Section 4: Rules for Individual Need and Requirement Statements and for Sets of Needs and Requirements

This section defines the rules for individual need statements and requirement statements, needs expressions and requirements expressions, and sets of needs and sets of requirements that help them to be formulated such that they will be well-formed helping to have the characteristics defined in this Guide.

Included with each rule is an explanation of the rule, any qualifications or exemptions to the rule, followed by selected examples of the application of the rule.

In addition to the rules outlined in this section, the reader is encouraged to follow the principles of good technical writing (as they apply to a need or requirement statement) such as those outlined in the Simplified Technical English (STE) specification (ASD-STE100).

Note: Because need statements are expressed at a higher level of abstraction, some of the rules in this section may not always apply as rigorously to need statements as they do to requirement statements. The key is that the need statements have the characteristics defined earlier that are appropriate to the level of abstraction at which the need statements are written. Appendix D provides a matrix that maps the applicability of each rule to need statements and requirement statements in order to provide readers with a clearer understanding of the applicability of the guidance provided by each rule.

Appendix E contains cross reference matrices mapping the rules to the characteristics discussed in this Guide as well as mapping concepts and activities and attributes discussed in the NRM to the characteristics which also contribute to the quality of the need statements and requirement statements.

4.1 Accuracy

4.1.1 R1 – Structured Statements

Definition:
Need statements and requirement statements must conform to one of the agreed patterns, thus resulting in a well-structured complete statement.

Elaboration:
Organizations should provide needs and requirements writers with a catalogue of agreed patterns to follow. Each different type of need or requirement should have, at least, one assigned pattern. Following a specific agreed pattern boosts consistency within the sets of needs and sets of requirements, where similar requirements (requirements of a same types) will always follow a similar structure. When one (and only one) pattern is followed for a given need statement or requirement statement, the quality characteristics defined in Section 2 will be leveraged. Ambiguity will be reduced (C3), completeness of each need statement and requirement statement is guaranteed (C4), each need statement or requirement statement will be singular (C5), verifiability and

validatability are enforced (C7), and each need statement and requirement statement will be conforming (C9).

If a condition that applies to a need or requirement is not stated explicitly within the need statement or requirement statement, the need or requirement will not be Unambiguous (C3), Complete (C4), Verifiable/Validatable (C7), nor Correct (C8).

Detailed information on the different blocks that comprise a need or requirement pattern and further information about patterns and examples can be found in Appendix C. See also R2, R3, R11, R18, and R27.

Examples:

See Appendix C, for examples of need statements and requirement statement patterns.

4.1.2 R2 – ACTIVE VOICE

Definition:

Use the active voice in the need statement or requirement statement with the responsible entity clearly identified as the subject of the sentence.

Elaboration:

The active voice requires that the entity performing the action is the subject of the sentence.

The active voice is important in writing needs and requirements since the onus for satisfying the requirement is on the subject, not the object of the statement. If the entity responsible for the action is not identified explicitly, it is unclear who or what should perform the action making it difficult to determine how to verify/validate the statement since it is unclear as to which entity should be subject to the verification/validation activity.

Including the entity in the subject also helps ensure the need statement or requirement statement refers to the appropriate level requirement set consistent with the entity name (see R3).

If the entity is not clearly identified as the subject of the statement, the need or requirement is not Complete (C4).

Often when phrases such as “shall be” are used followed by a main verb, the statement is in the passive voice, which is why all the patterns into the agreed catalog of patterns (see R1 and Appendix C) shall make use of the active voice.

Examples:

Unacceptable: The Identity of the Customer shall be confirmed.

[This is unacceptable because it does not identify the entity that is responsible/accountable for confirming the identity.]

Improved by adding the subject: The Identity of the Customer shall be confirmed by the Accounting_System.

[Although this statement makes the subject clear, it is still unacceptable because the statement is in the passive voice in that the entity that is responsible/accountable for recording the audio is at the end of the statement rather than the beginning.]

Improved: The Accounting_System shall confirm the Customer_Identity.

[Note that “Accounting_System”, “confirm”, and “Customer_Identity” must be defined in the glossary since there are a number of possible interpretations of these terms—see R4. Also note that this statement is improved because it is in the active voice but, before it is acceptable, it needs to be further improved through the addition of appropriate condition and qualifying clauses]

Unacceptable: While in the Operations_Mode, the Audio having the characteristics defined in <ICD xxxx> shall be recorded.

[This is unacceptable because the entity that is responsible/accountable for recording the audio is not stated, which makes it difficult to identify which entity is responsible for the action, or how verification is to occur. In addition, the characteristics of the Audio are not referenced, nor are the conditions under which the action specified.]

Improved by adding the subject: While in the Operations_Mode, the Audio having the characteristics defined in <ICD xxxx> shall be recorded by the <SOI>.

[Although this statement makes the subject clear, it is still unacceptable because the statement is in the passive voice in that the entity that is responsible/accountable for recording the audio is at the end of the statement rather than the beginning.]

Improved: While in the Operations_Mode, the <SOI> shall record Audio having the characteristics defined in <ICD xxxx>.

[This is improved because it is in the active voice, but appropriate performance parameters should be added before the statement is acceptable. Also note that “Audio” must be defined in the glossary or data dictionary and required performance added for the requirement to be complete. The characteristics of the Audio received from a <xxxx> source is defined in the ICD.]

4.1.3 R3 – Appropriate Subject-Verb

Definition:

Ensure the subject and verb of the need or requirement statement are appropriate to the entity to which the statement refers.

Elaboration:

Subject

The subject of a need or requirement statement must be appropriate to the entity to which the statement refers, as discussed in Section 1.7.

Requirements referring to the:

- business management level have the form “The <business> shall …”;
- business operations level have the form “The <business unit> shall …”;
- system level have the form "The <SOI> shall …";
- subsystem level have the form "The <subsystem> shall ..."; and
- system element level have the form "The <system element> shall ...".

All these different structures (with the different subjects) shall be included in the catalog of agreed patterns (see R1 and Appendix C).

As a general rule, sets of need statements and sets of requirement statements for a specific entity should only contain needs or requirements for that entity—that is:

- a set of business management requirements would contain only business management needs or requirements,
- a set of business operations requirements would contain only business operations needs or requirements,
- a set of system requirements would contain only needs or requirements that apply to the integrated system,
- a set of subsystem requirements would contain only needs or requirements that apply to that subsystem, and
- a set of system element requirements would contain only needs or requirements that apply to that system element.

In some cases, however, a higher-level entity may wish to prescribe needs and requirements that apply to a lower-level entity. For example, it may be important for a business to mandate that the new aircraft under development must use a particular engine (perhaps for support reasons) in which case they may make a statement at the business level that refers to an entity at the subsystem level.

Therefore, any set of entity needs or requirements can state, for that entity, needs or requirements that refer to itself, as well a lower-level entity to which the need or requirement applies (when there is a valid reason to do so). When this is the case, the prescriptive allocated need or requirement is treated as a constraint on the lower-level entity. In these cases, the lower-level entity will then include an entity specific child requirement and trace that child requirement to its parent or source. The reason for doing so should be included in the rationale attribute for the need or requirement.

To continue our aircraft example above, although many requirements at the business management level of the ACME Aircraft Company will begin with “The ACME Aircraft Company shall …”, the business may therefore wish to state at the business management level a requirement stating that all aircraft developed by the organization shall use an engine with specific characteristics. For a specific aircraft, child requirements will be written for the appropriate entity that implements the intent of the business management generic requirement. For the entity dealing with the engine specifically, the system level child requirement would begin “The Aircraft shall….”; and at the subsystem level the child requirement would begin “The Engine shall …” and trace back to the system level parent requirement, which, in turn, would trace back to the business management level constraint as the parent or source.

When talking about a quality or physical characteristic of an entity, some tend to write the requirements on the characteristic rather than the entity that has that characteristic which is not consistent with this rule nor rule and R3.

“The <entity characteristic> shall be <verb><value range>.”

While this may be appropriate for a design output, it is not appropriate for a design input. In accordance with this rule, the subject of the requirement should be the entity to which the requirement applies, and the characteristic should be the object.

“The <entity> shall <verb><characteristic> <value range>.”

Verb

As with the subject, the verb of a need statement or requirement statement must be appropriate to the subject of the need or requirement for the entity it is stated. For needs, the verbs such as

"support", "process", "handle", "track", "manage", and "flag" may be appropriate. However, they are too vague for requirement statements which therefore may not be Unambiguous (C3) nor Verifiable/Validatable (C7).

For example, at the business management level the use of a verb such as "safe", may be acceptable if it is unambiguous at that level, decomposed at the lower levels, and is verifiable at those levels.

When transforming these need statements into requirement statements the functions referenced by these verbs would be decomposed into specific functions the <SOI> can be verified to perform at stated performance levels and conditions of operations.

Examples:

Subject examples*:*

Business management requirements have the form "The <business> shall …". For example, "ACME_Transport shall …".

Business operations requirements on business elements have the form "The <business element> shall …". For example, requirements on personnel roles have the form: "The <personnel role> shall …"—such as, "The Production_Manager shall …"; "The Marketing_Manager shall …".

System level needs have the form "The <stakeholders> need the system to ……"

System requirements have the form "The <SOI> shall ..."—for example, "The Aircraft shall …"

Subsystem level needs have the form "The <stakeholders> need the <subsystem> to ……"

Subsystem requirements have the form "The <subsystem> shall …" - for example, once the subsystems are defined: "The Engine shall …"; and "The Landing_Gear shall …".

Verb examples*:*

System level stakeholder need: "The <stakeholders> need the <SOI> to process data received from <other system>."

System level requirement: "The <SOI> shall [process] data received from <another system>. having the characteristics defined in <Interface Definition XYZ.>"

Through analysis, the verb/function "process" could be decomposed into sub functions such as "receive", "store", "calculate", "report", and "display".

Then a decision needs to be made regarding the specific subsystem or system element in which these sub functions are to be performed.

If more than one subsystem or system element has a role in performing any one of the sub functions, that requirement should be communicated at the system level and allocated to the applicable subsystems or system elements.

If a sub function is to be implemented by a single subsystem or system element, then the sub function requirement should be communicated at the subsystem or system element set of requirements and traced back to the higher-level requirement from which it was decomposed.

Unacceptable system requirement: "The User shall ..."

[As discussed in Section 1.8, this is unacceptable because the requirement should be on the system, not the user or operator of the system. This wording is often the result of writing requirements directly from user stories or resulting need statements, without doing the

transformation of the use case or need into a requirement. Ask, what does the system have to do so that the need can be achieved?]

Improved: "The <SOI> shall …".

Unacceptable: The <SOI> shall display the Current_Time on the <Display Device> per <Display Standard xyz>.

[Again, "Current_Time" and "Display_Device" must be defined in the glossary or data dictionary. This statement is most likely unacceptable because the Display_Device is either a system element which is presumably at a level lower than the SOI (and is therefore not appropriate to this level), or the Display_Device is a system or system element outside the SOI (and therefore should be handled as an interface since requirements cannot be placed on it by this requirement set). The display standard would be developed by the organization that would define standards to be used when displaying all information by all applications the organization develops including fonts, font sizes, colors, spacing, brightness, human factors, etc.]

Improved: The <SOI> shall display the Current_Time per <Display Standard xyz>.

[This is improved because the action is appropriate to the level of the requirement set. Note that, before being an acceptable statement, appropriate condition and qualifying clauses must be added.]

4.1.4 R4 – Defined Terms

Definition:
Define all terms used within the need statement and requirement statement within an associated glossary and/or data dictionary.

Elaboration:
As systems become increasingly complex and software intensive, the ability to share data and information, including needs and requirements, across organizations both internal and external, is critical to project success. The definition and documentation of a common ontology by organizations and their projects is fundamental to the definition and documentation of a common ontology. This ontology includes the formal naming and definition of a set of terms, entities, data types, and properties as well as defining the relationships among these terms, entities, and data types that are fundamental to the project and organization of which the project is part. Having a documented, common ontology and an associated glossary and/or data dictionary, helps ensure consistent use of this data and information across all system lifecycle process activities and work products as well as across various groups within and external to the project. This common ontology is key to tool interoperability and the ability to share sets of data and information, including needs and requirements. Definitions of terms used within need statements and requirement statements must be agreed to, documented, and used consistently throughout the project and all SE artifacts developed during all lifecycle activities. Most natural languages are rich with words having several synonyms, each with a subtly different meaning based on context. For example, if a person states: "I need a cream for wrinkles." - what was the intended meaning of the word "for" – to "give", "remove", or "prevent"? If someone goes to a barber and states: "I

need my hair shorter – I want it over my ears." – what was the intended meaning of the word "over" - "cover" or "above"?

In need statements and requirement statements, shades of meaning will most likely lead to ambiguity and to difficulty during verification and validation activities across the lifecycle. Define terms in some form of ontology, including a glossary, data dictionary, or similar artifact that allows the reader of a need statement or requirement statement to know exactly what the writer's intended meaning was when the word was chosen.

The meaning of a term should be the same every time the word is used no matter the work product, SE tool, or artifact being developed across all lifecycle stages.

A standard format should be agreed within the organization to make the use of glossary terms identifiable in the need statements and requirements statements; for example, glossary items may be capitalized and multiple words in single terms joined by an underscore (such as "Current_Time"). This is essential for consistency to avoid using the word with its general meaning without context. This is the convention used in the examples in this Guide. This standard should be implemented and enforced within all SE tools used by the project to help ensure consistency.

For cases where needs and requirements will be translated into a different language, it is helpful to develop a "translation matrix" where terms in the originating language are listed along with the acceptable term to be used in the target language such that the original intent is communicated. The use of this matrix will help ensure consistency in the translations when multiple people are involved in the translations over time.

Examples:

Unacceptable: The <SOI> shall display the current time as defined in <Display Standard xyz>.

[This is unacceptable because it is ambiguous. What is "current"? In which time zone? To what degree of accuracy? In which format?]

Improved: The <SOI> shall display the Current_Time as defined in <Display Standard xyz>.

[Note that "Current_Time" must then be defined in the glossary or data dictionary in terms of accuracy, format, time zone, and units. Further, appropriate condition and qualifying clauses must be included before the statement is acceptable. In addition, the organization can define a display standard that applies to all products developed by the organizations concerning how information is to be displayed (fonts, colors, size, spacing, human factors, etc.]

4.1.5 R5 – Definite Articles

Definition:

Use the definite article "the" rather than the indefinite article "a".

Elaboration:

The definite article is "*the*"; the indefinite article is "*a*."

When referring to entities, use of the indefinite article can lead to ambiguity. For example, if the need or requirement refers to "a user" it is unclear whether it means any user or one of the defined users for which the system has been designed.

This causes further confusion during verification and validation activities across the lifecycle—for example, babies are arguably users of baby food, but the system would fail if the test agency

sought to verify or validate that a baby could order, receive, open, and serve (or even independently consume) baby food.

From a medical device perspective who is the user – nurse, doctor, or patient? Although the patient is the one receiving the medical device, it is the nurse or doctor that will order, receive, open, install, and monitor the device after installation. From a risk perspective it is the patient that is of the main concern.

On the other hand, if the requirement refers to "the User", the reference is explicitly to the nature of the user defined in the glossary—in the baby food example, the "User" is presumably the adult responsible for feeding the baby. In the medical device example, the specific intended user would be defined in the glossary to remove ambiguity as to the intended user reference in the need statement or requirement statement.

Examples:

Unacceptable: The <SOI> shall provide a time display.

[This is unacceptable for a requirement because it is ambiguous—is the intent to provide **a** device to display the time or to display **the** time? If a device to display the time, then this is not appropriate to level. If to display **a** time - will any time displayed do? Is a one-off display of the time satisfactory? The writer's intention was most likely that they wanted the system to display continuously the current time in a format that is readable, yet if the developer provided **a** constant display of "10:00 am" (or even a one-off display of any time), they could argue (albeit unreasonably) that they have met the requirement; yet they would have clearly failed to meet the customer's need and intent.

Note that, as a need statement: "The <stakeholders> need the <SOI> to display the time.", the statement is arguably acceptable. However, as part of the transformation process, the requirement writers must remove the ambiguity - resulting in the following requirement statement (amongst others).

Improved*:* The <SOI> shall display the Current_Time as defined in <display standard xyz>.

[Note that "Current_Time" must be defined in the glossary or data dictionary since there are a number of possible meanings and formats of the more-general term "current time." For completeness, condition and qualifying clauses must also be added before the requirement is acceptable. In addition, the organization can define a standard that applies to all products developed by the organizations concerning how information is to be displayed (fonts, colors, size, spacing, human factors, etc.]

Exceptions and relationships:

The purpose of this rule is to avoid the ambiguity that arises because "a" or "an" is tantamount to saying, "any one of". In some cases, however, the use of an indefinite article is not misleading. For instance, "… with an accuracy of less than 1 second" allows the phrase to read more naturally and there is no ambiguity because of the accuracy quoted.

4.1.6 R6 – Common Units of Measure

Definition:

When stating quantities, all numbers should have appropriate and consistent units of measure explicitly stated using a common measurement system in terms of the thing the number refers.

Elaboration:

When stating quantities, all numbers should have appropriate and consistent units of measure explicitly stated using a common measurement system in terms of the thing the number refers.

There are three primary measurement systems: British Imperial, US, Metric.

For temperatures, the following are most often used: Celsius, Fahrenheit, or Kelvin.

Within a project, a common measurement system must be used consistently. For example, do not mix both US and metric units of measure within any of the project's artifacts.

Define precisely and use consistently the same expression for the measurement unit—for example, use consistently either 'liter' or the abbreviation 'l'.

Once the measurement system has been chosen for each system or system element, it is fundamental to continue preserving consistency by using the same measurement unit for each property-element pair. For instance, the length of the screw should always be in mm and the length of the chair always in cm; never mix them within the same property-element pair.

A word of caution: When converting from one measurement system to another, state the resulting value with a number of significant digits that appropriately preserves the precision of the original number. See also R40.

Appropriate use of units can sometimes be more difficult that it may seem at first glance. For example, it should be noted that "liter" is the US spelling of the unit "litre", so a choice has to be made as to which spelling to use. Further, "liter/litre" is not an SI unit although it is one of the units (such as hours and days) that is accepted for use in the context of SI units—the correct SI unit for volume is *cubic metres (m^3)*, noting again that a choice has to be made since "meter" is the US spelling of "metre". It is clear, even from this tiny portion of the measurement domain, that care must be taken to avoid ambiguity by defining the unit in the project glossary and then using the unit consistently throughout the set of statements.

Exceptions and relationships:

In rare cases, projects may require integration between systems (or system elements) using different units of measure (for example when integrating COTS from one measurement system into an SOI based on another). If this is unavoidable, artifacts must clearly indicate which units of measure are used and for which elements.

Examples:

Unacceptable: With power applied, The Circuit_Board shall <…> a temperature of less than 30 degrees.

[This is unacceptable because the units used are stated without indicating the specific measurement system used.]

Improved: With power applied, The Circuit_Board shall <…> a temperature of less than 30 degrees Celsius.

Unacceptable: The <SOI> shall establish communications as defined in <ICD xyz> with at least 4 in less than or equal to 10 seconds.

[This is unacceptable because the units used are incomplete. "4" of what?]

Improved: The <SOI> shall <establish communications> as defined in <ICD xyz> with at least 4 satellites in less than or equal to 10 seconds

[Note that the phrase “establish communications” is probably acceptable at the business level when used in a need statement, but the phrase would need further elaboration at lower levels so that the need is decomposed into verifiable requirements relating to frequency, type of communications (voice? data?), quality, bandwidth, etc. which would typically be defined in an ICD that defines the characteristics of the communication signal to be established.]

Unacceptable:

“The Fuel_System shall have a maximum fuel volume of 60 l”.

“When the Fuel_System detects the Fuel_Tank volume is less than 500 ml, the Fuel_System shall notify the operator within 1 second.”

[This is unacceptable because the two requirements use different units of measure (“l” and “ml”).

Improved:

“The Fuel_System shall have a maximum Fuel_Tank volume of 60 l “.

“When the Fuel_System detects the Fuel_Tank volume is less than 0.5 l, the Fuel_System shall notify the operator within 1 second.”

[Now both requirements are written using the same units of measure(liters).]

Additional Acceptable: "The Fuel_System shall have a maximum Fuel_Tank volume of 60 l.” “When the Signal_System detects the Fuel_Tank volume is less than 50 ml, the Signal_System shall light the Low-Fuel_Iindicator within 1 second."

[In this particular case, while the property (“Fuel_Tank volume”) remains the same, the component/element is different ("Fuel_System" vs "Signal_System"). The measurement unit can vary because we are dealing with different component–property pairs.]

[Note that “Fuel_System” and “Fuel_Tank” must be defined in the glossary or data dictionary.]

4.1.7 R7 – Vague Terms

Definition:
Avoid the use of vague terms.

Elaboration:
Vague terms can lead to ambiguous, unverifiable needs and requirements where the true intent is not being communicated. Avoid words that provide vague quantification, such as “some”, “any”, “allowable”, “several”, “many”, “a lot of”, “a few”, “almost always”, “very nearly”, “nearly”, “about”, “close to”, “almost”, and “approximate”, Avoid vague adjectives such as “ancillary”, "relevant”, “routine”, “common”, “generic”, “significant”, “flexible”, “expandable”, “typical”, “sufficient”, “adequate”, “appropriate”, “efficient”, “effective”, “proficient”, “reasonable” and “customary.” Adverbs qualify actions in some way and are particularly troublesome if vague. Avoid vague adverbs, such as “usually”, “approximately”, “sufficiently”, and “typically”, which can lead to ambiguous, unverifiable requirements that do not reflect accurately the stakeholder expectations. As a general rule, words that end in “-ly” often result in ambiguity.

Examples:
Unacceptable: The <SOI> shall usually be online. [This is unacceptable because "usually" is ambiguous - is availability what is meant?] Improved: The <SOI> shall have an Availability of greater than xx% over a period of greater than yyyy hours. [Note that "Availability" must be defined in the glossary or data dictionary since there are a number of possible ways of calculating that measure. Alternately, the availability could be expressed as a need. Then the organization responsible for transforming the need into a requirement could develop feasible concepts for meeting the needed availability and derived one or more well-formed requirements that result in the need being met.] Unacceptable: The Flight_Information_System shall display per <Display Standard xyz> the Tracking_Information for relevant aircraft within <xxxx seconds> of detection. [This is unacceptable because it does not make explicit which aircraft are relevant. Additionally, the statement allows the developer to decide what is relevant; such decisions are in the province of the customer, who should make the requirement explicit.] Improved: The Flight_Information_System shall display per <Display Standard xyz> the Tracking_Information of each Aircraft located less than or equal to 20 kilometers from the Airfield when in the Operations_Mode within <xxxx seconds> of detection. [Now it is clear for which aircraft the information needs to be displayed. Note that "Aircraft", "Tracking_Information", "Airfield", and "Operations_Mode" must be defined in the glossary or data dictionary.]

Exceptions and relationships:
R3 points out that the use of a verb such as "safe", may be acceptable at the business management or operations level as long as it is unambiguous at that level, decomposed at the lower levels, and is verifiable at the level stated. Similarly, some vague adjectives may be allowable at the business management or operations level, providing they are not ambiguous at that level. NLP/AI tools providing automatic assessment of this rule shall be flexible enough and tailorable in order not to identify this issue as an error at a business level, while enforcing the absence of vague terms in other lower-level documents.

4.1.8 R8 – ESCAPE CLAUSES

Definition:
Avoid the inclusion of escape clauses that state vague conditions or possibilities, such as "so far as is possible", "as little as possible", "where possible", "as much as possible", "if it should prove necessary", "if necessary", "to the extent necessary", "as appropriate", "as required", "to the extent practical", and "if practicable".

Elaboration:
Escape clauses give an excuse to the developer of the system at lower levels not to implement a need or requirement. From a contracting standpoint, needs or requirements with these phrases could therefore be interpreted as being optional even if communicated in a "shall" requirement statement.

Escape clauses can lead to ambiguous needs that the SOI cannot be validated to meet and are open to interpretation and that do not reflect accurately lifecycle concepts, or other sources, from which they were transformed.

Escape clauses can lead to ambiguous, unverifiable requirements that are open to interpretation and that do not reflect accurately the needs, source, or higher-level requirements from which they were transformed.

Examples:

Unacceptable: The GPS shall, *where there is sufficient space*, display the User_Location in accordance with <Display Standard xyz>.

[This is unacceptable because whether there is sufficient space is vague, ambiguous, and unverifiable. The requirement is clearer without the escape clause.]

Improved: The GPS shall display the User_Location in accordance with <Display Standard XYZ>.

[Note that appropriate performance measures must also be specified such as within what time, format, and accuracy. Also note that "GPS" and "User_Location" must be defined in the glossary or data dictionary.]

4.1.9 R9 – Open-Ended Clauses

Definition:

Avoid open-ended, non-specific clauses such as "including but not limited to", "etc." and "and so on".

Elaboration:

Open-ended clauses imply there is more required without stating exactly what.

Open-ended clauses can lead to ambiguous, unverifiable needs and requirements that do not reflect accurately the stakeholder's expectations and needs and can create ambiguity in the mind of the reader.

Needs or requirements with open-ended clauses are not Complete (C4).

Use of open-ended clauses also violates the one-thought rule (R18) that leads to the singular characteristic. If more cases are required, then include additional needs and requirements that explicitly state those cases.

Depending on the contract type (fixed price versus level of effort or cost plus) open-ended requirements can lead to serious interpretation problems concerning what is in or out of scope of the contract; possibly resulting in expensive contract changes. For level of effort or cost-plus contracts, open-ended requirements can be used by the supplier to do and bill the customer for additional work not intended by the customer leading to budget overruns and expensive contract changes.

Examples:

Unacceptable: The ATM shall display the Customer Account_Number, Account_Balance, and so on per <Display Standard xyz>.

[This is unacceptable because it contains an opened list of what is to be displayed.]

Improved: *[Split into as many requirements as necessary to be complete. Note that the types of customer information to be displayed needs to be defined in the glossary.]:*

The ATM shall display the Customer Account_Number in accordance with <Display Standard xyz>.
The ATM shall display the Customer Account_Balance in accordance with <Display Standard xyz>.
The ATM shall display the Customer Account_Type in accordance with <Display Standard xyz>.
The ATM shall display the Customer Account_Overdraft_Limit in accordance with <Display Standard xyz>.

[Note: Some may feel that a bulleted list or table is acceptable. While, from a readability perspective, this may be true, from a requirement management perspective, each item in the bulleted list is still a requirement. Because of this requirement statements as in the example above should be written as individual statements especially when allocation, traceability, verification, and validation activities are specific to the single item. See also R17, R18, R22, and R28.]

[Note the above examples are incomplete in that expected performance and under what conditions has not been included in the requirement statements.]

4.2 Concision

4.2.1 R10 – Superfluous Infinitives

Definition:
Avoid the use of superfluous infinitives such as “to be designed to”, “to be able to”, “to be capable of”, “to enable”, “to allow”.

Elaboration:

An **infinitive** is a verbal consisting of the word “to” + “a verb”.

Use of infinitives within requirement statements:

When writing a requirement statement, using more verbs than necessary to describe a single action (verb), such as “The <SOI> shall be designed to be able to <do an action> ...” or “The <SOI> shall be designed to be capable of <action>...” rather than simply “The <SOI> shall <action>...”.

By forcing conformance with one of the agreed structured statements (See R1), the use of such superfluous infinitives is avoided.

Note that at the enterprise and business levels, requirements for an entity to “provide a capability” are acceptable. Where capability is made up of people, processes, and products; these requirements will be decomposed to address the people aspects (skill set, training, roles, etc.), processes (procedures, work instructions, etc.); and products (hardware and software systems). The enterprise and business level higher-level requirements will be allocated appropriately to the people, processes, and products (SOI), as appropriate.

When the resulting requirement sets are defined for all three areas, the capability will be communicated by a set of requirements for each (people, process, product) that will result in the needed capability to be provided.

The use of "be able to" or "have the capability to" is also ambiguous in terms of defining the success criteria for system verification. For example, if the SOI is tested 100 times, and is only successful once, the case could be made that it is "able to" or is "capable of", but that isolated performance is probably not acceptable to the customer.

As another example, the SOI may be capable of a given function provided that the customer buys an additional capability. In this case the SOI could be designed to be expandable to allow additional capabilities to be added. Again, although the SOI is capable of a given feature if additional functionality is acquired, that is probably not what the customer desires.

If the intent of wording such as "be able to" or "be capable of" is to communicate the system will only need to do the required action based on some condition, trigger, or state, then the state the condition, trigger, or state must be included as part of the requirement (see R1 and R11).

Use of infinitives within need statements:

The use of "to be able to", "to have the capability to", "to enable the user to", or "to allow the user to" may be acceptable within a need statement. For example: "The <stakeholders > need the system "to provide the capability for users to" <do some action>."

Those transforming the need statements into requirements will define a feasible concept that will result in the system being able to provide a capability or allow or enable an action. As part of the transformation process from need to one or more requirements, the requirement writer will determine what the SOI is required to do to provide that capability or allow or enable the action.

When using "allow" or "enable' in a need statement it is important to understand how the terms are interpreted and the intent of the resulting actions. "Allow" can be interpreted as "to not prohibit something, to let it happen, to remove any constraint that would prevent something from happening" and "enable" can be interpreted as "to make something possible". Understanding this distinction is important when deriving the requirements that will meet the intent of the need.

Examples:

Unacceptable: The Weapon_System shall *be able to* store the location of each Ordnance.

[This is unacceptable because it contains the superfluous infinitive "be able to" which implies the requirement is ubiquitous in that it can be applied at any time, but it therefore begs the question as to the condition under which the feature can be invoked. However, this is acceptable for the stakeholder need: "The stakeholders need the Weapon_System to be able to store the location of each Ordnance."]

Improved: When <condition>he Weapon_System shall store the Location of each Ordnance.

[This is better because the requirement applies only in a particular condition. Note that the terms "Weapon_System", "Location", and "Ordnance" must be defined in the glossary or data dictionary. To be complete the appropriate performance characteristics would also need to be defined and included within the requirement statement.]

4.2.2 R11 – Separate Clauses

Definition:

Use a separate clause for each condition or qualification.

Elaboration:

Each need or requirement should have a main verb describing a basic function or need. If appropriate, the main sentence may then be supplemented by clauses that provide conditions or

qualifications (performance values, trigger, or constraints). A single, clearly identifiable clause should be used for each condition or qualification expressed.

As mentioned in R1 and Appendix C, a need and requirement should match one, and only one, pattern from the catalog of agreed patterns.

If an applicable qualifying clause or condition is not stated explicitly within the need or requirement statement, the need or requirement statement is not Complete (C4), Verifiable/Validatable (C7), nor Correct (C8).

If a qualifying clause is not stated explicitly within the need or requirement statement, the need or requirement statement is not Complete (C4)), Verifiable/Validatable (C7), nor Correct (C8)—for example, performance associated with the action verb or for an interface requirement where a pointer to where the specific interaction is defined (such as in the ICD).

When using clauses, make sure the clause does not separate the object of the sentence from the verb. See also R1, R18, R27 and Appendix C.

Examples:

Unacceptable: The Navigation_Beacon shall provide Augmentation_Data having the characteristics defined in <ICD xyz> at an accuracy of less than or equal to 20 meters to each Maritime_User during Harbor_Harbor_Approach_Maneuvering (HHAM).

[This is unacceptable because it inserts a phrase in such a way that the object of the sentence is separated from the verb.]

Improved: The Navigation_Beacon shall provide Augmentation_Data having the characteristics defined in <ICD xyz> to each Maritime_User engaged in Harbor_Harbor_Approach_Maneuvering (HHAM), at an accuracy of less than or equal to 20 meters.

[This rewrite places the basic function in an unbroken clause followed by the sub-clause describing performance. Note that: "Navigation_Beacon", "Maritime_User", and "Harbor_Harbor_Approach_Maneuvering (HHAM)" must be defined in the glossary or data dictionary. There is also an issue in that when discussing accuracy, precision needs to also be addressed.]

4.3 Non-ambiguity

4.3.1 R12 – Correct Grammar

Definition:

Use correct grammar.

Elaboration:

We interpret language based on the rules of grammar. Incorrect grammar leads to ambiguity and clouds understanding. This is especially true when the recipient of the need statement or requirement statement is working in a second language relying on specific rules of grammar. If these rules are not followed, that person may misinterpret the meaning of the need statement or requirement statement.

Incorrect use of grammar may make the true intent unclear resulting in an incorrect requirement and thus make it difficult to verify the SOI meets the intent of the requirement.

Care must be taken when translating need statements and requirement statements from one language to another and when sentence structure differs depending on the language in which the

original need or requirement statement was written. Punctuation varies from language to language and even between dialects of a given language.

Be cautious when need statements and requirement statements must be translated. An interesting exercise is to translate a requirement statement from one language to another and translate the result back to the original language.

Many word processing applications are capable of applying a set of grammar rules based on a set of built-in rules that are selectable. When using an application specific set of grammar rules, project-specific grammar items may be flagged as incorrect as they may not be applicable to the project's use of grammar. Therefore, the application grammar checking can be regarded as a project-specific aspect. Many applications allow the user to "turn off" or "ignore" grammar rules that do not apply to the project.

See also R4 - Define terms.

Examples:

Unacceptable: The Weapon_System shall *storing* the location of all ordnance.

[This is unacceptable because the grammatical error leads to uncertainty about the meaning.]

Improved: The Weapon_System shall store the location of all Ordnance.

[Note that "Ordnance" must be defined in the glossary to be explicit about the types of weapons and ammunition. Also, where the location is to be stored and the format of the data, must be addressed. The action "store" needs to be further evaluated to determine if that is an appropriate action (verb) for the Weapon_System vs a user interacting with the Weapon_System as discussed in R3. Additionally, to be complete any specific performance measures and conditions should be included within the requirement statement.]

Unacceptable: When in the Active_State, the Record_Subsystem shall display each of the Names of the Line_Items, without obscuring the User_ID per <Display Standard xyz>.

[This is unacceptable because the grammatical error involving the inappropriate placement of "each of"—it is most likely that a Line_Item has only one name.]

Improved: When in the Active_State, the Record_Subsystem shall display on <Display Device> the Name of each Line_Item, without obscuring the User_ID per <Display Standard XYZ.>.

[This is acceptable the ambiguity has been addressed. The requirement is now more complete in that it references where the information is to be display and the standard to be used concerning the display of the information. If there are any performance measures that apply, they would also need to be addressed.]

Unacceptable: The <corporate website> shall only use Approved_Fonts.

[This is unacceptable because it mandates that the website shall *only* use the designated fonts—that is, it is not expected to perform any other function except to use those fonts. This is clearly not what is meant but becomes ambiguous by the inappropriate grammar and the incorrect placement of the word "only". What is most likely meant is that the only fonts to be used are the approved fonts defined in the organization's display standard.]

Improved: The <corporate website> shall display information using Approved_Fonts defined in <Display Standard xyz>.