

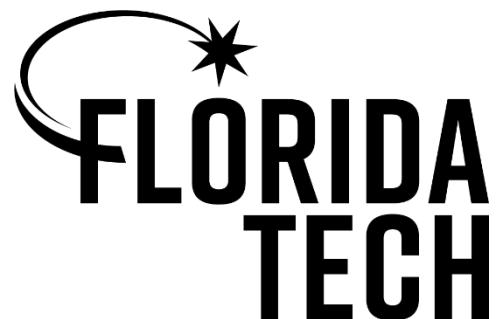
SYS 5460: Functional (cont.) & Non-Functional Requirements

Contents

- Overview of the Engineering Design Process
- Functional and Non-functional Requirement Formal Definitions
- Examples
 - Reliability
 - Response rate

Department of Computer & Engineering Sciences

College of Engineering
Florida Institute of Technology



Using Requirements Templates or Boiler Plates

- [when?] [under what conditions?] THE SYSTEM SHALL | SHOULD | WILL <process> <thing to be processed>[<process detail>*]

[Condition] [Subject] [Action] [Object] [Constraint]

EXAMPLE: When signal x is received [Condition], the system [Subject] shall set [Action] the signal x received bit [Object] within 2 seconds [Constraint].

Or

[Condition] [Action or Constraint] [Value]

EXAMPLE: At sea state 1 [Condition], the Radar System shall detect targets at ranges out to [Action or Constraint] 100 nautical miles [Value].

Or

[Subject] [Action] [Value]

EXAMPLE: The Invoice System [Subject], shall display pending customer invoices [Action] in ascending order [Value] in which invoices are to be paid.

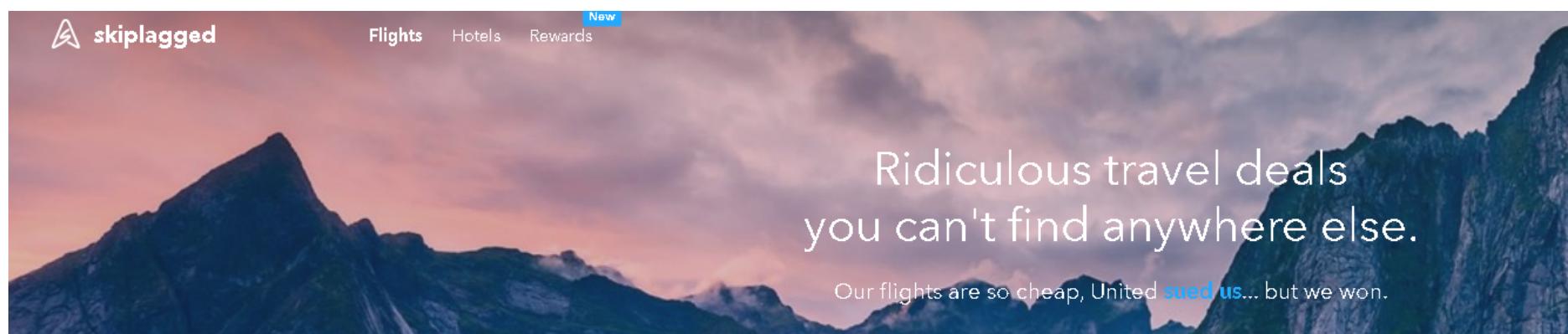
■ For Stakeholder capabilities

The <stakeholder type> shall be able to <capability> within <performance> of <event> while <operational condition>.

The <stakeholder> shall not be placed in breach of <applicable law>.

The weapons operator shall be able to fire a missile within 3 seconds of radar sighting while in severe sea conditions.

E.g. The ambulance driver shall not be placed in breach of national road regulations.



Find flights the airlines don't want you to see.
We're exposing **loopholes** in airfare pricing to save you money.

■ For System capabilities

The **<system>** shall **<function>**
not less than **<quantity>** **<object>**
while **<operational condition>**.

*E.g. The communications system shall sustain telephone contact
with not less than 10 callers
while in the absence of external power.*

The **<system>** shall **<function>** **<object>**
every **<performance>** **<units>**.

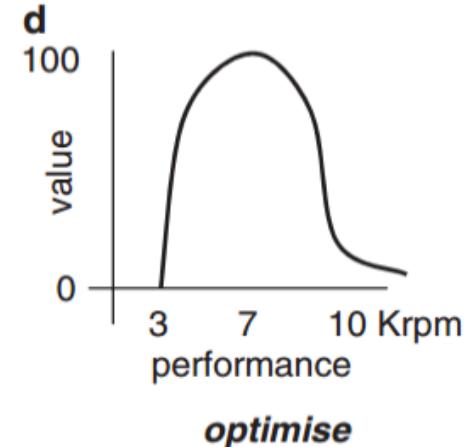
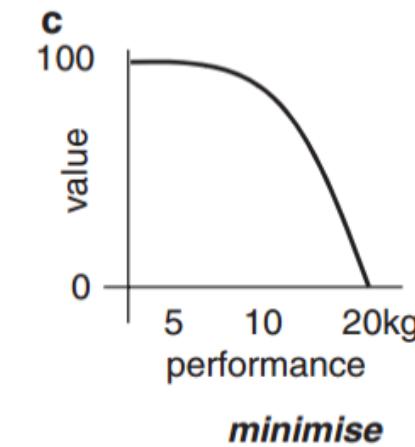
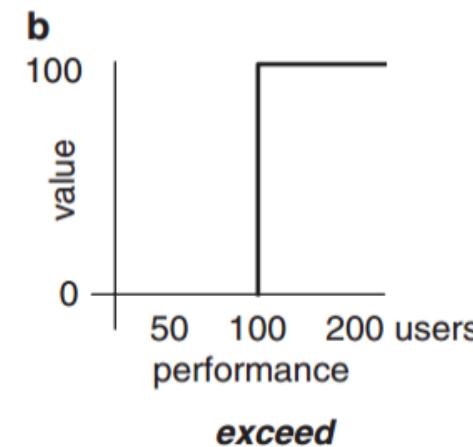
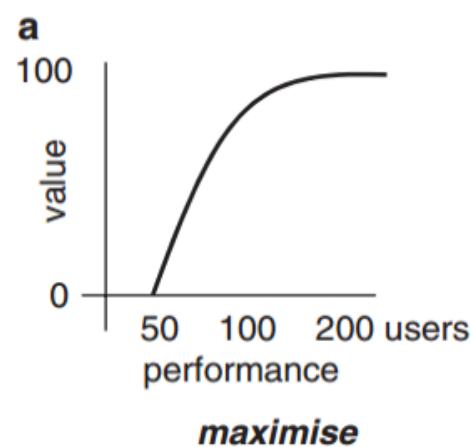
*E.g. The coffee machine shall produce a hot drink
every 10 seconds.*

Examples (Hull Ch 4)

Type of constraint	Boiler-plate
Performance/ capability	The <system> shall be able to <function> <object> not less than <performance> times per <units>.
Performance/ capability	The <system> shall be able to <function> <object> of type <qualification> within <performance> <units>.
Performance/ capacity	The <system> shall be able to <function> not less than <quantity> <object>.
Performance/ timeliness	The <system> shall be able to <function> <object> within <performance> <units> from <event>.
Performance/ periodicity	The <system> shall be able to <function> not less than <quantity> <object> within <performance> <units>.
Interoperability/ capacity	The <system> shall be able to <function> <object> composed of not less than <performance> <units> with <external entity>.
Sustainability/ periodicity	The <system> shall be able to <function> <object> for <performance> <units> every <performance> <units>.
Environmental/ operability	The <system> shall be able to <function> <object> while <operational condition>.

Requirements Value

- Functional requirements are non-negotiable. If they are not met , the product is of no use
- Other requirements may be negotiable
 - One approach is to provide performance values (Mandatory, Desired, Best)
 - Another proposed approach is to represent the value of a requirement value by supplying a function that maps performance to value



Technical Processes in the Design Stage

Technical process	Purpose
1. Stakeholder requirements definition	“Define the requirements for a system that can provide the services needed by users and other stakeholders in a defined environment” IEEE and ISO/IEC (2008, p. 36)
2. Requirements analysis	“Transform the stakeholder, requirement-driven view of desired services into a technical view of a required product that could deliver those services” IEEE and ISO/IEC (2008, p. 39)
3. Architectural design	“Synthesize a solution that satisfies system requirements” IEEE and ISO/IEC (2008, p. 40)
4. Implementation	“Realize a specified system element” IEEE and ISO/IEC (2008, p. 43)
5. Integration	“Assemble a system that is consistent with the architectural design” IEEE and ISO/IEC (2008, p. 44)
6. Verification	“Confirm that the specified design requirements are fulfilled by the system” IEEE and ISO/IEC (2008, p. 45)
7. Transition	“Establish a capability to provide services specified by stakeholder requirements in the operational environment” IEEE and ISO/IEC (2008, p. 46)
8. Validation	“Provide objective evidence that the services provided by a system when in use comply with stakeholders’ requirements, achieving its intended use in its intended operational environment” IEEE and ISO/IEC (2008, p. 47)

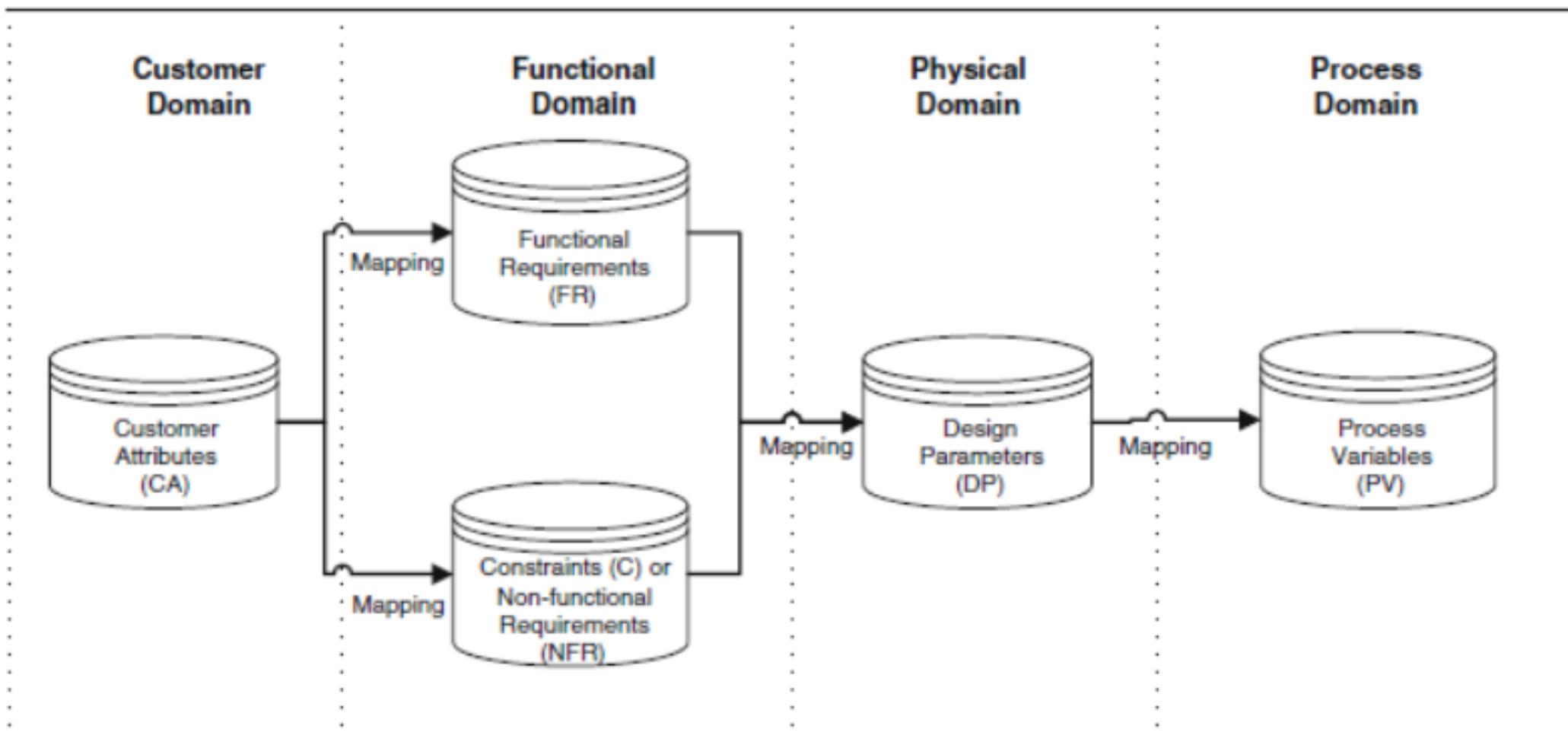
Functional Requirement

- A statement that identifies what a product or process must accomplish to produce required behavior and/or results.
- A requirement that specifies a function that a system or system component must be able to perform. (IEEE and ISO/IEC 2010, p. 153)
- **Functional** requirements have the following essential characteristics:
 - Define what the system should do
 - Be action oriented
 - Describe tasks or activities
 - Associated with the transformation of inputs to outputs

Non-Functional Requirements

- A software requirement that describes not what the software will do but how the software will do it (IEEE and ISO/IEC 2010, p. 231)
- Properties or qualities the product must have to facilitate its functionality (Robertson and Robertson 2005, p. 146)
- Non-Functional requirements have the following essential characteristics:
 - Define a property or quality that the system should have
 - Can be subjective, relative, and interacting
 - Describe how well the systems must operate
 - Are associated with the entire system

Formal Process of Engineering Design (DSM)



Stakeholder Requirements

Requirements Analysis

System Architecture

Production of the Solution

Axiomatic Design Methodology

- Independence Axiom: each functional requirement should be satisfied **without affecting any other functional requirement**.
- Information Axiom: The system design that requires the **least amount of information** to fulfill the design parameters is preferred.

$$a_{ij} = 1 \text{ if } FR\ i \text{ is satisfied by } DP\ j$$

Equation for functional requirements

$$[FR] = [A][DP]$$



FR: Functional requirement

A: Design matrix

DP: Design parameters

$$[A] = \begin{vmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{vmatrix}$$

$$FR_1 = A_{11}DP_1 + A_{12}DP_2 + A_{13}DP_3$$

$$FR_2 = A_{21}DP_1 + A_{22}DP_2 + A_{23}DP_3$$

$$FR_3 = A_{31}DP_1 + A_{32}DP_2 + A_{33}DP_3$$

What type of matrix is A to satisfy axiom 1

Examples

Canon TS9120 Wireless All-In-One Printer with Scanner and Copier: Mobile and Tablet Printing, with Airprint(TM) and Google Cloud Print compatible, Gray

by Canon

★★★★★ 530 customer reviews | 445 answered questions

List Price: \$199.99

Price: \$59.99 prime

You Save: \$140.00 (70%)

Get \$70 off instantly: Pay \$0.00 upon approval for the Amazon Prime Rewards Visa Card.

Color: Gray



Style: Printer

Printer **Printer and Ink Bundle** **Printer, Ink and Paper Bundle**

- Inspire your creativity with prints that will impress. From stunning photographs to detailed documents, put the 6-color individual ink system to work and never compromise on speed or quality.
- Enjoy the simplicity of connecting your smartphone, tablet and all your favorite devices with ease. Print hassle free - whether from the cloud, through Bluetooth, from social media or even on the go.
- The PIXMA TS9120 Wireless printer is the all-in-one that fits perfectly anywhere in the home, and looks great with any decor thanks to its two-tone design and multiple color options.
- With intuitive features like the 5.0" LCD touchscreen and enhanced user interface, Bluetooth printing, & document removal reminder, it's clear to see why the PIXMA TS9120 makes printing a breeze.



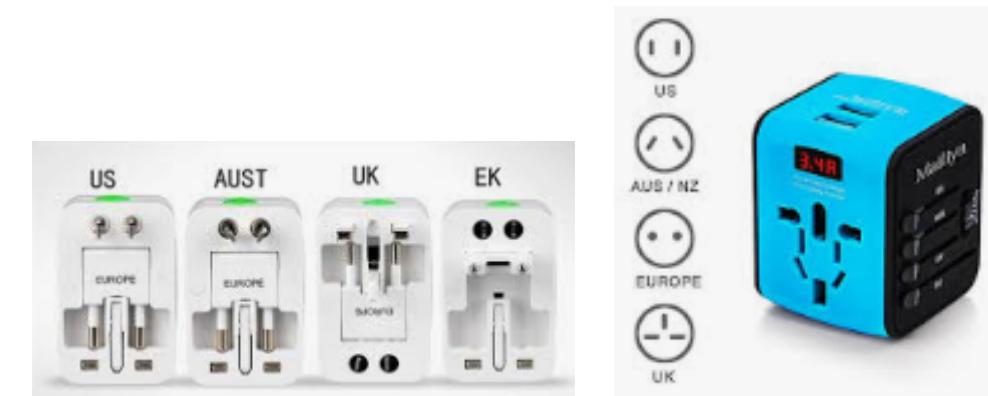
Re-Caulk, the Easy Way

Allway's new 3-in-1 Caulk Tool helps remove and re-apply Caulk Quickly and neatly. Stainless Steel Caulk Remover pulls out the old Caulk bead with either a push or pull motion.

ANYONE CAN DO IT. IT'S EASY.

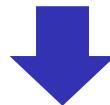
Triangular rubber blade is adjustable. There is a choice of 3 rounded tips to provide the desired profile to smooth the freshly applied Caulk bead, neatly around the tub, shower, sink, window, or exterior trim.

Allway Tools
Better by Design



Independence and Design Complexity

- The independence axiom may be used to evaluate design complexity
- Systems exhibit complexity as a result of excessive interaction between components within the system design
- If the number of DPs is less than the number of FRs:
 - The design has added complexity because DPs are satisfying more than one FR
 - The FR has not been satisfied



Your system requirement tool shall assist you with this

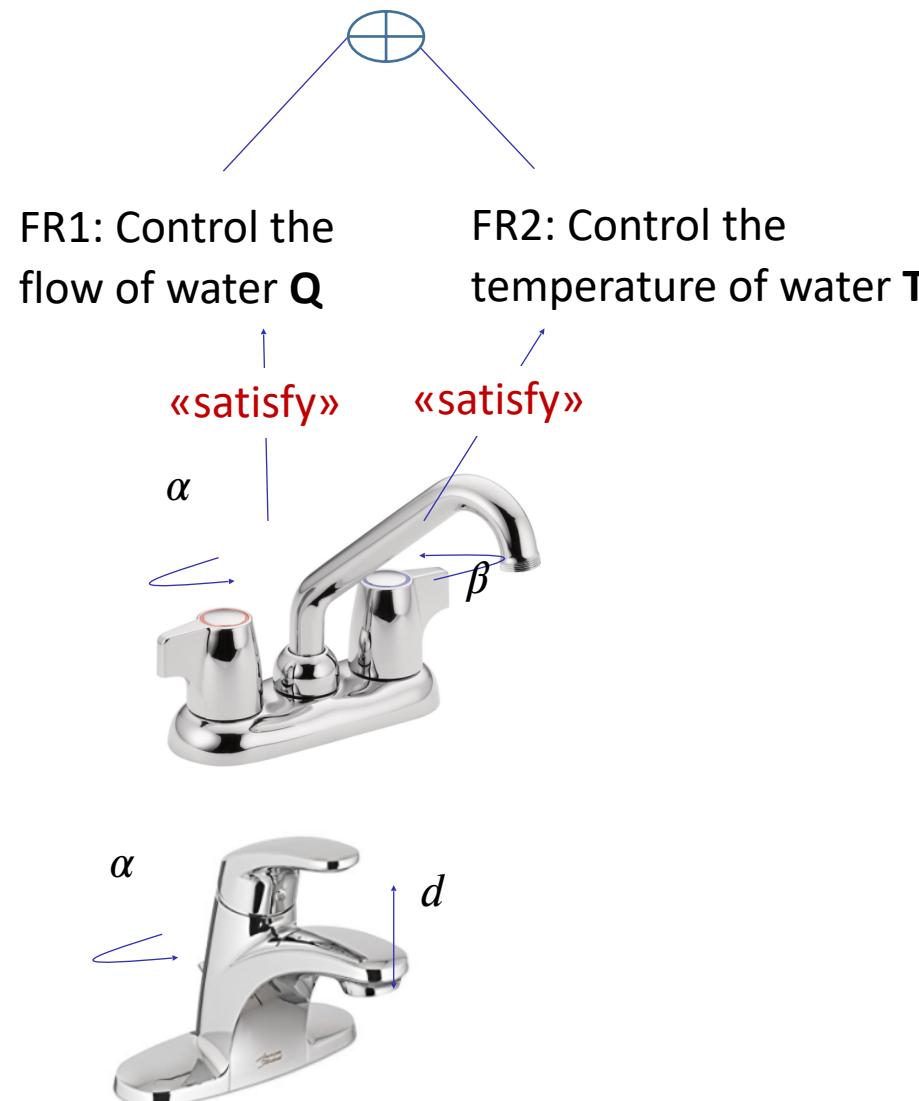


Relationships are build in the system using a requirements tool or MBSE tool

Relationship Name	Keyword	Source-> Target Interpretation
Satisfy	«satisfy»	«model element» satisfy «requirement»
Verify	«verify»	«test case» verify «requirement»
Refine	«refine»	«model element» refine «requirement» «requirement» refine «model element»
Derive Requirement	«deriveRqt»	«requirement» derived from «requirement»
Copy	«copy»	«requirement» copied from «requirement»
Trace	«Trace»	«requirement» traced from «requirement»
Containment	(Crosshair icon)	«requirement» is contained in «requirement»

Example: Water Faucet

The user shall be able to control the temperature and flow of water



Requirements and System Architecting

- Requirements should not anticipate the solution
- Requirements describe the what
- Architecture describes the how
- Solution-free requirements (arguably) may not be feasible in practice
- Every time we satisfy a requirement with an element we are deriving logical architectural components
- Each new component leads to a new requirements that are related to aspects of the solution
- Example:
 - The system shall transport 2 people from point A to point B
 - Solution could be a ground transport (say car->constraint)
 - Then we need to determine requirements specific to car
- The Architecture that lie in the background of the solution is the base system architecture

Non-Functional Requirements (Robertson Ch11)

- Non-functional requirements are properties that the **functionality** must have
- Non-functional requirements do not alter the product's essential functionality
 - Look and Feel: the spirit of the product's appearance
 - Usability and Humanity: the product's ease of use,
 - Performance: how fast, how safe, how many, how available, and how accurate the functionality must be
 - Operational: the operating environment of the product,
 - Maintainability and Support: expected changes, and the time needed to make them;
 - Security: access, confidentiality, recoverability, and auditability of the product
 - Cultural and Political: special requirements that come about because of the culture and customs of people who can come in contact with the product
 - Legal: the laws and standards that apply to the product

Look and Feel (Req Type 10)

- Appear to be simple to use
- Approachable, so people do not hesitate to use it
- Authoritative, user feel that they can rely on it and trust it
- Conforming to the client's other products
- Attractive to children
- Unobtrusive , so that people are not aware of it
- Innovative and appearing to be state of the art
- Professional looking
- Exciting
- Cool

Stakeholder Requirement:

The product shall be comply with windows user experience interactions guidelines for windows 8

Performance (Req Type 12)

- Speed to complete a task
- Accuracy of the results
- Safety of the operator
- Volumes of data to be stored
- Ranges of allowable values
- Throughput , rate of transactions
- Efficiency of resource usage
- Reliability (MTBF)
- Availability (uptime)
- Fault tolerance
- Scalability

Stakeholder Requirement:

The product shall schedule de-icing activities , so that the minimum necessary amounts of de-icing material are spread on roads

Operational and Environmental (Req Type 13)

- Operating environment (water, land, air ,space)
- Operating conditions (temp, pressure, wind speeds, tide severity, illumination)
- Partner of collaborating systems
- Threatening systems

Stakeholder Requirement:

The product shall survive being dropped from shoulder height

Maintainability and Support (Req Type 14)

- Expected changes in the product in the following areas:

- Organization
- Environment
- Laws that apply to the product
- Business rules

Stakeholder Requirement:

The product shall be translated into various foreign languages. As of yet the languages are unknown

Security (Req Type 15)

- Access: Data and functionality accessible to authorized users. Deny access to unauthorized people
- Privacy: Data is protected from unauthorized or accidental disclosure
- Integrity: Product data is the same as the source, or authority of the data. Protected from corruption

Stakeholder Requirement:

The product shall ensure that only authorized users have access to the products data

The product shall ensure its road temperature data corresponds to the data transmitted by the weather station

Cultural (Req Type 16)

- Avoid use of symbols associated with religion, politics , etc.
- Incorporate widely acceptable phrases when using foreign languages (e.g. Spanish, French, English)

Stakeholder Requirement:

The product shall not use any terms or icons that might possible offend anyone on the planet

Legal (Req Type 17)

- Examine adjacent systems or actors
- Consider legal rights or applicable laws:
 - Privacy protection
 - Non-disclosure
 - Consumer protections
 - Warranties

Stakeholder Requirement:

The product shall comply with the Americans with Disabilities Act of 1990 as amended

Non-Functional Requirement and System “ilities”

- There are over 200 recognized system *ilities* in systems engineering
- List compiled by Adams(2015)

Accessibility	Degradation of service	Modularity	Security
Accountability	Dependability	Naturalness	Sensitivity
Accuracy	Development cost	Nomadicity	Similarity
Adaptability	Development time	Observability	Simplicity
Additivity	Distributivity	Off-peak period performance	Software cost
Adjustability	Diversity	Operability	Software production time
Affordability	Domain analysis cost	Operating cost	Space boundedness
Agility	Domain analysis time	Peak-period performance	Space performance
Auditability	Efficiency	Performability	Specificity
Availability	Elasticity	Performance	Stability
Buffer space performance	Enhanceability	Planning cost	Standardizability
Capability	Evolvability	Planning time	Subjectivity
Capacity	Execution cost	Plasticity	Supportability
Clarity	Extensibility	Portability	Surety
Code-space performance	External consistency	Precision	Survivability
Cohesiveness	Fault-tolerance	Predictability	Susceptibility
Commonality	Feasibility	Process management time	Sustainability

68

Sample of Non-Functional Requirements (1)

Communication cost	Flexibility	Productivity	Testability
Communication time	Formality	Project stability	Testing time
Compatibility	Generality	Project tracking cost	Throughput
Completeness	Guidance	Promptness	Time performance
Component integration cost	Hardware cost	Prototyping cost	Timeliness
Component integration time	Impact analyzability	Prototyping time	Tolerance
Composability	Independence	Reconfigurability	Traceability
Comprehensibility	Informativeness	Recoverability	Trainability
Conceptuality	Inspection cost	Recovery	Transferability
Conciseness	Inspection time	Reengineering cost	Transparency
Confidentiality	Integrity	Reliability	Understandability
Configurability	Inter-operability	Repeatability	Uniform performance
Consistency	Internal consistency	Replaceability	Uniformity
Controllability	Intuitiveness	Replicability	Usability
Coordination cost	Learnability	Response time	User-friendliness
Coordination time		Responsiveness	Validity

64

Cumulative
total:132

Sample of Non-Functional Requirements (2)

	Main-memory performance		
Correctness	Maintainability	Retirement cost	Variability
Cost	Maintenance cost	Reusability	Verifiability
Coupling	Maintenance time	Risk analysis cost	Versatility
Customer evaluation time	Maturity	Risk analysis time	Visibility
Customer loyalty	Mean performance	Robustness	Wrappability
Customizability	Measurability	Safety	
Data-space performance	Mobility	Scalability	
Decomposability	Modifiability	Secondary-storage performance	

33

Cumulative
total:165

Sample of Non-Functional Requirements (3)

Analyzability	Demonstrability	Manageability
Anonymity	Durability	Performance
Atomicity	Effectiveness	Privacy
Attractiveness	Expandability	Provability
Augmentability	Expressiveness	Quality of service
Certainty	Extendability	Readability
Changeability	Functionality	Self-descriptiveness
Communicativeness	Immunity	Structuredness
Complexity	Installability	Suitability
Comprehensiveness	Integratability	Tailorability
Conformance	Legibility	Trustability
Debuggability	Likeability	Viability
Defensibility	Localizability	

Degradability	Heterogeneity	Reproducibility
Deployability	Homogeneity	Resilience
Determinability	Interchangeability	Securability
Disposability	Manufacturability	Serviceability
Distributability	Producability	Ubiquity
Expandability	Repairability	
Fidelity	Repeatability	

58

Cumulative
total:223

Example: Temperature



Moderna Announces Longer Shelf Life for its COVID-19 Vaccine Candidate at Standard Refrigerator Temperatures

November 16, 2020

Vaccine candidate now expected to remain stable at standard refrigerator temperatures of 2° to 8°C (36° to 46°F) for 30 days, up from previous estimate of 7 days

Shipping and long-term storage conditions at standard freezer temperatures of -20°C (-4°F) for 6 months

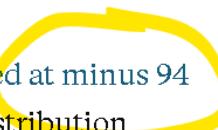
mRNA-1273 to be distributed using widely available vaccine delivery and storage infrastructure

No dilution required prior to vaccination

CAMBRIDGE, Mass.--(BUSINESS WIRE)--Nov. 16, 2020-- [Moderna, Inc.](#) (Nasdaq: MRNA), a biotechnology company pioneering messenger RNA (mRNA) therapeutics and vaccines to create a new generation of transformative medicines for patients, today announced new data showing that mRNA-1273, its COVID-19 vaccine candidate, remains stable at 2° to 8°C (36° to 46°F), the temperature of a standard home or medical refrigerator, for 30 days. Stability testing supports this extension from an earlier estimate of 7 days. mRNA-1273 remains stable at -20° C (-4°F) for up to six months, at refrigerated conditions for up to 30 days and at room temperature for up to 12 hours.

Pfizer has launched a pilot in Rhode Island, Texas, New Mexico, and Tennessee to test the distribution of its COVID-19 vaccine, which has to be kept at temperatures well below freezing.

Pfizer's vaccine candidate has to be shipped and stored at minus 94 degrees Fahrenheit (minus 70 degrees Celsius). Its distribution therefore relies on a "cold chain," a multipart pipeline that keeps the shots chilled, from manufacturer to injection.



PROJECT DELIVERABLES FOR NEXT WEEK

- Goals and project strategy
- Stakeholders
- Use Cases (Nominal or Off nominal scenarios)
- Requirements Tables:
 - Minimum of 15 requirements
 - Divided into functional and NF
- CONOPS (Describe the system)
- Verification and Validation
 - Verification Methods for 10 Requirements
 - Validation Methods for 5 Requirements
- 1 (poster file) Slide Marketing Material/ Display or Pitch
 - Font 8-10

References

- Chung, L., Nixon, B. A., Yu, E. S., & Mylopoulos, J. (2000). *Non-functional requirements in software engineering*. Boston: Kluwer Academic Publishers
- IEEE and ISO/IEC. (2008). *IEEE and ISO/IEC Standard 15288: Systems and software engineering —system life cycle processes*. New York: Institute of Electrical and Electronics Engineers and the International Organization for Standardization and the International Electrotechnical Commission.
- Adams, K. M. (2015). *Non-functional requirements in systems analysis and design* (Vol. 28). Heidelberg: Springer.