as applicable, the security/privacy environment in which the system must operate, the type and degree of security or privacy to be provided, the security/privacy risks the system must withstand, the safeguards required to reduce those risks, the security/privacy policy that must be met, the security/privacy accountability that the system must provide, and the criteria that must be met for security/privacy certification/accreditation.
- **Computer Resource**  
Computer hardware resource utilization may include the maximum allowable use of processor capacity, memory capacity, input/output device capacity, auxiliary storage device capacity, and communications/ network equipment capacity. The requirements (stated, for example, as percentages of the capacity of each computer hardware resource) shall include the conditions, if any, under which the resource utilization is to be measured.
- **Computer Software**  
These requirement types specify the requirements, if any, regarding computer software that must be used by, or incorporated into, the system. Examples include operating systems, user interface/presentation layer software, database management systems, communications/network software, utility software, input and equipment simulators, test and diagnostic software, and manufacturing software. The correct nomenclature, version, and documentation references of each such software item shall be provided.
- **Computer Communications**  
These requirement types specify the additional requirements, if any, concerning the computer communications that must be used by, or incorporated into, the system. Examples include geographic locations to be linked; configuration and network topology; transmission techniques; data transfer rates; gateways; required system use times; type and volume of data to be transmitted/received; time

boundaries for transmission/reception/ response; peak volumes of data; and diagnostic features.

➤ **Training-Related**

These requirement types specify the system requirements, if any, pertaining to training. Examples include training devices and training materials to be included in the system, or to be developed and maintained separately from the system, such as stand alone computer based training (CBT) programs, databases, and operational scripts.

➤ **Documentation**

These requirement types specify requirements for system documentation, such as specifications, drawings, technical manuals, test plans and procedures, and installation instruction data, if not covered in other contractual documents.

➤ **System Environment**

These requirement types specify the requirements, if any, regarding the environment in which the system must operate. Examples for a software system are the computer hardware and operating system on which the software must run. Examples for a hardware-software system include the environmental conditions that the system must withstand during transportation, storage, and operation, such as conditions in the natural environment (wind, rain, temperature, geographic location), the induced environment (motion, shock, noise, electromagnetic radiation), and other adverse operating environments (power outages, brown-outs, external network failures).

➤ **System Quality**

These requirement types specify the requirements, if any, pertaining to system quality factors. Examples include quantitative requirements concerning system functionality (the ability to perform all required functions), reliability (the ability to perform with correct, consistent results -- such as mean time between failure for equipment), maintainability (the ability to be easily serviced, repaired, or corrected), and availability (the ability to be accessed and operated when needed). Where necessary, the requirements for quantitative quality factors shall be assigned to account for the manner in which the system is expected to operate. For example, the assignment of quantitative requirements reliability/availability may be dependent on the system state, mode, or other operating paradigm (if more than one exists), since the potential points of failure for each operating condition may be different.

When applicable, the requirements shall be specified for capturing and storing quality parameter data, conducting trend and statistical capability analyses on this data, reporting quality status, and tracking corrective actions aimed at quality improvements.

➤ **Design and Construction**

These requirement types specify the requirements, if any, that constrain the design and construction of the system. For hardware-software systems, this paragraph shall include the physical requirements imposed on the system. These requirements may be specified by reference to appropriate commercial or military

standards and specifications. If these items are left to the design phase for their requirements specifications, this fact shall be so stated. Applicable constraint examples include:

- a. Use of a particular system architecture or requirements on the architecture, such as required subsystems; use of standard, custom design, or existing components
- b. Attributes of the system, such as:
  - Flexibility (the ability to be easily adapted to changing requirements);
  - portability of software (the ability to be easily modified for a new environment),
  - reusability (the ability to be used in multiple applications, or the ability to use a specific preexisting computer program, module, code objects, or subset of source code),
  - testability (the ability to be easily and thoroughly tested),
  - usability (the ability to be easily learned and used), and
  - other attributes.
- c. Use of particular design or construction standards; use of particular data standards; use of a particular programming language; workmanship requirements and production techniques
- d. Physical characteristics of the system (such as weight limits, dimensional limits, color, protective coatings); interchangeability of parts; ability to be transported from one location to another; ability to be carried or set up by one, or a given number of, persons
- e. Materials that can and cannot be used; requirements on the handling of toxic materials; limits on the electromagnetic radiation that the system is permitted to generate
- f. Use of nameplates, barcodes, part marking, serial and lot number marking, and other identifying markings
- g. Flexibility and expandability that must be provided to support anticipated areas of functional growth or changes in technology

➤ **Site Specific Adaptation**

Adaptation is defined as the process by which automated software is made unique to provide the required service at a specific site. These requirement types shall specify the requirements, if any, concerning installation-dependent data that the system is required to provide (such as site-dependent latitude and longitude or site-dependent state tax codes) and operational parameters that the system is required to use that may vary according to operational needs (such as parameters indicating operation-dependent constants or localized data sets).

➤ **Personnel-Related**

These requirement types specify the system requirements, if any, included to accommodate the number, skill levels, duty cycles, training needs, or other information about the personnel who will use or support the system. Examples include requirements for the number of workstations to be provided and for built-in help and training features. Also included shall be the human factors engineering requirements, if any, imposed on the system. These requirements

shall include, as applicable, considerations for the capabilities and limitations of humans, foreseeable human errors under both normal and extreme conditions, and specific areas where the effects of human error would be particularly serious. Examples include requirements for adjustable-height work stations, color and duration of error messages, physical placement of critical indicators or buttons, and use of auditory signals.

➤ **Safety**

These requirement types specify the system requirements, if any, concerned with preventing or minimizing unintended hazards to personnel, property, and the physical environment. Examples include the use of hazardous materials; providing weight classifications for purposes of shipping, handling, and storage; audible or visual warning devices; electrical system grounding; and electrical/mechanical interlocks. If these provisions do not apply for a given information systems development, implementation, or migration effort, that shall be explicitly stated.

➤ **Logistics-Related**

These requirement types specify the system requirements, if any, concerned with logistics considerations. These considerations may include: system maintenance, software support, system transportation modes, normal or special packaging needs not covered under packaging requirements, supply-system requirements, expedited delivery of backup recovery media, impact on existing facilities, and impact on existing equipment.

➤ **Packaging**

These requirement types specify the requirements, if any, for packaging, labeling, and handling of the system and its components for delivery. Applicable commercial or military specifications and standards may be referenced if appropriate.

➤ **System External Interface**

These requirement types specify system external interface requirements.

➤ **Other**

“Other” specifies additional system requirements, if any, that are not covered in the previous paragraphs.

The benefit of classifying the various types of requirements is the ability to group like requirements for reporting, tracking and validation purposes, plus it aids in identifying possible overlooked requirements.

The goal of the systems analyst is to define system requirements that meet the following criteria:

- Consistent — the requirements are not conflicting or ambiguous.
- Complete — the requirements describe all possible system inputs and responses.
- Feasible — the requirements can be satisfied based on the available resources and constraints.

- Required — the requirements are truly needed and fulfill the purpose of the system.
- Accurate — the requirements are stated correctly.
- Traceable — the requirements directly map to the functions and features of the system.
- Verifiable — the requirements are defined so they can be demonstrated during testing.

## **ANALYZING REQUIREMENTS**

More often than not fact-finding activities produce requirements that are often in conflict with each other. This is because requirements are solicited from many different sources and each person has their own opinions and desires for the functionality and features of the new system. The goal of the requirements analysis activity is to discover and resolve the problems with the requirements and reach agreement on any modifications to satisfy the stakeholders. The process is concerned with the “initial” requirements gathered from the stakeholders. These requirements are usually incomplete and documented in an informal way on instruments such as use cases, tables, and reports. The focus here is to reach agreement on the stakeholder’s needs, in other words it should answer the question “do we have the right system requirements for the project?” Inevitably these draft requirements contain many problems such as:

- Missing requirements
- Conflicting requirements
- Infeasible requirements
- Overlapping requirements
- Ambiguous requirements

These types of requirements problems are very common in many of the requirement documents written today and can be extremely costly to fix later in the development cycle if left unresolved.

## **SPECIFYING REQUIREMENTS**

System requirements are usually documented in a formal way to communicate the requirements to the key stakeholders of the system. This document serves as the contract between the system owners and the development team on what is going to be provided in terms of a new system. Thus, it may go through many revisions and reviews before everyone agrees and authorizes its contents. There is no standard name or format for this document. In fact many organizations use different names such as requirements statement, requirements specification, requirements definition, functional specification, etc., and the format is usually tailored to that organization’s needs. For those companies that provide information systems and software to the U.S. Government, the Government requires that they use the format and naming conventions specified in their published



standards document MIL-STD-498<sup>1</sup>. Many organizations have created their own standards adapted from MIL-STD-498 because of its thoroughness and because many people are already familiar with it. A requirements definition document should consist of the following.

- The functions and services the system should provide.
- Non-functional requirements including the system's features, characteristics, and attributes.
- The constraints, which restrict the development of the system, or under which the system, must operate.
- Information about other systems the system must interface with.

## **VALIDATING REQUIREMENTS**

Who will read the requirements definition document? This document is probably the most widely read and referenced document of all the project documentation. System owners and users use it to specify their requirements and any changes that may arise. Managers use it to prepare project plans and estimates and developers use it to understand what it required and to develop tests to validate the system. With this in mind it is important to note that requirements are read more often than they are written. Therefore taking the time to write them correctly, concisely, and clearly will not only save time from a schedule point of view, but also save costs and reduce the risk of costly requirements errors. By performing requirements validation will help us achieve that goal.

**Requirements Validation** is an activity that checks the requirements definition document for accuracy, completeness, consistency, and conformance to standards.

Requirements validation is performed on a final draft of the requirements definition document after all input has been solicited from the system owners and users. The purpose of this activity is for the systems analyst to ensure the requirements are written correctly. Examples of errors the systems analyst might find are:

- System models that contain errors
- Typographical or grammar errors
- Requirements that conflict with each other
- Ambiguous or poorly worded requirements
- Lack of conformance to quality standards required for the document

---

<sup>1</sup> MIL-STD-498 is a standard that merges DOD-STD-2167A and DOD-STD-7935A to define a set of activities and documentation suitable for the development of both weapon systems and Automated Information Systems.

## **DOCUMENTING REQUIREMENTS**

Please use the following attributes and guidelines when documenting each requirement using the requirements matrix appearing on the following page.

### **Requirement Number**

A good guideline to follow when assigning requirement Ids is to preface the number with the system acronym. Ex. MSS-1.0. This numbering format will allow the assignment and traceability of parent and child requirements. MSS-1.0 being the parent and MSS-1.1 being the child.

### **Requirement Title**

Assign a short phrase indicating the nature of the requirement.

### **Requirement Text**

Provide a textual statement of the requirement including details and constraints.

### **Requirement Parent**

If the requirement is part of a decomposition of a master complex requirement, indicate the number of the parent requirement, else enter "None."

### **Requirement Type**

Indicate the requirement type. The following list of requirement types provide the choices for this entry:

Functional, Performance, Data, Information Security and Privacy, Computer Resource, Computer Software, Computer Communications, Training-Related, Documentation, System Environment, System Quality, Design and Construction, Site Specific Adaptation, Personnel-Related, Safety, Logistics-Related, Packaging, System External Interface, and Other. The list can also be used as a checklist of potential system requirement types and provides guidance for each requirement type.

### **Revision Date and Revision Number**

Indicate the acceptance date and revision number of current (accepted/baselined) version. If new requirement, enter "New – Version 1.0"

### **Verification Method**

This communicates how you intend to test (or prove) that the requirement has been satisfied. Indicate "Demonstration," "Test," "Inspection," or "Analysis."

### **Precedence/Criticality**

If some requirements are more important or critical than other requirements, the precedence or criticality may be placed here. An example would be High, Medium, Low Criticality.

Normally the user would help you decide this entry, if no data is available, use your best judgment. Enter "Must," "Want," or "Nice to Have."

The following requirements matrix should be used when documenting requirements. I have provided three examples for your benefit.

#	Title	Text (details & constraints)	Parent	Type	Revision # & Date	Verification Method	Criticality
MSS-1.00	Process New Member Order	The system shall be able to process new member orders. Within this process it should be able to validate member demographic information, verify credit worthiness, inquire and modify inventory levels based on quantity of product ordered, initiate backorder process in the event of insufficient inventory to fulfill order, and send an order confirmation notice once the order has been placed. Member credit status will be obtained from the Account Receivable system. A picking ticket, containing the available ordered items, must be generated and routed to the warehouse.	None	Functional	New - Version 1.0	Test	Must
MSS-14.0	One Hour Order Confirmation Notice	An E-mail notice shall be generated and sent to the member, within one hour from the time the member placed the order. The member's E-mail address shall be stored on the system within the member's profile. The one-hour constraint applies only to the sending of the notification and not when it's received by the member. Related requirement(s): MSS-1.0	None	Performance	New - Version 1.0	Test	Must
MSS-14.0	Member Account Security	Members can only access and view information regarding their own accounts. Member accounts shall be password and ID protected. The Member Services System will generate IDs and passwords and send them to the members. In the event the member forgets their ID or password, a new set will need to be generated.	None	Information Security and Privacy	New - Version 1.0	Test	Must