TECHNICAL SPECIFICATION

ISO/IEC TS 30103

First edition
2015-09-15

Software and Systems Engineering — Lifecycle Processes — Framework for Product Quality Achievement

Ingénierie du logiciel et des systèmes — Processus du cycle de vie — Cadre pour la réalisation de la qualité du produit

Reference number
ISO/IEC TS 30103:2015(E)

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, SC 7, *Software and Systems Engineering*.

Introduction

This Technical Specification provides guidance on the application of ISO/IEC/IEEE 15288:2008[1] life cycle processes with specific reference to addressing quality in projects that deliver systems and software products and services. It focuses on a systematic approach to achieving quality, involving the development of certain information items, the inter-relationships between these information items and the maintenance and mutual consistency management of these information items. In particular, it describes how to develop detailed specifications of the collection of process instances needed to produce a specific product or system and achieve its quality goals. It describes how the guidance in life cycle process standards may be applied in conjunction with other standards (such as the ISO/IEC 25000[12] SQuaRE series of standards that address the specification, measurement and evaluation of product quality) to achieve quality during the development of a specific system.

Application of life cycle processes to develop a system involves the production of, among others, a collection of information items such as stakeholder requirements, system requirements, designs, plans, and technical strategies, as well as a collection of artefacts including the various system elements and enabling systems. The guidance in this Technical Specification is based on the following principles:

— **Localization of quality responsibility**: Requirements should be established for each system element and enabling system, derived from the overall system requirements; the set of process instances needed to develop each system element or enabling system should be identified; their responsibilities towards quality are demarcated by the outcomes defined for the corresponding life cycle process in ISO/IEC/IEEE 15288;

— **Creation of process instance descriptions**: For each process instance, success criteria should be established based on the outcomes defined for the corresponding life cycle process, the characteristics and requirements of the particular system element, and requirements and constraints arising from product decisions made in other process instances; the set of specific tasks and associated competencies needed to achieve these success criteria should be identified, particularly for system elements with significant quality risk;

— **Consistency with institutional knowledge:** Achievement of quality ultimately depends on correct technical decisions. Relevant institutional knowledge should be systematically identified and deployed for making product and process decisions, and the resulting information items and artefacts should be checked for consistency with the applicable bodies of institutional knowledge;

— **Maintenance of content consistency**: All the information items, including the process instance specifications themselves, may evolve concurrently throughout the development life cycle. Content consistency relationships among the various information items and artefacts should be tracked and managed as these information items and artefacts evolve concurrently.

The Technical Specification is applicable to any project involving the development, enhancement or re-engineering of systems with hardware, software and human elements. It is particularly useful to project organizations that operate in multiple application domains where the set of critical quality characteristics varies widely across projects, requiring a more systematic and detailed approach to planning the achievement of quality during the development stage of the system life cycle.

This Technical Specification is intended to provide guidance for two-party situations and may be equally applied where the two parties are from the same organization. This Technical Specification can also be used by a single party as self-imposed tasks. This Technical Specification can also serve as guidance in multi-party situations, where high risks are inherent in the supply and integration of complex systems, and procurement can involve several suppliers, organizations or contracting parties.

Software and Systems Engineering — Lifecycle Processes — Framework for Product Quality Achievement

1 Scope

1.1 Application

This Technical Specification provides guidance on

- applying processes from ISO/IEC/IEEE 15288:2008[1] in conjunction with other standards to contribute to achieving quality of systems and software products and services during the development stage of the life cycle (6.4),
- the information items that should be produced through the implementation of the relevant processes (6.5), and
- the new information items (Clauses 9 and 10).

The scope of this Technical Specification is to indicate how to apply the life cycle processes in ISO/IEC/IEEE 15288:2008 to achieve quality in the context of a specific product. It is independent of any tailoring that may be made to modify the generic process descriptions to suit the needs of a particular context. Even after any applicable tailoring, there is a need to apply the resulting process guidance and determine the specific detailed activities, tasks and associated success criteria for each of the process instances needed to deliver the target system. That is the focus of the framework described in this Technical Specification.

This Technical Specification is applicable to

- those who use or plan to use ISO/IEC/IEEE 15288:2008 on projects dealing with man-made systems, software-intensive systems, software products, and services related to those systems and products, regardless of project scope, product(s), service(s), methodology, size or complexity,
- those who use or plan to use ISO/IEC/IEEE 15289[5] on projects dealing with man-made systems, software-intensive systems, software products, and services related to those systems and products, regardless of project scope, product(s), methodology, size or complexity,
- anyone performing technical processes and tasks,
- those who are responsible for the technical management of projects concerned with the development of systems,
- those responsible for performing ISO/IEC/IEEE 15288:2008 life cycle processes at a project level,
- organizations and individuals subcontracting a project management effort,
- anyone developing systems engineering management documentation to complete technical planning aspects of their project processes,
- anyone performing systems engineering activities,
- project managers responsible for staffing projects and identifying competency development needs,
- anyone developing information items during the application of project and technical processes, and
- anyone performing project and technical processes to aid in ensuring that the information items developed during these processes conform to ISO/IEC/IEEE 15289.

1.2 Audience

The guidance in this Technical Specification is intended to be used in the development and maintenance stages of the life cycle by all organizations and projects that develop or maintain systems and software products and services. It is of particular value to organizations that work in a variety of application domains, where the set of critical quality characteristics and approaches to achieve them vary widely across projects.

1.3 Limitations

The achievement of quality ultimately depends on the competent performance of technical and management tasks. While this Technical Specification provides guidance on a systematic approach to quality achievement, including identification of needed competencies, the use of the approach alone is not sufficient to guarantee achievement of quality.

This Technical Specification provides guidance on developing detailed specifications of the collection of process instances needed to develop the product or service including the artefact-specific tasks within each process instance, but it does not address sequencing and information flow issues among these tasks and processes. The area of situational method engineering[8] addresses the problem of organizing the collection of tasks into a network with defined information flow patterns.

Tradeoffs among quality attributes are intentionally not addressed. Tradeoffs are part of the enactment of requirements, design, planning and other processes. Any iteration needed to address tradeoffs is part of concurrent elaboration. Consistency relationships must hold among information items and artefacts after tradeoffs have been made. Guidance on tradeoffs is not provided to avoid over-prescription.

The approach described herein is applicable to design and realization of services and service delivery systems. However, the achievement of service quality also depends on management of quality during interactive service delivery, and this approach does not address that aspect of service quality.

2 Motivation

ISO/IEC/IEEE 15288:2008 establishes a common framework for describing the life cycle of systems created by humans. It defines a set of processes and associated terminology. These processes can be applied at a project level and guidance is provided on the outcomes, activities and tasks needed to achieve quality.

These processes are intentionally defined to be product-agnostic so that they can be widely applicable to all projects involving systems and software products and services. ISO/IEC TR 24748-1[2], ISO/IEC TR 24748-2[3] and ISO/IEC/IEEE 29148[11] provide guidance on the application of these life cycle processes to a specific system, indicating that multiple recursive instances of each life cycle process may be needed corresponding to system elements at each hierarchical level of the system, and that multiple iterative instances may be needed to deal with dependencies and changes.

When applying the guidance from these standards to develop a specific system, there are challenges that can potentially lead to gaps in quality achievement which may not be detected until late in the life cycle.

— All the specific tasks needed to develop each artefact associated with each system element and enabling system must be identified. While the list of activities and tasks in each life cycle process provides a starting point, the specific tasks to be performed depend on the requirements and characteristics of the target system element. Any omissions in identifying the right set of specific tasks will lead to defects that may not be detected until late in the life cycle.

— The product decisions made in various process instances need to be mutually consistent. The decisions in some information items and artefacts impose requirements and constraints on others. Mutual consistency relationships among deliverables must be maintained as they evolve during the concurrent enactment of the various process instances. Gaps arise if changes are not propagated consistently among artefacts.

— Quality achievement depends on the right technical decisions being taken. There is considerable institutional knowledge from standards and other sources that can guide these technical decisions, but there are often gaps in identification and application of the relevant technical knowledge.

This Technical Specification closes these potential gaps, describing in detail how the life cycle processes may be used to achieve quality for a specific system.

3 Terms, definitions and abbreviated terms

3.1
concurrent elaboration
life cycle process instances are enacted concurrently during the project, and the information items and artefacts produced by these process instances evolve concurrently

Note 1 to entry: This is referred to as concurrent elaboration of information items.

3.2
content consistency
semantic consistency among the contents of information items

3.3
information item
separately identifiable body of information that is produced, stored, and delivered for human use

[SOURCE: ISO/IEC/IEEE 15289]

3.4
institutional knowledge
knowledge from accepted sources, including standards, academic sources, domain and industry bodies of knowledge and organizational knowledge

3.5
instantiation
identification, for each instance of a life cycle process, of the success criteria, artefact-specific activities and tasks needed to achieve the process outcomes, and the competencies needed to perform these tasks, based on the characteristics and requirements of the target system element

Note 1 to entry: This is referred to as instantiation of the life cycle processes for a specific system, product or service.

3.6
locality of quality responsibility
assignment of responsibility for specific quality-related requirements or decompositions thereof to a particular process instance

3.7
process instance
single specific and identifiable execution of a process

[SOURCE: ISO/IEC 33001:2015, 3.2.17]

Note 1 to entry: It is the result of instantiation of life cycle processes for a specific system, product or service.

3.8
quality responsibility
responsibility for achievement of a quality requirement or decomposition thereof

3.9
success criteria
set of conditions to be satisfied by a process instance at completion

Note 1 to entry: Information items and artefacts produced by the process instance must meet the success criteria. Success criteria are established based on the outcomes of the corresponding life cycle process, requirements of the system element to which the process instance contributes, and requirements and constraints arising from decisions in other process instances.

4 Quality achievement concepts

4.1 Overview of quality achievement

Figure 1 shows an overview of the relationships between the process and product decisions related to quality achievement in a project. Product decisions for a project include, among others, the system requirements, design, realization technologies, as well as verification, integration, validation, transition and measurement strategies. Process decisions include identification of the specific tasks needed to achieve quality for the specific system, as well as the practices, tools and technical management approach. The process decisions are influenced by previous product decisions, and in turn generate further product decisions.

The figure indicates the life cycle process standards that provide guidance on process decisions, and some of the standards that provide guidance on product decisions. There are also other standards that provide guidance on product and process decisions related to particular quality characteristics or application domains, such as security, safety and automotive standards.

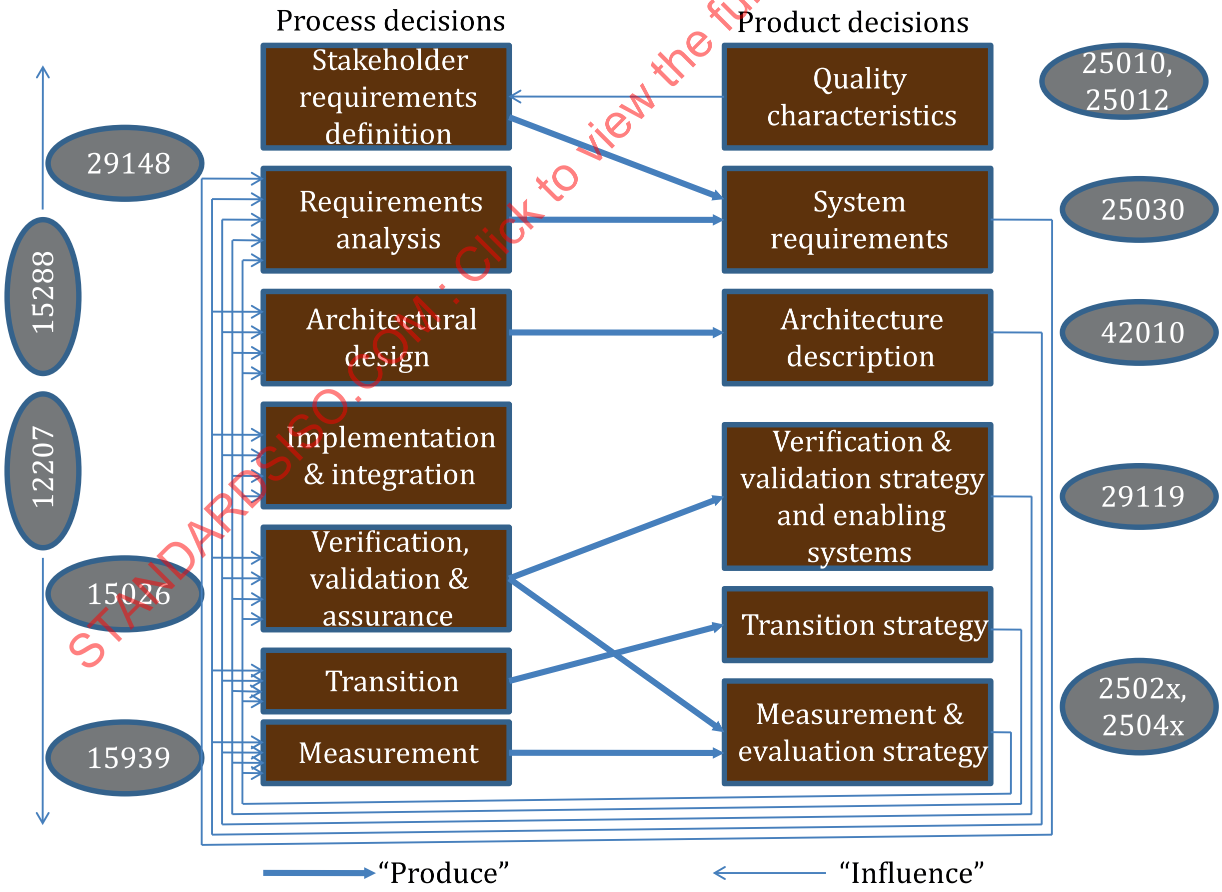

Figure 1 — Overview of quality achievement

It should be noted that the lines in the above diagram represent informational and generative relationships, not control flows. Life cycle processes are typically enacted concurrently and iteratively, so that the various product-related decisions and the related process decisions evolve continuously throughout the project. In general, it cannot be assumed either that the process decisions are all frozen prior to enactment of that process, or that information items and artefacts are produced sequentially. In complex projects, all of these evolve continuously and concurrently, and the consistency relationships among them must be managed.

4.2 Guiding principles and approach

The approach to quality achievement described herein incorporates four principles.

— **Localization of quality responsibility**: Quality requirements should be established for each system element and enabling system (herein referred to generically as artefacts), based on the overall system requirements and the architectural design; the responsibility for meeting these requirements should be further distributed among the collection of process instances that together produce the system element, in accordance with the outcomes defined for the corresponding life cycle processes.

— **Creation of process instance descriptions**: For each process instance, its responsibilities towards meeting the requirements of the system element should be captured in the form of success criteria. The detailed activities and tasks needed to achieve the success criteria, and the competencies to perform them should be identified, especially for system elements with significant quality risk.

— **Consistency with institutional knowledge**: Achievement of quality ultimately depends on correct technical decisions. Relevant institutional knowledge should be systematically identified and deployed for making product and process decisions, and the resulting information items and artefacts should be checked for consistency with the applicable bodies of institutional knowledge.

— **Maintenance of content consistency**: Content consistency relationships among the information items and artefacts (i.e. mutual consistency of product decisions) should be tracked and managed as they are concurrently elaborated during project enactment.

The principles establish each process instance as a locality of quality responsibility with defined success criteria that reflect quality requirements, along with the complementary concepts of ensuring consistency of decisions across localities and systematic application of institutional knowledge to process and product decisions. The creation of detailed descriptions of process instances is referred to as *instantiation* of the life cycle processes to the particular system element.

The resulting approach is shown in Figure 2.

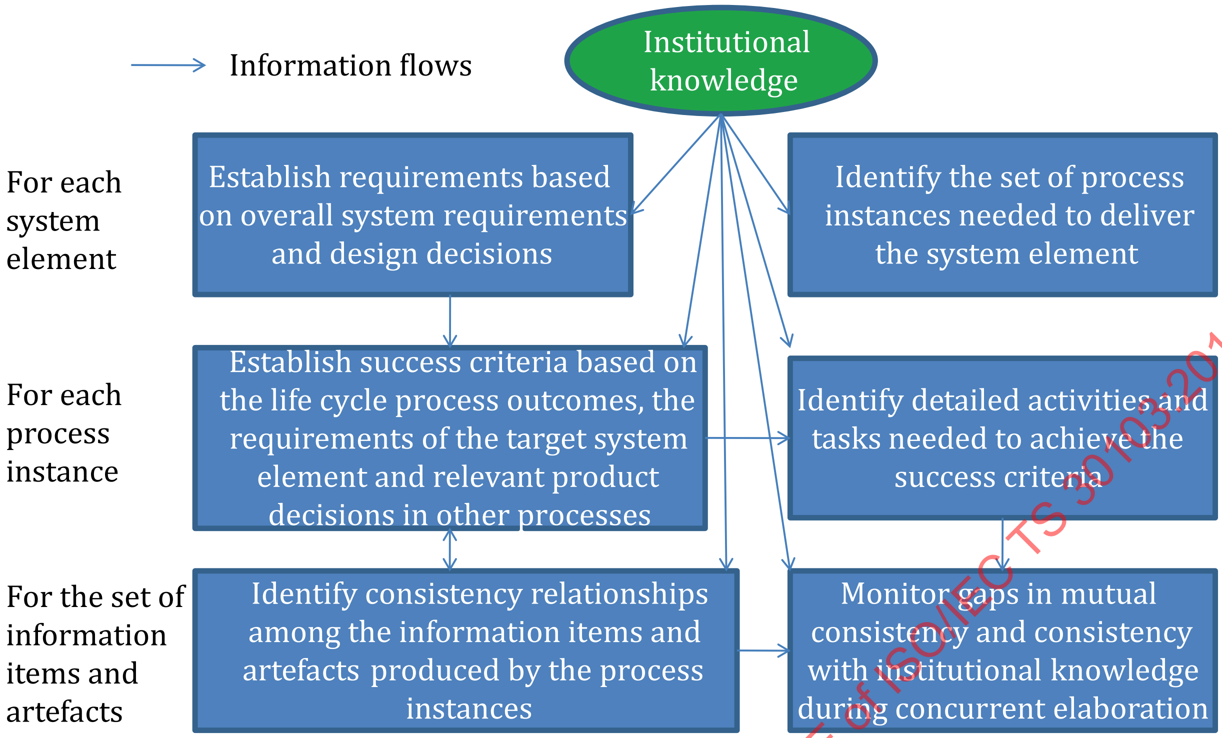

Figure 2 — Approach to quality achievement

The application of these principles should be tuned to the complexity of the system, development context, criticality of quality characteristics and identified technical risks to achieve the best balance between the benefits conferred by the systematic approach and the effort required to implement the approach. There are several dimensions of flexibility in the application of the approach, for each of which choices need to be made based on identified risks and other contextual factors:

— the granularity of system elements to which the approach is applied;

— whether to drill all system requirements down to the lowest level of the system hierarchy, or only those requirements that are critical to the particular system element;

— the set of process instances selected for detailed instantiation, and the level of detail needed in the description, based on identified risks;

— the set of content consistency relationships to be tracked and managed.

The theoretical foundations underlying these principles are discussed in the informational Annex E. This approach is similar to the concepts of Quality by Design[9] and aligned with the "Right First Time" concept[28].

4.3 Localization of quality responsibility

4.3.1 Establishing system element requirements

The system requirements and architecture description identify the collection of system elements that need to be developed, typically by decomposing the system-of-interest hierarchically into lower level system elements.

NOTE 1 These system elements may be modules, components, services, features or updates and changes to existing system elements (in the case of incremental development or re-engineering).

Recursive instances of the Requirements Analysis and Architectural Design processes decompose the overall system requirements to establish the requirements for each lower level system element. Requirements relating to quality characteristics should be hierarchically decomposed and allocated. Depending on the nature of the quality characteristic and the decomposition, a requirement may be directly assigned to a single system element, assigned to each of several system elements, or decomposed and distributed among multiple system elements.

NOTE 2 In incremental development, the allocated requirements may reflect the changes in characteristics needed for each system element, rather than the overall characteristics of the element.

The granularity to which requirements are drilled down depends on the tradeoffs between the cost of establishing requirements for lower level elements and the benefits of more granular quality management. This tradeoff may be managed at the level of individual requirements, so that more critical requirements are drilled down deeper in the system hierarchy.

In this Technical Specification, the approach to quality achievement is illustrated with the example of adding barcode scanning capabilities to a hospital Blood Bag Inventory System. Figure 3 shows an example hierarchical decomposition and some requirements for each system element.

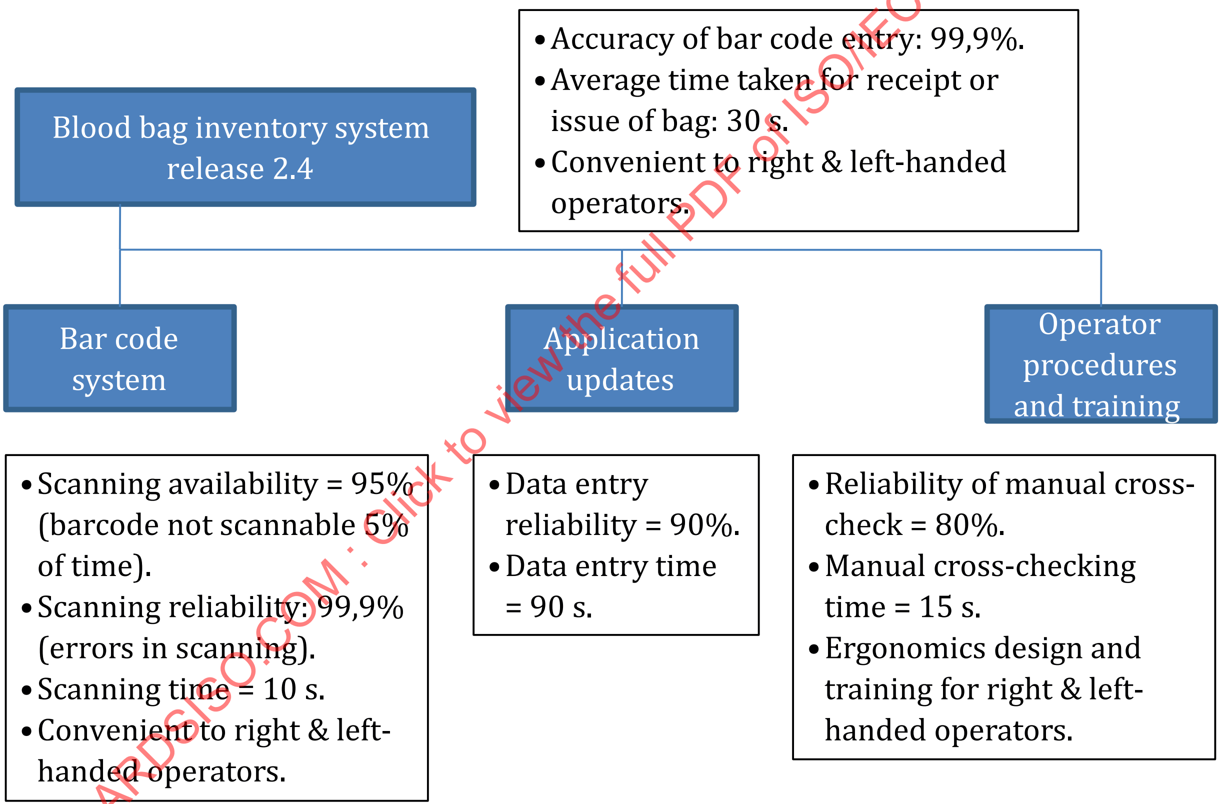

Figure 3 — Example (partial) system element requirements

Annex A describes the example problem in more detail, and indicates the approach for determining the requirements for each system element from the overall system requirements.

In addition to system elements, the development of the system-of-interest may also need enabling systems e.g. testbeds, deployment setup, swing hardware for transition etc. Requirements must also be established for each enabling system by application of the Architectural Design and Requirements processes in conjunction with the relevant Technical Process, i.e. Integration, Verification, Validation or Transition process.

4.3.2 Identification of process instances

For each system element or enabling system (referred to generically as artefacts herein), the collection of life cycle process instances needed to develop the artefact should be identified. This is based on the type of the artefact (e.g. hardware, software, service), and the realization approach (e.g. development, acquisition, adaptation of existing system elements or assets, integration from lower level system elements). For example, a system element to be acquired needs a recursive instance of the acquisition process, while system elements to be developed need instances of the Architectural Design and Implementation processes. Note that for software elements, ISO/IEC/IEEE 12207:2008 may be used to identify the collection of recursive process instances.

NOTE ISO/IEC TR 24748-1 also includes the concept of iterative process instances, wherein a process is applied iteratively to refine the information items and artefacts produced. For the purposes of this Technical Specification, iterative instances are subsumed by the concept of *concurrent elaboration*, according to which all process instances are assumed (in principle) to operate possibly concurrently throughout the project, resulting in continuous evolution of the set of information items and artefacts produced.

Annex B identifies a set of recursive process instances for the example problem.

ISO/IEC/IEEE 15288:2008 identifies the outcomes for each life cycle process. The responsibilities for each process towards quality can be inferred from these outcomes. The quality responsibilities of each process instance may be established by identifying success criteria for the process instance in accordance with the outcomes.

4.4 Creation of process instance descriptions

4.4.1 Establishment of success criteria

The success criteria for a process instance include the following:

— The outcomes for the life cycle process, applied to the requirements and critical quality characteristics of the particular artefact. Outcomes that are independent of the target artefact characteristics are carried forward without change;

— Additional “checklist” criteria or practices based on institutional knowledge, reflecting considerations specific to the particular artefact type;

EXAMPLE The success criteria for the Bar Code System Requirements Analysis process instance may include a checklist item to establish requirements on the material for the barcode sticker.

— Requirements and constraints based on decisions in other process instances.

EXAMPLE Applicable architecture conformance constraints and support for verification may be added as success criteria for Implementation Process instances.

For example, the Requirements Analysis process identifies the following outcomes[11]:

a) The required characteristics, attributes, and functional and performance requirements for a product solution are specified;
b) Constraints that will affect the architectural design of a system and the means to realize it are specified;
c) The integrity and traceability of system requirements to stakeholder requirements is achieved;
d) A basis for verifying that the system requirements are satisfied is defined.

For the Bar Code System in the example problem, it is known from institutional knowledge and the system requirements that availability, reliability, scanning accuracy and operator ergonomics are important quality characteristics. This leads to establishment of the following success criteria for the Bar Code System Requirements Analysis process:

a1) the target availability, reliability and accuracy for the barcode system are specified;

a2) scanner characteristics needed to support operator ergonomics are specified;

b1) constraints on integration of the barcode scanner with the inventory system are specified;

b2) constraints on the barcode system arising from the physical infrastructure are specified;

b3) fault modes that may affect the functioning of the barcode system are identified;

b4) constraints on the material for barcode stickers are identified;

c) the integrity and traceability of barcode system requirements to barcode stakeholder requirements and/or overall system requirements is achieved;

d) procedures for verifying the barcode system availability, reliability and accuracy, and that it meets operator ergonomics requirements, are defined;

e) scanning accuracy requirements are established for each deployment scenario: manual entry, initial operational phase, stabilized operations.

These success criteria are derived based on institutional knowledge about the types of constraints relevant to the system element (in this case, a physical device such as a barcode scanner) and about the requirements analysis outcomes relevant to a particular quality attribute (in this case, that reliability and availability requirements need an accompanying fault modes identification). The establishment of success criteria provides an opportunity to use knowledge about the domain and characteristics of the system element to pull in the relevant institutional knowledge and establish detailed quality responsibilities for each process instance.

It should be noted that outcome c) does not lead to the identification of new success criteria because it is largely independent of the specific artefact. Success criteria supplement the life cycle process outcomes with more detailed identification of quality responsibilities, where applicable.

The success criteria may be augmented based on product decisions made in other process instances, such as requirements for supporting features, technology and strategy decisions, measurement choices, risk mitigation strategies and conformance constraints. Success criterion e) is derived based on the establishment in the Transition process of three deployment scenarios: a manual operations scenario where operators enter barcodes manually based on the stickers, an initial operational phase during which operators are still gaining familiarity with the system and a stabilized operations scenario where they are fully familiar with the proper use of the system.

Success criteria capture internal and external quality requirements that relate to the behaviour of the system and its elements[13]. They can also be used to capture quality-in-use requirements that relate to emergent properties of the interaction between the system and its environment, including users. Monitoring quality-in-use success criteria requires validation activities and is part of monitoring the consistency relationship between the system requirements and stakeholder needs.

4.4.2 Identification of detailed activities and tasks

Life cycle processes identify generic activities and tasks needed to achieve the outcomes. Using this guidance, projects should identify the specific tasks needed to achieve the success criteria for each particular process instance. For example, the Requirements Process instance for the Bar Code System uses the task guidance provided in the ISO/IEC/IEEE 15288, Requirements Process:

Define necessary implementation constraints that are introduced by stakeholder requirements or are unavoidable solution limitations.

To identify the following specific task within the process instance:

Identify API support needed for verification of scanning correctness.

The detailed activities and tasks needed to achieve the success criteria for each process instance are identified based on the following:

— Activities and tasks in the life cycle process definition;

— Institutional knowledge about practices relevant to the particular application domain and target quality characteristics. Product decisions about technologies, methods, architectural and design patterns etc. may have associated guidance relating to specific activities and tasks that need to be performed, and best practices to achieve quality.

The competencies needed to perform the activities and tasks should also be identified based on institutional knowledge.

The detailed activities and tasks bind the activities and tasks of the life cycle processes to the specific practices and technologies selected for the particular project. The completeness and correctness of the list of detailed activities and tasks should be validated against the success criteria for the process instance. The activities and tasks for each process instance are combined to generate the list of technical tasks for the overall project plan.

4.4.3 Process instance descriptions

Figure 4 shows the schema for the creation of process instance descriptions: establishing its success criteria and identifying the detailed activities and tasks needed to achieve the success criteria. It should be noted that these descriptions also undergo concurrent elaboration, i.e. the success criteria and list of detailed activities and tasks for each process instance may themselves evolve throughout the project.

The creation of process instance descriptions involves considerable effort and is only performed in situations where the perceived quality risk is sufficient to warrant this effort, possibly only for a subset of system elements and process instances. It is a high-maturity practice that enables prevention and early detection of quality gaps, by creating localities of quality responsibility and visibility into the schema for achieving quality.

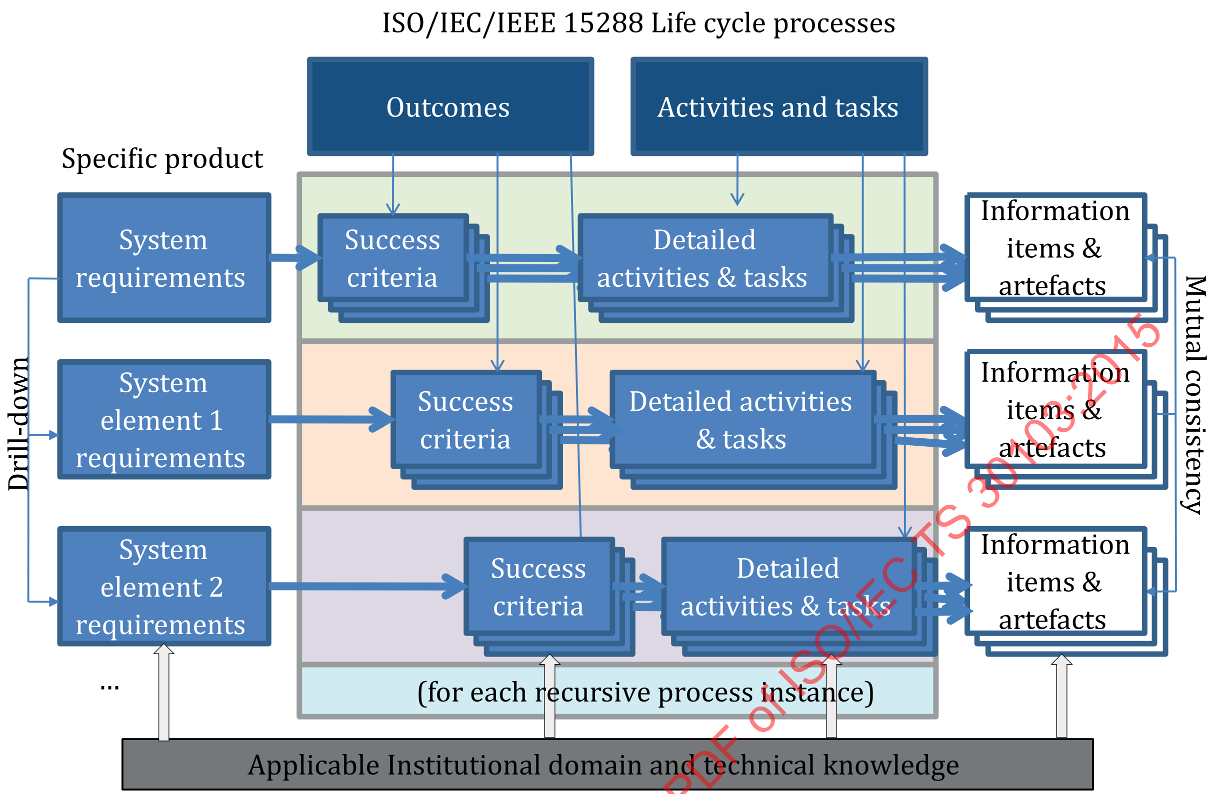

Figure 4 — Creation of process instance descriptions

Process instance descriptions enable in-process quality checks within each process and across the outputs of processes.

a) Verification activities are performed to check that the set of tasks and activities identified for each process instance are sufficient to achieve the success criteria.

b) Verification activities are performed to check that the outputs from each process instance meet the success criteria.

c) Verification activities are performed to ensure that the outputs from process instances are mutually consistent and also consistent with applicable institutional knowledge.

d) Monitoring gaps in the network of consistency relationships among the outputs of process instances provides visibility into product quality maturity for management purposes.

The creation of process instance descriptions could be included as an additional task within the Project Planning Process of ISO/IEC/IEEE 15288. 7.1 shows this potential augmentation to the Project Planning Process. Process Instance Descriptions constitute a new information item, as identified in 9.1.

Annex B shows an example of instantiation, including the identification of detailed activities and tasks. Annex D shows a process view to address a particular functional requirement (accuracy) as an exemplar of the complete end-to-end approach, covering multiple processes and system elements at different levels of the hierarchy, as well as showing the application of institutional knowledge in identifying instantiated outcomes and tasks.

4.5 Consistency with institutional knowledge

The correctness of product and process decisions can be improved through systematic deployment of relevant institutional knowledge. Institutional knowledge often takes the form of a collection of life cycle practices associated with an integrity level, quality characteristic or technology. Instantiation enables this knowledge to be systematically injected into projects.

Institutional knowledge includes

— standards,

— authoritative bodies of knowledge such as SWEBOK (software engineering body of knowledge) and SEBOK (system engineering body of knowledge),

— trusted sources, e.g. books, technical papers, product manuals,

— organizational knowledge, including assets and learnings from previous projects, and

— knowledge of experts.

It may be captured in various documents and repositories that facilitate identification and retrieval of relevant knowledge. Verification activities should include checking that process and product decisions are consistent with relevant institutional knowledge.

The concept of Organization and Application Normative Frameworks in ISO/IEC 27034-1:2011[6] provides an example of how institutional knowledge can be structured to facilitate systematic deployment into projects. At the organizational level, knowledge is organized relative to a set of quality concerns, and the collection of process and product practices required to address each concern to a defined stringency level are populated into the Organization Normative framework. Projects select applicable practices based on their quality concerns and associated stringency levels and populate them into the Application Normative Framework.

Another approach to systematic deployment of applicable institutional knowledge is the development and application of a standardized profile[31]. Relevant sections from multiple applicable standards applicable to a particular class of projects may be extracted and synthesized into a profile agreed to by the various stakeholders.

4.6 Maintenance of content consistency

While configuration management standards such as IEEE P828[7] provide for versioning of configuration items, ensuring mutual consistency of the product decisions contained within information items and artefacts requires verification activities.

The set of content consistency relationships among information items and artefacts is identified based on institutional knowledge. For example, ISO/IEC/IEEE 15288:2008 identifies several traceability requirements among information items and artefacts (system requirements, architectural design, implemented system). In addition, there are also content consistency relationships between

— system elements and the strategies and enabling systems used for their integration, verification, validation and transition,

— system elements and the descriptions of the process instances that develop them, and

— system requirements, architectural design and the measurement strategy.

A candidate set of content consistency relationships among information items and artefacts is shown in Annex F. The set of applicable consistency relationships for each information item can be identified based on such knowledge and a subset of them chosen for verification and monitoring based on perceived technical risk.

The Verification and Validation processes should monitor consistency relationships and identify gaps that need to be closed, as part of verifying the quality of any information item or artefact. This includes verifying consistency with applicable institutional knowledge, that success criteria are met, and mutual consistency relationships. These criteria are added to verification enablers such as review checklists.

Typically, verification activities are triggered whenever a process instance generates a new version of information items or artefacts.

The desire to maintain perfect consistency among information items and artefacts should be balanced against other concerns, including timeliness of delivery and the effort needed to make repeated changes. A new information item called the Consistency Tracker (an extension of traceability matrices) is proposed to facilitate the tracking and management of consistency gaps. It identifies the source and target items (information items or artefacts), expected consistency relationships and known gaps (if any).

Gaps in content consistency identified by verification and validation activities are entered into the Consistency Tracker and tracked to closure. Consistency tracking may be merged with defect tracking. The entries in the Consistency Tracker include not only defects but also intentional gaps arising from technical management decisions, e.g. incremental coverage of requirements, partial implemention of design, limited test coverage, etc.

The Consistency Tracker provides management visibility to gaps in content consistency and the status of progress towards quality. Quality is achieved when there is sufficient content consistency across all the information items from stakeholder needs to the operational system, as indicated by verification and validation activities.

Guidance on the Consistency Tracker is provided in Clauses 9 and 10. Annex C shows an example of the use of the Consistency Tracker.

Content consistency is an extension of traceability. While traceability identifies the informational relationships that drive a decision, with a view to making sure that there are no “orphan decisions” either with no driving rationale in the needs or with no footprint in the implementation, maintenance of content consistency goes further and continually monitors the informational relationships during concurrent elaboration. It also extends the scope of traceability, from traceability only among product-related artefacts and information items, to also include mutual consistency with information items relating to the processes, product and enabling systems.

5 Required background concepts

5.1 System and Software Concepts

This Technical Specification presupposes an understanding of system and software concepts.

NOTE 1 System concepts are introduced in ISO/IEC/IEEE 15288:2008, 5.1. Additional discussion, such as systems and system structure, is provided in ISO/IEC TR 24748-1:2010, 3.1.

NOTE 2 ISO/IEC TR 24748-2 provides more information on concepts related to system life cycle management.

NOTE 3 ISO/IEC 25000 SQuaRE series of International Standards provide more detail on product quality concepts.

5.2 Life cycle concepts

This Technical Specification presupposes an understanding of life cycle concepts.

NOTE Life cycle concepts are introduced in ISO/IEC/IEEE 15288:2008, 5.2. Additional discussion is in ISO/IEC TR 24748-1:2010, 3.2.

In accordance with ISO/IEC/IEEE 15288:2008, this Technical Specification assumes that life cycle processes may be performed concurrently during the life cycle. The information items and artefacts they produce may evolve continually throughout the life cycle. This is referred to as *concurrent elaboration* of information items and artefacts. For example, design decisions result in identification of the detailed tasks needed to further elaborate the design.

ISO/IEC TR 24748-1:2010, 3.4 and ISO/IEC/IEEE 29148:2011 describe iterative and recursive application of life cycle processes to address progressive refinement and different levels of detail in a system hierarchical structure respectively. This Technical Specification treats each recursive application as a process instance with associated success criteria and detailed tasks. Iterative application is not considered separately in this Technical Specification because it is covered by the concept of concurrent elaboration.

5.3 Process concepts

This Technical Specification presupposes an understanding of process concepts.

NOTE Process concepts are discussed in ISO/IEC/IEEE 15288:2008, 5.3 and ISO/IEC TR 24748-1:2010, 3.3.

5.4 Organizational concepts

This Technical Specification presupposes an understanding of organizational concepts.

NOTE Organizational concepts are discussed in ISO/IEC TR 24748-3:2011, 4.5.

This Technical Specification is particularly useful for organizations that take on projects in multiple application domains, where the set of relevant quality characteristics and the approaches to achieve them vary widely across projects.

5.5 Information Item Concepts

This Technical Specification presupposes an understanding of information item concepts.

NOTE ISO/IEC 15289 provides more detail on information items and specifies how life cycle data is managed in information items.

5.6 Notion of technical management

In the scope of this Technical Specification, technical management is responsible for decisions relating to quality achievement, including definition and tailoring of life cycle processes, and choice of the methods, technologies and tools for development. Project management is responsible for resources, schedules, stakeholders' management, risk management and achievement of stakeholder concerns other than technical concerns, such as profitability and market alignment.

Technical management responsibilities include the identification and performance of the right activities needed to develop the system, ensuring the correctness of each technical information item and artefact, and ensuring mutual consistency among information items and artefacts during concurrent elaboration. This Technical Specification primarily addresses these aspects.

6 Context of application

6.1 Relationship to other standards

This Technical Specification is a part of the ISO/IEC SC 7 set of life cycle standards and guidance documents on Systems Engineering. It depends on several ISO/IEC standards and technical reports, and is expected to be used in conjunction with others, as indicated in Table 1.

Table 1 — Related ISO Standards

Document	Relationship
ISO/IEC/IEEE 15288:2008	Defines the system engineering life cycle processes to which this approach applies.
ISO/IEC TR 24748-1:2010, ISO/IEC TR 24748-2:2011	Provide guidance on the application of the life cycle processes, including introducing the concept of recursive and iterative instances.

Table 1 *(continued)*

Document	Relationship
ISO/IEC 24748-4 (draft)	The System Engineering Management Plan defines the collection of practices used to manage the entire system engineering effort, including the approach to quality achievement. The approach described in this Technical Specification is a potential candidate for inclusion in the System Engineering Management Plan.
ISO/IEC/IEEE 12207:2008	The approach described here can potentially also be applied to the Software Engineering life cycle processes.
ISO/IEC/IEEE 29148:2011	Provides guidance on requirements engineering, and further elucidates the concept of recursive and iterative instances. This is useful foundational material to understand the approach and concepts in this Technical Specification.
ISO/IEC/IEEE 15289:2011	Identifies the information items produced by projects.
ISO/IEC/IEEE 15026:2010	Approach to assurance. The material in this Technical Specification can serve as additional inputs to quality-related assurance activities, since it provides more visibility into the scheme for quality achievement at the level of each system element and recursive process instance.
ISO/IEC 25000:2014	The SQuaRE series of standards (ISO/IEC 250xx) relate to product quality specification, measurement and evaluation. The approach described here presupposes the use of the SQuaRE series standards to capture product quality requirements, measure and evaluate their achievement.
ISO/IEC 25010:2011, ISO/IEC 25012:2008	ISO/IEC 25010 defines the quality model used for defining product quality requirements, while ISO/IEC 25012 defines the model for data quality. This approach is intended to be used in conjunction with these quality characteristic models.
ISO/IEC 25020:2007, ISO/IEC 25021:2012, ISO/IEC 25022:—, ISO/IEC 25023:—, ISO/IEC 25024:—	These documents provide guidance on the measurement of quality characteristics, including data quality and quality-in-use. This guidance may be used to design indicators of whether the success criteria for a given process instance are being met.
ISO/IEC 25030:2007	Guidance on the establishment and specification of product quality requirements. This must be applied not only for the system as a whole, but hierarchically to establish requirements for each system element and enabling system.
ISO/IEC 25040:2011, ISO/IEC 25041:2012	ISO/IEC 25040 establishes the process for the evaluation of product quality. ISO/IEC 25041 provides an evaluation guide for developers, acquirers and independent evaluators.
ISO/IEC 29119 (draft)	The Testing standard identifies the approach and suite of practices for verification and validation activities. This guidance can be used to select the mechanisms to verify whether success criteria are being met and to check for consistency relationships.
ISO/IEC/IEEE 42010:2011	The Architecture Description standard can be used to express how the design in intended to satisfy quality requirements. The design requirements and conformance constraints arising from the architectural design should be captured as success criteria for process instances.
ISO/IEC 15504:2004	The approach described here can facilitate organizational capability assessment, since it systematizes the application of institutional knowledge to both process and product decisions.
ISO/IEC 27034:2011	This information security standard introduces the concept of organization and application normative frameworks that illustrate how institutional knowledge can be organized for systematic deployment into projects.
ISO/IEC/IEEE 15939:2007	This standard defines the process that can be used for measurement of quality and other project attributes.

NOTE ISO/IEC 15504-5:2012 and ISO/IEC 15504-6:2013, which provide process definitions from ISO/IEC 12207 and ISO/IEC 15288 respectively including a set of process performance indicators called base practices for each process, can be helpful for the systematic generation of the mapping.

6.2 Organizational context

The guidance in this Technical Specification is applicable to any organization engaging in systems engineering activities. However, it is of particular value to project organizations where there are wide variations across projects in scope of engineering responsibility, in the type of systems and applications domains involved, in the scale and complexity of projects, in the quality characteristics of interest and their relative importance, and consequently the knowledge, competencies and practices required to achieve quality in each project. In product organizations where there is considerable similarity across projects, the mapping from life cycle process guidance to specific practices and activities is relatively stable, enabling the development of competencies and organizational capability to develop this mapping with repeatable accuracy. In project organizations where there are vast differences across projects, generating the mapping from organizational life process guidance or even tailored domain-specific life cycle processes to the particular practices, activities and tasks needed in the specific context is much more challenging because of the extreme case-by-case variations. The guidance provided in this Technical Specification on how to generate this mapping systematically can be particularly valuable for such organizations.

6.3 Stakeholder context

The primary stakeholders for this Technical Specification are project teams that need to perform life cycle processes. They are provided guidance on identifying the collection of recursive process instances needed, establishment of success criteria for each process instance, and working out the detailed set of activities and tasks needed to achieve the success criteria.

Technical Management professionals are provided guidance on managing and tracking consistency relationships among information items and artefacts during concurrent elaboration.

Customers and other stakeholders can obtain greater visibility into the detailed technical approach for achieving quality.

6.4 Stage context

The guidance provided in this Technical Specification applies primarily to the development stage of the life cycle. It also applies to incremental development, corrective maintenance and re-engineering efforts that may be needed during the maintenance stage.

6.5 Process context

The guidance in this Technical Specification covers identification and description of recursive process instances for some or all of the following technical processes in ISO/IEC/IEEE 15288:2008, related to the development stage of the system life cycle:

— Stakeholder Requirements Definition Process;

— Requirements Analysis Process;

— Architectural Design Process;

— Implementation Process;

— Integration Process;

— Verification Process;

— Transition Process;

— Validation Process.

It proposes the inclusion of one additional task in the Project Planning Process: creation of process instance descriptions (see 7.1). It also proposes that the Project Assessment Control Process should maintain an additional information item, the Consistency Tracker (see 7.2).

6.6 Information item context

This Technical Specification provides guidance on maintaining content consistency among the various information items and artefacts produced by the life cycle processes. It introduces two new information items:

— Process Instance Descriptions, to be produced by the Project Planning process (see 9.1);

— Consistency Tracker, an extension of traceability matrices, to be maintained by the Project Assessment and Control Process (see 9.2).

7 Potential process augmentations

7.1 Project planning process

Creation of process instance descriptions could be performed as part of the Project Planning process. The Project Planning process as defined in ISO/IEC/IEEE 15288:2008 includes the following:

4) Establish a work breakdown structure based on the evolving system architecture. NOTE Each element of the system architecture, and appropriate processes and activities are described with a level of detail that is consistent with the identified risks. Related tasks in the work breakdown structure are grouped into project tasks according to organizational responsibilities. Project tasks identify every work item being developed or produced and its associated tasks.

A fifth task could be added to this activity:

5) For each system element and enabling system, identify the collection of process instances needed to produce it, and create detailed process instance descriptions if the perceived quality risk is high enough to justify the creation of such descriptions.

7.2 Project assessment and control process

The Project Assessment and Control Process in ISO/IEC/IEEE 15288:2008 includes the following task:

2) Perform quality assurance in accordance with project plans.

This task could be modified to the following:

2) Perform quality assurance in accordance with project plans. Monitor gaps in content consistency among information items and artefacts using the Consistency Tracker.”

8 Guidelines for process augmentations

8.1 Project planning process

Creating process instance descriptions for an artefact involves the following:

— Identifying the set of applicable life process instances for the artefact. This depends on the realization strategy for the artefact (e.g. acquisition, design and build, reuse and/or adaptation of existing assets, modification of existing artefacts, integration from other artefacts). These decisions typically result from the performance of either the Project Planning process (for the artefact or its parent) or the Architectural Design process;

— Establishing the success criteria for each applicable life cycle process based on the requirements of the target artefact, defined outcomes for the life cycle process and relevant product decisions made in other processes;

— Identifying the detailed tasks needed to achieve the success criteria based on institutional knowledge relevant to the technologies and practices selected;

— Identifying the domain, technical and technological competencies required to perform the tasks.

Identification of the success criteria for each process instance, the detailed tasks needed to meet them and verifying that success criteria are met is a high maturity practice that establishes localities of quality responsibility and reduces quality risk.

Annex B shows an example of creating a process instance description. Annex D shows an example of a process view for meeting a particular system requirement, constructed by extracting relevant entries from the process instance descriptions.

8.2 Project assessment and control process

One aspect of project assessment and control is that of verifying, tracking and maintaining consistency among the information items produced. This includes the following:

— Identifying the set of mutual content consistency relationships to be monitored, for each information item and artefact. The selected set of consistency relationships is captured in the Consistency Tracker;

— Determining the mechanisms needed to verify the consistency relationships. This is part of the Verification Strategy;

— Recording technical management decisions that affect consistency maintenance in the Consistency Tracker. This includes decisions on incremental coverage of requirements, incremental implementation of design, test coverage decisions, deferred defect fixes etc.;

— Capturing the results of verification activities in the Consistency Tracker as defects or gaps to be closed;

— Providing visibility to management on the progress towards quality, in terms of the status of consistency relationships between information items: gaps resulting from management decisions, gaps that need to be closed immediately, and deferred gaps.

9 Potential information item augmentations

9.1 Process Instance Descriptions

Process Instance Descriptions could be produced by the Project Planning process. Each Process Instance Description could include the following content:

— a list of success criteria;

— detailed activities and tasks to achieve the success criteria;

— competencies needed for performing the tasks.

9.2 Consistency tracker

This information item could be produced by the Project Assessment and Control Process. Each entry could include the following:

— the information item or artefact, or the specific subpart thereof, to which the entry relates;

— the target information item or artefacts with which consistency is expected, if applicable. This field is empty for defects and gaps that are internal to a single item;

— the type of consistency relationship expected between the two items;

— description of the gap/defect.

— status:

  — *TBD*: The consistency relationship is yet to be established;

  — *OK*: No gaps;

  — *Pending*: Re-evaluation of the consistency relationship is pending, following an update to the current or target information item;

  — *Intentional*: Gap resulting from an explicit management decision;

  — *Deferred:* Known gaps, not to be closed immediately;

  — *To be closed*.

10 Guidelines for information item augmentations

10.1 Process instance descriptions

Process Instance Descriptions need to be created only for those system elements and process instances with significant quality risks. The set of success criteria can also be limited to only those aspects where identified risks warrant the effort to continually monitor those success criteria.

10.2 Consistency tracker

The Consistency Tracker is related to traceability matrices. Its scope extends beyond traceability among product artefacts to enabling systems and process artefacts. It is populated based on technical management decisions relating to mutual consistency, and verification activities which check whether information items and artefacts meet identified success criteria, are mutually consistent, and consistent with applicable institutional knowledge.

The information contained in the Tracker can be periodically processed and provided as reports on the status of progress towards quality. Achievement of an acceptable level of quality requires the closure of all gaps in the network of consistency relationships from stakeholder requirements to the operational system, except those resulting from technical management decisions and choices.

Annex A
(informative)

Example: Establishing System Element Requirements

This is an example to illustrate the concepts in this Technical Specification of a project to enhance the inventory system for managing blood bags in a hospital. The current system places handwritten stickers on each blood bag that includes an ID number, as well as some provenance information. The ID number serves as an index into a database. Currently acquisition and issue of blood bags involves manual entry of information into the database. The goal of the project is to replace the handwritten stickers with ones that include a scannable barcode. The scope of the project includes the acquisition of scanner devices, their integration into the software system, definition of revised operational procedures for issue and acquisition, and providing training services to the staff to use the new system. It also includes fixing some minor problems reported with the existing solution.

This Annex shows an example of how quality requirements are established by drilldown from system requirements. The salient points are the system decomposition for this incremental development project, and the table showing the decomposed quality requirements.

Annex B shows how these quality requirements are used in the creation of process instance descriptions, specifically the Requirements Analysis process for the system level. This includes the identification of success criteria and the identification of detailed tasks and competencies.

Annex C shows the usage of the Consistency Tracker.

Annex D shows a process view that depicts the approach to addressing the accuracy requirement. The emphasis here is on showing the end-to-end approach to product-specific instantiation of processes, including both multiple processes and multiple process instances. This shows relationships between processes (e.g. architecture is responsible for verification interfaces). The use of institutional knowledge in identifying tasks and outcomes is also highlighted.

Figure A.1 shows the system decomposition for the project (an incremental release of the blood bag inventory management system that adds the handheld scanner and related functionality). This is created from the development viewpoint, i.e. it decomposes the product into system elements that each represent capabilities to be delivered.

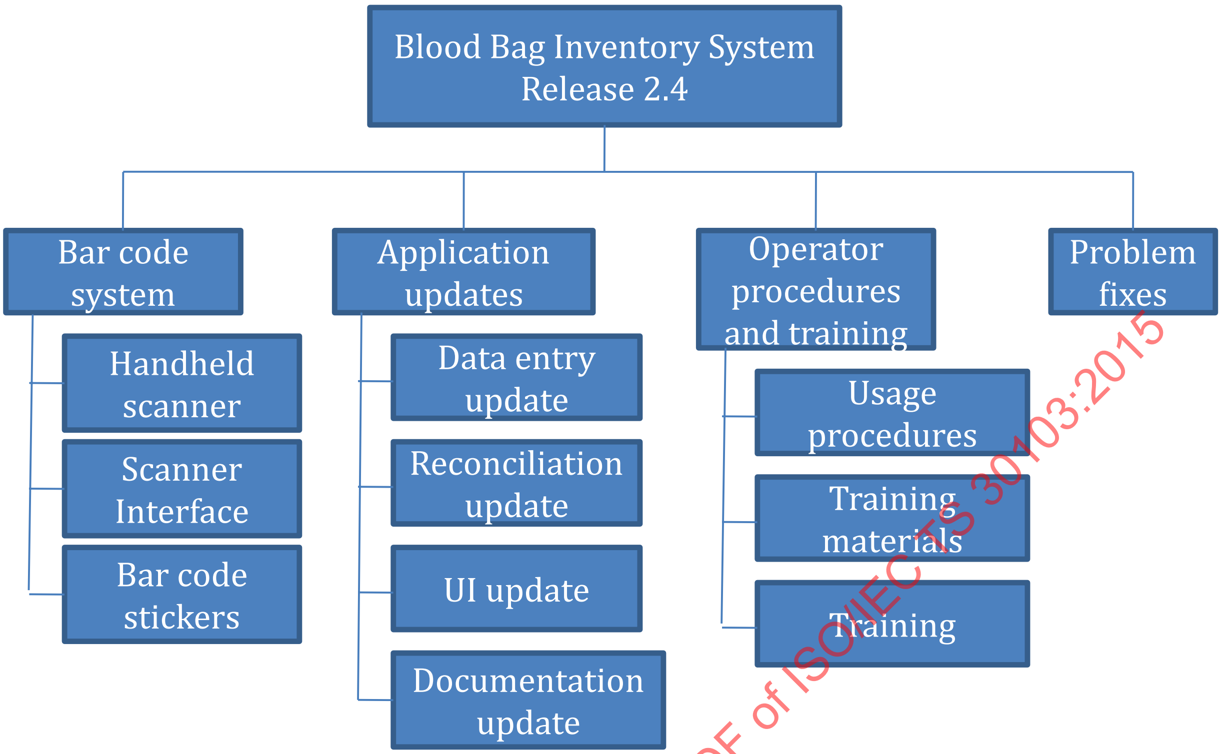

Figure A.1 — Example product breakdown structure

Requirements should be established for each system element, drilled down from the overall system requirements. Table A.1 shows an example of this drilldown for some requirements.

Table A.1 — Example requirements drilldown

System Element	Requirements (partial)	Basis for determination
Blood Bag Inventory System Release 2.4	— Accuracy of barcode entry: 99,9 %. Average time taken for receipt or issue of bag: 30 s. — Convenient to right & left-handed operators.	Business case for transitioning to new system, operational goals, user characteristics.
Barcode system	— Scanning availability = 95 % (barcode not scannable 5 % of time). — Scanning reliability: 99,9 % (errors in scanning). — Scanning time = 10 s. — Convenient to right & left-handed operators.	Accuracy goal is allocated among scanning function availability (95 %), scanner reliability (99,9 %), manual data entry (application, 90 %), and error detection through cross-checking procedures (user training, 80 %). This allocation has a combined error rate of (1-0,95)×(1-0,9) × (1-0,8) + 0,95 × (1-0,999) × (1-0 8) × (1-0,9) = 0,001 02 i.e. 99,9 % accuracy [i.e. undetected scanning error + undetected manual entry error]. Allocation of time: 10 s + 15 s + 5 % × 90 s = 29,5 s. User characteristics requirements assigned to all applicable system elements.
Application updates	— Data entry reliability = 90 %. — Data entry time = 90 s.	
Operator procedures and training	— Manual cross-checking reliability = 80 %. — Cross-checking time = 15 s. — Ergonomics design and training for right & left-handed operators.	

There are three system requirements shown in the Table A.1:

— Accuracy: Barcodes for blood bags added or removed from the inventory shall be entered into the system with at least 99,9 % accuracy (subject to operators following procedures);

— Latency: The average time to receive or issue a blood bag using the system shall be less than 30 s;

— Usability: The system shall provide support for both left- and right-handed operators.

The usage model is as follows. When a blood bag is received or issued, first the operator attempts to scan the barcode using the handheld scanner. If this fails, they enter the barcode manually. In either case, the operator is expected to cross-check whether the barcode was entered accurately into the system.

The scanning function is considered to be unavailable if the scanner is unable to scan the barcode for any reason. The scanning is unreliable if the barcode is scanned, but the result is incorrect for any reason (including poor positioning of the bag by the operator). Data entry reliability covers the possibility of operator errors in data entry. The operators may also fail to detect errors when they cross-check whether the barcodes were read correctly.

Two usage scenarios are identified. In scenario A, the barcode is scanned automatically, with the user checking the results. It produces an error if the scanning is unreliable and the operator fails to detect the error. The probability of this is (1 % to 99,9 %) × (1 % to 80 %) = 0,000 2, based on the chosen design requirements. In scenario B, the barcode cannot be scanned automatically, the data entry is erroneous, and the operator fails to detect the error. The probability of this is (1 % to 95 %) × (1 % to 90 %) × (1 % to 80 %) = 0,001 02. This corresponds to an accuracy of 99,9 %.

It is the responsibility of the design to decompose the accuracy target into target percentages for the four parameters (scanning function availability, scanner reliability, data entry reliability and checking reliability). The design choices are limited by feasibility of achievement. Institutional knowledge and benchmarking are used to decompose the accuracy requirements into these various design requirements.

Similarly, the latency requirement has also been decomposed by the design. The time taken for scenario A is scanning time + cross-checking time = 10 + 15 = 25 s. The time for scenario B is scanning time (failure) + data entry time + cross-checking time = 10 + 90 + 15 = 115 s. Since scenario B occurs only 5 % of the time, the average latency is 0,95 × 25 + 0,05 × 115 = 29,5 s. Again, the design sets these target parameters based on institutional knowledge about feasibility of achievement and the tradeoffs in effort and cost associated with trying to achieve tighter goals for various parameters.

The ergonomics requirement is assigned to both the Barcode system and to user training based on applicability. The example illustrates that product requirements may be allocated as design requirements to one or more system elements based on applicability, or decomposed and distributed among multiple system elements.

Annex B
(informative)

Example: Creation of Process Instance Descriptions

This Annex shows a sample of creating process instance descriptions. In this case, it is the instantiated Requirements Analysis process for the Barcode system. It is derived from the System Requirements Specification for the overall Blood Bag Inventory System Release 2.4, and based on the Stakeholder Requirements Definition.

First, the set of process instances need to be identified for each system element, as shown in Table B.1. The set of instances varies based on the element type and the realization approach for each system element.

Table B.1 — Example collection of process instances

Blood Bag Inventory System Release 2.4	Barcode System	Application Updates	Operator Procedures and Training
Stakeholder Requirements	Stakeholder Requirements	Stakeholder Requirements	Stakeholder Requirements
Requirements Analysis	Requirements Analysis	Requirements Analysis	Requirements Analysis
Architectural Design	Acquisition		Architectural Design
Implementation		Implementation	Implementation
Integration	Integration	Integration	Integration
Verification	Verification	Verification	Verification
Transition			Transition
Validation			

Process instance: Requirements Analysis for Blood Bag Inventory System Release 2.4.

Process outcomes, activities and tasks (from ISO/IEC/IEEE 15288:2008):

Outcomes: As a result of the successful implementation of the Requirements Analysis Process: a) The required characteristics, attributes, and functional and performance requirements for a product solution are specified. b) Constraints that will affect the architectural design of a system and the means to realize it are specified. c) The integrity and traceability of system requirements to stakeholder requirements is achieved. d) A basis for verifying that the system requirements are satisfied is defined. Activities and tasks: a) **Define System Requirements.** This activity consists of the following tasks: 1) Define the functional boundary of the system in terms of the behaviour and properties to be provided. 2) Define each function that the system is required to perform. 3) Define necessary implementation constraints that are introduced by stakeholder requirements or are unavoidable solution limitations. 4) Define technical and quality in use measures that enable the assessment of technical achievement. 5) Specify system requirements and functions, as justified by risk identification or criticality of the system, that relate to critical qualities, such as health, safety, security, reliability, availability and supportability. b) **Analyze and Maintain System Requirements**. This activity consists of the following tasks: 1) Analyze the integrity of the system requirements to ensure that each requirement, pairs of requirements or sets of requirements possess overall integrity. 2) Feed back the analyzed requirements to applicable stakeholders to ensure that the specified system requirements adequately reflect the stakeholder requirements to address the needs and expectations. 3) Demonstrate traceability between the system requirements and the stakeholder requirements. 4) Maintain throughout the system life cycle the set of system requirements together with the associated rationale, decisions and assumptions.

The success criteria for the process instance are derived from the outcomes above, the Bar Code System requirements, and domain knowledge relating to scannable information (e.g. barcodes, magnetic strips, RFID etc.) and characteristics of handheld devices.

Instantiation

Success criteria:

— A system integrity level is established for the system, using ISO/IEC/IEEE 15026-3;

— The target accuracy of barcode entry is defined, including target availability of the scanning function (ability to scan the barcode on the sticker) and target scanning reliability (code scanned incorrectly);

— Fault modes are identified for scanning of barcode labels, including possible operator errors, malfunctions, defects in the item being scanned and errors in the software system state;

— Productivity goals are defined for the barcode system in terms of average time for scanning a barcode;

— Capital and operating costs for the barcode system are identified, including one-time equipment acquisition and deployment costs, associated infrastructure and consumables;

— Confirmation of the original business case is created in terms of the comparison of systems costs with the ROI from productivity and reliability improvements;

— User characteristics are identified for Barcode system operators. Usability goals are defined;

— The changes needed in the data entry software to accommodate integration with the barcode system are identified;

— APIs are defined to support the verification of the scanned/manually entered data against the contents of the barcode.

Below are the detailed tasks needed to achieve the success criteria, also derived based on institutional knowledge.

NOTE Activities and tasks from 15288 are included to illustrate that the detailed tasks are derived from them.

a) **Define System Requirements**

1) Define the functional boundary of the system in terms of the behaviour and properties to be provided.

— Identify the interfaces to the existing data entry system and any changes needed to the existing data entry software to accommodate integration with the Barcode system.

— Identify the changes to the existing data entry system usage processes.

— Identify the desired behaviour and properties of the Barcode system.

2) Define each function that the system is required to perform.

3) Define necessary implementation constraints that are introduced by stakeholder requirements or are unavoidable solution limitations.

— Identify API support needed for verification of scanning correctness.

4) Define technical and quality in use measures that enable the assessment of technical achievement.

— Define scanning reliability and scanning function availability measures.

— Define measure of scanning productivity.

— Define measures of usability.

5) Specify system requirements and functions, as justified by risk identification or criticality of the system, that relate to critical qualities, such as health, safety, security, reliability, availability and supportability.

— Define system integrity level (tasks identified from ISO/IEC/IEEE 15026-3):

— Perform risk analysis and evaluation.

— Assign system integrity level.

— Specify scanning reliability, scanning function availability and average scanning time goals.

— Identify fault modes for scanning, including possible operator errors, malfunctions, defects in the item being scanned and errors in the software system state.

— Specify usability goals.

— Identify user characteristics for Barcode system operators.

b) **Analyse and Maintain System Requirements.** This activity consists of the following tasks:

1) Analyse the integrity of the system requirements to ensure that each requirement, pairs of requirements or sets of requirements possess overall integrity.

— Determine feasibility of integration with the data entry software and estimate effort needed.

   - Analyse the handling of fault modes to determine preliminary feasibility of meeting the reliability goals.
   - Preliminary analysis of Barcode system usage procedures to determine feasibility of achieving the expected productivity goals.

2) Feed back the analysed requirements to applicable stakeholders to ensure that the specified system requirements adequately reflect the stakeholder requirements to address the needs and expectations.
   - Feedback with operators, customer. Include ROI analysis feedback with customer.

3) Demonstrate traceability between the system requirements and the stakeholder requirements.

4) Maintain throughout the system life cycle the set of system requirements together with the associated rationale, decisions and assumptions.

Competencies needed to perform the tasks:

- Domain knowledge of handheld scanning.
- Knowledge of embedded device integration.
- Requirements elicitation and analysis.
- Typical goals and measures for quality characteristics, including reliability, accuracy, productivity and usability.

Annex C
(informative)

Example: Consistency Tracker

This Annex shows a few Consistency Tracker entries for the sample problem to illustrate the concept. Clause numbers may be used to identify the information item and target information item. The information contained here can be processed to derive a status for each information item and shown visually as a network of consistency gaps to be closed.

Table C.1 — Example Consistency Tracker

Information item or clause	Target Information item/clause/decision	Expected Relationship	Gap Description	Status
Barcode system requirements: reliability requirement	Inventory System requirements: accuracy requirement	Decomposition	none	OK
Barcode system requirements: availability requirement	Inventory System requirements : accuracy requirement	Decomposition	Unclear if unavailability due to infrastructure problems (e.g. power) is included	To be closed
			Feasibility of meeting requirement not yet checked against available products in the market	Deferred
Operator procedures and training architectural design	Operator procedures and training requirements : ergonomics requirement	Coverage of requirements	Procedure for left-handed operators has intentionally not been defined.	Intentional
Operator procedures requirements specification	Operator procedures stakeholder requirements	Derived from	Stakeholder feedback on operator procedures not yet reflected in requirements specification.	Pending
...				

Annex D
(informative)

Example: Process View for Specific Requirement

This Annex illustrates the use of process views to depict the collection of processes and tasks that address a specific system requirement and how this correlates with institutional knowledge. The process view extracts tasks from each process instance that relate to the particular product requirement.

This is a specialty engineering process view, as described in ISO/IEC/IEEE 15288:2008, Annex D. It shows the success criteria and detailed tasks corresponding to each life cycle process needed to implement the requirement below. These success criteria and detailed tasks are based on outcomes, activities and tasks defined in ISO/IEC/IEEE 15288:2008. Excerpts from ISO/IEC/IEEE 15288:2008 life cycle processes relevant to the instantiation are shown in text boxes, with the success criteria and tasks (for each instance of that life cycle process, where applicable) below it. The associated institutional knowledge is identified in italics.

These success criteria and detailed tasks supplement the outcomes, activities and tasks identified in ISO/IEC/IEEE 15288:2008. Only success criteria and tasks relevant to the accuracy requirement are shown.

Product Requirement: Expected accuracy of blood bag barcodes entered into the system (subsequently determined to be 99,9 %).

Stakeholder Requirements Definition process.

a) **Elicit Stakeholder Requirements.**

2) Elicit stakeholder requirements from the identified stakeholders.

b) **Define Stakeholder Requirements.**

1) Define the constraints on a system solution that are unavoidable consequences of existing agreements, management decisions and technical decisions.

2) Define a representative set of activity sequences to identify all required services that correspond to anticipated operational and support scenarios and environments.

3) Identify the interaction between users and the system.

4) Specify health, safety, security, environment and other stakeholder requirements and functions that relate to critical qualities.

Success criteria

— Barcode entry accuracy and scanning function availability requirements are elicited.

— Infrastructural requirements relating to scanners have been conveyed to the customer.

Detailed tasks

— Identify infrastructural requirements associated with handheld scanners (e.g. provisions for scanner holders, sticker inventory management) and communicate them to the customer.

<u>Institutional knowledge</u>: *Needs for infrastructure support.*

— Elicit operator requirements using the handheld devices operator requirements checklist.

<u>Institutional knowledge</u>: *Requirements checklist for operators of handheld devices.*

— Determine usage scenarios for the handheld scanner, including scenarios involving manual data entry.

Institutional knowledge: *Typical usage scenarios for handheld scanning devices (input to task).*

— Elicit stakeholder requirements for barcode entry accuracy and availability of handheld scanning function.

Requirements Analysis Process. a) **Define System Requirements.** 4) Define technical and quality in use measures that enable the assessment of technical achievement. 5) Specify system requirements and functions, as justified by risk identification or criticality of the system, that relate to critical qualities, such as health, safety, security, reliability, availability and supportability. b) **Analyze and Maintain System Requirements.** 1) Analyze the integrity of the system requirements to ensure that each requirement, pairs of requirements or sets of requirements possess overall integrity.

Success criteria

— The target accuracy of barcode entry, target availability of the scanning function (ability to scan the barcode on the sticker) and target scanning reliability (code scanned incorrectly) are defined.

— Fault modes and their handling for handheld scanning are identified.

Institutional knowledge: *Importance of fault modes identification for the handheld devices domain.*

Detailed tasks

— Define barcode entry accuracy, scanning reliability and scanning function availability measures.

Institutional knowledge: *Typical reliability and availability measures for handheld scanning.*

— Specify barcode entry accuracy, scanning reliability and scanning function availability goals.

Institutional knowledge: *ISO/IEC 25030 standard for specification of product quality requirements.*

— Identify fault modes for scanning, including possible operator errors, malfunctions, defects in the item being scanned and errors in the software system state.

Institutional knowledge: *Typical procedures and fault modes for handheld scanning.*

— Analyse the handling of fault modes to determine preliminary feasibility of meeting the reliability goals.

Institutional knowledge: *Typical handling actions. Handheld scanner reliability and scanning function availability benchmarks (including problems with unreadable barcodes).*

Architectural Design Process a) **Define the Architecture.** 2) Partition the system functions identified in requirements analysis and allocate them to elements of system architecture. Generate derived requirements as needed for the allocations. b) **Analyze and Evaluate the Architecture.** 1) Analyze the resulting architectural design to establish design criteria for each element. 2) Determine which system requirements are allocated to operators. 3) Determine whether hardware and software elements that satisfy the design and interface criteria are available off-the-shelf.

Success criteria – Release 2.4 overall

— The architectural design specified supports 99,9 % accuracy of blood bag barcodes entry.

— Verification interfaces are supported for determining the data obtained from the barcode scanner.

Detailed tasks – Release 2.4 overall

— Allocate the reliability and availability requirements, and the validation interface support requirement among the system elements, including operator training.

Institutional knowledge: *Benchmark scanner capabilities, barcode sticker quality, operator error rates. Approach for decomposition of reliability and availability requirements.*

— Create a fault tree for barcode entry and analyse it to ensure feasibility of achieving accuracy requirement, based on benchmarks of scanning availability and reliability, and data about human error rates.

Institutional knowledge: *Use of fault trees.*

Success criteria – Barcode system

— The architectural design specified supports 95 % availability of the scanning function and 99,9 % reliability of scanning (i.e. scanning rarely produces incorrect code entries).

Detailed tasks – Barcode system

— Allocate the reliability and availability requirements among the system elements, including the handheld scanner and barcode stickers.

Institutional knowledge: *Relative contribution of barcode stickers and handheld scanner to scanning function availability.*

— Determine whether hardware elements that satisfy the reliability criteria are available off-the-shelf.

Success criteria – Application

— The specified architectural design includes an interface for determining the data obtained from the barcode scanner.

— The specified architectural design supports manual data entry and user interfaces to support operators in checking correctness of barcode entry.

Detailed tasks – Application

— Design interface for providing the data obtained from the barcode scanner.

Success criteria – User Training

— Usage procedures are defined that support the goal of 99,9 % scanning accuracy.

Institutional knowledge: *Impact of usage ergonomics and operator practices on scanning accuracy.*

Detailed tasks – User Training

— Define operator procedures and guidelines for scanning and entry of barcodes.

Institutional knowledge: *Typical procedures and guidelines for data entry using handheld scanners. Special handling requirements associated with the blood bags domain, including any applicable regulatory requirements.*