Requirements Notes

General Requirement Statement Notes

A software requirement is a property that must be exhibited by something in order to solve some problem in the real world. A requirement can be written in the form of a natural language or some other language.

Product and Process Requirements

Product Requirement - Need or constraint on the software to be developed.

Process Requirement - A constraint on the development of the software

Process Requirements may also be imposed directly by the development organization, their customer, or a 3rd party.

Functional and Nonfunctional Requirements

Functional Requirements - describe the functions that the software will execute. These can be known as capabilities or features. To validate a functional requirements behavior a finite set of test steps can be written to validate its behavior.

Nonfunctional requirements - act to constrain the solution. Also known as constraints or quality requirements.

Nonfunctional requirements can be further classified into performance requirements, maintainability requirements, safety requirements, reliability requirements, security requirements, interoperability requirements.

Requirements Construct

Stakeholder requirements, system requirements, and system element requirements need to be developed to contribute to requirements validation with the stakeholders.

Specific requirements contain:

-Shall solve a problem, achieve an objective, or address a stakeholder concern

-Has measurable conditions

-Bounded by constraints

-Can be verified

-Defines the performance of the system when used by a specific stakeholder or the corresponding capability of the system, but not a capability of the user, operator or other stakeholder

If a requirement is written in a natural language, it should include a subject and a verb, as well as other information that adequately express the information content of the requirement.

-Requirements are mandatory binding provisions and use ‘shall’

-Non-requirements like descriptive text use verbs such as ‘are’, ‘is’, and ‘was'. Avoid using ‘must’ so it is not mistaken as a requirement.

-Statements of fact use ‘will’. ‘Will’ can also be used to establish context or limitations of use.

-Preferences or goals use ‘should’ and are not requirements.

-Avoid negative requirements such as ‘shall not’.

-Use an active voice.

-Avoid terms such as ‘shall be able to’.

Constraints - restrict the design solution or implementation of the systems engineering process.

Works Cited: IEEE 29148-2018 5.2.4

Characteristics of Individual Requirements

Each stakeholder, system, and system element shall possess these characteristics:

-Necessary: The requirement defines an essential capability, characteristic, constraint, or quality factor.

-Appropriate: The specific intent and amount of detail of the requirement is appropriate for what it is referring to.

-Unambiguous: The requirement is stated in a way that it is able to only be interpreted in one way.

-Complete: The requirement sufficiently describes necessary capability, characteristics, constraints, or quality factors so that no additional information is needed to understand the requirement.

-Singular: The requirement states a single capability, characteristic, constraint or quality factor.

-Feasible: The requirement is within system constraints and within acceptable risk.

-Verifiable: The requirement is structured and worded so that it can be proven to the customer's satisfaction at the level the requirement exists.

-Correct: The requirement is an accurate representation of what is needed.

-Conforming: The requirement conforms to an approved standard template and style for writing.

Works Cited: IEEE 29148-2018 5.2.5

Requirements Language Criteria

Requirements should state 'what' is needed, and not 'how'.

Vague and general terms should be avoided in requirements.

Unbounded or ambiguous terms to avoid:

-Superlatives ('best', 'most')

-Subjective language ('user friendly', 'easy to use')

-Vague pronouns ('it', 'this', 'that')

-Ambiguous terms ('almost always', 'significant', 'minimal', 'or', 'and/or')

-Open-ended, non-verifiable terms ('provide support', 'but not limited to')

-Comparative phrases ('better than', 'higher quality')

-Loopholes ('if possible', 'as appropriate')

-Terms that imply totality ('all', 'never', 'every')

Works Cited: IEEE 29148-2018 5.2.7

Requirements attributes

Well-formed requirement should include descriptive attributes defined to assist in identifying relevant requirements and to help in understand and managing requirements.

Examples of requirement attributes:

-Each requirement should be uniquely identified.

-Version number to make sure that the correct version of the requirement is being implemented.

-Owner.

-Stakeholder priority should be identified.

-Risk value assigned to each requirement based on risk factor.

-Rationale for establishing each requirement.

-Difficulty for each requirement should be noted.

-Type attribute to aid in identifying relevant requirements and categorizing requirements into groups for analysis and allocation.

Examples of type attributes:

-Functional/Performance requirements describe the system or system element functions or tasks.

-Interface requirements are the definition of how the system is required to interact with external systems.

-Process Requirements. These are stakeholder requirements imposed thought the contract or statement of work.

-Quality Requirements are non-functional requirements.

-Usability/Quality-in-Use Requirements. Provide the basis for the design and evaluation of systems to meet the user needs.

-Human Factors Requirements - State required characteristics for the outcomes of interaction with human users in terms of safety, performance, effectiveness, efficiency, reliability, maintainability, health, well-being and satisfaction.

Works Cited: IEEE 29148-2018 5.2.8

Characteristics of a Set of Requirements

- Sets of requirements - provide a consistent solution that meets stakeholders intentions and constraints
- Required characteristics for a set of requirements
    - Complete - the set of requirements stands alone without needing further explanation. It describes necessary capabilities, characteristics, and constraints.
    - Consistent - the set of requirements contains unique requirements that don’t conflict with other requirements. All requirements are homogenous.
    - Feasible - the set of requirements is able to be completed within given constraints and acceptable risk.
    - Comprehensible - set of requirements is clear about what is expected and its relationship to the given system.
    - Able to be validated - completion of the requirements will lead to the achievement of the entity needs within constraints.
- Check the requirements set to make sure they follow the characteristic to avoid requirements change and growth throughout the development life cycle

Set of Requirements

Key	Summary	Description	T	Linked Issues	P	Labels
AA-62	The previous version of the scheduling software shall be shut down upon rollout of the new version.	The previous version of the scheduling software needs to be taken down when the new one rolls out so that pilots don’t get confused. Source: Director of Safety	▤		⌄	non-functional
AA-61	The scheduling software shall be installed on all company computers upon rollout.	Scheduling software will be installed onto all company computers for easy access. Source: Director of Safety	▤		⌃	
AA-60	The scheduling software shall implement data from previous scheduling portal into the new scheduling portal.	Since the necessary data about employees and trips already exists is will be transferred. Source: Director of Safety	▤		⌄	functional
AA-59	The scheduling software shall not go down for maintenance for more than one hour per day.	Pilots and flight attendants must be able to access the portal at all times of the day and from different time zones so the portal may only do down for 1 hour a day. Source: Director of Safety	▤		=	
AA-58	The scheduling software shall be approved by the safety department, scheduling department, and pilot union.	Before rollout of the software it must be approved. Source: Director of Safety	▤		⏫	functional
AA-57	The scheduling software shall be compatible with windows, apple, and android devices.		▤		⌃	
AA-56	The scheduling software shall provide interactive user training after each employees first log in	Giving employees user training after their first time logging into the new scheduling portal will allow them to get familiar with new functionality. Source: Director of Safety	▤		⌄	
AA-55	The scheduling software shall include a database that holds pilot and trip information		▤		⌃	non-functional

AA-54	The scheduling software shall not exceed the budget determined by Alaska Airlines	The cost of developing and implementing the software cannot go above the cost allocated for the project. Source: Director of Safety	▤		⇈	functional
AA-48	The scheduling software shall allow pilots to pick up a trip from a trip listed on the open time page of the scheduling portal..	Source: User	▤	AA-23	=	functional
AA-47	The scheduling software shall provide an option to download a pdf of bid sheets.		▤		⇊	non-functional
AA-46	Pilots shall view bid sheets, using Alaska Airlines scheduling portal, from the bid subpage.	Pilots need to view their personalized bid sheets before bidding for their schedule. Source: User	▤		⇈	functional
AA-45	The scheduling software shall provide a filtration system on the trip trade board.	When a pilot or flight attendant is looking for a trip to pick up they will be able to filter the available trips by length, location, and pay. Source: Scheduling department	▤		⇊	functional
AA-44	The scheduling software shall allow pilots to post a trip they have been assigned to the trading board on the schedule portal.	Source: User	▤	AA-17	=	functional
AA-43	The scheduling software shall require an employee ID and unique password to log into the scheduling portal.	Source: User	▤	AA-16	↑	non-functional
AA-42	The software shall provide scheduling agents a means to approve or deny an employee time off request.	Scheduling agents must be able to approve or deny any requests a pilot or flight attendant makes for time off. Source: Scheduling department	▤		↓	functional
AA-41	The scheduling software shall send all employee time-off requests to scheduling agents.	When a pilot or flight attendant requests time off, the request must be sent to a scheduling agent so they can approve or deny the request. Source: Scheduling department	▤		=	functional
AA-40	The scheduling software shall allow pilots to request a set of days off, using the scheduling portals time off page.	Source: User	▤	AA-19	↓	functional
AA-39	Scheduling agents shall be able to remove a trip, using Alaska Airlines scheduling software, from an individual pilot's schedule.	Source: Scheduling department	▤	AA-12	↑	functional

19 issues

Characteristics of this set of system/software requirements

Characteristics of a set of requirements:

- Complete - each requirement describes necessary capabilities, characteristics, constraints, or quality factors to meet without the need for more information after reading the requirement.
- Consistent - the requirements in the set don’t conflict with or overlap and the terminology used within the set of requirements is consistent.
- Feasible - the set of requirements is achievable within the stated constraints.
- Comprehensible - the set of requirements is clear about what is expected from each requirement and its relation to the system
- Able to be validated - it is practical that the set of requirements will lead to the achievement of the the business goals.

Checklist Item	
The system-level technical requirements are traceable to the user requirements.	✓
Each system requirement describes something relevant: a function the system must perform, performance a function must provide, a constraint on the design, or a reference such as to an interface definition.	✓

The level of detail that the requirements provide about system functionality is appropriate. The requirements are sufficient to describe what the overall system must do, what its performance must be, and what constraints an engineer should consider. There are few requirements that specifically affect the design of only one component of the system. The major requirements drivers (e.g., those stressing the design) and associated risks should be identified.	☑
The requirements include any legal or regulatory constraints within which the system must perform. Example: There may be restrictions on the use or quantity of certain hazardous materials in a system.	☑
The requirements include enterprise architecture constraints within which the system must integrate (or toward which the system is desired to migrate). Requirements include appropriate open systems and modularity standards. Examples: DoD Net-Ready requirements, modular open system architecture concepts, Electronic Systems Center strategic technical plan goals	☑
Environmental design requirements are specified. Example: A control unit may be in a controlled office environment and the other major components may be outdoors, thus two environments must be defined and associated with the functionality operating in each environment.	☑
All external interfaces for the system are included. Major internal interfaces may also be included if they are important to system modularity, or future growth in capability. These may include physical (mechanical fastening, electrical wiring, connectors), functional (mechanical stress transfer points, cooling, power sources, antennas, wire message formats, data exchanges), and software (software interface specifications, library calls, data formats, etc.). Remember that an internal interface between two subsystems that use a transport mechanism that is not part of the system is a hidden external interface. For example, two subsystems that communicate internally with each other over a sensitive but unclassified network as the internal interface (the data exchanged between them) and an external interface (the wiring and internet protocols to enable the data exchanges with the network).	☑
Requirement statements use the word “shall” or “should.” The word “shall” has meaning in contractual language and is enforceable legally. Other words like “will,” “may,” “should,” and “must” may show intent but are not legally binding in contracts. In some situations, it may be desirable to use “should” to show the government’s intent and preference while at the same time allowing flexibility and latitude. Example: “The system shall have a mean time between failures of greater than 500 hours.”	☑
Requirements statements are unambiguous. Terminology is clear without the use of informal jargon. Statements are short and concise.	☑
Performance requirements statements (including logistics/sustainment/support) are quantifiable, testable, and/or verifiable. Avoid the phrase “shall not.” It is very difficult to prove a negative. Avoid qualitative words like “maximize” or “minimize.” They force an engineer to judge when the design is good enough. The user may think that the engineer did not “minimize enough” and get into a legal argument with the contractor. Note: Every user requirements document includes: “the system shall be easy to use” requirement. Talk to other MITRE staff for examples from other projects and seek out a human factors specialist for requirements wording that is suitable both for specifying these requirements and methodologies for verifying them. Avoid specific, one-point values when defining requirements. Use ranges (minimum of, more than, less than, maximum of, within, etc.) to accommodate appropriate interpretation. Using a single point value may cause arguments if the system is tested at that exact value only, or if a test appears to be successful from an intent perspective, but does not meet the exact value stated in the system requirement. Example: The system shall process a minimum of 100 transactions/sec. Example: The system shall be operable up to and including 30,000 ft. Example: The system shall operate in temperatures between 5 and 35 degrees Celsius.	☑
If objective performance values are included as goals, ensure they are clearly identified and distinguished from firm requirements. User requirement documents refer to threshold requirements (those that must be provided), and objective requirements (better performance has value to the user, but not above the objective requirement). Example: The system shall detect and display up to 100 targets within the surveillance volume with a goal of detecting and displaying up to 125 targets.	☑
The operational and support environment is described and defined. Example: The system shall be maintainable by an Air Force level 5 technician. Example: The system shall be reparable while in flight.	☑
The requirements include appropriate use of Government and industry specifications, standards, and guides. Only include them if they are relevant and ensure that the correct version is listed in a list of reference documents.	☑
Verification approaches for all system performance and sustainability requirements are complete and appropriate. Every requirement must have a verification method identified. If a requirement cannot easily be verified by a direct inspection, measurement, or onetime demonstration of the requirement, the verification requirement should include an expanded test criteria description to ensure that there is no disagreement later in the program. This can include describing the number of trials, statistical criteria to be used, conditions of the test such as simulated inputs, etc.	☑

Sample Requirement Statement

Schedule.Add Version 1.0

Priority:High

Type: Functional

Owner: Stirling Sites

Scheduling agents shall be able to remove a trip, using Alaska Airlines scheduling software, from an individual pilot's schedule.

Characteristics of this Individual Requirement

Necessary: The requirement specifies an essential capability for the pilot scheduling system to work as intended.

Appropriate: Level of detail in the requirement is specific and descriptive.

Unambiguous: Requirement is only able to be interpreted in the way it is intended to be.

Complete: The capability of the scheduling agent to add a trip to a pilot's schedule is stated in a way where no additional information is needed to understand the requirement.

Singular: The requirement states a single capability of the scheduling system.

Feasible: The requirement is within the constraints of the system.

Verifiable: The wording of the requirement makes it so the functionality can be proven to the customer’s satisfaction.

Correct: The requirement is an accurate description of what is needed.

Conforming: The requirement conforms to IEEE 29148-2018 standards.

Requirements Language Criteria

Vague, general, and ambiguous terms are avoided in the requirement statement. States what is needed and not how it is going to be done.

Requirements Attributes

Identification: Schedule.Add

Stakeholder Priority:High

Type: The requirement is a functional requirement because it describes a task of the scheduling system. It is also a process requirement because it is a constraint on the development of the software.