NuWave TECHNOLOGIES

Job Aid: Requirements Development Guidelines

CONTENTS

I. Introduction

This job aid outlines how to approach requirements development. You need requirements to define and analyze potential solutions to business and system needs. You identify requirements to solve a problem or achieve a specific objective. Requirements do not explain how you will implement a function; rather they define performance expectations and desired performance levels. Requirements also address constraints to take into consideration. Identify these constraints while developing the requirements for a new application or for an existing system.

II. Requirements Development Process

Requirements development includes eliciting, documenting, verifying, analyzing, prioritizing, validating, and communicating customer and business needs. You develop and baseline requirements in the Plan and Analyze stages. Thereafter, each stage of the application development life cycle references the requirements. You accept and record additional requirements and updates to existing requirements using the configuration management process.

- During the Plan stage, you develop high-level requirements based on stakeholder input. You elicit stakeholder needs, expectations, constraints, and interfaces (internal and external), and then analyze, refine, elaborate, and translate them into high-level requirements. These requirements cover all stages of the application development life cycle. The high-level requirements may include needs, expectations, and constraints with regard to verification and validation. The high-level requirements provide the starting point for the Analyze stage. Note that high-level requirements here are equivalent to CMMI's customer requirements.
- During the Analyze stage, you collect stakeholder feedback and information on system behaviors. The focus is to elicit, verify, validate, and baseline the product requirements to provide the foundation for design, build, and test. In the Analyze stage, you analyze high-level requirements in conjunction with operational concepts (e.g. business processes, use cases, user scenarios) to drive out a more detailed and precise set of product

requirements. In this stage, you group the product requirements and allocate them to high-level requirements and product functions or process areas. You then further allocate them in the Design and Build stages to product components which may include objects, services, processes, or interfaces. The project team also uses the high-level and product requirements in developing the test scenarios and conditions. Note that product requirements here are equivalent to CMMI's product and product component requirements. Product requirements are also called detailed requirements.

- During the Design stage, you translate product requirements into application functional designs. Also in this stage, you detail out test conditions and expected results based on the requirements and designs.
- During the Build stage, you translate the functional designs into detailed technical designs and build out product components. The product you create in this stage is a realization of the requirements.
- During the Test stage you prepare and execute assembly, product, performance, user acceptance, and operational readiness tests. You also verify and validate that the product is operating in accordance with all of the requirements.

    - Product Test verifies coverage of the product requirements.
    - Performance Test verifies coverage of the performance requirements, and may additionally addresses other product requirements.
    - User Acceptance Test validates that the product fulfills the business intent.
    - Operational Readiness Test validates that the product operates properly in the intended environment.

Verify and validate requirements throughout each stage of the application development life cycle to ensure that the work products meet the requirements and fulfill the business intent. Involve stakeholders (internal and external) in the requirements development and analysis process. The more visibility they have into the requirements process, the more assurance they will have that you are properly identifying requirements.

III. Requirements Gathering

You gather high-level customer requirements during the Plan stage. But you gather and analyze the more detailed product requirements during the Analyze stage of the application development life cycle. It is during this stage that the requirements team investigates the specific goals and constraints of the project. The team gathers feedback from various stakeholders, experts, and end users to determine how the application should solve the original business problem and fulfill the high-level requirements.

It is possible that requirements may surface in later stages of the project, especially during user acceptance testing sessions. It is important to collect and analyze these requirements and evaluate them for merit in current or future releases.

Complete, accurate requirements gathering is an important first step in a successful project. Clear requirements, referenced in all stages of the application development life cycle, ensure that the product ultimately meets the needs of both the users and the stakeholders.

Requirements Gathering Process

Requirements gathering involves the following:

- *Eliciting.* Gathering input from a cross section of individuals which represent the full stakeholder community. Note that eliciting goes beyond collecting requirements. It involves proactively identifying additional requirements not explicitly provided by customers and other stakeholders.
- *Documenting.* Compiling the input collected into written form, following the guidelines for requirements writing, for future prioritization and approval.
- *Verifying.* Confirming with the stakeholders that the documented requirements accurately capture the inputs of the participants. For a complete definition of verification, see Quality Management Guidelines.
- *Analyzing.* Reviewing the requirements using operational concepts (descriptions of how the solution operates, e.g. business processes, use cases, and user scenarios) to:

    - Balance stakeholder needs and constraints
    - Ensure that the requirements are necessary and sufficient
    - Allocate the requirements to business processes/areas, product functions and product components using the Requirements Traceability Matrix

    - Determine the impact of the operating environment on meeting the documented requirements.

Also, when analyzing requirements, consider impact to scope, quality, effort, resources, timing, and risks.

- *Prioritizing.* Ordering and eliminating the gathered and documented requirements according to a set of parameters and grouping them into releases. Take into account considerations such as business case, feasibility, cost, timing, logical sequence, and other constraints.
- *Validating.* Confirming the prioritized set of requirements with stakeholders, experts, and end users to ensure this set solves the original business problem and will perform as intended in the user's environment. For a complete definition of validation, see Quality Management Guidelines.

IV. Types of Requirements

You develop and baseline high-level requirements during the Plan stage. You develop and baseline product requirements during the Analyze stage.

High-level requirements are an elaboration of stakeholder needs, expectations, constraints, and interfaces (internal and external) at a level customers and business stakeholders can easily understand. They provide a high-level description of what the product must do. They describe the major business processes, capabilities, interfaces (internal and external), functions, and business performance metrics in scope. High-level requirements provide the basis for developing the more detailed and precise set of product requirements.

Product requirements guide the design, building, and testing of the product. You develop them during the Analyze stage. They are a more detailed and precise set of requirements which elaborate on the high-level requirements created in the Plan stage. Product requirements cover the application, as well as technical architecture, training and performance support, deployment, and service/support requirements. All product requirements must trace back to the high-level requirements.

Product requirements fall into the following types:

Application

- *Functional.* These requirements specify what the user should be able to do with the application. For example, the user can compare two products from the same manufacturer. Use cases are a best practice to elicit and communicate functional requirements. Functional requirements must be grouped and allocated to product functions and product components which may include objects, services, processes, or interfaces.
- *Quality.* These requirements are sometimes called "non-functional requirements." These requirements do not focus on what the application does, but focus on:
    - Flexibility. How easy it is to understand, fix, test, maintain, enhance, and port/migrate the application. These are also called scalability requirements.
    - Performance. The ability of the application to process all the business events within a certain time frame. These requirements specify measurable objectives that describe the speed of a system. These typically also include capacity (e.g., 20,000 concurrent users may use the system) and availability (e.g., 24x7) requirements.
    - Reliability. The application's ability to function correctly under both normal and abnormal operating conditions.
    - Usability. The ease and efficiency for someone to learn, interact with, and continuously use the application. For example, the user can return to the home page from any other page on the site.
- *Interface/Data.* These requirements define the overall integration (internal and external), data, and conversion requirements. These include support for internal application integration, data synchronization, external integration with legacy systems, trading partners, and/or service providers. For each interface, capture the type of interface (synchronous, asynchronous, batch) and direction of information flow. Identify transformation and formatting requirements. Identify requirements pertaining to the translation and transformation of data between applications, such as the support for company and/or industry standard message formats and message layouts (i.e., HIPAA, CIDX, ebXML, etc.). Also define requirements for data display, validation, conversion, cleansing, retention, etc.
- *Security and Control.* These requirements may be software related (authentication, administration, access control, auditing

and logging, communication and data transmission, privacy, session handling, data security, input validation, etc.) or non-software related (building security, management procedures, system administration, certificate registration, etc.). Review the customer's security, privacy, and data classification policies and procedures as well as applicable legal and regulatory constraints. Control requirements specify the internal controls that the application must implement to prevent, identify, isolate, and correct the errors at the earliest logical point and to ensure that all business transactions submitted to the application are processed.

- *Content.* These requirements specify the type of content and any presentation restrictions. For example, comparison charts are always presented in tabular format to improve legibility. This type of requirement is most relevant in website development when building a custom application; it is less relevant to other projects such as a packaged implementation.

Technical

These requirements specify any technical requirements and constraints to be met by the application. For example, the application is accessible via IE 4+ and Netscape 4+ browsers. Furthermore, these the technical requirements focus on:

- Development Environment. Requirements for technology services, tools, techniques, and standards for designing, building, and testing system components.
- Execution Environment. Requirements for run-time technology services, control structures, and supporting infrastructure upon which system runs.
- Operations Environment. Requirements for technology services, tools, standards, and controls required to keep a business application production or development environment operating at the designed service level.

Training and Performance Support

These requirements enable users and support staff to effectively execute new processes and use new applications. Typically you create these based on the understanding of training, performance support, and communication needs.

Service Introduction

These requirements indicate hours of coverage, geographic coverage requirements, technology platform, levels of support required, operability requirements, and user demographics.

Deployment

These requirements address the sequencing, resourcing, timing, dependencies, and contingency related to the deploying the solution. You typically document deployment requirements at the business capability level.

In some cases, you can combine some of these categories; in other cases other categories may be appropriate depending on the type of application you are building. Furthermore, not all requirement types are required for a given application.

V. Who Contributes Requirements

Requirements can come from a variety of sources, ranging from end users to help-desk personnel to C-level (e.g., CFO, CEO) executives. Use all sources to which you have access when applying any of the techniques described in this document.

Existing documentation, such as brand guidelines, results from previous user testing sessions and feedback from the production application, can be useful input into the requirements gathering process. Customer relationship management systems and market research may also provide insight into the users' wants and needs.

Involve technical stakeholders and users as part of the requirements gathering and review process. These technical resources help identify constraints to functionality due to technical infeasibility and can provide input to the technical, support, and deployment requirements.

VI. Requirements Writing

Requirements are the basis for design. The project team uses them to help build accurate specifications for new systems and applications. For this reason, it is important that you write them in a clear, concise, and easily understood format to avoid misinterpretation or ambiguity.

Below are guidelines for writing high quality requirements:

- Keep sentences and paragraphs short and concise.
- Clearly define the requirement so that it is easily understood by the customer/user.
- Document individually testable requirements. Avoid long narrative paragraphs containing more than one requirement.
- Use complete sentences for each requirement (not a bullet list of buzzwords, lists of acronyms, or sound bites on a slide).
- Write each requirement so that it contains a subject and predicate where the subject is a user type or the system under discussion, and the predicate is a condition, action, or intended result.
- Confirm that the whole requirement specifies a desired end goal or result.
- Never use "and/or" or "etc." in a requirements statement. These words suggest that you may be combining several requirements into one.
- Record requirements at a consistent level of detail
- Make sure that the stated requirement does not contradict other requirements.
- Only define requirements that are necessary to the application (i.e., if the requirement were to be removed or deleted, a deficiency would exist that could not be fulfilled by other requirements).
- Verify each requirement through inspection, analysis, demonstration, or test.

VII. Requirements Analysis, Verification, and Validation

The project team performs analysis, verification, and validation on requirements throughout the system development life cycle. Analyze, verify, and validate requirements to ensure that they represent how the system being developed will satisfy the stakeholder needs and will operate properly in the intended environment.

When issues are discovered, analyze them to determine the origination point of the issue. Once you have identified the point of origin, send the issue back to the area of origination for resolution and possible process correction.

You perform requirements analysis, verification, and validation in the context of operational concepts. Operational concepts describe the conceptual operation of the application. You refine them as the team makes solution decisions and develops lower-level, detailed requirements. You continue to develop them and make them more detailed during the analysis and validation process. At the completion of the process, the project team will have well-defined operational concepts that define the interaction of the system, end users, and the environment; and that satisfy the operational, maintenance, and support needs. Review operational concepts periodically to determine their continued applicability.

Operational Concepts

You can describe operational concepts using the following:

Use Cases and User Scenarios

Use cases are made up of scenarios that satisfy customer/user goals. You use them to elicit functional requirements and explain the intended application behaviors from a customer/user point of view. You also use them for creating test conditions, prioritizing requirements, and bridging the conceptual gap between requirements and design.

User scenarios are similar to use cases. A scenario is a sequence of events that might occur when using an application. Scenarios identify the needs of the stakeholders and the end users.

Functional Prototype

You can use a functional prototype to elicit and confirm the functional requirements of the application with stakeholders. It is a partial implementation of the application built to ensure understanding of key requirements and gain confidence in the proposed solution. A prototype can be low-fidelity or high-fidelity, depending on the scope of the prototyping exercise. Another term for prototyping is conference room pilot (CRP). Typically, you throw away prototypes and do not evolve them into the final product.

Technical Proof-of-Concept

You build a technical proof-of-concept, sometimes also called a prototype, to mitigate technical risks. For high-risk technical areas of the project, it is recommended that proof-of-concept exercises be conducted to verify the ease of implementation, feasibility, extensibility, and scalability of the proposed solution.

Analysis, Verification, and Validation Techniques

Use the following techniques when analyzing, verifying, and validating requirements with stakeholders:

Interviews

Interviews are one-on-one discussions with the customer/user. You conduct interviews to learn how the current application works and what its related problems are. Interviews are most appropriate when an application already exists and analysis of the existing application is needed.

Focus Groups

Focus groups are made up of end users and/or stakeholders. A facilitator leads the group in a discussion on a specific topic related to the new or existing system or application. Participant responses are recorded and analyzed. Focus groups are most helpful during the beginning stages of a project.

Joint Application Design (JAD) Sessions

JAD sessions are similar to focus groups in that they are a form of group interviewing. You can use these sessions to obtain a better understanding of requirements from user/customer input and feedback regarding an application. The sessions are smaller than focus groups and are more focused and structured in the exchange of information.

Scenario Building/Visualization

In this technique, a facilitator asks users to visualize their current application and its related processes and then asks them to imagine their ideal application experience.

Requirements Prioritization

Requirements prioritization involves ordering the gathered and documented requirements according to a set of parameters and grouping them into releases. Take into account considerations such as business case, feasibility, cost, timing, logical sequence, and other constraints. In this process, you may eliminate some of the requirements.

Cost, schedule, and risk analyses help prioritize requirements. These analyses further identify requirements that deliver the most value for the least cost and have minimal impact on project's identified risks. Use the following aids when performing cost, schedule, and risk analyses:

Estimating Worksheet
Estimating worksheets help analyze potential cost and schedule implications related to requirements implementation.

Estimating Models and Simulations
Estimating models reduce the cost of the product and the risk in developing the product. An estimate represents a prediction regarding the human resource effort required to complete a set of tasks related to a defined set of deliverables.

Risk Management
Risk management is the recognition, assessment, and control of uncertainties that may result in schedule delays, cost overruns, performance problems, adverse environmental impacts, or other undesired consequences. You create a Risk Management Plan to capture the project's approach to risk management. The overall goal of the Risk Management Plan is to reduce the project's exposure to events that threaten the accomplishment of its objectives.

VIII. Requirements Traceability

It is important to establish and maintain the bi-directional traceability between the high-level customer requirements, the detailed product requirements, and the various analysis, design, build, and test components (e.g. use cases, designs, test conditions), product functions/components, business areas, etc., throughout all stages of the project.

Bidirectional Traceability

This primarily applies to vertical traceability and at a minimum needs to be implemented both forward and backward (i.e., from requirements to end product and from end product back to requirements).

Vertical Traceability

Vertical traceability identifies the origin of items (e.g., customer needs) and follows these same items as they travel through the hierarchy of the solution blueprint, to the tasks in the work plan, to the product components, and eventually back to the customer. When the requirements are managed well, you can establish traceability from the high-level customer requirements to the detailed product requirements and from the product requirements back to the high-

level requirements. Such bidirectional traceability helps determine that the product completely addresses all high-level requirements and that you can trace all product requirements to valid high-level requirements.

Horizontal Traceability

Horizontal traceability is also important but it is not required to satisfy bidirectional traceability. Horizontal traceability identifies the relationships among related items across work groups or product components for the purpose of avoiding potential conflicts. It enables the project to anticipate potential problems (and mitigate or solve them) before integration testing. For example, horizontal traceability would follow related requirements across two work groups working on two associated components of a product. The traceability across these two work groups enables the work groups to see when and how a change in a requirement for one of the components may affect the other component. Thus, horizontal traceability enables the project team to anticipate potential problems (and mitigate or solve them) before integration testing.