33rd Annual INCOSE international symposium

hybrid event

Honolulu, HI, USA
July 15 - 20, 2023

Dr. Tami Katz, INCOSE Requirements Working Group

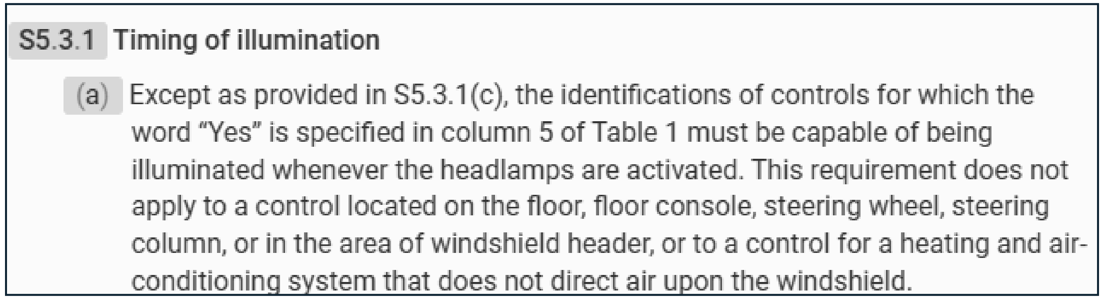
What is the Point of Requirements?

What is a Requirement?

- A Requirement is:
  - A statement which translates or expresses a need and its associated constraints and condition. [ISO/IEC/IEEE 15288, ISO/IEC/IEEE 29148]
  - A statement that identifies a system, product, or process characteristic or constraint, which is unambiguous, clear unique, consistent, stand-alone (not grouped), and verifiable, and is deemed necessary for stakeholder acceptability. [INCOSE SE Handbook v4]
  - The result of a formal transformation of one or more needs or parent requirements into an agreed-to-obligation for an entity to perform some function or possess some quality. [INCOSE SE Handbook v5]

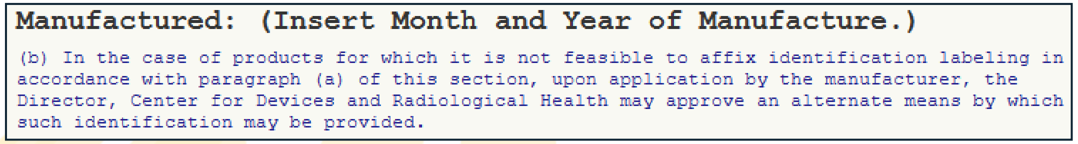
INTERNATIONAL STANDARD
ISO/IEC/ IEEE 15288
First edition 2015-05-15
ms and software engineering — m life cycle processes
INTERNATIONAL STANDARD
ISO/IEC/ IEEE 29148
Second edition 2018-11
Systems and software engineering — Life cycle processes — Requirements engineering
Ingénierie des systèmes et du logiciel — Processus du cycle de vie — Ingénierie des exigences
INCOSE
SYSTEMS ENGINEERING HANDB
A GUIDE FOR SYSTEM LIFE CYCLE PROCESSES AND ACTI
FOURTH EDITION
For INCOSE member, Corporate Advisory Board, and Academic Cou
Do not distribute.
INCOSE Systems Engineering HANDBOOK
SYSTEMS ENGINEERING HANDBOOK
A GUIDE FOR SYSTEM LIFE CYCLE PROCESSES AND ACTIVITIES
FIFTH EDITION
INCOSE International Council on Systems Engineering
WILEY

What Requirements Look Like



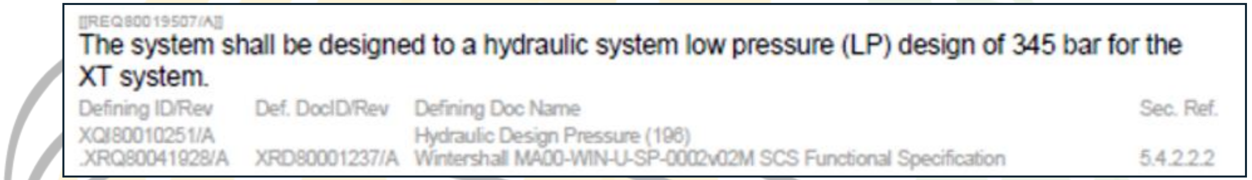
S5.3.1 Timing of illumination

(a) Except as provided in S5.3.1(c), the identifications of controls for which the word “Yes” is specified in column 5 of Table 1 must be capable of being illuminated whenever the headlamps are activated. This requirement does not apply to a control located on the floor, floor console, steering wheel, steering column, or in the area of windshield header, or to a control for a heating and air-conditioning system that does not direct air upon the windshield.



Manufactured: (Insert Month and Year of Manufacture.)

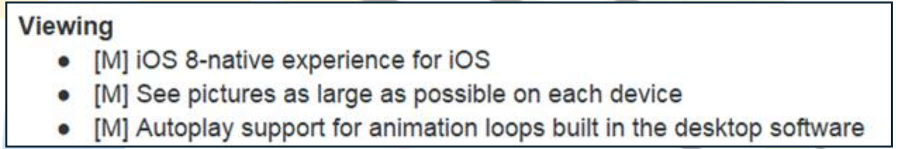
(b) In the case of products for which it is not feasible to affix identification labeling in accordance with paragraph (a) of this section, upon application by the manufacturer, the Director, Center for Devices and Radiological Health may approve an alternate means by which such identification may be provided.



[[REQ80019507/A]]

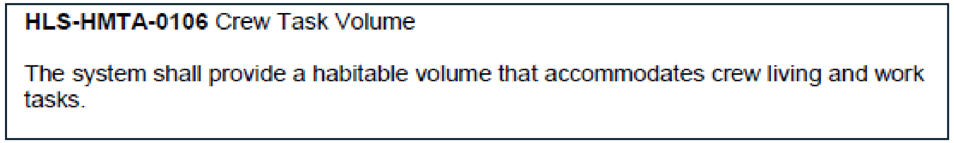
The system shall be designed to a hydraulic system low pressure (LP) design of 345 bar for the XT system.

Defining ID/Rev	Def. DocID/Rev	Defining Doc Name	Sec. Ref.
XQI80010251/A		Hydraulic Design Pressure (196)	
.XRQ80041928/A	XRD80001237/A	Wintershall MA00-WIN-U-SP-0002v02M SCS Functional Specification	5.4.2.2.2



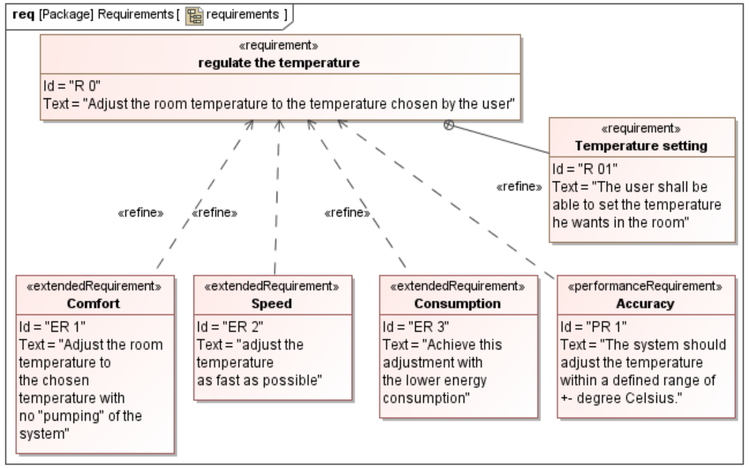
Viewing

- [M] iOS 8-native experience for iOS
- [M] See pictures as large as possible on each device
- [M] Autoplay support for animation loops built in the desktop software



HLS-HMTA-0106 Crew Task Volume

The system shall provide a habitable volume that accommodates crew living and work tasks.



«requirement»
regulate the temperature
Id = "R 0"
Text = "Adjust the room temperature to the temperature chosen by the user"

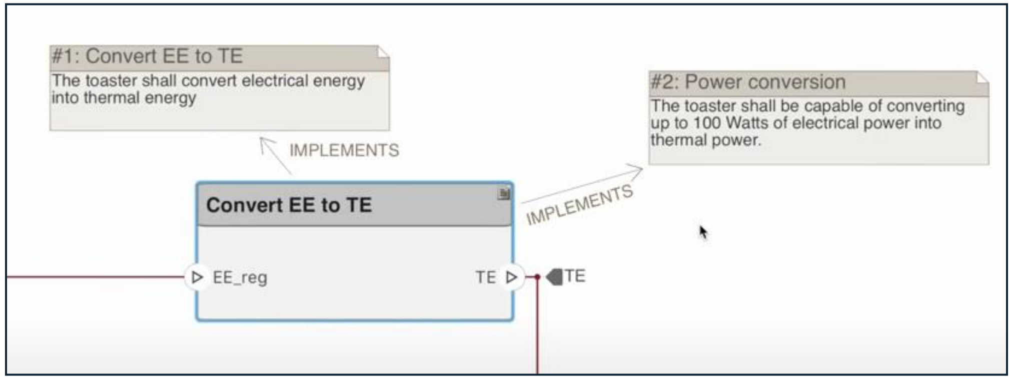
«requirement»
Temperature setting
Id = "R 01"
Text = "The user shall be able to set the temperature he wants in the room"

«extendedRequirement»
Comfort
Id = "ER 1"
Text = "Adjust the room temperature to the chosen temperature with no "pumping" of the system"

«extendedRequirement»
Speed
Id = "ER 2"
Text = "adjust the temperature as fast as possible"

«extendedRequirement»
Consumption
Id = "ER 3"
Text = "Achieve this adjustment with the lower energy consumption"

«performanceRequirement»
Accuracy
Id = "PR 1"
Text = "The system should adjust the temperature within a defined range of +- degree Celsius."

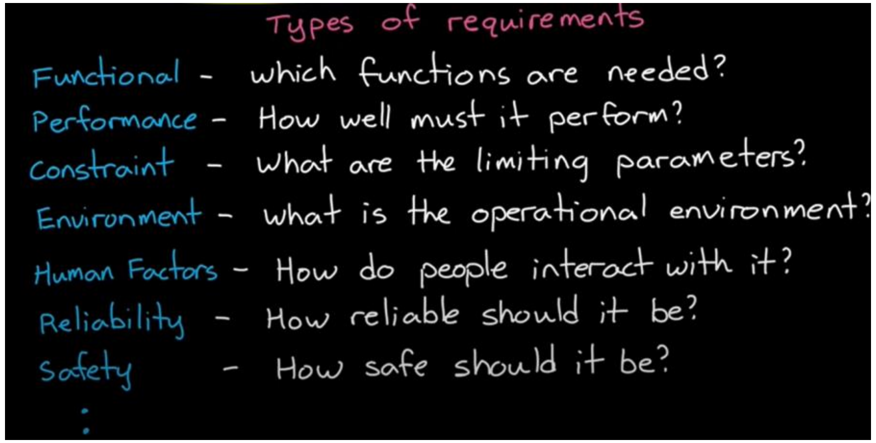


#1: Convert EE to TE
The toaster shall convert electrical energy into thermal energy

#2: Power conversion
The toaster shall be capable of converting up to 100 Watts of electrical power into thermal power.

Why Requirements Exist

- Requirements…
  - communicate to the system developers what the stakeholder is expecting.
  - support decision-making (criteria for trade of options, architecture decisions).
  - define how project success is determined.
  - document legal agreements between buyer/supplier.
  - ensure conformity to law (regulations).
  - ensure conformity to industry standards (quality).



Types of requirements

Functional - which functions are needed?

Performance - How well must it perform?

Constraint - what are the limiting parameters?

Environment - what is the operational environment?

Human Factors - How do people interact with it?

Reliability - How reliable should it be?

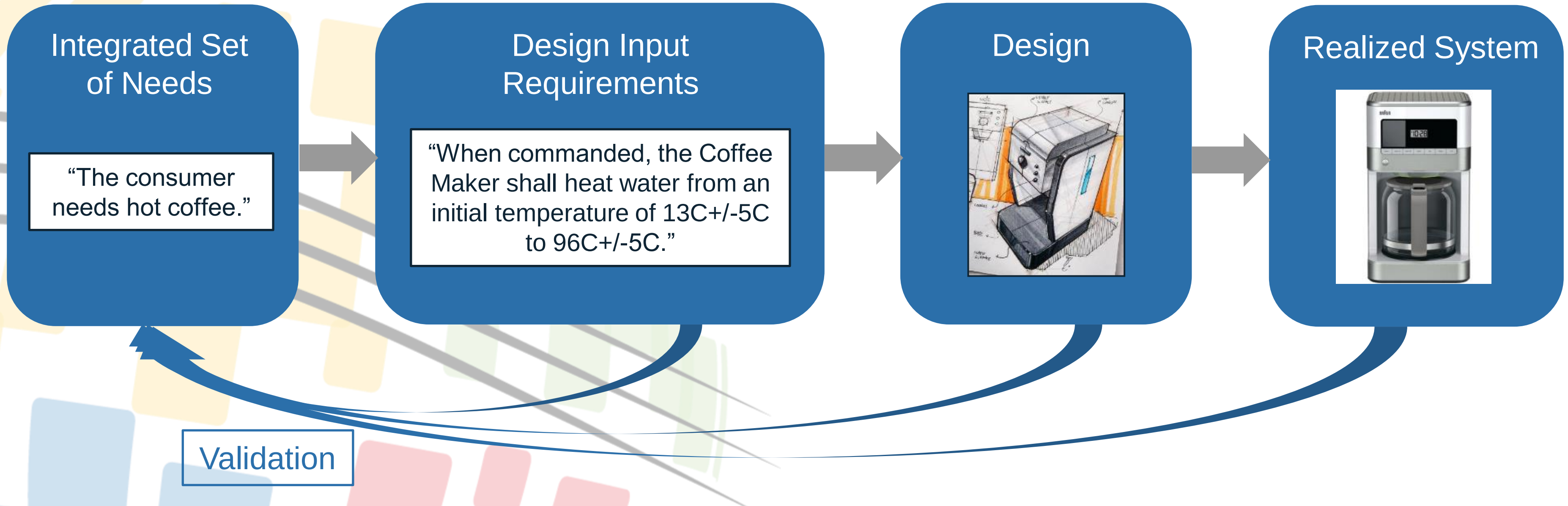
Safety - How safe should it be?

:

[(201) An Introduction to Requirements | Systems Engineering, Part 4 - YouTube]

Requirements Ensure the “Right” System is Developed

- Requirements are the mechanism to translate the stakeholder “needs” perspective to a perspective of what the system does to accomplish those needs; these are used as input to the design process.
- Poor requirements could result in a system that does not meet the stakeholder needs, resulting in the wrong system being developed (the system does not meet validation).
- Requirements are only as good as the Needs they are transformed from; capturing an “Integrated Set of Needs” identifies the stakeholder needs as well as the system lifecycle concepts (usage, maintenance, etc.), constraints, interfaces, risks, and business objectives to ensure all perspectives and use cases are considered at the starting point for the system development.



Integrated Set of Needs

“The consumer needs hot coffee.”

Design Input Requirements

“When commanded, the Coffee Maker shall heat water from an initial temperature of 13C+/-5C to 96C+/-5C.”

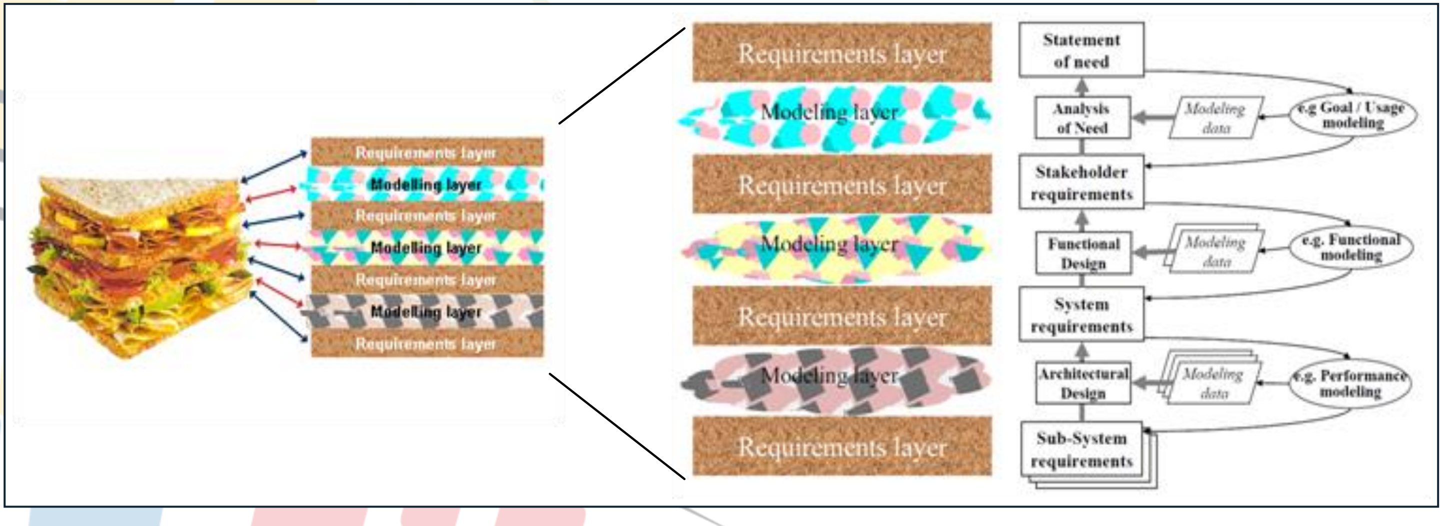
Design

Realized System

Validation

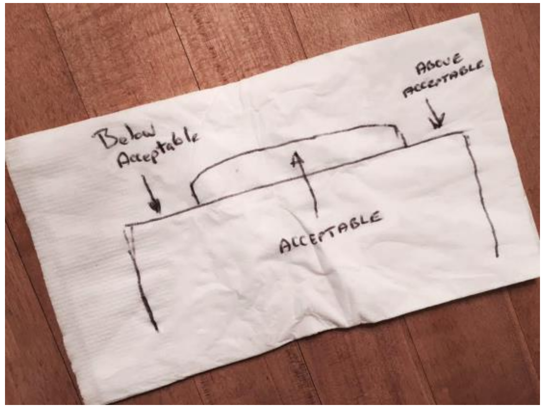
What comes first, Architecture or Requirements?

- This is an iterative and recursive process!
- As part of the architecture development process, a starting set of requirements are applied.
- As the architecture matures, continued analysis provides further definition of required capabilities and functions.
- This is referred to as the “Systems Engineering Sandwich” [Dick, 2017], where the modeling layer is the refinement of the architecture (elements, behaviors, performance and capabilities) and produces additional data, functions and features to support further requirement decomposition.



When are Requirements Not Needed?

- Risk and Complexity can determine when requirements may not be needed, such as:
  - A low risk development
  - Simple products
  - Small teams
  - Dictated solutions (fabrication effort only)
  - Pre-existing products (COTS)
  - Lack of law/regulations



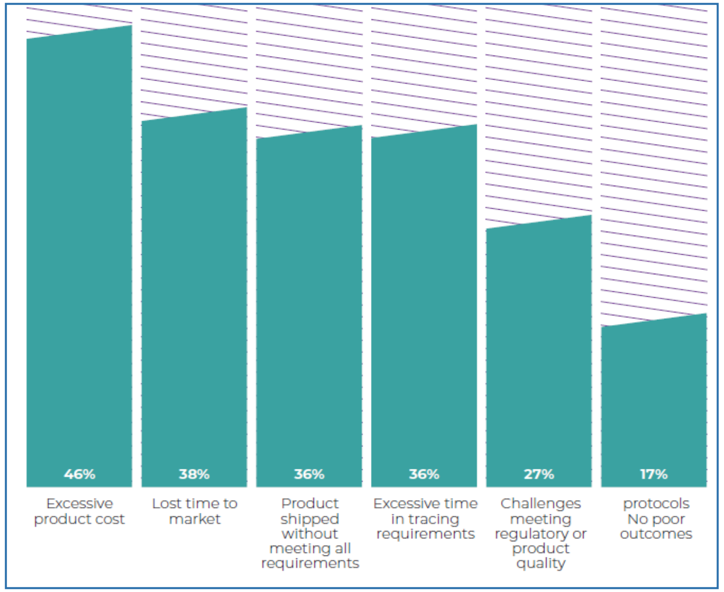
- Data has shown that for complex systems, there are costs associated with poor, or lack of, requirements (next slide).

Example Outcomes of Poor / No Requirements

- Complex projects which spend less than 5% of total project costs on the requirements engineering process experienced an **80% to 200% cost overrun**, whereas those that invested 8% to 14% were able to meet their costs or incur less than 60% overrun (Gruhl, 1992).

- A 2018 Engineering.com survey report noted **impacts to project success** based on poor requirements development and management (Engineering.com, 2018).

- Research into software safety found that most **accidents** related to software in the aviation industry stem from requirements problems, particularly related to incompleteness of the requirements (Howard & Anderson, 2002).



Types of Failures due to Poor Requirements Management. (Engineering.com, 2018)

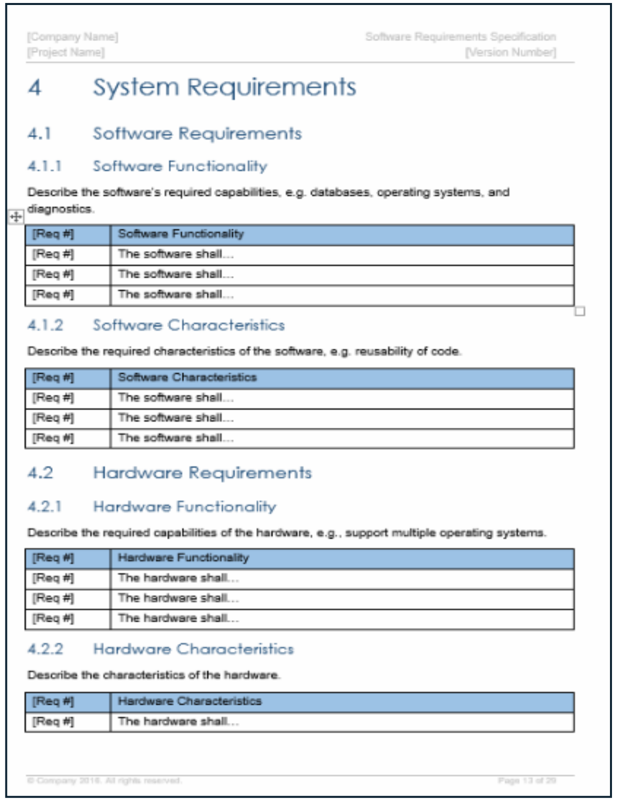
Why Are Requirements Difficult to Get Right?

- Based on system complexity, there could be hundreds (or thousands) of requirement statements used as input to the design and production efforts.
- Requirements are only effective if developed, communicated and managed correctly.
  - Poorly written, errors in derivation, conflicting, and challenges with management using difficult tools have all contributed to a negative results from requirements.
- There are multiple approaches, textbooks, methods, tools to guide the SE practitioners, it can be overwhelming!
- There is not one single method that works for all industries and applications, and if a method is misapplied it may result in poor outcomes.
- Examples of different methods are highlighted in the following slides to highlight various approaches and their limitations.

Requirements get a bad reputation when poor efforts are used – let's address the processes and methods!

Communicating Requirements (Documents)

- One method to capture and communicate a set of requirements is through a document or spreadsheet.
- Requirements in textual forms can use a structured, natural language following an agreed-to convention, and can be used to formalize agreements between buyers/suppliers (or any type of interfacing organizations).
- Attributes can be used to provide additional information on the requirements (identifier number, short title, trace to source, priority, verification method, etc.) [INCOSE NRM, 2022]
- Document-based methods have limited capability to capture trace of requirements to their source, to the architecture, to verification artifacts.
- Requirements could be generated and placed in a software-based requirements management tool, and then exported to a document form, enabling a more data-centric approach of managing the requirements while still using textual formats for communication.



[Company Name] Software Requirements Specification
[Project Name] [Version Number]

4 System Requirements

4.1 Software Requirements

4.1.1 Software Functionality

Describe the software's required capabilities, e.g. databases, operating systems, and diagnostics.

[Req #]	Software Functionality
[Req #]	The software shall…
[Req #]	The software shall…
[Req #]	The software shall…

4.1.2 Software Characteristics

Describe the required characteristics of the software, e.g. reusability of code.

[Req #]	Software Characteristics
[Req #]	The software shall…
[Req #]	The software shall…
[Req #]	The software shall…

4.2 Hardware Requirements

4.2.1 Hardware Functionality

Describe the required capabilities of the hardware, e.g., support multiple operating systems.

[Req #]	Hardware Functionality
[Req #]	The hardware shall…
[Req #]	The hardware shall…
[Req #]	The hardware shall…

4.2.2 Hardware Characteristics

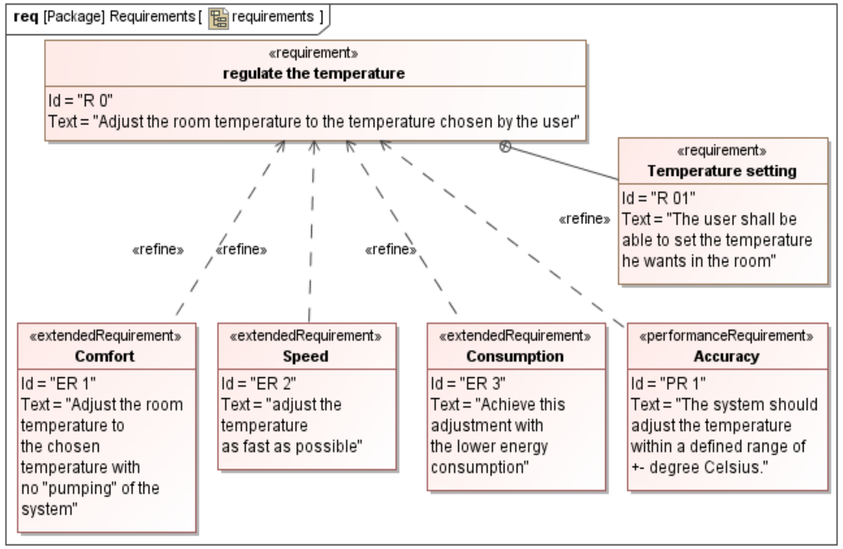
Describe the characteristics of the hardware.

[Req #]	Hardware Characteristics
[Req #]	The hardware shall…

© Company 2016. All rights reserved. Page 13 of 29

Communicating Requirements (Models)

- Model Based Systems Engineering (MBSE) enables the development and modeling of requirements, tracing them to use cases, stakeholder needs, architecture and functions.
- This data-centric approach enables the project to understand how the requirements connect to many aspects of the system, assessing completeness and impacts of any changes.
- Model-only requirements have limitations; diagrams, and models focus on functional, performance, and interfaces, and are not often suited to specify quality (-ilities), environments, regulations, standards, and physical characteristics.
- Depending on what is being developed and communicated, both textual requirements and modeled requirements could be used to maximize communication and trace of the requirements for the system.[INCOSE NRM, 2022]



req [Package] Requirements [requirements]

«requirement»
regulate the temperature
Id = "R 0"
Text = "Adjust the room temperature to the temperature chosen by the user"

«requirement»
Temperature setting
Id = "R 01"
Text = "The user shall be able to set the temperature he wants in the room"

«extendedRequirement»
Comfort
Id = "ER 1"
Text = "Adjust the room temperature to the chosen temperature with no "pumping" of the system"

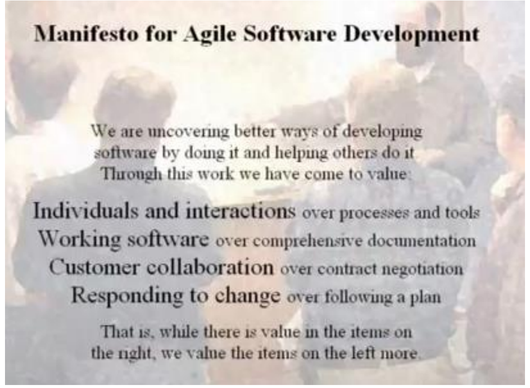
«extendedRequirement»
Speed
Id = "ER 2"
Text = "adjust the temperature as fast as possible"

«extendedRequirement»
Consumption
Id = "ER 3"
Text = "Achieve this adjustment with the lower energy consumption"

«performanceRequirement»
Accuracy
Id = "PR 1"
Text = "The system should adjust the temperature within a defined range of +- degree Celsius."

Requirements in Software Agile Development

- There are some who advocate that requirements have no place in a fast, Agile approach to software development.
- Agile development still utilizes the concept of requirements, but they may be communicated as user stories with acceptance criteria, versus a formal documented specification.
- Changes to the requirements are expected as the system matures and the organization establishes methods to enable quick requirement updates and associated configuration control.
- Formalization of these requirements may look different than on other system development, but the intent is still the same: Communicate what the system must do to accomplish the stakeholder needs.

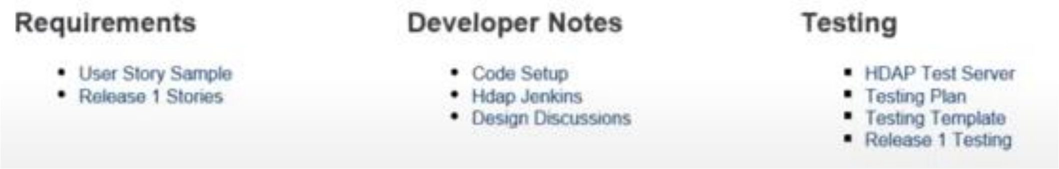


Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.



Requirements

- User Story Sample
- Release 1 Stories

Developer Notes

- Code Setup
- Hdap Jenkins
- Design Discussions

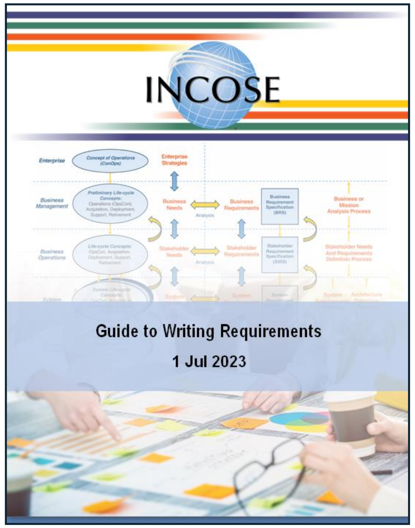
Testing

- HDAP Test Server
- Testing Plan
- Testing Template
- Release 1 Testing

[(201) Agile Requirements Documentation: Tips and Tricks for Modern Teams - YouTube]

How Important is Requirement Syntax?

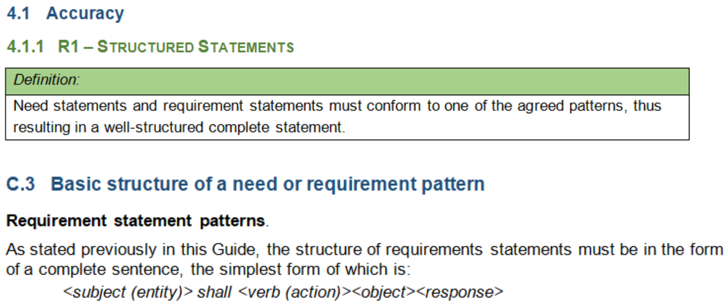
- Requirements need to be constructed in a way that they are Necessary, Appropriate, Singular, Correct, Conforming, Unambiguous, Complete, Feasible, and Verifiable. [INCOSE GtWR, 2022]
- The INCOSE Guide to Writing Requirements (GtWR) provides guidance to construct textual need and requirement statements, offering characteristics and rules.
  - The GtWR syntax has been incorporated into requirement management tools for quality check of the requirement phrasing, to help uncover requirements which do not meet the characteristics noted above.
- This syntax may not apply to all situations!
  - Based on industry conventions, other syntax approaches may be more applicable (use of “must”, model parameters, tabular parameter and value, etc.)
- It is important that the team define terminology to be used and incorporate approaches which follow conventions that are agreed upon by the organizations involved.



INCOSE

Guide to Writing Requirements

1 Jul 2023



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.

C.3 Basic structure of a need or requirement pattern

Requirement statement patterns.

As stated previously in this Guide, the structure of requirements statements must be in the form of a complete sentence, the simplest form of which is:

<subject (entity)> shall <verb (action)><object><response>

Bottom line: Use an agreed-to set of conventions that clearly communicate the requirements between giver/receiver organizations.

Requirement Best Practices

- The Requirements are based on an Integrated Set of Needs, which are determined from an analysis of stakeholders, lifecycle concepts, constraints, business goals and risks.

- Do not just copy paste requirements from a similar project, perform the due diligence to ensure the Needs are fully understood before transforming them into the correct requirements for the system of interest.

- Based on the industry, there are different conventions of requirements communication and management – it is best to investigate what works well for the type of system being developed; a consumer product has different approaches than a space system!

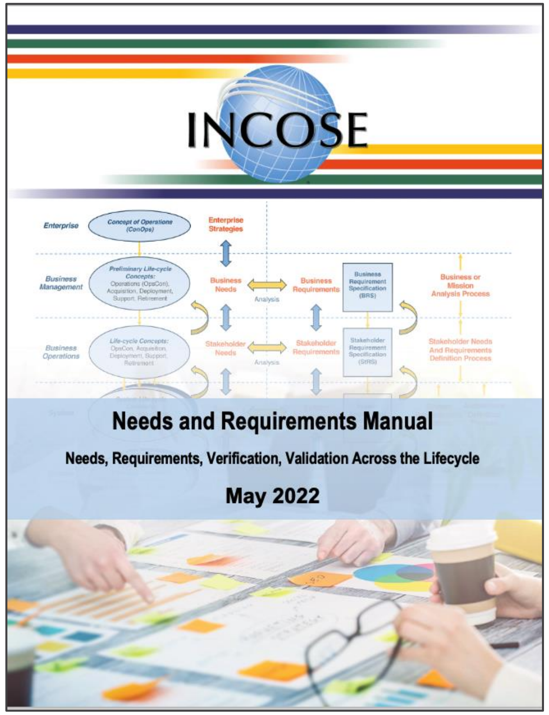
- For best results, the Needs and Requirements should be Necessary, Appropriate, Singular, Correct, Conforming, Unambiguous, Complete, Feasible, and Verifiable.

- Attributes can be used to supplement requirement statements and add additional information.

- Data-centric approaches are valuable for needs / requirements development (and management) to enable trace to other parts of the system and project records.

Reference the Guide to Writing Requirements v4 for recommended approaches.

Where to Learn More!


INCOSE
Enterprise
Concept of Operations (ConOps)
Enterprise Strategies
Business Management
Preliminary Life-cycle Concepts: Operations (OpsCon), Acquisition, Deployment, Support, Retirement
Business Needs
Analysis
Business Requirements
Business Requirement Specification (BRS)
Business or Mission Analysis Process
Business Operations
Life-cycle Concepts: OpsCon, Acquisition, Deployment, Support, Retirement
Stakeholder Needs
Analysis
Stakeholder Requirements
Stakeholder Requirement Specification (StRS)
Stakeholder Needs And Requirements Definition Process

Needs and Requirements Manual

Needs, Requirements, Verification, Validation Across the Lifecycle

May 2022

- INCOSE Requirements Working Group (RWG)
  - INCOSE RWG YouTube
  - Technical Products
    - Needs and Requirements Manual
    - Guide to Writing Requirements (v4 just released!)
    - Guide to Needs and Requirements
    - Guide to Verification and Validation
    - Available for free download from INCOSE Store for INCOSE members
- Open RWG Meeting: Wednesday, July 19, 1:30 pm, room 305A

https://www.incose.org/incose-member-resources/working-groups/process/requirements

References

- Automotive Safety Regulations: www.ecfr.gov
- Baker, J. (2020). Sparx Systems - Extending the Requirement Type. Retrieved from https://www.eaglobalsummit.com/sessions/extending-the-requirement-type/
- Dick, J., Hull, E., & Jackson, K. (2017). Requirements Engineering (Fourth Edition ed.). Springer;
- Engineering.com. (2018). Engineering.com. Retrieved 10 22, 2020, from Research Report: Design Teams: Requirements Management and Product Complexity: https://www.engineering.com/ResourceMain?resid=873
- FDA Regulations: www.accessdata.fda.gov
- Gruhl, W. (1992). Lessons Learned, Cost/Schedule Assessment. NASA.
- Helle, K., Engen, S., & Falk, K. (2020). Towards Systemic Handling of Requirements in the Oil and Gas Industry – a Case Study. INCOSE IS2020. INCOSE.
- Howard, J., & Anderson, P. (2002). The Safety Risk of Requirements Incompleteness. In S. Engineering (Ed.), ISSC.
- IEEE. (2019). IEEE 15288. Systems and Software Engineering - System Life Cycle Processes .
- IEEE. (2018). IEEE 29148. Systems and Software Engineering Life Cycle Processes - Requirements Engineering . IEEE.
- INCOSE. (2015). INCOSE Systems Engineering Handbook (INCOSE-TP-2003-002-04). Wiley.
- INCOSE. (2022). INCOSE Needs and Requirements Manual. INCOSE. INCOSE. (2022). INCOSE Needs and Requirements Manual. INCOSE.
- INCOSE. (2023). INCOSE Guide to Writing Requirements. INCOSE.
- MATLAB. An Introduction to Requirements. https://www.youtube.com/watch?v=Iblo2Il-pOA
- Product Requirements Template: https://productschool.com/blog/product-management-2/product-template-requirements-document-prd/?utm_source=template&utm_medium=Template_Doc_PRD&utm_campaign=ProductManagementTemplates

33rd Annual INCOSE
international symposium

hybrid event

Honolulu, HI, USA
July 15 - 20, 2023

www.incose.org/symp2023