Department of Health and Human Services   
Centers for Medicare & Medicaid Services  
Target Life Cycle

CMS Requirements Writer’s Guide

Version 5.0

December 16, 2021

Record of Changes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number | Date | Reference | A = Add,  M = Modify,  D= Delete | Description of Change | Change Request (CR) # |
| 1 | October 6, 2008 | All | A | Initial Release, v4.0 | NA |
| 2 | April 20, 2009 | Pages 13-14 | M | Updated Sections 3.1.5,  3.1.6 and 3.3. | NA |
| 3 | August 31, 2009 | Page 21 | M | Updated example scenario, inserted “user requirements” where applicable, and corrected metamorphosis of requirements document. | 57 |
| 3 | March 4, 2013 | Cover page, pages 2, 5 – 7, 9 & 29 | M & D | Changed references of “CMS Integrated IT Investment & System Lifecycle Framework” to “CMS Expedited Life Cycle.” Removed references to CMS DRAV. Deleted DRAV from acronym list. Replace “and” with “&.” Change “IS” to “Information Security.” | NA |
| 4 | April 25, 2013 | All | A & M | Added new content on: writing reporting requirements; attributes of high quality requirements; and, global requirements writing.  Updated formatting, edited grammar and punctuation throughout document. | NA |
| 4 | February 02, 2015 | Cover Page | M | Update CMS logo. | NA |
| 5 | December 16, 2021 | All | M | Changed references of “CMS Expedited Life Cycle (XLC)” to “CMS Target Life Cycle (TLC).” | NA |

Table of Contents

[1 INTRODUCTION 1](#_Toc354672594)

[1.1 Background & Vision 1](#_Toc354672595)

[1.2 Purpose 1](#_Toc354672596)

[1.3 Scope 2](#_Toc354672597)

[1.4 Intended Audience 2](#_Toc354672598)

[1.5 Reference Documents 2](#_Toc354672599)

[1.6 Document Organization 2](#_Toc354672600)

[1.7 Process Hierarchy 4](#_Toc354672601)

[1.8 Requirements Hierarchy 4](#_Toc354672602)

[1.9 Glossary 6](#_Toc354672603)

[2 USERS OF THIS GUIDE 8](#_Toc354672604)

[2.1 Business Owners 8](#_Toc354672605)

[2.2 Project Managers 9](#_Toc354672606)

[2.2.1 Scoping Requirements 9](#_Toc354672607)

[2.2.2 Managing Requirements 9](#_Toc354672608)

[2.3 Business Analysts 9](#_Toc354672609)

[2.4 Subject Matter Experts 10](#_Toc354672610)

[3 REQUIREMENTS DEVELOPMENT TASKS 11](#_Toc354672611)

[3.1 Project Strategy 11](#_Toc354672612)

[3.1.1 Writing a Business Purpose 11](#_Toc354672613)

[3.1.2 Writing a Functional Purpose 11](#_Toc354672614)

[3.1.3 Determining the Measures of Success 12](#_Toc354672615)

[3.1.4 Identifying the Stakeholders 12](#_Toc354672616)

[3.1.5 Determining the Project Assumptions & Constraints 13](#_Toc354672617)

[3.1.6 Determining the Project Risks 14](#_Toc354672618)

[3.1.7 Assessing the Project Priorities 15](#_Toc354672619)

[3.2 Business Requirements and Business Rules Documentation 15](#_Toc354672620)

[3.2.1 Writing Business Requirements 15](#_Toc354672621)

[3.2.2 Writing Business Rules 17](#_Toc354672622)

[3.3 Writing User, Functional and Nonfunctional Requirements 18](#_Toc354672623)

[3.3.1 Creating the Project Diagrams 18](#_Toc354672624)

[3.3.2 Writing User Requirements 19](#_Toc354672625)

[3.3.3 Documenting Scenarios 20](#_Toc354672626)

[3.3.4 Writing Functional & Nonfunctional Requirements 23](#_Toc354672627)

[3.3.5 Determining Pass/Fail Statements 26](#_Toc354672628)

[4 Global Requirements 27](#_Toc354672629)

[4.1 Overview 27](#_Toc354672630)

[4.2 Reporting Requirements 27](#_Toc354672631)

[4.2.1 Business Requirements 27](#_Toc354672632)

[4.2.2 User/Functional/Nonfunctional Requirements 28](#_Toc354672633)

[5 STYLE GUIDELINES 30](#_Toc354672634)

[5.1 Writing Style 30](#_Toc354672635)

[5.2 Quality Attributes of Requirements 31](#_Toc354672636)

[5.2.1 Correct 31](#_Toc354672637)

[5.2.2 Complete 31](#_Toc354672638)

[5.2.3 Clear 31](#_Toc354672639)

[5.2.4 Consistent 31](#_Toc354672640)

[5.2.5 Verifiable 31](#_Toc354672641)

[5.2.6 Traceable 32](#_Toc354672642)

[5.2.7 Feasible 32](#_Toc354672643)

[5.2.8 Modular 32](#_Toc354672644)

[5.2.9 Design-independent 32](#_Toc354672645)

[Appendix A. ACRONYM LIST A-1](#_Toc354672646)

[Appendix B. Components OF REQUIREMENTS AND RULES B-1](#_Toc354672647)

[Appendix C. Best Practices for Verbs and Phrases C-1](#_Toc354672648)

[Appendix D. ADVANCED STYLISTIC APPROACHES D-1](#_Toc354672649)

[Appendix E. Requirements Elicitation Approaches E-1](#_Toc354672650)

List of Tables

[Table 1: Document Organization 3](#_Toc354672521)

[Table 2: Glossary 6](#_Toc354672522)

[Table 3: Sample Project Risks 14](#_Toc354672523)

[Table 4: Sample Project Priorities 15](#_Toc354672524)

[Table 5: Sample Business Requirements 17](#_Toc354672525)

[Table 6: Commonly Used Verbs for Business and User Requirements C-1](#_Toc354672526)

[Table 7: Commonly Used Verbs for Functional/Nonfunctional Requirements C-1](#_Toc354672527)

[Table 8: Words to Avoid C-1](#_Toc354672528)

List of Figures

[Figure 1: Process Hierarchy 4](#_Toc354606038)

[Figure 2: Requirements Hierarchy 5](#_Toc354606039)

[Figure 3: Sample Business Process Model Fragment 16](#_Toc354606040)

[Figure 4: Sample Work Context Diagram 19](#_Toc354606041)

# INTRODUCTION

## Background & Vision

The Centers for Medicare & Medicaid Services (CMS) desire to apply a more uniform methodology to the documentation of proposed information systems that service the Agency’s Information Technology (IT) needs. This initiative spans the CMS enterprise to include all development activities for new or redesigned systems. The Agency continues to evolve, elaborate, and enhance its methodology and approach to the capture and documentation of requirements that meet the criteria of being complete, clear, consistent, verifiable, feasible, modular, singular, and design independent so that the enterprise technical vision and architecture return the highest value to its beneficiaries.

Whether building a new system, making changes to existing software, or using commercial off-the-shelf software, exploring, capturing, and communicating requirements is necessary. Requirements are commonly used to communicate ideas. They are usually derived from a variety of informational sources, and can represent many things to many people. Because requirements can be interpreted in many ways, defining them is a form of art that is commonly used to express the needs and wants of a system in its simplest form that removes all ambiguity.

In the world of systems development, requirements represent the things that need to be known and the things that need to be done that are critical for project success. The right product cannot be built unless the developer knows precisely what the product should do and how the product’s success will be measured. Projects succeed when there are clearly defined goals and attainable results which focus on satisfying the need and delivering results users want. This is only possible when those needs are clearly defined and the problem is fully documented in writing.

The person responsible for writing requirements must understand the business goals and have knowledge of the intended product, what it must do, what qualities it must have, what constraints it must conform to, and what interfaces it must have to other systems and the outside world. As development continues, knowledge of the product increases and the analytical activity should continue to grow until all the requirements are known and analyses at all phases are complete. In this way, the entire analysis process deals with identifying the “problem.”

## Purpose

The purpose of the *CMS Requirements Writer’s Guide* (RWG)is to provide guidance to managers, analysts, users, developers, and others who are responsible for scoping, writing, or reviewing requirements. It seeks to describe the level of detail that should be included in each written requirement and what stakeholders must look for when conducting formal reviews of internally or externally developed requirements. It also intends to standardize the nomenclature associated with writing requirements. The document employs an object-oriented methodology for organizing requirements.

The RWG will assist in effectively managing requirements and providing guidelines for measuring and accepting requirements. Lastly, the document provides a clearly defined structure for written requirements and a standardized set of criteria to assess each requirement.

This guide provides guidelines to assist in the development of requirements. Although the person writing requirements is encouraged to rigorously follow these guidelines, the unique characteristics of each IT project may warrant slight modifications to the guidelines.

## Scope

This guide addresses how to write and organize business, user, functional, and nonfunctional requirements for IT projects. It is the Agency’s goal that all IT investments identified as new projects, system redesign, platform ports, or the incorporation of major new business functions use the guidance provided in the RWG. It is not the intent that systems currently in production and performing routine maintenance must convert their requirements baseline to this format.

## Intended Audience

The RWG is appropriate for all stakeholders of the CMS Target Life Cycle but especially for: business owners, project managers, business analysts, and subject matter experts charged with developing business, user, functional, and nonfunctional requirements.

## Reference Documents

CMS follows the guidance specified in Section 6.26.3 of IEEE/EIA 12207.1 1997 Guide for Information Technology with respect to requirements. The Institute of Electrical and Electronics Engineers (IEEE) standard recommends tailoring its methodology to fit the needs of the organization using it. Captured within the RWG are the results of this tailoring. CMS also adopts best business practices as specified in Chapter 5 of the Business Analysis Body of Knowledge v2.0. The following documents were used in developing this guide:

* IIBA Business Analysis Body of Knowledge, Version 2.0
* IEEE/EIA Guide for Information Technology, Std 12207.1
* IEEE Developing System Requirements Specification, Std 1233
* IEEE Recommended Practice for Software Requirements Specifications, Std 830
* The Art and Discipline of Writing Requirements, Revision 2

## Document Organization

It is strongly recommended that all readers familiarize themselves with the terms listed in the glossary. Novice guide users may find it helpful to refer to the glossary as the terms are used throughout the document. Experienced requirements documenters should review the glossary to determine if their understanding of a term coincides with how the guide intends to use it.

Readers unfamiliar with their role in the requirements writing process should first examine Section 2 where the tasks and responsibilities for the various people involved in the requirements writing process are identified. Following this, the reader may refer to the appropriate sections in Section 3 that guides them through the process of fulfilling each of the tasks they are responsible for completing. By contrast, if readers are familiar with what is expected of them, they should be able to refer to the Table of Contents to quickly locate specific relevant sections.

All readers are encouraged to familiarize themselves with Section 4, where stylistic and syntactical writing considerations are discussed. Stylistic considerations are of importance to all writers. Syntactical rules are especially important to business analysts because they standardize communication between the business owner and the designer, thus providing more clarity.

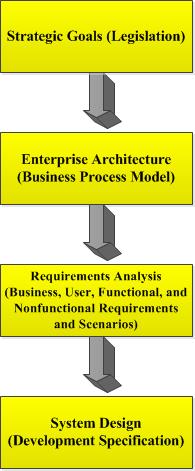
This document is organized as follows:

| Table : Document Organization | |
| --- | --- |
| Section | Overview |
| Section 2: Users of this Guide | Identifies each type of user and what is expected of them throughout the requirements documentation process. |
| Section 3: Requirements Development Tasks |  |
| Section 4: Global Requirements | How some requirements may be treated as global in scope. |
| Section 5: Style Guidelines | Contains a discussion of a “middle level” investigation of requirements, including the proper layout, form, and phrasing of requirements. |
| Appendix A: Acronym List | Provides a list of acronyms used in the document. |
| Appendix B: Requirements and Rules Components |  |
| Appendix C: Best Practices for Verbs and Phrases | A listing of strong, transitive verbs for shall statements and words/phrases to avoid. |
| Appendix D: Advanced Stylistic Approaches | Contains a discussion on methods that may be used to reduce ambiguity in written Requirements Documents (RDs). |
| Appendix E: Requirements Elicitation Approaches | Best practices for various requirements elicitation techniques. |

## Process Hierarchy

This guide is based on a process of building a system that starts at a very broad level (Strategic Goals) and generates more detail about the system in successive levels. The need for a new system usually begins with either legislation from Congress or a strategic initiative from CMS Management. The business owner then translates these goals into business requirements. The business owner uses the business requirements to create more detailed user requirements, scenarios, and functional and nonfunctional requirements with the help of business analysts and subject matter experts. The requirements document is then turned over to the systems developer where the process of converting business requirements into a new system or solving an existing problem begins.

Figure : Process Hierarchy



## Requirements Hierarchy

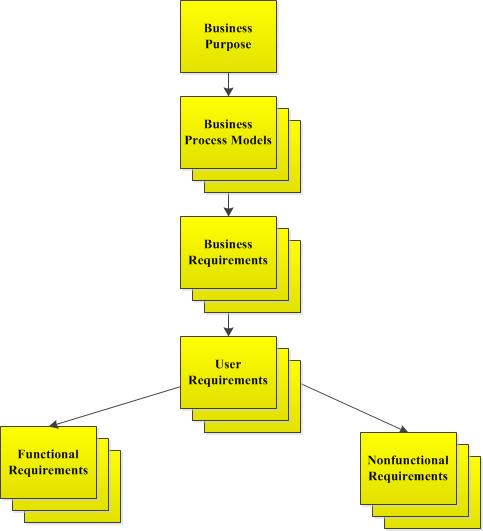
The requirements hierarchy, as diagrammed below, represents the customary progressive elaboration that takes place in a waterfall environment which elicits requirements in the early phases of a project. Ideally, business owners first identify all the business requirements necessary for supporting their business purpose. These requirements should be very broad and inclusive of all the work to be performed by the business unit so that all potential opportunities for leveraging IT investments are recognized. Along with documenting the business requirements, the business unit captures the business rules.

At this point, the business unit has the information needed to begin defining how an IT investment can be used to fulfill the business requirements identified. At the next level, the user requirements characterize the scope of the IT investment. Following this, functional and nonfunctional requirements define what the proposed system will actually do.

The CMS Target Life Cycle shows the creation of the requirements document spanning the concept, planning, and requirements analysis phases. At first the requirements document may elaborate down to just the business requirements level.

As analysis continues, user and some functional/nonfunctional requirements are captured. The requirements document is considered complete when all requirements down to the functional/nonfunctional requirements have been documented therein. The final requirements document requires sign off by the business owner.

Figure : Requirements Hierarchy



Historically, the term “system requirements” was used to categorize all lower level requirements intended to describe an IT deliverable. In order to ensure that they are more completely defined, requirements are now described as functional, nonfunctional, or design (see Glossary). In addition, a layer of requirements (user requirements) was added to establish the initial scope of the new system.

## Glossary

| Table : Glossary | | |
| --- | --- | --- |
| Term | Process Phase | Definition |
| Alternative Scenario | Requirements Analysis | A scenario identifying the steps that occur when the outcome of a scenario differs from the expected outcome. |
| Business Requirements (BR) | Enterprise Architecture & Requirements Analysis | Business requirements are statements of the functions or program needs that must be met in order to accomplish the business objectives of the project. These business objectives address legislative mandates or strategic business goals (such as improved customer service, business efficiencies, business process reengineering, etc.). |
| Business Process Model (BPM) | Enterprise Architecture | A graphical depiction of either the existing (As Is) business work flow or the desired (To Be) business work flow. |
| Business Rule (RU) | Requirements Analysis & Requirements Analysis | The conditions under which a business requirement must be fulfilled. |
| Constraint | Requirements Analysis | A specific type of requirement that limits design, development, or deployment options. Constraints mandate the way the system must be produced and may exist at both the business requirements level and the functional/nonfunctional requirements level. |
| Domain | Requirements Analysis | A user or system that interacts with the system being designed (actor). |
| Event | Requirements Analysis | Historically, a synonym for user requirement. |
| Functional Requirement (FR) | Requirements Analysis | A statement of action that describes the behavior and information that the solution will manage. They describe capabilities the system will be able to perform in terms of behaviors or operations – a specific system action or response. |
| Nonfunctional Requirement (NR) | Requirements Analysis | A statement that describes conditions that do not directly relate to the behavior or functionality of the solution, but rather describe environmental conditions under which the solution must remain effective or qualities that the systems must have. See Section 3.3.4.1 for a list of various types. |
| Pass/Fail Statement | Requirements Analysis | A testable statement that describes how to know if a requirement is met. |
| Precondition | Requirements Analysis | The state the system must be in before a scenario can start. |
| Requirement | All levels | A measurable statement of intent about something that the system must do, a property the system must have, or a constraint on the system. |
| Scenario | Requirements Analysis | A sequence of steps taken to complete a user requirement, similar to a use case. |
| Subject Matter Expert (SME) | All levels | A subject matter expert is an employee of CMS or a contractor with specific knowledge regarding a domain that interacts with the system. |
| Stakeholder | Enterprise Architecture | A person, system, or entity that interacts with, is responsible for, or has some interest/influence in the system being developed. |
| System Requirement | Requirements Analysis | A historically umbrella term, which included functional nonfunctional, and design requirements. |
| Traceability | Requirements Analysis | The ability to trace relationships between two or more requirements. |
| Trigger | Requirements Analysis | An action or activity that occurs that causes the scenario to begin. |
| Use Case | Requirements Analysis | A description of a system’s behavior as it responds to a request that originates from outside of that system. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. The use case should contain all system activities that have significance to the users. Use cases typically avoid technical jargon, preferring instead the language of the subject matter expert. |
| User Requirement (UR) | Requirements Analysis | A requirement that describes how a user interacts with the system in order to fulfill a Business Requirement. The UR could also be internally initiated by the system. |

# USERS OF THIS GUIDE

There are primarily four types of users this guide is intended to help:

* Business Owners
* Project Managers
* Business Analysts
* Subject Matter Experts

The guide will help each type of user identify what is expected of them throughout the requirements documentation process.

## Business Owners

The business owner(s) is the entity or entities responsible for defining, promoting, endorsing, and upholding the business needs and user requirements for the system and for performing user acceptance testing of the final product(s) based on those business needs and requirements. The business owner(s) may represent the interests of external partners such as Medicare contractors, fiscal intermediaries, or other Federal agencies and/or the interests of internal CMS Centers, Offices, or Consortia. The business owner(s) define and verify the following:

* System functionality
* Access rights
* Business rules and their privacy classification
* Timeliness
* Completeness and accuracy of data.

The business owner may also be the project manager.

Below are the tasks for which the business owner is responsible in the requirements writing process. If a Business Process Model (BPM) was created for the project, the business owner should be thoroughly familiar with it. This will be of great assistance and in some cases will provide the information necessary to perform the following tasks:

* Writing business requirements
* Writing business rules
* Writing a business purpose
* Writing a functional purpose
* Determining the measures of success
* Assessing the project priorities
* Identifying the stakeholders
* Determining the risks
* Determining the assumptions and constraints

Sections 3.1 – 3.3 of this document detail these tasks.

## Project Managers

The project manager is the individual at CMS who has the day-to-day responsibility for the success and management of the IT project. This individual may be at any level in the CMS organizational structure. The project manager is the primary point of contact for the project with relation to the CMS Target Life Cycle (TLC). The project manager communicates regularly with management on the needs and progress of the project and is responsible for obtaining and managing project resources. In those cases where contract support is utilized, the project manager may also serve as the Government Task Leader (GTL) and/or the Project Officer, or otherwise provide support to the assigned GTL and Project Officer for the contract. The project manager may also operate as a business analyst.

### Scoping Requirements

It is essential that the project manager have the business strategy documentation produced by the business owner in order to begin the requirements documentation process. Typically, the first step in the requirements documentation process is to review the BPM and supporting narrative to capture any business requirements documented or implied there. It is the responsibility of the project manager to ensure the requirements document is complete.

If the project manager is also operating as part of the business analysis team, the project manager should refer to Sections 3.1 – 3.3 to start the requirements writing process.

### Managing Requirements

The project manager is responsible for managing the requirements during requirements development as well as after the team has finished its work. This includes ensuring requirement traceability, reviewing requirements, base lining requirements (both formal and informal), tracking change requests (when appropriate), and enforcing writing standards. It is not the intent of this document to provide guidance on how to manage or maintain requirements.

It is also the project manager’s responsibility to manage updates to the requirements document. A template for this document is available on the CMS website. The various parts of the requirements document are discussed throughout this guide.

## Business Analysts

The business analyst is an individual tasked with gathering and writing requirements for the IT project. The person may also be the project manager, an employee of the CMS organization or a contractor. The business analyst communicates regularly with the project team and the subject matter experts while gathering and writing the requirements.

After performing the requirements gathering process, the business analyst is responsible for putting the requirements into a formal document that can be handed over to system designers to continue the project. The steps an analyst must take to accomplish this are as follows:

* Documenting business requirements & business rules
* Documenting user requirements
* Creating scenarios
* Writing functional requirements
* Writing nonfunctional requirements
* Capturing the requirement attributes
* Importing requirements into the CMS Requirements Repository

These tasks are described in Section 3.3.

## Subject Matter Experts

A Subject Matter Expert (SME) should generally be a stakeholder in the project, though not all stakeholders are SMEs. Their expertise may come from either a business or a technical perspective.

While a SME may not be responsible for writing requirements or the supporting documentation, they should be familiar with what the project team will expect from them during this process so that the team’s deliverables will accurately reflect the SME’s needs and knowledge.

SMEs should be involved throughout the entire process of requirements documentation and should specifically be familiar with:

* Section 3.1: Project Strategy tasks
* Section 3.2: Writing Business Requirements tasks
* Section 3.3.: Writing User, Functional and Nonfunctional Requirements tasks

# REQUIREMENTS DEVELOPMENT TASKS

This section describes specific tasks and deliverables to complete as determined by the reader’s role in the project. Section 2 describes the role of the participants in the project.

## Project Strategy

All the stakeholders that are identified in the BPM or during the project kickoff should participate in performing the business strategy tasks. Project managers cannot begin their work until these tasks are complete. The information collected in these tasks will be part of the requirements document.

### Writing a Business Purpose

The business purpose describes *why* CMS would fund the project. The business purpose is always related to the mission of the organization and/or financial concerns. CMS projects are often the result of legislative initiatives, which are therefore related to the mission of the organization. Upgrades and enhancements to systems often involve financial concerns, allowing the agency to take advantage of current (more efficient) technology. The business purpose may be used to capture the business drivers if this information is not already covered in the introduction/purpose of the requirements document.

An example of a good business purpose would be:

The business purpose is to comply with the provisions set by Congress in The Medicare Prescription Drug, Improvement and Modernization Act of 2003 (MMA) by reimbursing care providers for services rendered to beneficiaries. In accomplishing this, the process of making payments will be more efficient in terms of both time and money.

It is possible that there are multiple business purposes for funding the project, and each should be written in a complete sentence. When stating more than one purpose, ensure they do not compete. For example, complying with a law and updating technology to reduce maintenance costs are valid business purposes that do not conflict. If there are more than two business purposes, the project team may want to consider breaking the project up into smaller components.

### Writing a Functional Purpose

The functional purpose describes *what* the project shall do for the business in a single, complete sentence. It is not unusual for stakeholders to have competing notions regarding what the functional purpose should be. While a compound sentence is acceptable, the purpose should still be limited to one sentence.

The purpose should be a statement containing strong transitive verbs like: build, compare, explore, investigate, perform, recommend, or replace. This is the statement that gives the entire project its direction. When formulating the statement, consider the seriousness of the problem and why it needs to be solved.

An example of a functional purpose might be:

The functional purpose is to replace the existing functionality of the APPS system by leveraging the technical architecture already in place at CMS and to add functionality that will reduce existing manual processes.

If a functional purpose cannot be reduced to a single sentence, the project team may want to consider if the project is too big and should be broken up into separate projects.

### Determining the Measures of Success

The measures of success list metrics on how the project shall be measured against the functional and business purposes. The measures of success must always be concrete and measurable. This can be especially challenging if the processes used in an existing system have never been measured. The measures must also tie back to the business and functional purposes. Some types of measures include:

* Customer satisfaction: does it meet an expectation or improve the customer’s perception of the business?
* Work cycle times: how efficiently should the system work or affect other work?
* Product and service quality: how does the system improve the business process?
* Cost performance: does the system reduce business expenses?
* Employee satisfaction: does it meet an expectation or improve the employee’s ability to perform the business functions?

Some examples of these include:

* The system processes payments with 100% accuracy. (service quality)
* The system processes a transaction within .5 seconds. (work cycle time)
* The system reduces maintenance costs by 25%. (cost performance)

### Identifying the Stakeholders

Stakeholders of the project are initially identified by the BPMs and are expanded here. They fall into one or more of the following four categories:

* Those who will pay for the project: these are the people who have the final say over the completeness of the requirements.

Those who will use or influence the end product (including other systems): these are the people and systems where much of the desired functionality will be elicited. Stakeholders who also exert influence on the system design include those who set technical architecture policy or who deliver data exchange files (defined by Interface Control Documents).

* Those who will be positively affected by the project’s implementation: these represent the most cooperative stakeholders and it would be natural to expect some scope creep from them.
* Those who will perceive to have limited benefit by the project’s implementation: these represent the most resistant stakeholders and satisfying their requirements can be challenging.

It is important to list and discuss all categories to accurately determine each group’s concerns and needs. Identify those who will use the system by describing how they will use it and from where they will use it. It may also help to determine approximately how many users are in each grouping.

An example of the stakeholders for a project might be:

Payers

* CBC

Users (people and systems that interact with or influence the end product)

* CMS
* CBC Management: responsible for approving payments
* CBC Staff: responsible for processing payments
* OIS: responsible for setting technical architecture policy
* MARx system: provides beneficiary level payment information
* FACS: notifies the Treasury of payments to plans
* Department of the Treasury: responsible for transferring funds to plans
* Plan Sponsors: receives payments and payment information

Positively Affected

* Plan Sponsors will have payments processed more accurately and timely.
* CBC Staff/Management will have more robust tools to perform the monthly payment process.

Limited Benefit

* Plans owing CMS money will be more carefully tracked.

It is important to consider how the entire user and support staff will be affected by the new system. Many times, while a new system will ultimately make a user more productive, it often places additional burdens or demands on support groups. Understanding how they are adversely affected can uncover new assumptions that may need to be addressed.

### Determining the Project Assumptions & Constraints

It is valuable to list the constraints and assumptions that the project team and developers identify about the environment in which they will base their project planning decisions. False or unstated assumptions and constraints can result in risks being realized and potential failure of the project.

An assumption is a statement believed to be true for the purposes of planning but is presently unverifiable. Over time, assumptions can evolve as they become validated (at which time they may become constraints or are incorporated into the business or technical requirements). It may be helpful to assess each assumption and state the likelihood that it is correct. Some assumptions might include:

* The new system will utilize the same ICDs as the existing system.
* The new system will not be treated as a system of record.

A constraint is an internal or external limitation on the project that affects its performance. It can be related to the business (e.g., budget or human resources) or it may be technical in nature (such as architecture standards). Typical types of constraints are both business and technical including:

* Compliance: what legislative initiates must be met?
* Solution: distinct from technological restraints, it defines the boundaries within which the problem may be solved and are absolutely non-negotiable.
* Schedule: how much time is allotted to build the system? Is there a window of opportunity that must be taken advantage of or deadlines that have to be met?
* Resources: how do the dollars, skills, or people available to the project affect requirements capture and prioritization?
* Current Implementation: if applicable, a description of the components (automated, mechanical, organizational and otherwise) of the current system.
* Architectural: what technical decisions regarding the implementation are already in place?

For example, constraints might be:

* The system is compliant with the MMA.
* The system is ready for processing payments by the first business day of each quarter.
* The system runs under the CMS 3 zone architecture.

Assumptions and constraints can also be interrelated such as when the project is constrained by certain facts and it assumed that those facts will be acceptable. In such cases, it may be helpful to state it as both an assumption and a constraint. For example:

* Assumption: the HITSP specification will be adopted by providers.
* Constraint: the HITSP specification must be employed.

Constraints and assumptions should be revisited periodically during requirements development to determine if any updates are necessary.

### Determining the Project Risks

Constraints and assumptions will usually have associated risks. It is helpful to ask such questions as what would be the result should an assumption be proven false or a constraint prevents the project from meeting a measure of success. Identify the project risks that the organization faces in such instances. Assess the level of impact and likelihood of risk (high, medium, low) and the consequences should the risk become a reality, as well as some methods for reducing the risk. Categorize the risk if appropriate (technical, operational, user acceptance, etc.).

A project might include the following risks:

| Table 3: Sample Project Risks | | | | |
| --- | --- | --- | --- | --- |
| Risk Area | Risk | Impact | Likelihood | Mitigation |
| Schedule | The risk is that the project will not meet the scheduled deadline because the subject matter experts are already overloaded with work and have little extra time for requirements elicitation. | High | Medium | Prepare requirements for SME review and use less interview time. |
| Customer Satisfaction | The risk is that the new system may not have a similar look and feel to the existing system because the project has engaged a new developer. | Low | Low | Do screen mock ups for the SMEs to review. |

Risks identified in the project’s business case should be evaluated for inclusion in the requirements document.

### Assessing the Project Priorities

Successful completion of a project depends on several factors. Among these are the resources available to do the work, the proper scope of the work, the work being completed on schedule to meet business needs, and a quality solution that is free from defects. However, there is always an inherent conflict between scope, resources, schedule, and allowable defects.

The project priorities help the team determine what is most important among the competing priorities, should a choice need to be made. The team is allowed to select only one of these as being of highest importance to the project. Ideally, only one of the four should be designated as second highest. Use a chart similar to the following to discuss the four priorities with the team. After making the selections on the chart, document why the team made its selections. The table below presents a sample project priority:

| Table 4: Sample Project Priorities | |
| --- | --- |
| Product Quality Dimension | Priority Level (High, Medium, Low) |
| Scope (features) | Low |
| Schedule | High |
| Defects | Medium |
| Resources (manpower, budget) | Low |

For this example, schedule has been determined to be the most important priority. It will drive whether the team uses some existing, proven systems vs. investing in new technology. Minimal scope must be reached (enrolling, submitting payment request, calculating payment, payment), but complexity and depth of each task is negotiable, based on time/budget.

## Business Requirements and Business Rules Documentation

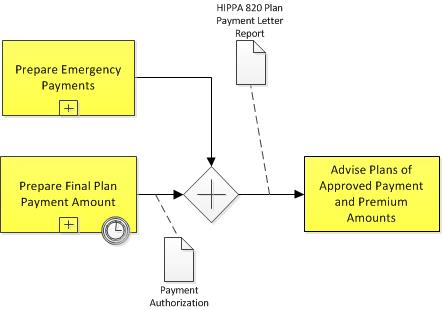
Business requirements are general statements describing what the stakeholders must do in order to fulfill the business objectives. These business goals address legislative mandates or strategic objectives that the core functionality of the system must support (such as improved customer service, business efficiencies, business process reengineering, etc.).

Business rules are documented with business requirements to indicate how policies, guidelines, standards, or regulations impact the business activity. A business rule is a statement that defines or constrains some aspect of the business. It is intended to assert business structure, or to control or influence the behavior of the business. Taken together, they address what is needed from the stakeholder, not from the system.

### Writing Business Requirements

The first place to begin deriving business requirements is the Business Process Model. Consider the following portion of a BPM:

Figure : Sample Business Process Model Fragment



From this portion of the model, the following Business Requirements (BRs) may be identified:

* The business owner shall process payments for plans.
* The business owner shall process emergency payments for plans.
* The business owner shall notify the plans of prospective payments.
* The business owner shall notify the plans of prospective premium amounts.
* The business owner shall secure plan payment information.

A business requirement never identifies what is expected of the system but only addresses what the stakeholder requires. For example, the first requirement below is a business requirement but the second one is a nonfunctional (performance) requirement:

* The user shall perform their duties at any time during the business week including Saturday.
* The system shall be available for use 99.999% of the time.

Each business requirement is further elaborated by business rules and user requirements. Use the following guidelines for writing business requirements:

* Start every requirement with “The [stakeholder] shall…” to support the business activities that must be performed.
* The requirement should be written in clear, concise English language only and not reference any data elements, value codes, or other such constructs found in functional requirements or system design documents.
* Business requirements are derived from legislation, Business Process Models, or re- engineering efforts. Indicate the specific model if this is the case. Business requirements can also specify expected standards or performance criteria.

The business requirement may represent the synthesis of the intent of multiple user requirements.

While requirements are still under development, the typical way of identifying them is by using a temporary unique identifier. The final requirement numbers should be assigned by the Requirements Repository and *not* in the requirements document. Examples of business requirements and the identified business process activities include:

Table : Sample Business Requirements

| Business Requirement | Business Process Activity |
| --- | --- |
| CMS shall process Monthly Payments for each applicable Plan. | Monthly Plan Payment Process |
| CMS shall maintain an accounting of each payment made to a Plan. | Record Payments & Adjustments, Record Approvals |
| CMS shall trace all Payments that are disbursed or received from a Plan. | Record Payments & Adjustments |
| CMS shall generate Financial reports based on a Plans’ Payment data. | Produce Management Reports |
| CMS shall trace all Payments that are disbursed or received from a Plan. | Record Approvals, Record Denials |
| CMS shall track the available funds in the Medicare Trust Fund Accounts for Plans. | Review Trust Fund Balances and Payment Amounts |

Refer to the Style Guide in Section 5 to better understand how requirements should be phrased.

### Writing Business Rules

A business rule describes a policy, guideline, standard, or regulation upon which the business operates. A business rule is a statement that defines or constrains some aspect of the business. It is intended to assert business structure, or to control or influence the behavior of the business. The business rules that concern the project are atomic in that they cannot be further decomposed and they are not process-dependent, so that they apply at all times.

Business rules may apply to more than one business requirement and are uniquely identified with the prefix of RU. They typically fall under one of the following categories:

* Term: the meaning of a word or phrase.
* Fact: the relationship between two or more terms.
* Derivation: the result of a calculation or the result of a logical inference based on known information.
* Assertion: the values of a term or fact are considered valid by the business. They may be further broken down as authorization (who may perform an action), condition (the circumstance under which another business rule may be applied) or an Integrity Constraint (which values of a term or fact are permissible).
* Action Enabler: trigger an activity or a message if a certain condition becomes true. Most business rules employ the use of a “must” statement.

More complex business rules may require a truth table or diagram such as a flowchart or activity diagram. To understand how the rule impacts the business requirement, it is useful to document the rule with the requirement. For example, consider the following business rules for a plan payment process:

* The business owner shall process monthly payments for each plan.
* Plans must be paid by the 15th of the month. [*action enabler*]
* Payments must be approved by management. [*assertion*]
* Payments must be calculated by the total payment amounts by each payment status by payee. [*derivation*]
* The Payment Detail Report must be sorted by Tax Identification Number (TIN). [*fact*]
* A TIN is an identification number used for the administration of tax laws. [*term*]

## Writing User, Functional and Nonfunctional Requirements

Writing requirements is a difficult task, but the benefits summarized below describe the purpose of capturing requirements:

* To communicate the needs of a user or organization
* To provide a solid foundation for the system or product
* To provide the first view of what the intended product must do
* To provide clear descriptions of how the system should perform
* To serve as a foundation for testing and user acceptance
* To provide a basis for design
* To provide clear goals and measurable objectives
* To show traceability back to the requirement sources
* To define compliance requirements (e.g., the Federal Information Security Management Act (FISMA), the Health Insurance Portability and Accountability Act (HIPAA), Federal Information Processing Standards (FIPS), etc.)

Typically the business analyst and the SME will work together closely to produce the requirements.

### Creating the Project Diagrams

At the beginning of a system development project, it is best to define the boundaries of the system to be analyzed and built. Within the requirements writing process, it is essential to know the scope of this system. One example of a good tool for this purpose is the Work Context Diagram.

The context diagram defines the boundaries of the system being analyzed by showing how it connects to the outside world. The diagram is largely created before starting to break the system into its functional pieces. It shows how the system under analysis connects to the other stakeholders in the diagram. These connections help convey the precise scope of analysis. The diagram demonstrates the scope of the systems analysis to the users. Without an agreed context, there is no way of telling if the correct system is being analyzed. There is no ideal place to start the analysis and no real place to stop.

The sample work context diagram below shows all entities that will have knowledge of the system and that will interact with it. The direction of the arrows indicates which entity will *initiate* the user requirement. After a user requirement is initiated, there is usually a two-way communication. The arrows of the work context diagram simply show who begins the user requirement. Note that it is possible for the system to initiate a user requirement that interacts with itself or the stakeholders. Timed processes such as an automated backup would fall under this category.

Figure : Sample Work Context Diagram



### Writing User Requirements

Writing User Requirements (URs) is the first step in decomposing business requirements. A user can represent either a stakeholder who interacts with the system or another system that exchanges data with the new system. URs describe how a user intends to interact with the system in order to fulfill a business requirement. A BR may be decomposed into several URs. For example:

* Business Requirement: CMS shall process emergency payments for each applicable Plan.
* User Requirement: The user shall create emergency payments.
* User Requirement: The user shall modify emergency payments.
* User Requirement: The user shall delete emergency payments.
* User Requirement: The user shall view emergency payments.
* User Requirement: The user shall authorize emergency payments.

Using the work context diagram, the project team identifies the list of user requirements initiated by that domain for each line and identifies the business requirement that that it derives from. For the sample work context diagram, these included:

* Staff to System (line 1)
* Staff shall issue monthly payments.
* Staff shall issue interim payments.
* Staff shall issue emergency payments.
* Staff shall manage accounts receivable.
* Staff shall authenticate themselves to the system.

Writing a purpose for each UR will help reduce any ambiguity as to why the UR exists. All URs should use an action verb and indicate who is doing what to whom. For example, use “The [stakeholder] shall view reports” or “The [stakeholder] shall create a report.” In the examples above, the Staff is doing something (creating) to payments. The project manager is responsible for identifying these URs with the assistance of subject matter experts. Be wary of losing focus by adding too much detail here.

From the URs, scenarios will be generated that describe in more detail how the system will fulfill the UR. Functional and nonfunctional requirements can later be generated for both URs and scenarios.

In some cases, when the number of scenarios is expected to be very limited, it may make the most sense to create a higher level UR to encompass all the scenarios envisioned. For example:

* Business Requirement: CMS shall process emergency payments for each applicable Plan.
* User Requirement: The user shall maintain emergency payments.
* Scenario 1: Create Payments
* Scenario 2: Modify Payments
* Scenario 3: Delete Payments
* Scenario 4: View Payments
* Scenario 5: Authorize Payments

Not all stakeholders will necessarily have user requirements identified. For example, the Office of Information Services (OIS) sets the policy for the technical architecture of a system and hence only nonfunctional requirements will be generated from that stakeholder.

The business analyst uses their best judgment on how to clearly organize the URs and scenarios given the time and budget allowed. Note that in order to avoid identifying duplicative requirements or scenarios, it is permissible to categorize scenarios and requirements that are applicable for all users within a global requirements section in the requirements document. Examples of global requirements include performance and authentication requirements. Adding a brief flow diagram to indicate the context of the user requirement can make it easier for the reader to understand the context of when the UR occurs.

### Documenting Scenarios

Scenarios, which are similar to use cases, describe the steps a user takes to satisfy a user requirement. They are a convenient structure around which to organize requirements. There are usually several scenarios associated with a user requirement. The steps of a scenario typically encompass all the functional requirements for that scenario. Scenarios are particularly helpful because they encourage the complete analysis of the system’s failure or error conditions. By starting with the user’s goals and understanding not only how the system guarantees the successful achievement of these goals but also how the system responds when it cannot achieve these goals, the resulting specifications and testing scenarios are more complete.

All the user requirements identified in the work context diagram should have at least two scenarios associated with them. The first scenario is the primary scenario and is usually the most common way the subject matter expert believes the scenario will play out.

Before writing the formal description, it often helps to diagram a scenario using a flow chart. Any diagramming methodology the team is familiar with is acceptable. When documenting a scenario the following aspects must be considered:

* What are the preconditions that must be met before the scenario can occur? For example, before an emergency payment can be made, the plan information must exist in the system. Thinking through the preconditions can sometimes lead to new URs.
* What are the triggers that cause the scenario to occur? Is there a schedule that triggers the scenario? Multiple triggers will often indicate a need for multiple scenarios.
* What is the expected result of the scenario? Knowing this will help in identifying alternative scenarios, aid in testing the scenario, and clarify the purpose of the scenario.
* Steps are the process that must be followed in order to go from the trigger to the expected result. This typically is a back and forth process between one or more users interacting with the system or performing some business process.

A portion of a sample scenario is outlined below. Note one means of how to handle conditional situations along the way. In some cases, the condition results in the scenario terminating before the scenario is complete – this is normal. Note too that functional requirements are documented as part of the scenario in order to best understand the context of the requirement.

**2. Send Request for EP Mailing Address for IRS 1099 Forms to MCS**

2.1 The NLR shall request updated mailing addresses from MCS for distributing IRS 1099 forms.

2.2 Purpose

The purpose is to provide a means for the NLR to request updated EP mailing addresses from MCS.

2.3 Scenario: Receive an Updated EP Mailing Address for IRS 1099 Forms from MCS

This scenario covers the process the NLR goes through to receive the updated EP’s mailing address from MCS.

**2.3.1 Scenario Flowchart/Use Case Diagrams**

None

**2.3.2 Precondition**

EHR incentive payments have been disbursed during the calendar year and the updated mailing addresses are requested from MCS.

**2.3.3 Steps**

| Steps | Descriptions |
| --- | --- |
| 1. | The NLR receives updated EP mailing address from MCS. |
| 2. | The NLR matches the EP identifying information received from MCS with the existing information in the NLR. |
| 3. | The NLR stores the updated EP mailing address. |
| 4. | The NLR logs audit trail information. |

**2.3.4 Scenario/Use Case Functional & Nonfunctional Requirements**

2.3.4.1 The NLR shall receive the EP’s updated mailing address from MCS.

2.3.4.2 The NLR shall match EP identifying information with existing information in the system.

2.3.4.3 The NLR shall match the EP identifying information based on the following criteria:

* NPI
* TIN
* Payee NPI
* Payee TIN

2.3.4.4 The NLR shall update the EP mailing address in MCS.

2.3.4.5 The NLR shall store the updated EP mailing address.

2.3.4.6 The NLR shall store the EP mailing address with the following information:

* NPI
* TIN
* Payee NPI
* Payee TIN
* Payee Individual Legal Business Name
* Payee Organization Legal Business Name
* Payee 1099 Address

2.3.4.7 The NLR shall log audit trail information.

Expect to document complex URs using several scenarios. Name the scenario and include a description to help distinguish what is unique about each scenario.

1.1.5 Alternative Scenario: Multiple Records Returned from MCS for an EP IRS 1099 Mailing Address

This scenario depicts the disposition process when the NLR receives multiple records in response to an EP mailing address request.

**1.1.5.1** **Scenario Flowchart / Use Case Diagram**

None

**1.1.5.2 Precondition**

EHR incentive payments have been disbursed during the calendar year and updated mailing addresses are requested from MCS.

**1.1.5.3 Trigger**

Multiple addresses for a single EP are received from MCS in response to a 1099 request.

**1.1.5.4 Expected Result**

The first updated EP mailing address information received is successfully stored in the NLR.

**1.1.5.5 Steps**

| Steps | Descriptions |
| --- | --- |
| 1. | The NLR receives multiple records to update an EP mailing address from MCS. |
| 2. | The NLR uses the first 1099 address record received from MCS that matches a stored EP as the 1099 mailing address. |
| 3. | The NLR stores the matching updated EP mailing address. |
| 4. | The NLR logs audit trail information. |

**1.1.5.6 Scenario/Use Case Functional & Nonfunctional Requirements**

1.1.5.6.1 The NLR shall select the first address that matches a stored EP.

1.1.5.6.2 The NLR shall use the following information to update EP mailing address for IRS 1099s:

* NPI
* TIN
* Payee NPI
* Payee TIN

1.1.5.6.3 The NLR shall store the selected updated EP IRS 1099 mailing address information.

1.1.5.6.4 The NLR shall store the following EP IRS 1099 information:

* NPI
* TIN
* Payee NPI
* Payee TIN
* Payee Individual Legal Business Name
* Payee Organization Legal Business Name
* Payee 1099 Address

1.1.5.6.5 The NLR shall store audit log information.

### Writing Functional & Nonfunctional Requirements

As the scenarios are fleshed out, requirements should be identified along the way. Requirements will always clearly state who is doing what to whom. They identify *what* is expected. For example:

The system shall allow access to only credentialed users.

Functional and nonfunctional requirements should always be stated in the form of what is required of the system, not what is required of the user. For example, after writing a login scenario the following requirements may apply:

1. The system shall require a user to change their password monthly.
2. The system shall require that a user not use the same password twice in a row.

This makes it clear what is expected from the system as well as the conditions/constraints that are imposed on the capability. Contrast requirements 1 and 2 above with the following:

1. The user will change his password monthly.
2. A user cannot use the same password twice in a row.

Requirements a and b sound like business rules, not functional requirements.

When documenting a requirement, it is also helpful to consider the following attributes of the requirement.

* What is the purpose of the requirement? Is the requirement written in such a way that it is clear why the requirement exists?
* Does the requirement conflict with another requirement? If it does, there may be some additional scenarios or a business decision that needs to be made.
* Does the requirement depend on another requirement? If it does, does the scenario account for this?
* Is the requirement modular enough so that it can be changed without excessive impact on the system?
* What is the precedence and criticality of the requirement? Low precedence/critical requirements could be postponed for later releases should the schedule become a problem. Requirements may be rated as Essential, Conditional (enhances final system), or Optional (may not be worthwhile).
* Track the history of the requirement. When was it approved and by whom? When was it changed? When was it deleted? Who acted on it? What was the rationale?

It is generally simpler to document the attributes of a requirement using a requirements repository.

Avoid creating requirements that specify a particular design or implementation method (unless there is a policy that constrains these options). For example, this requirement defines how the system will be designed, it is a “how” statement:

The System shall use a Web interface.

While this might be a valid requirement if the organization’s policy is to use Web applications only, the more acceptable way to write this would be:

The system shall allow access for remote, non-network users.

Finally, ensure that the requirement is feasible. Can it be done within the scope of budget and time allotted to the project? If not, several options exist:

* Throw out or postpone the requirement (for a later release)
* Change the requirement or its parent dependency
* Cancel the project (assuming the requirement is critical)

Refer to the style guidelines in Section 4 for more information on phrasing requirements.

#### Types of Nonfunctional Requirements

Nonfunctional requirements will detail the properties of the system. In the process of documenting these requirements, it is not unusual to uncover additional functional requirements which may need to be captured. Nonfunctional requirements include:

* Required States and Modes: identify the various states or modes in which the system can exist.
* Performance Requirements: this is the expected response time of critical system functions or intervals at which functions are expected to trigger. Other types of performance requirements include safety, up time and load.
* Security Requirements: identify the security measures (management, operational, and technical) that provide adequate protection against threats and vulnerabilities to the system based on its system security level. Consider any audit checks or logging that must be performed (e.g., content and handling of log files). Information Security requirements should be based on the CMS information security policies and standards found at: <http://www.cms.hhs.gov/InformationSecurity>, which provides guidance on how to implement federal security standards such as FISMA, FIPS and National Institute of Standards and Technology (NIST).
* Privacy Requirements: this would include privacy (HIPAA) and data protection requirements (e.g., media control, files, data transmission, archives and encryption). CMS policies on privacy requirements can be found at: [http://www.cms.hhs.gov/InformationSecurity](http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/InformationSecurity/index.html?redirect=/InformationSecurity).
* Section 508 Requirements: this would include Section 508 requirements that apply to all government systems based on the specifications of the contract/task order or the technologies in use. Please note that depending upon the scope of a project, it is possible that the project may need to comply with additional CMS Section 508 standards. These standards currently can be found at: [http://www.hhs.gov/Section508](http://www.hhs.gov/web/508/accessiblefiles/index.html).
* Legal Requirements: these include other rules and regulations imposed by the government or other regulatory agencies outside the Agency.
* Human-factors Engineering (Ergonomics) Requirements: these are also referred to as Look and Feel requirements. They may take the form of interface sketches or more detailed scenario models. They will help the designer understand how the user works with the functionality of the system. Identify areas that need concentrated human attention and are sensitive to human errors. Also consider manual operations to be performed that affect the system.
* Design Constraints and Qualification Requirements: identify what laws the system must uphold as well as any applicable standards the system must comply with. Consider any organizational policies that will affect the operation or performance of the system. List any existing components that could be reused.
* System Quality Characteristics: quantify the expected accuracy of the results produced by the system such as precision or units of measure.
* Internal Data Requirements: identify the essential objects/entities/classes that are germane to the system. This could be in the form of a first cut data model or a domain model. Consider integrity constraints, as well as retention and data replication requirements. Determine if any data conversion is required or any Interface Control Documents that must be negotiated or adhered to.
* Usability: identify the ease of learning for the expected users of the system. This would include statements of how easy the system is to use from the perspective of all users. A description of the help services that the system will provide would also be appropriate.
* Operational: specify the physical environment in which the system will operate. This would include statements on the expected technological environment and partner applications that the system will interface with.
* Maintainability: identify the amount of time necessary to make specified changes to the system or if there are special conditions that apply to the maintenance of the system.

### Determining Pass/Fail Statements

The use of pass/fail statements is optional based on project schedule and budget. When used, the pass/fail statement describes how to determine if the requirement was met. Include the pass/fail statement with the description of the requirement for easy reference. For example:

* The system shall require a user to change their password monthly. (Requirement)

Pass/Fail statement: Upon login, the system will check to see if more than 1 month has passed since the password was last changed and prompt the user to choose a different password when that occurs.

Writing the pass/fail statement can also help show if there are extra steps (or a whole scenario) that were overlooked and if the requirement is ambiguous. In this example, the pass/fail statement reveals an additional requirement about how to handle an error condition:

* The system shall require that a user select a different password when required to change their password. (Requirement)

Pass/Fail statement: If the user enters their existing password as their new password, the system will notify them that their entry is invalid and request a different password.

Writing the pass/fail statement now not only helps the designer understand the requirement better, it makes writing test procedures simpler. It may also help determine whether this is a requirement or an assumption. For example, consider the following requirement:

* The system shall utilize the user’s network username as their logon ID.

If the system lacks the ability to test whether or not the user’s network username actually exists, an assumption has been uncovered rather than a requirement.

Pass/Fail statements may also be written from a perspective of whether a requirement from the standard scenario or the alternative scenario is being evaluated. For example, these two requirements are applicable in both scenarios but the pass/fail statements reflect what is being evaluated in light of the expected result of the scenario:

Standard Scenario:

* The system shall authenticate a user after credentials are validated.

Pass/Fail statement: The user is authenticated to the system when valid credentials are submitted.

Alternative Scenario:

* The system shall authenticate a user after credentials are validated.

Pass/Fail statement: Invalid user credentials are rejected when submitted.

# Global Requirements

## Overview

Global requirements are not a type of requirement but relate to the scope of a requirement. Global requirements are user, functional, and nonfunctional requirements that apply to the entire system. They are documented in a global requirements section to avoid repeating the requirements in other sections of the document. Global requirements are grouped by requirement category or by applicable scenarios. For example: the global requirements section holds nonfunctional requirements related to security, privacy, and Section 508 compliance.

User level requirements are functional and nonfunctional requirements that are like global requirements except that they pertain to a specific user. User level requirements are often captured when a repetitive requirement is identified across multiple scenarios for a specific user requirement.

## Reporting Requirements

Reporting requirements provide an important cross check for the business analyst. As stakeholders and users identify the need to report on specific data points, the business analyst should be able to trace these reporting needs to their entry point in the system. At the business and functional level, the analyst should not be concerned with how the report is created or generated (pre-defined, batch, ad hoc, etc.). Rather, the analyst should concentrate on the content and currency of the data.

Since requirements are documented for functionality that must be repeated many times over and in the same manner, capturing requirements for purely ad hoc reports (reports that are defined and executed as needed) is not done at a detailed level. By their nature, ad hoc reports do not conform to the rigor outlined in this guidance. The desire to support ad hoc reporting should be captured by including a global design requirement that states the system must support that feature. Keep in mind, however, that doing this introduces the risk that some necessary data element will not be captured by the system. It is incumbent on the users and stakeholders to identify where all necessary data elements must be stored in the system.

While it is preferable to not repeat requirements in order to minimize change management issues, documenting reporting requirements often poses special problems due to the number of shared requirements among many users of the system. This is not as significant a problem for business requirements, but trying to organize reports by user can lead to confusion when documenting functional and non-functional requirements. As such, it may be more straightforward to organize reporting requirements a little differently, especially for larger systems. Section 4.2.1 below outlines this approach.

### Business Requirements

Business reporting requirements should state the purpose of the report as opposed to naming specific reports to generate. The goal is to capture what purpose the information in the report will satisfy for the business owner. Business rules are then used to specify other significant parameters. For example, the business rules can identify:

* Key data points – the primary information the stakeholder requires in order to satisfy the requirement
* Frequency – how often the report is required, such as weekly, monthly, or yearly
* Calculations – what counts, totals, quotients, means, etc. must be performed
* Sorting and grouping – how the information should be ordered or subtotaled
* Constraints – if applicable, configurable parameters such as a specific time and date range or filtering criteria
* Access – which users have full or partial view of the report, e.g., detail or summary data, PHI data, segmented data sets

The number of business owners may affect the outcome of reporting requirements. If several business components require similar reports, the best practice would be to treat each of them as having separate business needs represented as separate requirements. This maintains the ownership of the requirement and may capture subtle nuances. When the functional and nonfunctional requirements are written later, additional details are elaborated and similar reporting requirements can be merged.

### User/Functional/Nonfunctional Requirements

The number of reports required is generally driven by the number of different user roles in a system and the number of business processes being supported. As the number of reports grows, it is not uncommon to find increased similarity in the reporting requirements by various user roles. This can result in redundant reporting requirements across user roles and make management of the requirements more complicated. In these situations, it may be beneficial to consolidate and organize reporting requirements in a different manner. In a situation where there are fewer user roles, it is best practice to organize reporting requirements using the approach already described. That is, within each user role section of the requirements document, a set of scenarios is defined which lay out the functional/nonfunctional requirements for each report. For example:

1. User Role 1
   1. User Requirement 1
   2. User Requirement 2
   3. Reporting Requirements
      1. Scenario 1 - Report 1
      2. Scenario 2 - Report 2
      3. Scenario 3 - Report 3
2. User Role 2
   1. Etc.

In a more complex situation, other approaches should be considered that will make it easier to manage the reporting requirements while also reducing redundancy. For example, reports could be consolidated without regard to which user role owns which report:

1. User 1
2. User 2
3. Reports
   1. Global Reporting Requirements
   2. Scenario 1 - Report 1
   3. Scenario 2 - Report 2

When using this approach, a matrix should be included in an appendix to define which user roles have access to the reports and a requirement included in the Global Reporting Requirements to affirm that the system shall implement the privileges assigned therein. One axis of the matrix lists the reports and the other axis identifies all the user roles. The intersection provides guidance on whether any given report can be viewed by a specific user role. Following this would be as many subsections as needed to describe all the reports required.

Taking this approach will mean that change control will have to be managed by a group of users later in the project, but the advantages include a higher reuse of requirements, reduced design and development costs, and potentially reduced system utilization costs.

# STYLE GUIDELINES

The process of documenting requirements is a never-ending cycle. When using thorough writing techniques, it is quite easy to spend a considerable amount of time eliminating ambiguity in order to ensure all assumptions, as well all the requirements, have been identified. This process inevitably leads to new requirements that should be examined as well. It falls on the project manager to decide what level of detail and investigation is required for the requirements document before a diminishing amount of return is gained from effort put forth.

This section of the guide discusses what might be considered a “middle level” investigation of requirements. Proper layout, form, and phrasing of requirements can help ensure the document is complete and clear. For an even deeper level of requirements specification, please refer to Appendix B.

## Writing Style

CMS uses the Government Printing Office (GPO) Style Manual as the basis for writing. The GPO Style Manual is available through the following URL: <http://www.gpoaccess.gov/stylemanual/browse.html>. In some cases, the procedures in this guide depart from GPO rules. This guide takes precedence over conflicting GPO rules.

A properly structured requirement must be a complete sentence that has only one interpretation. Any individual who writes requirements should avoid using phrases, single words, or a collection of acronyms to express requirements. Each requirement should consist of a subject, verb, and predicate as noted in the requirement written below:

Requirement example:

The Medicare Managed Care System shall assign a disenrollment date for a beneficiary in a Managed Care Organization when a termination date for the beneficiary’s Part B eligibility is received.

The subject of the requirement is “The Medicare Managed Care System” and the predicate is “assign a disenrollment date for a beneficiary in a Managed Care Organization when a termination date for the Beneficiary’s Part B eligibility is received.” The subject of any requirement should imply ownership and shows the entity to which the requirement should be related. The predicate should be an action phrase or expression of something that must be accomplished for, by, or to the subject. The verb used in requirements is always “shall.” This implies that the requirement must be adhered to.

Keep the following in mind:

* Write in such a way that the requirement does not pose a specific solution. It should always be design independent.
* Write in such a way that there is only one interpretation of the requirement.
* Increase readability by using grammatically correct language and symbology, using simple words/phrases/concepts, defining unique words or symbols.
* Write a simple, complete, well-structured sentence that states one thing and states it well.

## Quality Attributes of Requirements

All requirements must have the following quality attributes:

### Correct

Correctness is defined as conforming to or agreeing with fact, logic, or known truth without error. Each requirement must accurately describe what needs to be done; it is accurate, technically precise, and legally without error.

BR1: The Medicare contractor shall pay for acupuncture services.

This BR is not correct because Medicare doesn’t cover acupuncture services.

### Complete

Completeness is defined as expressing a whole idea or statement. Each requirement must fully describe what needs to be done and provide everything the reader needs to know; it includes all necessary parts, elements, or steps.

BR1: The Medicare contractor shall pay for mammography services.

This BR is incomplete because it does not include any information about the conditions for payment, frequency, provider type, place of service, mammography type, or service dates.

### Clear

Clarity is defined as being unambiguous and not confusing. All readers of a requirement should arrive at a single, consistent interpretation and conclusion; it is free from cloudiness, haziness, or muddiness.

BR1: The Medicare contractor shall not issue demand letters or recoup Part A payments made to fee-for-service providers who have received payments on behalf of these individuals who have been terminated from the program retroactively for Part A.

This BR needs to be clarified, then divided into more than one requirement; additionally, a requirement should be written in the positive.

### Consistent

Consistency is defined as not being in conflict with other requirements. Consistent requirements do not conflict or contradict other requirements; they are free from variation.

BR1: The business owner shall process payments.

BR2: The contractor shall process payments.

One BR refutes the other.

### Verifiable

Verifiability is defined as being able to meet the requirement. A requirement should be testable, validated either in an automated system or manually by observation, and should be written in quantifiable (measurable) terms.

BR1: The Medicare contractor shall ensure that they pay the correct amount for covered and correctly coded services in a timely manner for eligible beneficiaries by legitimate providers.

This requirement is not verifiable because the conditions are not measurable. The usage of vague words such as timely, suitable, appropriate, applicable, usually, generally, often, typically, and flexible should be avoided.

### Traceable

Traceability is defined as being able to be uniquely identified and tracked. It allows tracking a requirement from its source and allows the creation of a Traceability Matrix. Every requirement is uniquely identified:

BR1

BR2

Etc.

### Feasible

Feasibility is defined as being able to be accomplished within cost and schedule. Requirements should be written so that it is possible to implement them within the scheduled budget and time frame.

BR1: The Medicare contractor shall process home health claims within ten days of receipt.

The current time frame is forty five days. Ten days is not a realistic requirement.

Avoid creating requirements that specify a particular design or implementation method (unless there is a policy that constrains these options). For example, this requirement defines how the system will work; it is a “how” statement:

BR1: The System shall send an e-mail.

This might be a valid requirement if the organization’s policy is to use only web notifications. The more acceptable way to write this would be:

BR1: The System shall send a notification.

### Modular

Modularity is defined as a unit being able to be changed without excessive impact. Requirements should be revisable with minimal impact to other requirements in a specified unit. Updates to one requirement should not have a major impact on other requirements. Organize like requirements together by process or function; e.g., monthly payment cycle, annual election period, enrollment, registration, etc.

### Design-independent

Design-independent is defined as not posing a specific solution. Business requirements should describe what needs to be done and not how it should be done.

BR1: The standard system maintainer shall create a bypass-able edit to not allow the number of units for ECT to exceed the number of covered days.

This requirement states how to perform the function.

1. ACRONYM LIST

| Acronym | Description |
| --- | --- |
| BPM | Business Process Model |
| BR | Business Requirement |
| CMS | Centers for Medicare & Medicaid Services |
| DOORS | Dynamic Object Oriented Requirements System |
| EIA | Electronic Industries Association |
| FIPS | Federal Information Processing Standards |
| FISMA | Federal Information Security Management Act |
| FR | Functional Requirement |
| GPO | Government Printing Office |
| GTL | Government Task Leader |
| HIPAA | Health Insurance Portability and Accountability Act |
| IEEE | Institute of Electrical and Electronics Engineers |
| IIBA | International Institute of Business Analysis |
| IT | Information Technology |
| NIST | National Institute of Standards and Technology |
| NR | Nonfunctional Requirement |
| OIS | Office of Information Services |
| RD | Requirements Document |
| RU | Business Rule |
| RWG | Requirements Writer’s Guide |
| SME | Subject Matter Expert |
| TLC | Target Life Cycle |
| UR | User Requirement |

1. Components OF REQUIREMENTS AND RULES

These attributes help ensure completeness and to help standardize how requirements and rules are documented and tracked in the Requirements Repository. The unique characteristics of each project, however, may necessitate the tailoring of attributes within this appendix. For example: The Dynamic Object Oriented Requirements System (DOORS) does not support adding a prefix to a requirement identification number. Therefore the requirements numbering or unique identifier scheme may be adapted based on the situation.

B.1 Business Requirement Attributes

Directions

**Requirement Number –** Provide a unique identifying number. This number will be important when establishing traceability.

**Requirement (the “shall” statement) –** Provide a complete sentence using the word “shall” that describes the capability the business needs to meet their objective.

**Source** (optional) – List the legislation or strategic objective document, the BPM, information regarding meetings, or emails from which this requirement originated.

**Priority** (optional) – Identify the requirement as “High” if essential to the project/system, “Medium” if desirable but not essential, and “Low” if optional.

**Purpose** (optional) **–** Provide a brief rationale for the requirement.

**Comments** (optional) – Capture any additional information associated with the requirement under the comments column for the requirement.

Example of a business requirement

**Requirement Number** – BR-001

**Requirement** **–** CMS shall implement its payment calculation using established Medicare processes to ensure the integrity of the payment calculation methodology.

**Priority** **–** High

**Purpose** **–** This is why CMS must interact with Grouper, OCE, OCE for Indian Health, MCE, Pricer, UPIN, Oscar, Physician Fee Schedule, and Ambulance Fee Schedule, and EDB

**Source** **–** Medicare Modernization Act of 2003

B.2 Business Rule Attributes

Directions

**Rule Number –** Provide a unique identifying number. This number will be important when establishing traceability.

**Rule –** Provide a complete sentence that defines or constrains some aspect of the business.

**Associated Business Requirement(s)** – Provide the business requirement number(s) of the parent business requirement(s).

**Type** – Provide the category of the rule (Term, Fact, Derivation, Assertion, Action Enabler).

**Source** – Provide the legislation, strategic objective, or policy guidance document.

Example of a Business Rule

**Rule Number –** RU-0099

**Rule –** A Plan Sponsor is an entity that sponsors a health plan. This can be an employer, union, or some other entity.

**Associated Business Requirement(s)** – BR-001, BR-002

**Type** – Term

**Source** – CMS Website

B.3 User Requirement Attributes

Directions

**Requirement Number –** Provide a unique identifying number. This number will be important when establishing traceability.

**Requirement (the “shall” statement) –** Provide a complete sentence using the word “shall” that describes the capability needed by the business to meet their objective.

**Associated Business Requirement** – Provide the user requirement number of the associated business requirement.

**Source** (optional) – Provide the stakeholder who identified or requested the requirement.

**Priority** (optional) – Identify the requirement as “High” if essential to the project/system, “Medium” if desirable but not essential, and “Low” if optional.

**Purpose** (optional) **–** Provide a brief rationale for the requirement.

Example of a User Requirement

**Requirement Number –** UR-001.

**Requirement (the “shall” statement) –** CBC shall issue monthly payments.

**Associated Business Requirement** – BR-100

**Purpose** (optional) **–** This allows for plans to be paid for services.

**Source** (optional) – CBC

B.4 Functional/Nonfunctional Requirement Attributes

Directions

**Requirement Number –** Provide a unique identifying number. This number will be important when establishing traceability.

**Requirement (the “shall” statement) –** Provide a complete sentence using the word “shall” that describes the action or expectation of what the system will do (functional requirement), or the behavioral property the system must have.

**Purpose** (optional) **–** Provide a brief rationale for the requirement

**Validation Measure** (Demonstration, Test, Analysis, Inspection, Special Qualification Method, or Not Applicable) – Provide a description of how the requirement will be validated.

**Pass/Fail Statement –** A testable statement that describes how to know if a requirement was met

**Associated User Requirement** – Provide the user requirement number of the associated user requirement.

**Dependent Requirement –** List any requirements that this requirement is dependent on.

**Priority (Essential, Conditional, Optional) –** Provide the priority of implementing this requirement from the user perspective. Essential means “must have,” Conditional means “like to have,” and Optional means “could do without.”

**Source** – Provide the name of the stakeholder who identified the requirement.

**Requirement Type** – For nonfunctional requirements, this is the category of requirement it falls within.

**Comments –** Provide any comments if necessary.

**Note1**: Nonfunctional requirements will be a subset of the RD. Be sure to include which category (from Section 3.3.2.1) the nonfunctional requirement falls into.

**Note2:** These attributes convey the recommended information to capture in each requirement. Project managers are encouraged to add any other information that is appropriate for their projects. If the requirements are initially written in software other than DOORS, the formatting of this information is left to the discretion of the project manager. Multiple requirements may be displayed on one page. A tabular format (e.g., MS Word or MS Excel tables) may also be used.

Example of a Functional Requirement

**Requirement Number** – FR-001

**Requirement (the “shall” statement) –** The system shall confirm that a check has not been cashed.

**Purpose –** This allows the Customer Service Representative to know if the customer has already received payment.

**Validation Measure** (Demonstration, Test, Analysis, Inspection, Special Qualification Method, or Not Applicable) – Test

**Pass/Fail Statement –** Examining the current status of the check will indicate if the check has been cashed.

**Associated User Requirement** – UR-001

**Dependent Requirement** – None

**Priority (Essential, Conditional, Optional)** – Essential

**Source** – John Doe

**Comments** – Implement in Release 2

Example of a Nonfunctional Requirement

**Requirement Number** – NR-009

**Requirement (the “shall” statement) –** The system shall meet Section 508 compliance for all interface screens.

**Purpose –** Section 508 is a legal requirement which enhances the accessibility of electronic and information technology.

**Validation Measure** (Demonstration, Test, Analysis, Inspection, Special Qualification Method, or Not Applicable) – Test

**Pass/Fail Statement –** Each screen is evaluated according to the HHS checklist for web-based applications.

**Associated User Requirement** – UR-002

**Dependent Requirement** – None

**Priority** (Essential, Conditional, Optional) – Essential

**Source** – Jane Smith

**Requirement Type** – Legal compliance

**Comments** – None

1. Best Practices for Verbs and Phrases

The following table presents action verbs commonly used to write business and user requirements.

| Table : Commonly Used Verbs for Business and User Requirements | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Apply | Approve | Attest | Audit | Authorize | Collaborate | Complete | Conduct |
| Confirm | Define | Designate | Determine | Disburse | Disposition | Evaluate | Fulfill |
| Initiate | Maintain | Manage | Modify | Monitor | Notify | Obtain | Post |
| Receive | Record | Request | Research | Respond | Review | Select | Send |
| Store | Submit | Subscribe | Summarize | Support | Suspend | Track | Update |
| Verify | View | Withdraw |  |  |  |  |  |

The following table presents action verbs commonly used to write functional/nonfunctional requirements.

| Table : Commonly Used Verbs for Functional/Nonfunctional Requirements | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Accept | Access | Activate | Assign | Authorize | Calculate | Cancel | Complete |
| Contain | Control | Convert | Create | Deactivate | Default | Delete | Display |
| Exclude | Extract | Generate | Insert | Log | Notify | Populate | Prohibit |
| Prompt | Reject | Require | Retrieve | Save | Search | Set | Sort |
| Supply | Suspend | Validate |  |  |  |  |  |

The following table presents words to avoid when writing requirements.

| Table : Words to Avoid | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Achievable | Adequate | Adopt | Allow | And/or | Approximately | Assemble | Assure |
| Bad | But | Can | Clear | Complete | Consider | Could | Damaged |
| Degraded | Distinguish | Easy | Effective | Efficient | E-mail | Ensure | Except |
| Far | Fast | Flexible | Generally | Good | Identify | Locate | Low |
| Maximize | May | Might | Minimize | Modular | Near | New | Nominal |
| Normal | Normally | Often | Old | Optimum | Optionally | Ought | Perhaps |
| Probably | Render | Satisfy | Scalable | Should | Significant | Slow | Strong |
| Suggest | Supervise | Survive | Tailor | Timely | Typically | Useful | User-friendly |
| Usually | Versatile | Weak | Well |  |  |  |  |

Additional phrases to avoid include:

* “but not limited to…”
* “be capable of…”
* “be designated to…”
* “to the maximum extent…”
* “as much as possible…”

1. ADVANCED STYLISTIC APPROACHES

Natural language is widely used in industry to state requirements for all types of systems because it is flexible and universal. However, natural language has one major drawback, which is its inherent ambiguity. This appendix describes methods that may be used to reduce ambiguity in written requirements documents.

D.1 Deletion

The process of deletion reduces a person’s perception to a scope they can deal with. In other words, things that seem to be unimportant are not recognized by a person or, in the context of requirements, not mentioned in a requirements document.

Implicit Assumptions

Implicit assumptions are those statements that are obvious or banal to the writer and are therefore omitted. Employ these techniques to aid in making implicit assumptions explicit.

Determine the main verb of a sentence. Form a new sentence by negation of the verb. Ask the question: Which statements must be true in order to guarantee that both sentences make sense? All statements found may indicate an implicit assumption.

Requirement example:

The Medicare Managed Care System shall assign a disenrollment date for a beneficiary in a Managed Care Organization when a termination date for the beneficiary’s Part B eligibility is received.

The negation of this requirement would be:

The Medicare Managed Care System shall not assign a disenrollment date for a beneficiary in a Managed Care Organization when a termination date for the beneficiary’s Part B eligibility is not received.

The statements (assumptions) that must be true in order to guarantee that both sentences make sense include:

* That the beneficiary’s Part B eligibility will be transmitted
* That the beneficiary is part of a Managed Care Organization

Under-specified process words are verbs, adjectives, and adverbs that are not sufficiently quantifiable.

To detect under-specified process words, determine the verbs of the sentence. If it is possible to have sentences with multiple actors, something was deleted in the original sentence (e.g., the word transmit implies the questions what is transmitted and what are the aim and the source of the transmission).

Process Words

Process words are verbs, adjectives, and adverbs. These types of words are often ambiguous and should be replaced by quantifiable measures.

* The system shall supervise the resources assigned to the patient care to ensure the maximum limit is not reached.

Use questions about the process words to determine if they are sufficiently specified:

* Supervise: what exactly is the system supervising and how?
* Resources Assigned: who or what can be assigned?
* Ensure: when and how is the maximum limit ensured?
* Maximum: what is the maximum limit?

Incomplete Superlatives and Comparisons

Incomplete superlatives and comparisons need a reference point to be completely specified. Moreover, there must be a metric and a predefined scale to compare certain aspects.

Use the following questions to detect incomplete superlatives and comparisons: Is there a reference point? What is the metric to measure a certain characteristic? What is the scale of the metric? For example:

* The functions should be easily changeable.

“Easily changeable” infers a reference point to something that is not easily changeable. What is the feature being compared to? Of course, “easily” is also a process word that should be examined (see above).

Modal Operators of Possibility

In addition to “what” a system shall do, the means to realize the “what” must be specified. Examples of modal operators of possibility include: *may*, *can*,etc. Statements that describe a possibility or impossibility shall be analyzed by using the question: Which means enable or prevent a certain system reaction? For example:

* Updating a patient’s pre-existing conditions may not overwrite any existing history. By what means is the update prevented?

Modal Operators of Necessity

Ensure that statements regarding the system’s behavior contain the desired behavior and the exception behavior. Examples for modal operators of necessity include: must, shall, and should. Each sentence that contains a modal operator of necessity shall be inspected to determine if it is necessary to define the behavior of the system in the case of an exception. For example:

* The system shall forward confirmation of the activity to the user.
* What if forwarding services are not available or an invalid forwarding destination is provided?

D.2 Generalization

Generalization is defined as a process that leads to a detachment of an experience from its context and to assume that the experience is overall valid. In the requirements analysis process, generalization may lead to the omission of several behavior aspects of the system (e.g., special cases or error exceptions).

Universal Quantifiers

These elements are statements about a certain amount. They summarize a set of objects. Using universal quantifiers may lead to the problem that some objects summarized in the set do not realize the specified behavior. Universal quantifiers include: *never, ever, not any, everyone, no one, someone, or nothing.*

Find all universal quantifiers. Check if the specified behavior of the system is really valid for all objects of the set summarized by the universal quantifier. Furthermore, check each sentence to determine if there is an explicitly defined set of objects the specified behavior is valid for. For example:

* Each communication shall be marked with a timestamp.

Every communication? Are there no exceptions?

Incomplete Specified Conditions

Conditions have the following sentence structure: *if condition then behavior1 else behavior2*. In requirements specifications the second case is often omitted. Key words to look for are: *when, then, if, in the case of, dependent on.*

Find the condition of the statement. Be sure that the requirement specifies the behavior for all possible cases (the “then” and the “else” branch, respectively). Additionally, the analyst may check the following questions: Are all the possible conditions covered? Are all possible variants described? For example:

* If the automatic data communication is disrupted, it should be possible to edit the individual fields of the transmission received.

What is possible if no disruption occurs?

Nouns without a Reference

This omission occurs when a noun of a statement is not specified sufficiently. Examples for insufficiently specified nouns are: *the user, the controller, the system, the message, the data, and the function*.

Check each noun of the requirements specification to determine if it specifies a defined person, a defined group of persons, or a defined real world object (or group of objects). Insufficiently specified nouns can be identified by means of the following questions: *Who exactly? What in detail? Which part of the set?* Each noun specified insufficiently must be extended by a completion, which specifies the noun exactly. For example:

* The system shall display the data to the user electronically.

Which data exactly? Which users exactly?

D.3 Distortion

This phenomenon is related to the so-called nominalization, i.e., a noun stands for a complex process. Nominalization does not cause any problems as long as the process is unique, exactly specified, and unambiguous in the current context. Examples of nominalizations include: the recording, the playback, the takeoff, etc.

Check each noun of all sentences, asking the question: Is it possible to find a verb which describes a process and which is similar to the noun (for example, a “recording” is both a noun and a verb)? Try using the following phrases:

* Is it possible to add the substantive to the phrase: a continuous...?
* Does the noun describe something that is not touchable?

If a question is answered with “yes”, the analyst has to check to see if any information was lost when describing a process by means of a noun. For example:

* The system shall report on the status of the activity.

What is reported? Where is it reported?

D.4 Other Guidance

An overall quality aspect of requirements statements is non-ambiguity. In the English language the frequent use of the gerund leads to misunderstandings and ambiguities. When using a gerund, the relation of the verb and the subject performing the verb is lost. For example:

* Reporting violations should be immediate.

This has two interpretations:

* Violations should be reported immediately.
* Reports of violations should appear immediately.

D.5 Writing Stronger Sentences

The federal government encourages using simpler words or phrases and avoiding the use of words or phrases that may unnecessarily complicate documentation.

1. Requirements Elicitation Approaches

Before business requirements can be written, they must be discovered. This appendix presents a variety of techniques the analyst team may use to enable the stakeholders to explain their business requirements.

E.1 Requirements Workshop Sessions

Definition – A requirements workshop is a structured meeting conducted in a group setting that consists of a broad spectrum of participants including SMEs and IT specialists. Requirements workshops are also referred to as: Joint Application Design (JAD) Sessions, Elicitation Workshops, or Facilitated Workshops. They should follow a detailed agenda to ensure key topics are discussed efficiently in the group setting.

Requirements Workshop Advantages

* Requirements workshop sessions bring experts together, giving them a chance to share their views and understand the views of others.
* Decreases the time associated with the requirements elicitation process.
* Miscommunication issues are identified and resolved early in the design process.

Requirements Workshop Disadvantages/Risks

* The participant selection is critical to the success. The wrong invitees may prevent key topics from being adequately addressed. Likewise, the lack of key people absent from the workshop may prevent key decisions from being made.
* The facilitator’s role is a key feature to the success of the requirements workshop. The facilitator must ensure the discussion stays on track, that meeting rules are followed, and that the meeting is as productive as possible. If the facilitator is not properly trained, the requirements workshop will become inefficient at identifying the requirements.

When to Use Requirements Workshop Sessions

* Requirements workshop sessions should be used when a variety of sources must reach consensus on the requirements in a time-efficient manner.

E.2 Interviews

Definition – Interviews are the technique of conducting a set of meetings with each stakeholder group. Interviews elicit requirements strictly from one stakeholder’s point of view.

Interview Advantages

* Permits a focused exploration of the stakeholder activities required to support the business process.
* Obtains a richer understanding of the stakeholder's unique business processes, decision-relevant business rules, and perceived needs that is often not elicited in JAD sessions.

Interview Disadvantages/Risks

* Interviews are time-consuming. It usually takes a number of sessions to complete the requirements identification. The original time estimates may get extended throughout the interview schedule based on emergent discoveries of processes and stakeholder interactions.
* Cross-functional impacts on the process may not be identified. There is a risk that the result may omit relationships and dependencies between requirements.
* The analyst must spend time developing a robust set of interview questions. The analyst must be highly skilled in conducting interviews that drill down into the stakeholder activities.

When to Use Interviews

* Typically, processes that have a small stakeholder population are best suited to this technique.

E.3 Brainstorming

Definition – Brainstorming is a technique that narrowly focuses on finding a solution to a topic or problem by eliciting a high volume of ideas.

Brainstorming Advantages

* Encourages the building upon ideas expressed by others.
* Produces very rapid idea generation. The quantity is more important than quality.

Brainstorming Disadvantages/Risks

* Improper facilitating results in too many ideas that are not properly evaluated.
* Debating ideas during the session inhibits members from expressing themselves.

When to Use Brainstorming

* Brainstorming should address a specific question or a narrow area of study. The session must be conducted to generate ideas rather than judgment. The conference should include experts and novices to ensure a broad range of ideas are generated.

E.4 Document Analysis

Definition – A means to elicit requirements from business documents or an existing system by studying available documentation and identifying relevant information. These sources of information allow the analyst to gather details of existing solutions (the “as is” situation) to see if they have components that can be used or should be changed for the new solution that is being proposed. Additionally, policy or rulemaking documentation can be analyzed to identify the “to-be” business rules and requirements.

Document Analysis Advantages

* Permits acquiring information without the need to involve stakeholders.
* Reduces the amount of time required to involve SMEs.
* Provides analysts with a foundation for more efficient use of other elicitation techniques.

Document Analysis Disadvantages/Risks

* The documentation is not current or is ambiguous.
* All facets of the process are not recorded or considered in the documentation.

When to Use Document Analysis

* Document Analysis is best applied when the documentation is available and the SMEs are not available.
* This technique should be used prior to starting other elicitation tasks.

E.5 Focus Group

Definition – A focus group is composed of individuals who are qualified to discuss or comment on a topic.

Focus Group Advantages

* Focus groups can provide accurate feedback to an existing or proposed solution.
* Isolating the discussion to a focused topic results in an in-depth qualitative research activity.

Focus Group Disadvantages/Risks

* Limiting the number of participants may have the group react in a group-think manner.
* The results may not represent the requirements of the stakeholders at large.

When to Use Focus Groups

* Focus groups can be used during any time during the requirements development life cycle.

E.6 Interface Analysis

Definition – Interface Analysis is initiated by analysts to reach agreement with stakeholders on what interface requirements are needed.

Interface Analysis Advantages

* This technique allows the definition of the interface requirements that enable the process components to interact with external systems.
* Helps clarify system boundaries, separates which system provides specific functionality.
* Ensures compatibility with other systems.

Interface Analysis Disadvantages/Risks

* Employing this technique too early in the process may result in capturing the design solution prior to defining the business or functional need.
* Analysis of the interfaces captures a limited scope of the requirements.

When to Use Interface Analysis

* Interface analysis must be used when a component is impacted by systems that exist outside of the processing boundaries.
* Typically used towards the end of the requirements analysis life cycle.

E.7 Observation

Definition – Observation relies on studying people performing their jobs. It is sometimes called “job shadowing.”

Observation Advantages

* Permits the analyst to observe existing activities to provide a basis of knowledge from which to create process improvement requirements.
* The bias effects are reduced and sometimes eliminated, which ensures that the collected data is objective and accurate.

Observation Disadvantages/Risks

* It may take some time before a specified action takes place for the analyst to observe because activities are not meant to be provoked.
* May be disruptive to the person being shadowed.

When to Use Observation

* This technique may be used for existing (as-is) processes where the majority of the current processes involve work that is easily observable.
* This technique is not recommended when most of the work involves the user applying a high level of intellectual analysis activity.

E.8 Prototyping

Definition – Prototyping is a means to visualize requirements before the application is designed or developed. Typically, “mock ups” of screen displays or report layouts are presented for an application.

Prototype Advantages

* Promotes early user interaction and feedback. Not only will the users correct any misconceptions by the analyst, but they will likely recognize misconceptions or requirements they did not anticipate on their own.

Prototype Disadvantages/Risks

* The prototype may lead to unrealistic stakeholder expectations. To mitigate this risk, the analyst must have a solid understanding of what may be produced based on the target architecture and resource availability.
* The prototype may result in user feedback to include features that are out of scope from the original project plan (i.e., "scope creep").

When to Use Prototyping

* Prototyping is a desired technique when reaching a solution calls for simulation, experimentation, or incremental evaluation.

E.9 Reverse Engineering

Definition – The process of analyzing the technical components of an existing system or product to identify underlying business processes, data, and rules.

Reverse Engineering Advantages

* Permits extracting design decisions from end products with little or no additional knowledge about the procedures involved in the original production.
* Requires minimal interaction with SMEs and stakeholders.

Disadvantages/Risks

* The end product may contain incorrect or obsolete features. The risk is that these features are perpetuated in the new requirements.
* Requires the analyst to have specialized technical and analytical skills. The analyst must be able to abstract from specific software platforms to generalized requirements.
* Existing tools that support reverse engineering have limited capabilities, require training to use, and may incur additional expense.

When to Use Reverse Engineering

* Reverse engineering is useful when there is existing software to analyze.
* It is applied in situations where the SMEs for the existing system are unavailable.

E.10 Survey

Definition - The means of eliciting information from many people in a relatively short time. A survey administers a standard set of questions to the stakeholders and SMEs.

Survey Advantages

* Efficiently gathers information from a large group of representative stakeholders in a short timeframe.
* Surveys can be administered anonymously thus ensuring an uninhibited response.

Survey Disadvantages/Risks

* The survey questions must be skillfully constructed. A bad set of survey questions will yield misleading data.
* Surveys must result in enough responses to provide meaningful data.

When to Use Surveys

* Surveys may be used when the study requires feedback from a large number of stakeholder participants.
* Surveys are effective and efficient when stakeholders are not located in one location.
* Surveys are also effective when requirements are desired without peer influence (i.e., anonymous surveys).